chanhvinh.com

User's Manual

32

SH7262 Group, SH7264 Group

User's Manual: Hardware

Renesas 32-Bit RISC Microcomputer
SuperH™ RISC engine Family / SH7260 Series

SH7262	R5S72620	SH7264	R5S72640
	R5S72621		R5S72641
	R5S72622		R5S72642
	R5S72623		R5S72643
	R5S72624		R5S72644
	R5S72625		R5S72645
	R5S72626		R5S72646
	R5S72627		R5S72647

Renesas Electronics
www.renesas.com

Rev.3.00 2011.03

chanhvinh.com

chanhvinh.com

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of 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. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. 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 of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may 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.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

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

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, 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, please evaluate the safety of the final products or system 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. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

chanhvinh.com

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins

   Handle unused pins in accord 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 LSI, an associated shoot-through current flows internally, and malfunctions may occur due to the false recognition of the pin state as an input signal. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment 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 moment 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 moment 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 moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has 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. Moreover, 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.

5. Differences between Products

   Before changing from one product to another, i.e. to one with a different type number, confirm that the change will not lead to problems.

   — The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

chanhvinh.com

Configuration of This Manual

This manual comprises the following items:

1. General Precautions in the Handling of MPU/MCU Products
2. Configuration of This Manual
3. Preface
4. Contents
5. Overview
6. Description of Functional Modules
   - CPU and System-Control Modules
   - On-Chip Peripheral Modules

   The configuration of the functional description of each module differs according to the module. However, the generic style includes the following items:

   i) Feature
   ii) Input/Output Pin
   iii) Register Description
   iv) Operation
   v) Usage Note

When designing an application system that includes this LSI, take notes into account. Each section includes notes in relation to the descriptions given, and usage notes are given, as required, as the final part of each section.

7. List of Registers
8. Electrical Characteristics
9. Appendix
   - Package Dimensions, etc.
10. Main Revisions for This Edition (only for revised versions)

The list of revisions is a summary of points that have been revised or added to earlier versions. This does not include all of the revised contents. For details, see the actual locations in this manual.

11. Index

chanhvinh.com

Preface

This LSI is an RISC (Reduced Instruction Set Computer) microcomputer which includes a Renesas-original RISC CPU as its core, and the peripheral functions required to configure a system.

Target Users: This manual was written for users who will be using this LSI in the design of application systems. Target users are expected to understand the fundamentals of electrical circuits, logical circuits, and microcomputers.

Objective: This manual was written to explain the hardware functions and electrical characteristics of this LSI to the target users.
Refer to the SH-2A, SH2A-FPU Software Manual for a detailed description of the instruction set.

Notes on reading this manual:

- In order to understand the overall functions of the chip
  Read the manual according to the contents. This manual can be roughly categorized into parts on the CPU, system control functions, peripheral functions and electrical characteristics.
- In order to understand the details of the CPU's functions
  Read the SH-2A, SH2A-FPU Software Manual.
- In order to understand the details of a register when its name is known
  Read the index that is the final part of the manual to find the page number of the entry on the register. The addresses, bits, and initial values of the registers are summarized in section 36, List of Registers.

chanhvinh.com

- Description of Numbers and Symbols

  Aspects of the notations for register names, bit names, numbers, and symbolic names in this manual are explained below.

(1) Overall notation

In descriptions involving the names of bits and bit fields within this manual, the modules and registers to which the bits belong may be clarified by giving the names in the forms "module name"."register name"."bit name" or "register name"."bit name".

(2) Register notation

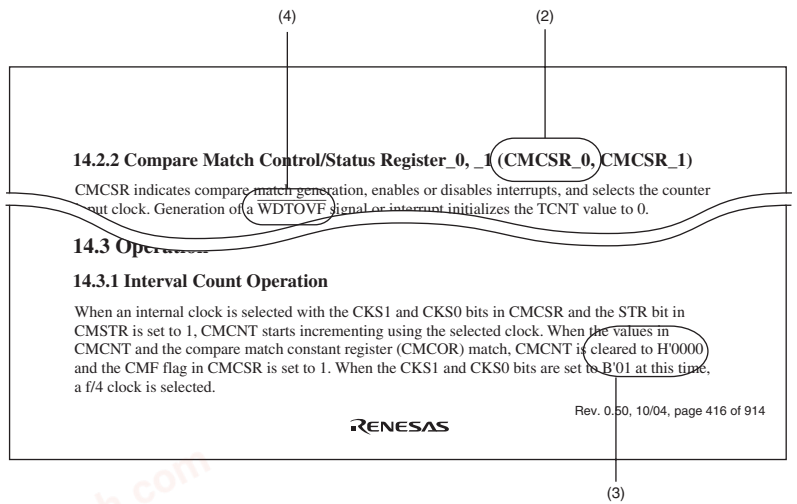
The style "register name"_"instance number" is used in cases where there is more than one instance of the same function or similar functions.
[Example] CMCSR_0: Indicates the CMCSR register for the compare-match timer of channel 0.

(3) Number notation

Binary numbers are given as B'nnnn (B' may be omitted if the number is obviously binary), hexadecimal numbers are given as H'nnnn or 0xnnnn, and decimal numbers are given as nnnn.
[Examples] Binary: B'11 or 11
Hexadecimal: H'EFA0 or 0xEFA0
Decimal: 1234

(4) Notation for active-low

An overbar on the name indicates that a signal or pin is active-low.
[Example] $\overline{\text{WDTOVF}}$



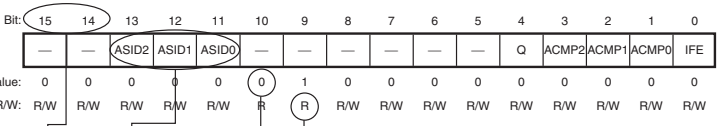
Note: The bit names and sentences in the above figure are examples and do not refer to specific data in this manual.

chanhvinh.com

- Description of Registers

  Each register description includes a bit chart, illustrating the arrangement of bits, and a table of bits, describing the meanings of the bit settings. The standard format and notation for bit charts and tables are described below.

[Bit Chart]



[Table of Bits]

(1) (2) (3) (4) (5)

Bit	Bit Name	Initial Value	R/W	Description
15	–	0	R	Reserved
14	–	0	R	These bits are always read as 0.
13 to 11	ASID2 to ASID0	All 0	R/W	Address Identifier These bits enable or disable the pin function.
10	–	0	R	Reserved This bit is always read as 0.
9	–	1	R	Reserved This bit is always read as 1.
	–	0		

Note: The bit names and sentences in the above figure are examples, and have nothing to do with the contents of this manual.

(1) Bit
Indicates the bit number or numbers.
In the case of a 32-bit register, the bits are arranged in order from 31 to 0. In the case of a 16-bit register, the bits are arranged in order from 15 to 0.

(2) Bit name
Indicates the name of the bit or bit field.
When the number of bits has to be clearly indicated in the field, appropriate notation is included (e.g., ASID[3:0]).
A reserved bit is indicated by "–".
Certain kinds of bits, such as those of timer counters, are not assigned bit names. In such cases, the entry under Bit Name is blank.

(3) Initial value
Indicates the value of each bit immediately after a power-on reset, i.e., the initial value.
0: The initial value is 0
1: The initial value is 1
–: The initial value is undefined

(4) R/W
For each bit and bit field, this entry indicates whether the bit or field is readable or writable, or both writing to and reading from the bit or field are impossible.
The notation is as follows:

R/W: The bit or field is readable and writable.
R/(W): The bit or field is readable and writable.
However, writing is only performed to flag clearing.
R: The bit or field is readable.
"R" is indicated for all reserved bits. When writing to the register, write the value under Initial Value in the bit chart to reserved bits or fields.
W: The bit or field is writable.

(5) Description
Describes the function of the bit or field and specifies the values for writing.

All trademarks and registered trademarks are the property of their respective owners.

chanhvinh.com

Contents

Section 1 Overview 1
1.1 SH7262/7264 Features 1
1.2 Product Lineup 12
1.3 Block Diagram 14
1.4 Pin Assignment 15
1.5 Pin Functions 19
1.6 List of Pins 29

Section 2 CPU 49
2.1 Register Configuration 49
2.1.1 General Registers 49
2.1.2 Control Registers 50
2.1.3 System Registers 52
2.1.4 Register Banks 53
2.1.5 Initial Values of Registers 53
2.2 Data Formats 54
2.2.1 Data Format in Registers 54
2.2.2 Data Formats in Memory 54
2.2.3 Immediate Data Format 55
2.3 Instruction Features 56
2.3.1 RISC-Type Instruction Set 56
2.3.2 Addressing Modes 60
2.3.3 Instruction Format 65
2.4 Instruction Set 69
2.4.1 Instruction Set by Classification 69
2.4.2 Data Transfer Instructions 75
2.4.3 Arithmetic Operation Instructions 79
2.4.4 Logic Operation Instructions 82
2.4.5 Shift Instructions 83
2.4.6 Branch Instructions 84
2.4.7 System Control Instructions 85
2.4.8 Floating-Point Operation Instructions 87
2.4.9 FPU-Related CPU Instructions 89
2.4.10 Bit Manipulation Instructions 90
2.5 Processing States 92

chanhvinh.com

Section 3 Floating-Point Unit (FPU) 95
3.1 Features 95
3.2 Data Formats 96
3.2.1 Floating-Point Format 96
3.2.2 Non-Numbers (NaN) 99
3.2.3 Denormalized Numbers 100
3.3 Register Descriptions 101
3.3.1 Floating-Point Registers 101
3.3.2 Floating-Point Status/Control Register (FPSCR) 102
3.3.3 Floating-Point Communication Register (FPUL) 104
3.4 Rounding 105
3.5 FPU Exceptions 106
3.5.1 FPU Exception Sources 106
3.5.2 FPU Exception Handling 106

Section 4 Boot Mode 109
4.1 Features 109
4.2 Boot Mode and Pin Function Setting 109
4.3 Operation 110
4.3.1 Boot Mode 0 110
4.3.2 Boot Modes 1 and 3 110
4.3.3 Boot Mode 2 111
4.4 Notes 114
4.4.1 Boot Related Pins 114

Section 5 Clock Pulse Generator 115
5.1 Features 115
5.2 Input/Output Pins 118
5.3 Clock Operating Modes 119
5.4 Register Descriptions 122
5.4.1 Frequency Control Register (FRQCR) 122
5.5 Changing the Frequency 125
5.5.1 Changing the Division Ratio 125
5.6 Usage of the Clock Pins 126
5.6.1 In the Case of Inputting an External Clock 126
5.6.2 In the Case of Using a Crystal Resonator 127
5.6.3 In the Case of Not Using the Clock Pin 127
5.7 Oscillation Stabilizing Time 128
5.7.1 Oscillation Stabilizing Time of the On-chip Crystal Oscillator 128

chanhvinh.com

5.7.2 Oscillation Stabilizing Time of the PLL circuit 128
5.8 Notes on Board Design 129
5.8.1 Note on Using a PLL Oscillation Circuit 129

Section 6 Exception Handling 131
6.1 Overview 131
6.1.1 Types of Exception Handling and Priority 131
6.1.2 Exception Handling Operations 132
6.1.3 Exception Handling Vector Table 134
6.2 Resets 137
6.2.1 Input/Output Pins 137
6.2.2 Types of Reset 137
6.2.3 Power-On Reset 139
6.2.4 Manual Reset 140
6.3 Address Errors 142
6.3.1 Address Error Sources 142
6.3.2 Address Error Exception Handling 143
6.4 Register Bank Errors 143
6.4.1 Register Bank Error Sources 143
6.4.2 Register Bank Error Exception Handling 144
6.5 Interrupts 144
6.5.1 Interrupt Sources 144
6.5.2 Interrupt Priority Level 145
6.5.3 Interrupt Exception Handling 146
6.6 Exceptions Triggered by Instructions 147
6.6.1 Types of Exceptions Triggered by Instructions 147
6.6.2 Trap Instructions 148
6.6.3 Slot Illegal Instructions 148
6.6.4 General Illegal Instructions 149
6.6.5 Integer Division Exceptions 149
6.6.6 FPU Exceptions 150
6.7 When Exception Sources Are Not Accepted 151
6.8 Stack Status after Exception Handling Ends 151
6.9 Usage Notes 153
6.9.1 Value of Stack Pointer (SP) 153
6.9.2 Value of Vector Base Register (VBR) 153
6.9.3 Address Errors Caused by Stacking of Address Error Exception Handling 153
6.9.4 Interrupt Control via Modification of Interrupt Mask Bits 153
6.9.5 Note before Exception Handling Begins Running 154

chanhvinh.com

Section 7 Interrupt Controller 157
7.1 Features 157
7.2 Input/Output Pins 159
7.3 Register Descriptions 160
7.3.1 Interrupt Priority Registers 01, 02, 05 to 22 (IPR01, IPR02, IPR05 to IPR22) 162
7.3.2 Interrupt Control Register 0 (ICR0) 164
7.3.3 Interrupt Control Register 1 (ICR1) 166
7.3.4 Interrupt Control Register 2 (ICR2) 167
7.3.5 IRQ Interrupt Request Register (IRQRR) 168
7.3.6 PINT Interrupt Enable Register (PINTER) 169
7.3.7 PINT Interrupt Request Register (PIRR) 170
7.3.8 Bank Control Register (IBCR) 171
7.3.9 Bank Number Register (IBNR) 172
7.4 Interrupt Sources 173
7.4.1 NMI Interrupt 173
7.4.2 User Debugging Interface Interrupt 174
7.4.3 IRQ Interrupts 174
7.4.4 PINT Interrupts 175
7.4.5 On-Chip Peripheral Module Interrupts 176
7.5 Interrupt Exception Handling Vector Table and Priority 177
7.6 Operation 191
7.6.1 Interrupt Operation Sequence 191
7.6.2 Stack after Interrupt Exception Handling 194
7.7 Interrupt Response Time 195
7.8 Register Banks 201
7.8.1 Banked Register and Input/Output of Banks 202
7.8.2 Bank Save and Restore Operations 202
7.8.3 Save and Restore Operations after Saving to All Banks 204
7.8.4 Register Bank Exception 205
7.8.5 Register Bank Error Exception Handling 205
7.9 Data Transfer with Interrupt Request Signals 206
7.9.1 Handling Interrupt Request Signals as Sources for CPU Interrupt but Not Direct Memory Access Controller Activating 207
7.9.2 Handling Interrupt Request Signals as Sources for Activating Direct Memory Access Controller but Not CPU Interrupt 207
7.10 Usage Note 208
7.10.1 Timing to Clear an Interrupt Source 208

chanhvinh.com

Section 8 Cache 209
8.1 Features 209
8.1.1 Cache Structure 209
8.2 Register Descriptions 212
8.2.1 Cache Control Register 1 (CCR1) 212
8.2.2 Cache Control Register 2 (CCR2) 214
8.3 Operation 218
8.3.1 Searching Cache 218
8.3.2 Read Access 220
8.3.3 Prefetch Operation (Only for Operand Cache) 220
8.3.4 Write Operation (Only for Operand Cache) 221
8.3.5 Write-Back Buffer (Only for Operand Cache) 221
8.3.6 Coherency of Cache and External Memory or Large-Capacity On-Chip RAM 223
8.4 Memory-Mapped Cache 224
8.4.1 Address Array 224
8.4.2 Data Array 225
8.4.3 Usage Examples 227
8.4.4 Usage Notes 228

Section 9 Bus State Controller 229
9.1 Features 229
9.2 Input/Output Pins 232
9.3 Area Overview 234
9.3.1 Address Map 234
9.3.2 Data Bus Width and Endian Specification of Each Area and Related Pin Settings Depending on Boot Mode 235
9.4 Register Descriptions 237
9.4.1 Common Control Register (CMNCR) 239
9.4.2 CSn Space Bus Control Register (CSnBCR) (n = 0 to 6) 242
9.4.3 CSn Space Wait Control Register (CSnWCR) (n = 0 to 6) 247
9.4.4 SDRAM Control Register (SDCR) 279
9.4.5 Refresh Timer Control/Status Register (RTCSR) 283
9.4.6 Refresh Timer Counter (RTCNT) 285
9.4.7 Refresh Time Constant Register (RTCOR) 286
9.4.8 AC Characteristics Switching Register (ACSWR) 287
9.4.9 AC Characteristics Switching Key Register (ACKEYR) 288
9.4.10 Sequence to Write to ACSWR 289
9.5 Operation 290
9.5.1 Endian/Access Size and Data Alignment 290

chanhvinh.com

9.5.2 Normal Space Interface ... 293
9.5.3 Access Wait Control ... 297
9.5.4 $\overline{\text{CSn}}$ Assert Period Expansion ... 299
9.5.5 MPX-I/O Interface ... 300
9.5.6 SDRAM Interface ... 303
9.5.7 Burst ROM (Clocked Asynchronous) Interface ... 339
9.5.8 SRAM Interface with Byte Selection ... 341
9.5.9 PCMCIA Interface ... 345
9.5.10 Burst ROM (Clocked Synchronous) Interface ... 352
9.5.11 Wait between Access Cycles ... 353
9.5.12 Bus Arbitration ... 361
9.5.13 Others ... 363

Section 10 Direct Memory Access Controller ... 367
10.1 Features ... 367
10.2 Input/Output Pins ... 370
10.3 Register Descriptions ... 371
10.3.1 DMA Source Address Registers (SAR) ... 380
10.3.2 DMA Destination Address Registers (DAR) ... 380
10.3.3 DMA Transfer Count Registers (DMATCR) ... 381
10.3.4 DMA Channel Control Registers (CHCR) ... 381
10.3.5 DMA Reload Source Address Registers (RSAR) ... 392
10.3.6 DMA Reload Destination Address Registers (RDAR) ... 392
10.3.7 DMA Reload Transfer Count Registers (RDMATCR) ... 393
10.3.8 DMA Operation Register (DMAOR) ... 394
10.3.9 DMA Extension Resource Selectors 0 to 7 (DMARS0 to DMARS7) ... 398
10.4 Operation ... 404
10.4.1 Transfer Flow ... 404
10.4.2 DMA Transfer Requests ... 406
10.4.3 Channel Priority ... 413
10.4.4 DMA Transfer Types ... 413
10.4.5 Number of Bus Cycles and DREQ Pin Sampling Timing ... 422
10.5 Usage Notes ... 426
10.5.1 Timing of DACK and TEND Outputs ... 426
10.5.2 Notes on Using Flag Bits ... 426

Section 11 Multi-Function Timer Pulse Unit 2 ... 427
11.1 Features ... 427
11.2 Input/Output Pins ... 432
11.3 Register Descriptions ... 433

chanhvinh.com

11.3.1 Timer Control Register (TCR) 437
11.3.2 Timer Mode Register (TMDR) 441
11.3.3 Timer I/O Control Register (TIOR) 444
11.3.4 Timer Interrupt Enable Register (TIER) 462
11.3.5 Timer Status Register (TSR) 465
11.3.6 Timer Buffer Operation Transfer Mode Register (TBTM) 470
11.3.7 Timer Input Capture Control Register (TICCR) 471
11.3.8 Timer A/D Converter Start Request Control Register (TADCR) 473
11.3.9 Timer A/D Converter Start Request Cycle Set Registers (TADCORA_4 and TADCORB_4) 476
11.3.10 Timer A/D Converter Start Request Cycle Set Buffer Registers (TADCOBRA_4 and TADCOBRB_4) 476
11.3.11 Timer Counter (TCNT) 477
11.3.12 Timer General Register (TGR) 477
11.3.13 Timer Start Register (TSTR) 478
11.3.14 Timer Synchronous Register (TSYR) 479
11.3.15 Timer Read/Write Enable Register (TRWER) 481
11.3.16 Timer Output Master Enable Register (TOER) 482
11.3.17 Timer Output Control Register 1 (TOCR1) 484
11.3.18 Timer Output Control Register 2 (TOCR2) 487
11.3.19 Timer Output Level Buffer Register (TOLBR) 490
11.3.20 Timer Gate Control Register (TGCR) 491
11.3.21 Timer Subcounter (TCNTS) 493
11.3.22 Timer Dead Time Data Register (TDDR) 494
11.3.23 Timer Cycle Data Register (TCDR) 494
11.3.24 Timer Cycle Buffer Register (TCBR) 495
11.3.25 Timer Interrupt Skipping Set Register (TITCR) 495
11.3.26 Timer Interrupt Skipping Counter (TITCNT) 497
11.3.27 Timer Buffer Transfer Set Register (TBTER) 498
11.3.28 Timer Dead Time Enable Register (TDER) 500
11.3.29 Timer Waveform Control Register (TWCR) 501
11.3.30 Bus Master Interface 502
11.4 Operation 503
11.4.1 Basic Functions 503
11.4.2 Synchronous Operation 509
11.4.3 Buffer Operation 511
11.4.4 Cascaded Operation 515
11.4.5 PWM Modes 520
11.4.6 Phase Counting Mode 525
11.4.7 Reset-Synchronized PWM Mode 532

chanhvinh.com

11.4.8 Complementary PWM Mode 535
11.4.9 A/D Converter Start Request Delaying Function 574
11.4.10 TCNT Capture at Crest and/or Trough in Complementary PWM Operation 578
11.5 Interrupt Sources 579
11.5.1 Interrupt Sources and Priorities 579
11.5.2 Activation of Direct Memory Access Controller 581
11.5.3 A/D Converter Activation 581
11.6 Operation Timing 583
11.6.1 Input/Output Timing 583
11.6.2 Interrupt Signal Timing 590
11.7 Usage Notes 594
11.7.1 Module Standby Mode Setting 594
11.7.2 Input Clock Restrictions 594
11.7.3 Caution on Period Setting 595
11.7.4 Contention between TCNT Write and Clear Operations 595
11.7.5 Contention between TCNT Write and Increment Operations 596
11.7.6 Contention between TGR Write and Compare Match 597
11.7.7 Contention between Buffer Register Write and Compare Match 598
11.7.8 Contention between Buffer Register Write and TCNT Clear 599
11.7.9 Contention between TGR Read and Input Capture 600
11.7.10 Contention between TGR Write and Input Capture 601
11.7.11 Contention between Buffer Register Write and Input Capture 602
11.7.12 TCNT2 Write and Overflow/Underflow Contention in Cascade Connection 602
11.7.13 Counter Value during Complementary PWM Mode Stop 604
11.7.14 Buffer Operation Setting in Complementary PWM Mode 604
11.7.15 Reset Sync PWM Mode Buffer Operation and Compare Match Flag 605
11.7.16 Overflow Flags in Reset Synchronous PWM Mode 606
11.7.17 Contention between Overflow/Underflow and Counter Clearing 607
11.7.18 Contention between TCNT Write and Overflow/Underflow 608
11.7.19 Cautions on Transition from Normal Operation or PWM Mode 1 to Reset-Synchronized PWM Mode 608
11.7.20 Output Level in Complementary PWM Mode and Reset-Synchronized PWM Mode 609
11.7.21 Interrupts in Module Standby Mode 609
11.7.22 Simultaneous Capture of TCNT_1 and TCNT_2 in Cascade Connection 609
11.7.23 Notes on Output Waveform Control During Synchronous Counter Clearing in Complementary PWM Mode 610
11.8 Output Pin Initialization for Multi-Function Timer Pulse Unit 2 612
11.8.1 Operating Modes 612
11.8.2 Reset Start Operation 612

chanhvinh.com

11.8.3 Operation in Case of Re-Setting Due to Error during Operation, etc. 613
11.8.4 Overview of Initialization Procedures and Mode Transitions in Case of Error during Operation, etc. 614

Section 12 Compare Match Timer 645
12.1 Features 645
12.2 Register Descriptions 646
12.2.1 Compare Match Timer Start Register (CMSTR) 647
12.2.2 Compare Match Timer Control/Status Register (CMCSR) 648
12.2.3 Compare Match Counter (CMCNT) 650
12.2.4 Compare Match Constant Register (CMCOR) 650
12.3 Operation 651
12.3.1 Interval Count Operation 651
12.3.2 CMCNT Count Timing 651
12.4 Interrupts 652
12.4.1 Interrupt Sources and DMA Transfer Requests 652
12.4.2 Timing of Compare Match Flag Setting 652
12.4.3 Timing of Compare Match Flag Clearing 653
12.5 Usage Notes 654
12.5.1 Conflict between Write and Compare-Match Processes of CMCNT 654
12.5.2 Conflict between Word-Write and Count-Up Processes of CMCNT 655
12.5.3 Conflict between Byte-Write and Count-Up Processes of CMCNT 656
12.5.4 Compare Match between CMCNT and CMCOR 656

Section 13 Watchdog Timer 657
13.1 Features 657
13.2 Input/Output Pin 658
13.3 Register Descriptions 659
13.3.1 Watchdog Timer Counter (WTCNT) 659
13.3.2 Watchdog Timer Control/Status Register (WTCSR) 660
13.3.3 Watchdog Reset Control/Status Register (WRCSR) 663
13.3.4 Notes on Register Access 664
13.4 Usage 666
13.4.1 Canceling Software Standby Mode 666
13.4.2 Using Watchdog Timer Mode 666
13.4.3 Using Interval Timer Mode 668
13.5 Usage Notes 669
13.5.1 Timer Variation 669
13.5.2 Prohibition against Setting H'FF to WTCNT 669
13.5.3 Interval Timer Overflow Flag 669

chanhvinh.com

13.5.4 System Reset by $\overline{\text{WDTOVF}}$ Signal 670
13.5.5 Manual Reset in Watchdog Timer Mode 670

Section 14 Realtime Clock 671
14.1 Features 671
14.2 Input/Output Pin 673
14.3 Register Descriptions 673
14.3.1 64-Hz Counter (R64CNT) 674
14.3.2 Second Counter (RSECCNT) 675
14.3.3 Minute Counter (RMINCNT) 676
14.3.4 Hour Counter (RHRCNT) 677
14.3.5 Day of Week Counter (RWKCNT) 678
14.3.6 Date Counter (RDAYCNT) 679
14.3.7 Month Counter (RMONCNT) 680
14.3.8 Year Counter (RYRCNT) 681
14.3.9 Second Alarm Register (RSECAR) 682
14.3.10 Minute Alarm Register (RMINAR) 683
14.3.11 Hour Alarm Register (RHRAR) 684
14.3.12 Day of Week Alarm Register (RWKAR) 685
14.3.13 Date Alarm Register (RDAYAR) 686
14.3.14 Month Alarm Register (RMONAR) 687
14.3.15 Year Alarm Register (RYRAR) 688
14.3.16 Control Register 1 (RCR1) 689
14.3.17 Control Register 2 (RCR2) 691
14.3.18 Control Register 3 (RCR3) 693
14.3.19 Control Register 5 (RCR5) 694
14.3.20 Frequency Register H/L (RFRH/L) 695
14.4 Operation 697
14.4.1 Initial Settings of Registers after Power-On 697
14.4.2 Setting Time 697
14.4.3 Reading Time 698
14.4.4 Alarm Function 699
14.5 Usage Notes 700
14.5.1 Register Writing during Count 700
14.5.2 Use of Realtime Clock Periodic Interrupts 700
14.5.3 Transition to Standby Mode after Setting Register 700
14.5.4 Usage Notes when Writing to and Reading the Register 701

Section 15 Serial Communication Interface with FIFO 703
15.1 Features 703

chanhvinh.com

15.2 Input/Output Pins 706
15.3 Register Descriptions 707
15.3.1 Receive Shift Register (SCRSR) 711
15.3.2 Receive FIFO Data Register (SCFRDR) 711
15.3.3 Transmit Shift Register (SCTSR) 712
15.3.4 Transmit FIFO Data Register (SCFTDR) 712
15.3.5 Serial Mode Register (SCSMR) 713
15.3.6 Serial Control Register (SCSCR) 716
15.3.7 Serial Status Register (SCFSR) 720
15.3.8 Bit Rate Register (SCBRR) 728
15.3.9 FIFO Control Register (SCFCR) 734
15.3.10 FIFO Data Count Set Register (SCFDR) 737
15.3.11 Serial Port Register (SCSPTR) 738
15.3.12 Line Status Register (SCLSR) 741
15.3.13 Serial Extension Mode Register (SCEMR) 742
15.4 Operation 743
15.4.1 Overview 743
15.4.2 Operation in Asynchronous Mode 746
15.4.3 Operation in Clock Synchronous Mode 757
15.5 Interrupts 766
15.6 Usage Notes 767
15.6.1 SCFTDR Writing and TDFE Flag 767
15.6.2 SCFRDR Reading and RDF Flag 767
15.6.3 Restriction on Direct Memory Controller Usage 768
15.6.4 Break Detection and Processing 768
15.6.5 Sending a Break Signal 768
15.6.6 Receive Data Sampling Timing and Receive Margin (Asynchronous Mode) 768
15.6.7 Selection of Base Clock in Asynchronous Mode 770

Section 16 Renesas Serial Peripheral Interface 771
16.1 Features 771
16.2 Input/Output Pins 774
16.3 Register Descriptions 775
16.3.1 Control Register (SPCR) 777
16.3.2 Slave Select Polarity Register (SSLP) 779
16.3.3 Pin Control Register (SPPCR) 780
16.3.4 Status Register (SPSR) 782
16.3.5 Data Register (SPDR) 785
16.3.6 Sequence Control Register (SPSCR) 786
16.3.7 Sequence Status Register (SPSSR) 788

chanhvinh.com

16.3.8 Bit Rate Register (SPBR) ... 789
16.3.9 Data Control Register (SPDCR) ... 791
16.3.10 Clock Delay Register (SPCKD) ... 793
16.3.11 Slave Select Negation Delay Register (SSLND) ... 794
16.3.12 Next-Access Delay Register (SPND) ... 795
16.3.13 Command Register (SPCMD) ... 796
16.3.14 Buffer Control Register (SPBFCR) ... 801
16.3.15 Buffer Data Count Setting Register (SPBFDR) ... 803
16.4 Operation ... 804
16.4.1 Overview of Operations ... 804
16.4.2 Pin Control ... 805
16.4.3 System Configuration Example ... 806
16.4.4 Transfer Format ... 809
16.4.5 Data Format ... 811
16.4.6 Error Detection ... 823
16.4.7 Initialization ... 828
16.4.8 SPI Operation ... 829
16.4.9 Error Handling ... 842
16.4.10 Loopback Mode ... 843
16.4.11 Interrupt Sources ... 844

Section 17 I^2C Bus Interface 3 ... 845
17.1 Features ... 845
17.2 Input/Output Pins ... 847
17.3 Register Descriptions ... 848
17.3.1 I^2C Bus Control Register 1 (ICCR1) ... 849
17.3.2 I^2C Bus Control Register 2 (ICCR2) ... 852
17.3.3 I^2C Bus Mode Register (ICMR) ... 854
17.3.4 I^2C Bus Interrupt Enable Register (ICIER) ... 856
17.3.5 I^2C Bus Status Register (ICSR) ... 858
17.3.6 Slave Address Register (SAR) ... 861
17.3.7 I^2C Bus Transmit Data Register (ICDRT) ... 861
17.3.8 I^2C Bus Receive Data Register (ICDRR) ... 862
17.3.9 I^2C Bus Shift Register (ICDRS) ... 862
17.3.10 NF2CYC Register (NF2CYC) ... 863
17.4 Operation ... 864
17.4.1 I^2C Bus Format ... 864
17.4.2 Master Transmit Operation ... 865
17.4.3 Master Receive Operation ... 867
17.4.4 Slave Transmit Operation ... 869

chanhvinh.com

17.4.5 Slave Receive Operation 872
17.4.6 Clocked Synchronous Serial Format 873
17.4.7 Noise Filter 877
17.4.8 Example of Use 878
17.5 Interrupt Requests 882
17.6 Bit Synchronous Circuit 883
17.7 Usage Notes 885
17.7.1 Note on Setting for Multi-Master Operation 885
17.7.2 Note on Master Receive Mode 885
17.7.3 Note on Setting ACKBT in Master Receive Mode 886
17.7.4 Note on the States of Bits MST and TRN when Arbitration is Lost 886
17.7.5 Note on I^2C-bus Interface Master Receive Mode 886
17.7.6 Note on IICRST and BBSY bits 886

Section 18 Serial Sound Interface 887
18.1 Features 887
18.2 Input/Output Pins 890
18.3 Register Description 891
18.3.1 Control Register (SSICR) 893
18.3.2 Status Register (SSISR) 900
18.3.3 Transmit Data Register (SSITDR) 904
18.3.4 Receive Data Register (SSIRDR) 904
18.3.5 FIFO Control Register (SSIFCR) 905
18.3.6 FIFO Status Register (SSIFSR) 908
18.3.7 Transmit FIFO Data Register (SSIFTDR) 911
18.3.8 Receive FIFO Data Register (SSIFRDR) 911
18.4 Operation Description 912
18.4.1 Bus Format 912
18.4.2 Non-Compressed Modes 913
18.4.3 Operation Modes 923
18.4.4 Transmit Operation 924
18.4.5 Receive Operation 927
18.4.6 Serial Bit Clock Control 930
18.5 Usage Notes 931
18.5.1 Limitations from Underflow or Overflow during DMA Operation 931
18.5.2 Note on Changing Mode from Master Transceiver to Master Receiver 931

Section 19 Serial I/O with FIFO 933
19.1 Features 933
19.2 Input/Output Pins 935

chanhvinh.com

19.3 Register Descriptions 936
19.3.1 Mode Register (SIMDR) 937
19.3.2 Control Register (SICTR) 939
19.3.3 Transmit Data Register (SITDR) 942
19.3.4 Receive Data Register (SIRDR) 943
19.3.5 Status Register (SISTR) 944
19.3.6 Interrupt Enable Register (SIIER) 949
19.3.7 FIFO Control Register (SIFCTR) 951
19.3.8 Clock Select Register (SISCR) 953
19.3.9 Transmit Data Assign Register (SITDAR) 954
19.3.10 Receive Data Assign Register (SIRDAR) 956
19.4 Operation 957
19.4.1 Serial Clocks 957
19.4.2 Serial Timing 958
19.4.3 Transfer Data Format 959
19.4.4 Register Allocation of Transfer Data 960
19.4.5 FIFO 962
19.4.6 Transmit and Receive Procedures 964
19.4.7 Interrupts 969
19.4.8 Transmit and Receive Timing 971

Section 20 Controller Area Network 975
20.1 Summary 975
20.1.1 Overview 975
20.1.2 Scope 975
20.1.3 Audience 975
20.1.4 References 975
20.1.5 Features 976
20.2 Architecture 977
20.3 Programming Model - Overview 980
20.3.1 Memory Map 980
20.3.2 Mailbox Structure 982
20.3.3 Control Registers 998
20.3.4 Mailbox Registers 1019
20.3.5 Timer Registers 1033
20.4 Application Note 1046
20.4.1 Test Mode Settings 1046
20.4.2 Configuration of This Module 1048
20.4.3 Message Transmission Sequence 1052
20.4.4 Message Receive Sequence 1067

chanhvinh.com

20.4.5 Reconfiguration of Mailbox 1069
20.5 Interrupt Sources 1071
20.6 DMAC Interface 1072
20.7 CAN Bus Interface 1073
20.8 Setting I/O Ports 1074
20.9 Usage Notes 1076
20.9.1 Notes on Port Setting for Multiple Channels Used as Single Channel 1076

Section 21 IEBus™ Controller 1077
21.1 Features 1077
21.1.1 IEBus Communications Protocol 1078
21.1.2 Communications Protocol 1082
21.1.3 Transfer Data (Data Field Contents) 1090
21.1.4 Bit Format 1093
21.1.5 Configuration 1094
21.2 Input/Output Pins 1095
21.3 Register Descriptions 1096
21.3.1 IEBus Control Register (IECTR) 1098
21.3.2 IEBus Command Register (IECMR) 1099
21.3.3 IEBus Master Control Register (IEMCR) 1101
21.3.4 IEBus Master Unit Address Register 1 (IEAR1) 1103
21.3.5 IEBus Master Unit Address Register 2 (IEAR2) 1104
21.3.6 IEBus Slave Address Setting Register 1 (IESA1) 1104
21.3.7 IEBus Slave Address Setting Register 2 (IESA2) 1105
21.3.8 IEBus Transmit Message Length Register (IETBFL) 1106
21.3.9 IEBus Reception Master Address Register 1 (IEMA1) 1107
21.3.10 IEBus Reception Master Address Register 2 (IEMA2) 1108
21.3.11 IEBus Receive Control Field Register (IERCTL) 1109
21.3.12 IEBus Receive Message Length Register (IERBFL) 1110
21.3.13 IEBus Lock Address Register 1 (IELA1) 1110
21.3.14 IEBus Lock Address Register 2 (IELA2) 1111
21.3.15 IEBus General Flag Register (IEFLG) 1112
21.3.16 IEBus Transmit Status Register (IETSR) 1115
21.3.17 IEBus Transmit Interrupt Enable Register (IEIET) 1119
21.3.18 IEBus Receive Status Register (IERSR) 1121
21.3.19 IEBus Receive Interrupt Enable Register (IEIER) 1125
21.3.20 IEBus Clock Selection Register (IECKSR) 1127
21.3.21 IEBus Transmit Data Buffer 001 to 128 (IETB001 to IETB128) 1128
21.3.22 IEBus Receive Data Buffer 001 to 128 (IERB001 to IERB128) 1129
21.4 Data Format 1130

chanhvinh.com

21.4.1 Transmission Format 1130
21.4.2 Reception Format 1131
21.5 Software Control Flows 1132
21.5.1 Initial Setting 1132
21.5.2 Master Transmission 1133
21.5.3 Slave Reception 1134
21.5.4 Master Reception 1135
21.5.5 Slave Transmission 1136
21.6 Operation Timing 1137
21.6.1 Master Transmit Operation 1137
21.6.2 Slave Receive Operation 1138
21.6.3 Master Receive Operation 1139
21.6.4 Slave Transmit Operation 1140
21.7 Interrupt Sources 1141
21.8 Usage Notes 1143
21.8.1 Note on Operation when Transfer is Incomplete after Transfer of the Maximum Number of Bytes 1143

Section 22 Renesas SPDIF Interface 1145
22.1 Overview 1145
22.2 Features 1145
22.3 Functional Block Diagram 1146
22.4 Input/Output Pins 1147
22.5 Renesas SPDIF (IEC60958) Frame Format 1147
22.6 Register 1149
22.7 Register Descriptions 1150
22.7.1 Control Register (CTRL) 1150
22.7.2 Status Register (STAT) 1155
22.7.3 Transmitter Channel 1 Audio Register (TLCA) 1159
22.7.4 Transmitter Channel 2 Audio Register (TRCA) 1160
22.7.5 Transmitter DMA Audio Data Register (TDAD) 1161
22.7.6 Transmitter User Data Register (TUI) 1162
22.7.7 Transmitter Channel 1 Status Register (TLCS) 1163
22.7.8 Transmitter Channel 2 Status Register (TRCS) 1165
22.7.9 Receiver Channel 1 Audio Register (RLCA) 1167
22.7.10 Receiver Channel 2 Audio Register (RRCA) 1168
22.7.11 Receiver DMA Audio Data (RDAD) 1169
22.7.12 Receiver User Data Register (RUI) 1170
22.7.13 Receiver Channel 1 Status Register (RLCS) 1171
22.7.14 Receiver Channel 2 Status Register (RRCS) 1173

chanhvinh.com

22.8 Functional Description—Transmitter 1175
22.8.1 Transmitter Module 1175
22.8.2 Transmitter Module Initialization 1176
22.8.3 Initial Settings for Transmitter Module 1176
22.8.4 Transmitter Module Data Transfer 1177
22.9 Functional Description—Receiver 1179
22.9.1 Receiver Module 1179
22.9.2 Receiver Module Initialization 1180
22.9.3 Receiver Module Data Transfer 1180
22.10 Disabling the Module 1183
22.10.1 Transmitter and Receiver Idle 1183
22.11 Compressed Mode Data 1183
22.12 References 1183
22.13 Usage Notes 1184
22.13.1 Clearing TUIR 1184
22.13.2 Frequency of Clock Input for Audio 1184

Section 23 CD-ROM Decoder 1185
23.1 Features 1185
23.1.1 Formats Supported by CD-ROM Decoder 1186
23.2 Block Diagrams 1187
23.3 Register Descriptions 1191
23.3.1 Enable Control Register (CROMEN) 1194
23.3.2 Sync Code-Based Synchronization Control Register (CROMSY0) 1195
23.3.3 Decoding Mode Control Register (CROMCTL0) 1196
23.3.4 EDC/ECC Check Control Register (CROMCTL1) 1198
23.3.5 Automatic Decoding Stop Control Register (CROMCTL3) 1199
23.3.6 Decoding Option Setting Control Register (CROMCTL4) 1200
23.3.7 HEAD20 to HEAD22 Representation Control Register (CROMCTL5) 1202
23.3.8 Sync Code Status Register (CROMST0) 1203
23.3.9 Post-ECC Header Error Status Register (CROMST1) 1204
23.3.10 Post-ECC Subheader Error Status Register (CROMST3) 1205
23.3.11 Header/Subheader Validity Check Status Register (CROMST4) 1206
23.3.12 Mode Determination and Link Sector Detection Status Register (CROMST5) 1207
23.3.13 ECC/EDC Error Status Register (CROMST6) 1208
23.3.14 Buffer Status Register (CBUFST0) 1210
23.3.15 Decoding Stoppage Source Status Register (CBUFST1) 1211
23.3.16 Buffer Overflow Status Register (CBUFST2) 1212
23.3.17 Pre-ECC Correction Header: Minutes Data Register (HEAD00) 1212

chanhvinh.com

23.3.18 Pre-ECC Correction Header: Seconds Data Register (HEAD01)..... 1213
23.3.19 Pre-ECC Correction Header: Frames (1/75 Second) Data Register (HEAD02)1213
23.3.20 Pre-ECC Correction Header: Mode Data Register (HEAD03)..... 1214
23.3.21 Pre-ECC Correction Subheader: File Number (Byte 16) Data Register (SHEAD00) 1214
23.3.22 Pre-ECC Correction Subheader: Channel Number (Byte 17) Data Register (SHEAD01) 1215
23.3.23 Pre-ECC Correction Subheader: Sub-Mode (Byte 18) Data Register (SHEAD02) 1215
23.3.24 Pre-ECC Correction Subheader: Data Type (Byte 19) Data Register (SHEAD03) 1216
23.3.25 Pre-ECC Correction Subheader: File Number (Byte 20) Data Register (SHEAD04) 1216
23.3.26 Pre-ECC Correction Subheader: Channel Number (Byte 21) Data Register (SHEAD05) 1217
23.3.27 Pre-ECC Correction Subheader: Sub-Mode (Byte 22) Data Register (SHEAD06) 1217
23.3.28 Pre-ECC Correction Subheader: Data Type (Byte 23) Data Register (SHEAD07) 1218
23.3.29 Post-ECC Correction Header: Minutes Data Register (HEAD20) 1218
23.3.30 Post-ECC Correction Header: Seconds Data Register (HEAD21) 1219
23.3.31 Post-ECC Correction Header: Frames (1/75 Second) Data Register (HEAD22)..... 1219
23.3.32 Post-ECC Correction Header: Mode Data Register (HEAD23) 1220
23.3.33 Post-ECC Correction Subheader: File Number (Byte 16) Data Register (SHEAD20) 1220
23.3.34 Post-ECC Correction Subheader: Channel Number (Byte 17) Data Register (SHEAD21) 1221
23.3.35 Post-ECC Correction Subheader: Sub-Mode (Byte 18) Data Register (SHEAD22) 1221
23.3.36 Post-ECC Correction Subheader: Data Type (Byte 19) Data Register (SHEAD23) 1222
23.3.37 Post-ECC Correction Subheader: File Number (Byte 20) Data Register (SHEAD24) 1222
23.3.38 Post-ECC Correction Subheader: Channel Number (Byte 21) Data Register (SHEAD25) 1223
23.3.39 Post-ECC Correction Subheader: Sub-Mode (Byte 22) Data Register (SHEAD26) 1223
23.3.40 Post-ECC Correction Subheader: Data Type (Byte 23) Data Register (SHEAD27) 1224

chanhvinh.com

23.3.41 Automatic Buffering Setting Control Register 0 (CBUFCTL0) 1224
23.3.42 Automatic Buffering Start Sector Setting: Minutes Control Register (CBUFCTL1) 1226
23.3.43 Automatic Buffering Start Sector Setting: Seconds Control Register (CBUFCTL2) 1226
23.3.44 Automatic Buffering Start Sector Setting: Frames Control Register (CBUFCTL3) 1227
23.3.45 ISY Interrupt Source Mask Control Register (CROMST0M) 1227
23.3.46 CD-ROM Decoder Reset Control Register (ROMDECRST) 1228
23.3.47 CD-ROM Decoder Reset Status Register (RSTSTAT) 1229
23.3.48 Serial Sound Interface Data Control Register (SSI) 1229
23.3.49 Interrupt Flag Register (INTHOLD) 1232
23.3.50 Interrupt Source Mask Control Register (INHINT) 1233
23.3.51 CD-ROM Decoder Stream Data Input Register (STRMDIN0) 1234
23.3.52 CD-ROM Decoder Stream Data Input Register (STRMDIN2) 1234
23.3.53 CD-ROM Decoder Stream Data Output Register (STRMDOUT0) 1235
23.4 Operation 1236
23.4.1 Endian Conversion for Data in the Input Stream 1236
23.4.2 Sync Code Maintenance Function 1237
23.4.3 Error Correction 1242
23.4.4 Automatic Decoding Stop Function 1243
23.4.5 Buffering Format 1244
23.4.6 Target-Sector Buffering Function 1246
23.5 Interrupt Sources 1248
23.5.1 Interrupt and DMA Transfer Request Signals 1248
23.5.2 Timing of Status Registers Updates 1250
23.6 Usage Notes 1250
23.6.1 Stopping and Resuming Buffering Alone during Decoding 1250
23.6.2 When CROMST0 Status Register Bits are Set 1250
23.6.3 Link Blocks 1251
23.6.4 Stopping and Resuming CD-DSP Operation 1251
23.6.5 Note on Clearing the IREADY Flag 1251
23.6.6 Note on Stream Data Transfer (1) 1252
23.6.7 Note on Stream Data Transfer (2) 1252

Section 24 A/D Converter 1253
24.1 Features 1253
24.2 Input/Output Pins 1255
24.3 Register Descriptions 1256
24.3.1 A/D Data Registers A to H (ADDRA to ADDRH) 1257

chanhvinh.com

24.3.2 A/D Control/Status Register (ADCSR) ... 1258
24.4 Operation ... 1262
24.4.1 Single Mode ... 1262
24.4.2 Multi Mode ... 1265
24.4.3 Scan Mode ... 1267
24.4.4 A/D Converter Activation by External Trigger or Multi-Function Timer Pulse Unit 2 ... 1270
24.4.5 Input Sampling and A/D Conversion Time ... 1270
24.4.6 External Trigger Input Timing ... 1273
24.5 Interrupt Sources and DMA Transfer Request ... 1274
24.6 Definitions of A/D Conversion Accuracy ... 1275
24.7 Usage Notes ... 1276
24.7.1 Module Standby Mode Setting ... 1276
24.7.2 Setting Analog Input Voltage ... 1276
24.7.3 Notes on Board Design ... 1276
24.7.4 Processing of Analog Input Pins ... 1277
24.7.5 Permissible Signal Source Impedance ... 1278
24.7.6 Influences on Absolute Precision ... 1279
24.7.7 A/D Conversion in Deep Standby Mode ... 1279
24.7.8 Note on Usage in Scan Mode and Multi Mode ... 1279

Section 25 NAND Flash Memory Controller ... 1281
25.1 Features ... 1281
25.2 Input/Output Pins ... 1285
25.3 Register Descriptions ... 1286
25.3.1 Common Control Register (FLCMNCR) ... 1287
25.3.2 Command Control Register (FLCMDCR) ... 1291
25.3.3 Command Code Register (FLCMCDR) ... 1294
25.3.4 Address Register (FLADR) ... 1295
25.3.5 Address Register 2 (FLADR2) ... 1297
25.3.6 Data Counter Register (FLDTCNTR) ... 1298
25.3.7 Data Register (FLDATAR) ... 1299
25.3.8 Interrupt DMA Control Register (FLINTDMACR) ... 1300
25.3.9 Ready Busy Timeout Setting Register (FLBSYTMR) ... 1305
25.3.10 Ready Busy Timeout Counter (FLBSYCNT) ... 1306
25.3.11 Data FIFO Register (FLDTFIFO) ... 1307
25.3.12 Control Code FIFO Register (FLECFIFO) ... 1308
25.3.13 Transfer Control Register (FLTRCR) ... 1309
25.3.14 Bus Hold Time Setting Register (FLHOLDCR) ... 1310
25.3.15 4-Symbol ECC Processing Result Register n (FL4ECCRESn) (n = 1 to 4) ... 1311

chanhvinh.com

25.3.16 4-Symbol ECC Control Register (FL4ECCCR) 1312
25.3.17 4-Symbol ECC Error Count Register (FL4ECCCNT) 1314
25.4 Operation 1316
25.4.1 Access Sequence 1316
25.4.2 Operating Modes 1316
25.4.3 Register Setting Procedure 1317
25.4.4 Command Access Mode 1318
25.4.5 Sector Access Mode 1321
25.4.6 ECC Error Correction 1326
25.4.7 Status Read 1331
25.5 Interrupt Sources 1332
25.6 DMA Transfer Specifications 1332
25.7 Usage Notes 1333
25.7.1 External Bus Mastership Release Timing 1333
25.7.2 Writing to the Control-Code Area when 4-Symbol ECC Circuit is in Use 1335
25.7.3 Usage Notes for the SNAND Bit 1336

Section 26 USB 2.0 Host/Function Module 1337
26.1 Features 1337
26.2 Input/Output Pins 1339
26.3 Register Description 1341
26.3.1 System Configuration Control Register (SYSCFG) 1344
26.3.2 CPU Bus Wait Setting Register (BUSWAIT) 1348
26.3.3 System Configuration Status Register (SYSSTS) 1349
26.3.4 Device State Control Register (DVSTCTR) 1350
26.3.5 Test Mode Register (TESTMODE) 1356
26.3.6 DMA-FIFO Bus Configuration Registers (D0FBCFG, D1FBCFG) 1359
26.3.7 FIFO Port Registers (CFIFO, D0FIFO, D1FIFO) 1360
26.3.8 FIFO Port Select Registers (CFIFOSEL, D0FIFOSEL, D1FIFOSEL) 1362
26.3.9 FIFO Port Control Registers (CFIFOCTR, D0FIFOCTR, D1FIFOCTR) 1370
26.3.10 Interrupt Enable Register 0 (INTENB0) 1374
26.3.11 Interrupt Enable Register 1 (INTENB1) 1376
26.3.12 BRDY Interrupt Enable Register (BRDYENB) 1378
26.3.13 NRDY Interrupt Enable Register (NRDYENB) 1379
26.3.14 BEMP Interrupt Enable Register (BEMPENB) 1381
26.3.15 SOF Output Configuration Register (SOFCFG) 1382
26.3.16 Interrupt Status Register 0 (INTSTS0) 1384
26.3.17 Interrupt Status Register 1 (INTSTS1) 1389
26.3.18 BRDY Interrupt Status Register (BRDYSTS) 1395
26.3.19 NRDY Interrupt Status Register (NRDYSTS) 1397

chanhvinh.com

26.3.20 BEMP Interrupt Status Register (BEMPSTS) 1399
26.3.21 Frame Number Register (FRMNUM) 1400
26.3.22 µFrame Number Register (UFRMNUM) 1403
26.3.23 USB Address Register (USBADDR) 1404
26.3.24 USB Request Type Register (USBREQ) 1405
26.3.25 USB Request Value Register (USBVAL) 1406
26.3.26 USB Request Index Register (USBINDX) 1407
26.3.27 USB Request Length Register (USBLENG) 1408
26.3.28 DCP Configuration Register (DCPCFG) 1409
26.3.29 DCP Maximum Packet Size Register (DCPMAXP) 1410
26.3.30 DCP Control Register (DCPCTR) 1411
26.3.31 Pipe Window Select Register (PIPESEL) 1421
26.3.32 Pipe Configuration Register (PIPECFG) 1422
26.3.33 Pipe Buffer Setting Register (PIPEBUF) 1429
26.3.34 Pipe Maximum Packet Size Register (PIPEMAXP) 1432
26.3.35 Pipe Timing Control Register (PIPEPERI) 1434
26.3.36 PIPEn Control Registers (PIPEnCTR) (n = 1 to 9) 1436
26.3.37 PIPEn Transaction Counter Enable Registers (PIPEnTRE) (n = 1 to 5) 1457
26.3.38 PIPEn Transaction Counter Registers (PIPEnTRN) (n = 1 to 5) 1459
26.3.39 Device Address n Configuration Registers (DEVADDn) (n = 0 to A) 1461
26.3.40 USB AC Characteristics Switching Register 1 (USBACSWR1) 1464
26.4 Operation 1465
26.4.1 System Control and Oscillation Control 1465
26.4.2 Interrupt Functions 1469
26.4.3 Pipe Control 1492
26.4.4 FIFO Buffer Memory 1502
26.4.5 Control Transfers (DCP) 1516
26.4.6 Bulk Transfers (PIPE1 to PIPE5) 1520
26.4.7 Interrupt Transfers (PIPE6 to PIPE9) 1522
26.4.8 Isochronous Transfers (PIPE1 and PIPE2) 1523
26.4.9 SOF Interpolation Function 1535
26.4.10 Pipe Schedule 1536
26.5 Usage Notes 1538
26.5.1 Procedure for Setting the USB Transceiver 1538
26.5.2 Power Supply for USB Transceiver 1538

Section 27 Video Display Controller 3 1539
27.1 Overview 1539
27.2 Features 1539
27.3 Input/Output Pins 1541

chanhvinh.com

27.4 Configuration 1542
27.5 Input Video Interface 1544
27.5.1 BT.601 Video Input 1544
27.5.2 BT.656 Video Input 1545
27.6 Functional Descriptions 1548
27.6.1 Video Display Function 1548
27.6.2 Video Recording Function 1553
27.6.3 Panel Control Signal Output Function 1555
27.7 Register Descriptions 1558
27.7.1 Video Operating Mode Register (VIDEO_MODE) 1562
27.7.2 Video Interrupt Control Register (VIDEO_INT_CNT) 1566
27.7.3 Video Input Timing Control Register (VIDEO_TIM_CNT) 1569
27.7.4 Valid Video Size Register (VIDEO_SIZE) 1571
27.7.5 Vertical Valid Video Start Position Register (VIDEO_VSTART) 1572
27.7.6 Horizontal Valid Video Start Position Register (VIDEO_HSTART) 1573
27.7.7 Timing Control Register 1 for Vertical Sync Signal for Video (VIDEO_VSYNC_TIM1) 1575
27.7.8 Video Storing Field Count Register (VIDEO_SAVE_NUM) 1576
27.7.9 Video Scaling and Correction Register (VIDEO_IMAGE_CNT) 1577
27.7.10 Video Base Address Register (VIDEO_BASEADR) 1579
27.7.11 Video Line Offset Register (VIDEO_LINE_OFFSET) 1580
27.7.12 Video Field Offset Register (VIDEO_FIELD_OFFSET) 1581
27.7.13 Video Line Buffer Count Register (VIDEO_LINEBUFF_NUM) 1582
27.7.14 Video Display and Recording Size Register (VIDEO_DISP_SIZE) 1583
27.7.15 Horizontal Video Display Position Register (VIDEO_DISP_HSTART) 1584
27.7.16 Graphics Block Control Registers (GRCMEN1 and GRCMEN2) 1585
27.7.17 Bus Control Registers (GRCBUSCNT1 and GRCBUSCNT2) 1587
27.7.18 Graphics Block Interrupt Control Registers (GRCINTCNT1 and GRCINTCNT2) 1590
27.7.19 Graphics Image Base Address Registers (GROPSADR1 and GROPSADR2) 1591
27.7.20 Graphics Image Size Registers (GROPSWH1 and GROPSWH2) 1591
27.7.21 Graphics Image Line Offset Registers (GROPSOFST1 and GROPSOFST2) 1593
27.7.22 Graphics Image Start Position Registers (GROPDPHV1 and GROPDPHV2) 1594
27.7.23 α Control Area Size Registers (GROPEWH1 and GROPEWH2) 1595
27.7.24 α Control Area Start Position Registers (GROPEDPHV1 and GROPEDPHV2) 1596
27.7.25 α Control Registers (GROPEDPA1 and GROPEDPA2) 1597

chanhvinh.com

27.7.26 Chroma-Key Control Registers (GROPCRKY0_1 and GROPCRKY0_2) 1600
27.7.27 Chroma-Key Color Registers (GROPCRKY1_1 and GROPCRKY1_2) 1601
27.7.28 Color Registers for Outside of Graphics Image Area (GROPBASERGB1 and GROPBASERGB2) 1602
27.7.29 SG Mode Register (SGMODE) 1603
27.7.30 Interrupt Output Control Register (SGINTCNT)..... 1604
27.7.31 Sync Signal Control Register (SYNCNT) 1605
27.7.32 Panel Clock Select Register (PANEL_CLKSEL)..... 1607
27.7.33 Sync Signal Size Register (SYN_SIZE) 1609
27.7.34 Vertical Sync Signal Timing Control Register (PANEL_VSYNC_TIM) 1610
27.7.35 Horizontal Sync Signal Timing Control Register (PANEL_HSYNC_TIM) .. 1611
27.7.36 Timing Control Register 2 for Vertical Sync Signal for Video (VIDEO_VSYNC_TIM2)..... 1612
27.7.37 Timing Control Register for Vertical Sync Signal for Graphics Image (GRA_VSYNC_TIM) 1613
27.7.38 AC Modulation Signal Toggle Line Count Register (AC_LINE_NUM) 1614
27.7.39 DE Area Size Register (DE_SIZE)..... 1615
27.7.40 DE Area Start Position Register (DE_START)..... 1616
27.8 Usage Notes..... 1618
27.8.1 Input Video Format Settings..... 1618
27.8.2 How to Use Video Recording Mode..... 1619
27.8.3 How to Use Video Display Mode..... 1620
27.8.4 How to Use Graphics Display..... 1624
27.8.5 How to Use Control Signal Output to Panel..... 1627
27.9 Interrupt Requests..... 1628
27.10 Usage Note..... 1628
27.10.1 The Procedure of Disabling the Video Receiving Block Operation 1628

Section 28 Sampling Rate Converter 1629
28.1 Features..... 1629
28.2 Register Descriptions..... 1631
28.2.1 Input Data Register (SRCID)..... 1632
28.2.2 Output Data Register (SRCOD)..... 1633
28.2.3 Input Data Control Register (SRCIDCTRL)..... 1635
28.2.4 Output Data Control Register (SRCODCTRL) 1637
28.2.5 Control Register (SRCCTRL)..... 1639
28.2.6 Status Register (SRCSTAT) 1645
28.3 Operation 1650
28.3.1 Initial Setting 1650
28.3.2 Data Input 1651

chanhvinh.com

28.3.3 Data Output 1653
28.4 Interrupts 1655
28.5 Usage Notes 1656
28.5.1 Notes on Accessing Registers 1656
28.5.2 Notes on Flush Processing 1656
28.5.3 Notes on Using Two Channels at the Same Time 1657

Section 29 SD Host Interface 1659

Section 30 Decompression Unit 1661

Section 31 On-Chip RAM 1663
31.1 Features 1663
31.2 Usage Notes 1667
31.2.1 Page Conflict 1667
31.2.2 RAME and RAMWE Bits 1667
31.2.3 Data Retention 1668

Section 32 General Purpose I/O Ports 1669
32.1 Features 1669
32.2 Register Descriptions 1678
32.2.1 Port A I/O Register 0 (PAIOR0) 1681
32.2.2 Port A Data Registers 1, 0 (PADR1, PADR0) 1681
32.2.3 Port A Port Register 0 (PAPR0) 1683
32.2.4 Port B Control Registers 0 to 5 (PBCR0 to PBCR5) 1683
32.2.5 Port B I/O Registers 0, 1 (PBIOR0, PBIOR1) 1693
32.2.6 Port B Data Registers 0, 1 (PBDR0, PBDR1) 1694
32.2.7 Port B Port Registers 0, 1 (PBPR0, PBPR1) 1696
32.2.8 Port C Control Registers 0 to 2 (PCCR0 to PCCR2) 1698
32.2.9 Port C I/O Register 0 (PCIOR0) 1701
32.2.10 Port C Data Register 0 (PCDR0) 1702
32.2.11 Port C Port Register 0 (PCPR0) 1704
32.2.12 Port D Control Register 0 to 3 (PDCR0 to PDCR3) 1705
32.2.13 Port D I/O Register 0 (PDIOR0) 1711
32.2.14 Port D Port Registers 0 (PDDR0) 1712
32.2.15 Port D Port Registers 0 (PDPR0) 1714
32.2.16 Port E Control Registers 0, 1 (PECR0, PECR1) 1715
32.2.17 Port E I/O Register 0 (PEIOR0) 1717
32.2.18 Port E Data Register 0 (PEDR0) 1717
32.2.19 Port E Port Register 0 (PEPR0) 1719

chanhvinh.com

32.2.20 Port F Control Register 0 to 3 (PFCR0 to PFCR3) ... 1720
32.2.21 Port F I/O Register 0 (PFIOR0) ... 1726
32.2.22 Port F Data Register 0 (PFDR0) ... 1726
32.2.23 Port F Port Register 0 (PFPR0) ... 1728
32.2.24 Port G Control Register 0 to 7 (PGCR0 to PGCR7) ... 1729
32.2.25 Port G I/O Registers 0, 1 (PGIOR0, PGIOR1) ... 1739
32.2.26 Port G Data Register 0, 1 (PGDR0, PGDR1) ... 1739
32.2.27 Port G Port Register 0, 1 (PGPR0, PGPR1) ... 1742
32.2.28 Port H Control Register 0, 1 (PHCR0, PHCR1) ... 1744
32.2.29 Port H Port Register 0 (PHPR0) ... 1747
32.2.30 Port J Control Register 0 to 2 (PJCR0 to PJCR2) ... 1748
32.2.31 Port J I/O register 0 (PJIOR0) ... 1751
32.2.32 Port J Data Register 0 (PJDR0) ... 1752
32.2.33 Port J Port Register 0 (PJPR0) ... 1753
32.2.34 Port K Control Register 0 to 2 (PKCR0 to PKCR2: Available Only in the SH7264 Group) ... 1754
32.2.35 Port K I/O Register 0 (PKIOR0: Available Only in the SH7264 Group) ... 1758
32.2.36 Port K Data Register 0 (PKDR0: Available Only in the SH7264 Group) ... 1758
32.2.37 Port K Port Register 0 (PKPR0: Available Only in the SH7264 Group) ... 1760
32.3 Usage Notes ... 1760
32.3.1 Notes on Output Function from PE0 to PE5 ... 1760

Section 33 Power-Down Modes ... 1761
33.1 Features ... 1761
33.1.1 Power-Down Modes ... 1761
33.2 Register Descriptions ... 1764
33.2.1 Standby Control Register 1 (STBCR1) ... 1765
33.2.2 Standby Control Register 2 (STBCR2) ... 1766
33.2.3 Standby Control Register 3 (STBCR3) ... 1767
33.2.4 Standby Control Register 4 (STBCR4) ... 1769
33.2.5 Standby Control Register 5 (STBCR5) ... 1772
33.2.6 Standby Control Register 6 (STBCR6) ... 1774
33.2.7 Standby Control Register 7 (STBCR7) ... 1776
33.2.8 Standby Control Register 8 (STBCR8) ... 1778
33.2.9 Software Reset Control Register (SWRSTCR) ... 1779
33.2.10 System Control Register 1 (SYSCR1) ... 1781
33.2.11 System Control Register 2 (SYSCR2) ... 1782
33.2.12 System Control Register 3 (SYSCR3) ... 1783
33.2.13 System Control Register 4 (SYSCR4) ... 1785
33.2.14 System Control Register 5 (SYSCR5) ... 1786

chanhvinh.com

33.2.15 On-Chip Data-Retention RAM Area Setting Register (RRAMKP) 1788
33.2.16 Deep Standby Control Register (DSCTR) 1790
33.2.17 Deep Standby Cancel Source Select Register (DSSSR) 1791
33.2.18 Deep Standby Cancel Edge Select Register (DSESR) 1793
33.2.19 Deep Standby Cancel Source Flag Register (DSFR) 1795
33.2.20 XTAL Crystal Oscillator Gain Control Register (XTALCTR) 1797
33.3 Operation 1798
33.3.1 Sleep Mode 1798
33.3.2 Software Standby Mode 1799
33.3.3 Software Standby Mode Application Example 1802
33.3.4 Deep Standby Mode 1803
33.3.5 Module Standby Function 1809
33.3.6 Adjustment of XTAL Crystal Oscillator Gain 1809
33.4 Usage Notes 1810
33.4.1 Usage Notes on Setting Registers 1810
33.4.2 Usage Notes when the Realtime Clock is not Used 1810

Section 34 User Debugging Interface 1811
34.1 Features 1811
34.2 Input/Output Pins 1812
34.3 Description of the Emulation Command Registers 1813
34.3.1 Bypass Register (SDBPR) 1813
34.3.2 Instruction Register (SDIR) 1813
34.3.3 Enable Register (SDENR) 1815
34.4 Operation 1816
34.4.1 TAP Controller 1816
34.4.2 Reset Configuration 1817
34.4.3 TDO Output Timing 1817
34.4.4 User Debugging Interface Reset 1818
34.4.5 User Debugging Interface Interrupt 1818
34.5 Usage Notes 1819

Section 35 Motor Control PWM Timer 1821
35.1 Features 1821
35.2 Input/Output Pins 1823
35.3 Register Descriptions 1824
35.3.1 PWM Control Register_n (PWCR_n) (n = 1, 2) 1825
35.3.2 PWM Polarity Register_n (PWPR_n) (n = 1, 2) 1826
35.3.3 PWM Counter_n (PWCNT_n) (n = 1, 2) 1827
35.3.4 PWM Cycle Register_n (PWCYR_n) (n = 1, 2) 1827

chanhvinh.com

35.3.5 PWM Duty Registers_nA, nC, nE, nG (PWDTR_nA, PWDTR_nC, PWDTR_nE, PWDTR_nG) (n = 1, 2) 1828
35.3.6 PWM Buffer Registers_nA, nC, nE, nG (PWBFR_nA, PWBFR_nC, PWBFR_nE, PWBFR_nG) (n = 1, 2) 1831
35.3.7 PWM Buffer Transfer Control Register (PWBTCR) 1832
35.4 Bus Master Interface 1833
35.4.1 16-Bit Data Registers 1833
35.4.2 8-Bit Data Registers 1833
35.5 Operation 1834
35.5.1 PWM Operation 1834
35.5.2 Buffer Transfer Control 1835
35.6 Usage Note 1836
35.6.1 Conflict between Buffer Register Write and Compare Match 1836

Section 36 List of Registers 1837
36.1 Register Addresses (by functional module, in order of the corresponding section numbers) 1838
36.2 Register Bits 1872
36.3 Register States in Each Operating Mode 1948

Section 37 Electrical Characteristics 1953
37.1 Absolute Maximum Ratings 1953
37.2 Power-On/Power-Off Sequence 1954
37.3 DC Characteristics 1955
37.4 AC Characteristics 1966
37.4.1 Clock Timing 1966
37.4.2 Control Signal Timing 1971
37.4.3 Bus Timing 1973
37.4.4 Direct Memory Access Controller Timing 2007
37.4.5 Multi-Function Timer Pulse Unit 2 Timing 2008
37.4.6 Watchdog Timer Timing 2009
37.4.7 Serial Communication Interface with FIFO Timing 2010
37.4.8 Renesas Serial Peripheral Interface Timing 2011
37.4.9 I^2C Bus Interface 3 Timing 2015
37.4.10 Serial Sound Interface Timing 2016
37.4.11 Serial I/O with FIFO Timing 2018
37.4.12 A/D Converter Timing 2020
37.4.13 NAND Type Flash Memory Controller Timing 2021
37.4.14 USB 2.0 Host/Function Module Timing 2026
37.4.15 Video Display Controller 3 Timing 2029

chanhvinh.com

37.4.16 SD Host Interface Timing 2031
37.4.17 General Purpose I/O Ports Timing 2032
37.4.18 User Debugging Interface Timing 2033
37.4.19 Motor Control PWM Timer 2035
37.4.20 AC Characteristics Measurement Conditions 2036
37.5 A/D Converter Characteristics 2037

Section 38 States and Handling of Pins 2039
38.1 Pin States 2039
38.2 Treatment of Unused Pins 2049
38.3 Handling of Pins in Deep Standby Mode 2051
38.4 Recommended Combination of Bypass Capacitor 2053

Appendix 2055
A. Package Dimensions 2055

Main Revisions for This Edition 2057

Index 2109

chanhvinh.com

chanhvinh.com

Section 1 Overview

1.1 SH7262/7264 Features

This LSI is a single-chip RISC (reduced instruction set computer) microcontroller that includes a Renesas-original RISC CPU as its core, and the peripheral functions required to configure a system.

The CPU in this LSI is an SH-2A CPU, which provides upward compatibility for SH-1, SH-2, and SH-2E CPUs at object code level. It has a RISC-type instruction set, superscalar architecture, and Harvard architecture, for superior rates of instruction execution. In addition, the 32-bit internal-bus architecture that is independent from the direct memory access controller enhances data processing power. This CPU brings the user the ability to set up high-performance systems with strong functionality at less expense than was achievable with previous microcontrollers, and is even able to handle realtime control applications requiring high-speed characteristics.

This LSI has a floating-point unit and cache. In addition, this LSI includes on-chip peripheral functions necessary for system configuration, such as a 64-Kbyte RAM for high-speed operation, a 1-Mbyte or 640-Kbyte large-capacity RAM (32-Kbytes for 1-Mbyte and 320-Kbytes for 640-Kbyte versions are shared by the data-retention RAM), multi-function timer pulse unit 2, compare match timer, realtime clock, serial communication interface with FIFO, I^2C bus interface 3, serial sound interface, serial I/O with FIFO, controller area network interface*[2], IEBus™*[1] controller*[2], Renesas SPDIF interface, CD-ROM decoder , A/D converter, NAND flash memory controller, USB 2.0 host/function, video display controller 3, sampling rate converter, SD host interface, decompression unit, motor control PWM timer, and interrupt controller modules, and general I/O ports.

This LSI also provides an external memory access support function to enable direct connection to various memory devices or peripheral LSIs. These on-chip functions significantly reduce costs of designing and manufacturing application systems.

The features of this LSI are listed in table 1.1.

Notes: 1. IEBus (Inter Equipment Bus) is a trademark of Renesas Electronics Corporation.
2. This module is included or not depending on the product code.

chanhvinh.com

Table 1.1 SH7262/7264 Features

Items	Specification
CPU	• Renesas original SuperH architecture • Compatible with SH-1, SH-2, and SH-2E at object code level • 32-bit internal data bus • Support of an abundant register-set — Sixteen 32-bit general registers — Four 32-bit control registers — Four 32-bit system registers — Register bank for high-speed response to interrupts • RISC-type instruction set (upward compatible with SH series) — Instruction length: 16-bit fixed-length basic instructions for improved code efficiency and 32-bit instructions for high performance and usability — Load/store architecture — Delayed branch instructions — Instruction set based on C language • Superscalar architecture to execute two instructions at one time including a floating-point unit • Instruction execution time: Up to two instructions/cycle • Address space: 4 Gbytes • Internal multiplier • Five-stage pipeline • Harvard architecture

chanhvinh.com

Items	Specification
Floating-point unit	• Floating-point co-processor included • Supports single-precision (32-bit) and double-precision (64-bit) • Supports data type and exceptions that conforms to IEEE754 standard • Two rounding modes: Round to nearest and round to zero • Two denormalization modes: Flush to zero • Floating-point registers — Sixteen 32-bit floating-point registers (single-precision × 16 words or double-precision × 8 words) — Two 32-bit floating-point system registers • Supports FMAC (multiplication and accumulation) instructions • Supports FDIV (division) and FSQRT (square root) instructions • Supports FLDI0/FLDI1 (load constant 0/1) instructions • Instruction execution time — Latency (FMAC/FADD/FSUB/FMUL): Three cycles (single-precision), eight cycles (double-precision) — Pitch (FMAC/FADD/FSUB/FMUL): One cycle (single-precision), six cycles (double-precision) Note: FMAC only supports single-precision • Five-stage pipeline
Cache memory	• Instruction cache: 8 Kbytes • Operand cache: 8 Kbytes • 128-entry/way, 4-way set associative, 16-byte block length configuration each for the instruction cache and operand cache • Write-back, write-through, LRU replacement algorithm • Way lock function available (only for operand cache); ways 2 and 3 can be locked
Interrupt controller	• Seventeen external interrupt pins (NMI, IRQ7 to IRQ0, and PINT7 to PINT0) • On-chip peripheral interrupts: Priority level set for each module • 16 priority levels available • Register bank enabling fast register saving and restoring in interrupt processing

chanhvinh.com

Items	Specification
Bus state controller	• Address space divided into seven areas (0 to 6), each a maximum of 64 Mbytes • The following features settable for each area independently — Bus size (8 or 16 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 are also available. — PCMCIA interface — Outputs a chip select signal ($\overline{CS0}$ to $\overline{CS6}$) 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 in the SH7262 Group, and for two of them in the SH7264 Group. • Can be activated by on-chip peripheral modules • Burst mode and cycle steal mode • Intermittent mode available (16 and 64 cycles 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 12 (max.) by the internal PLL circuit • Three types of clocks generated: — CPU clock: Maximum 144 MHz — Bus clock: Maximum 72 MHz — Peripheral clock: Maximum 36 MHz
Watchdog timer	• On-chip one-channel watchdog timer • A counter overflow can reset the LSI

chanhvinh.com

Items	Specification
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
Multi-function timer pulse unit 2	• Maximum 16 lines of pulse inputs/outputs based on fix 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
Compare match timer	• Two-channel 16-bit counters • Four types of clock can be selected (Pϕ/8, Pϕ/32, Pϕ/128, and Pϕ/512) • 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 1/256 s by the 32.768-kHz on-chip crystal oscillator

chanhvinh.com

Items	Specification
Serial communication interface with FIFO	• Eight channels • Clocked synchronous or asynchronous mode selectable (SH7262: channels 0 to 2, SH7264: channels 0 to 3) • Simultaneous transmission and reception (full-duplex communication) supported • Dedicated baud rate generator • Separate 16-byte FIFO registers for transmission and reception • Modem control function (SH7262: channel 1, SH7264: channels 1 and 3, in asynchronous mode)
Renesas serial peripheral interface	• Two 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: 36 Mbps
I^2C bus interface 3	• Three channels • Master mode and slave mode supported
Serial sound interface	• Four-channel bidirectional serial transfer • Duplex communication (channel 0) • Support of various real 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
Serial I/O with FIFO	• Support of 16-stage 32-bits FIFOs independently for transmission and reception • 8-bit monaural/16-bit monaural/16-bit stereo audio input and output • Connectable to linear, audio, or A-Law or μ-Law CODEC chip • Support of master and slave functions

chanhvinh.com

Items	Specification
Controller area network Note: This module is included or not depending on the product code.	• Two channels • TTCAN level 1 supports for all channels • BOSCH 2.0B active compatible • Buffer size: transmit/receive × 31, receive only × 1 • Two or more controller area network channels can be assigned to one bus to increase number of buffers with a granularity of 32 channels • 31 Mailboxes for transmission or reception
IEBus™ controller Note: This module is included or not depending on the product code.	• IEBus protocol control (layer 2) supported — Half-duplex asynchronous communications — Multi-master system — Broadcast communications function — Selectable mode (three types) with different transfer speeds • On-chip buffers (dual port RAM) for data transmission and reception that enable up to 128 bytes of consecutive transmit/reception (maximum number of transfer bytes in mode 2) • Operating frequency — 12 MHz, 12.58 MHz (1/2 divided clocks of Pϕ, AUDIO_X1, or AUDIO_X2.) — 18 MHz, 18.87 MHz (1/3 divided clocks of Pϕ, AUDIO_X1, or AUDIO_X2.) — 24 MHz (1/4 divided clocks of Pϕ, AUDIO_X1, or AUDIO_X2.) — 25.16 MHz (1/4 divided clocks of Pϕ) — 30 MHz, 31.45 MHz (1/5 divided clocks of Pϕ) — 36 MHz (1/6 divided clocks of Pϕ)
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	• 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.
NAND flash memory controller	• Direct-connected memory interface with NAND-type flash memory • Read/write in sectors • Two types of transfer modes: Command access mode and sector access mode (512-byte data + 16-byte management code: with ECC) • Interrupt request and DMA transfer request • Supports flash memory requiring 5-byte addresses (2 Gbits and more)
USB 2.0 host/function module	• 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

chanhvinh.com

Items	Specification
Video display controller 3	• Video input Input format: BT601, BT656 (NTSC/PAL) • Video recording Can be saved in RGB565 format, at 1/2 field rate. • Video processing Video scaling: 1/2, 1/3, 1/4 Horizontal scaling: 1/2, 1/3, 2/3, 1/4 (Each scaling value can be multiplied by 6/7 for PAL.) Adjustment of contrast and brightness • Graphic image 1 and 2 (two layers) Input format: RGB565 (16 bits), αRGB4444 (16 bits) • Overlap function α-blending window function: Mixes input image, layer 1, and layer 2 in a specified area according to the transmittance α (fade-in and face-out possible). Chromakey function: Mixes images according to the specified RGB color and transmittance α. Dot a function: Mixes αRGB4444-format graphic images according to the transmittance α. • Output image Resolution: VGA (640 × 480), WQVGA (480 × 240), QVGA (320 × 240), QVGA (240 × 320) Format: RGB565 (16 bits)
Sampling rate converter	• Data format: 32-bit stereo (16 bits each to L/R), 16-bit monaural for channel 0, and 16-bit monaural for channel 1 • Input sampling rate: 8/11.025/12/16/22.05/24/32/44.1/48kHz (channel 0), 44.1kHz (channel 1) • Output sampling rate: 44.1/48 kHz (channel 0), 8/16 kHz (channel 1)

chanhvinh.com

Items	Specification
SD host interface	• SD memory I/O card interface (1-/4-bits 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 detect function, write protect supported
Decompression unit	• Input data Data compressed with run-length encoding • Output data RAW data
General I/O ports	• SH7262: 89 I/Os, 6 inputs with open-drain outputs, and 4 inputs • SH7264: 115 I/Os, 6 inputs with open-drain outputs, and 8 inputs • Input or output can be selected for each bit
A/D converter	• 10-bit resolution • SH7262: four input channels, SH7264: eight input channels • A/D conversion request by the external trigger or timer trigger
Motor control PWM timer	• Two 10-bit PWM channels, each with eight outputs
User debugging interface	• E10A emulator support • JTAG-standard pin assignment
On-chip RAM	• 64-Kbyte memory for high-speed operation (16 Kbytes × 4) • 1-Mbyte or 640-Kbyte large capacity memory for video display/recording and work (32-Kbytes for 1-Mbyte and 320-Kbytes for 640-Kbyte versions are used for data retention) • 1-Mbyte version: 32-Kbyte memory for data retention (16 Kbytes × 2) • 640-Kbyte version: 320-Kbyte memory for data retention (16 Kbytes× 2, 128 Kbytes×1, 160 Kbytes×1)

chanhvinh.com

Items	Specification
Boot modes	• Four boot modes Boot mode 0: Booting from memory connected to $\overline{\text{CS0}}$ area Boot mode 1: Booting from a serial flash memory (high-speed communication) Boot mode 2: Booting from a NAND flash memory Boot mode 3: Booting from a serial flash memory (low-speed communication)
Power supply voltage	• Vcc: 1.1 to 1.3 V • PVcc: 3.0 to 3.6 V
Packages	SH7262 • 176-pin QFP, 24-mm square, 0.5-mm pitch JEITA package code: P-LQFP176-24 × 24-0.50 Renesas code: PLQP0176KB-A SH7264 • 208-pin QFP, 28-mm square, 0.5-mm pitch JEITA package code: P-LQFP208-28 × 28-0.50 Renesas code: PLQP0208KB-A

chanhvinh.com

1.2 Product Lineup

Table 1.2 Product Lineup

Product Classification	Product Code		IEBus Controller	Controller Area Network	Operating Temperature	Package
SH7262 Group	R5S72620W144FPU	1 Mbyte*[1]	Not included	Not included	Regular specifications (-20 to +85 °C)	176-pin QFP
	R5S72621W144FPU		Not included	Included		
	R5S72622W144FPU		Included	Not included		
	R5S72623W144FPU		Included	Included		
	R5S72620P144FPU		Not included	Not included	Wide-range specifications (-40 to +85 °C)	
	R5S72621P144FPU		Not included	Included		
	R5S72622P144FPU		Included	Not included		
	R5S72623P144FPU		Included	Included		
	R5S72624W144FPU	640 Kbyte*[2]	Not included	Not included	Regular specifications (-20 to +85 °C)	
	R5S72625W144FPU		Not included	Included		
	R5S72626W144FPU		Included	Not included		
	R5S72627W144FPU		Included	Included		
	R5S72624P144FPU		Not included	Not included	Wide-range specifications (-40 to +85 °C)	
	R5S72625P144FPU		Not included	Included		
	R5S72626P144FPU		Included	Not included		
	R5S72627P144FPU		Included	Included		

chanhvinh.com

Product Classification	Product Code		IEBus Controller	Controller Area Network	Operating Temperature	Package
SH7264 Group	R5S72640W144FPU	1 Mbyte*[1]	Not included	Not included	Regular specifications (-20 to +85 °C)	208-pin QFP
	R5S72641W144FPU		Not included	Included		
	R5S72642W144FPU		Included	Not included		
	R5S72643W144FPU		Included	Included		
	R5S72640P144FPU		Not included	Not included	Wide-range specifications (-40 to +85 °C)	
	R5S72641P144FPU		Not included	Included		
	R5S72642P144FPU		Included	Not included		
	R5S72643P144FPU		Included	Included		
	R5S72644W144FPU	640 Kbyte*[2]	Not included	Not included	Regular specifications (-20 to +85 °C)	
	R5S72645W144FPU		Not included	Included		
	R5S72646W144FPU		Included	Not included		
	R5S72647W144FPU		Included	Included		
	R5S72644P144FPU		Not included	Not included	Wide-range specifications (-40 to +85 °C)	
	R5S72645P144FPU		Not included	Included		
	R5S72646P144FPU		Included	Not included		
	R5S72647P144FPU		Included	Included		

Notes: 1. Hereinafter referred to as "1-Mbyte version" in this manual.
2. Hereinafter referred to as "640-Kbyte version" in this manual.

chanhvinh.com

1.3 Block Diagram

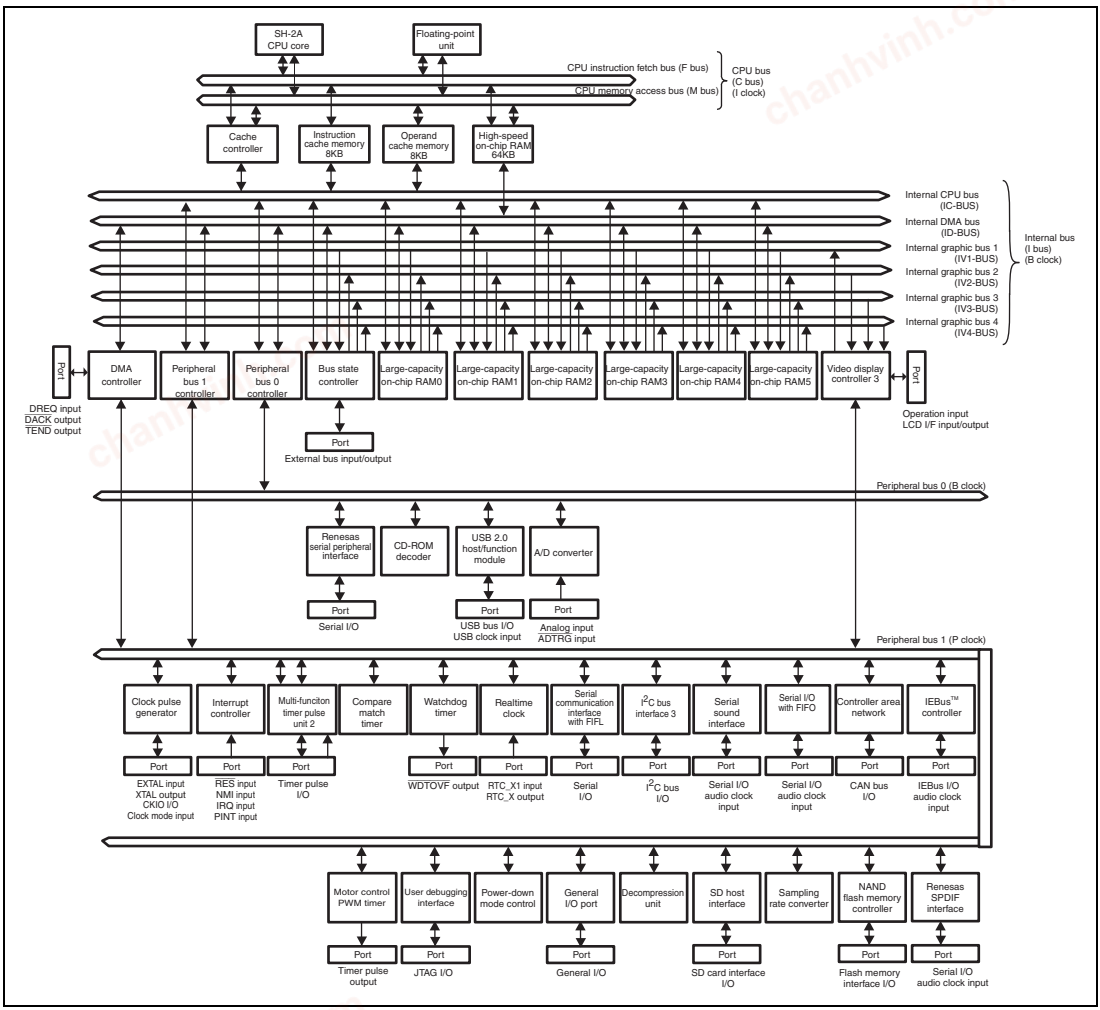


Figure 1.1 Block Diagram

chanhvinh.com

1.4 Pin Assignment

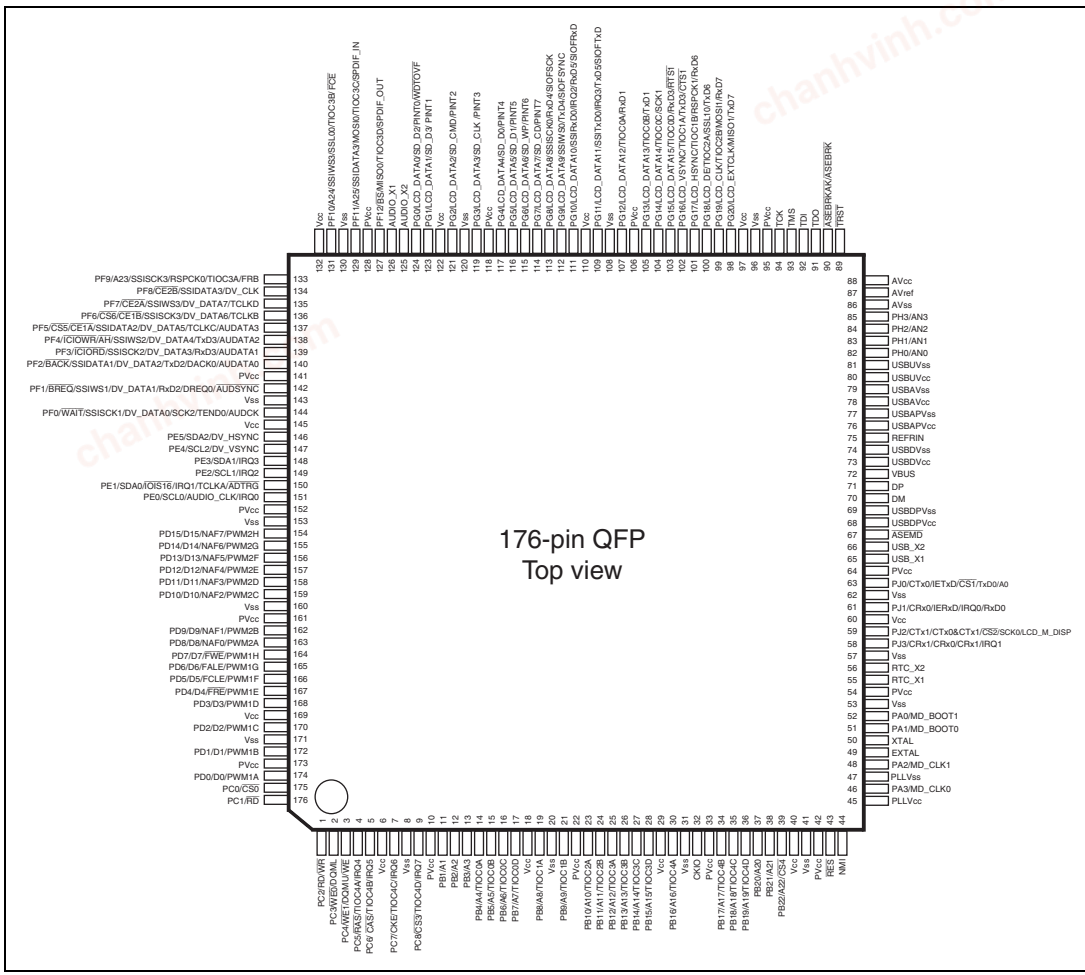


Figure 1.2 (1) Pin Assignment for the SH7262 Group (1-Mbyte Version)

chanhvinh.com

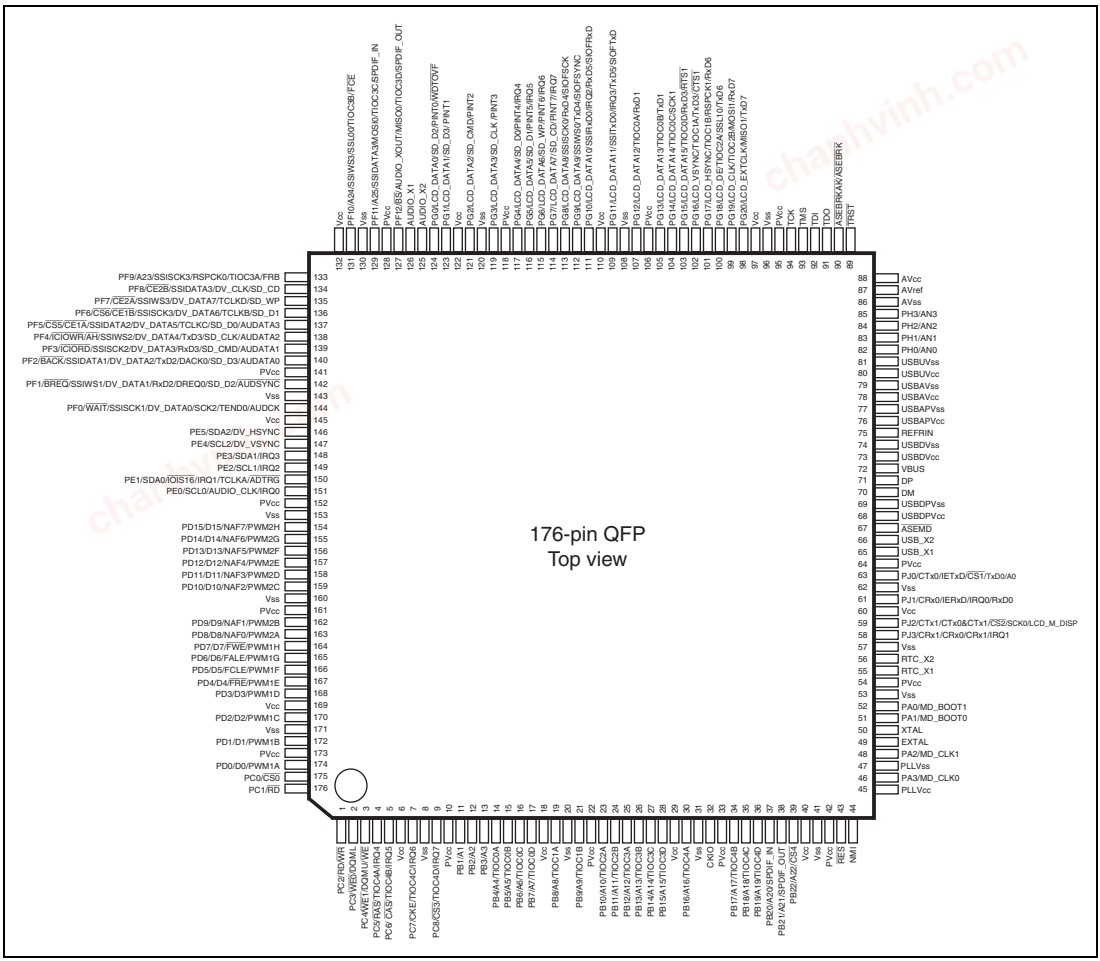


Figure 1.2 (2) Pin Assignment for the SH7262 Group (640-Kbyte Version)

chanhvinh.com

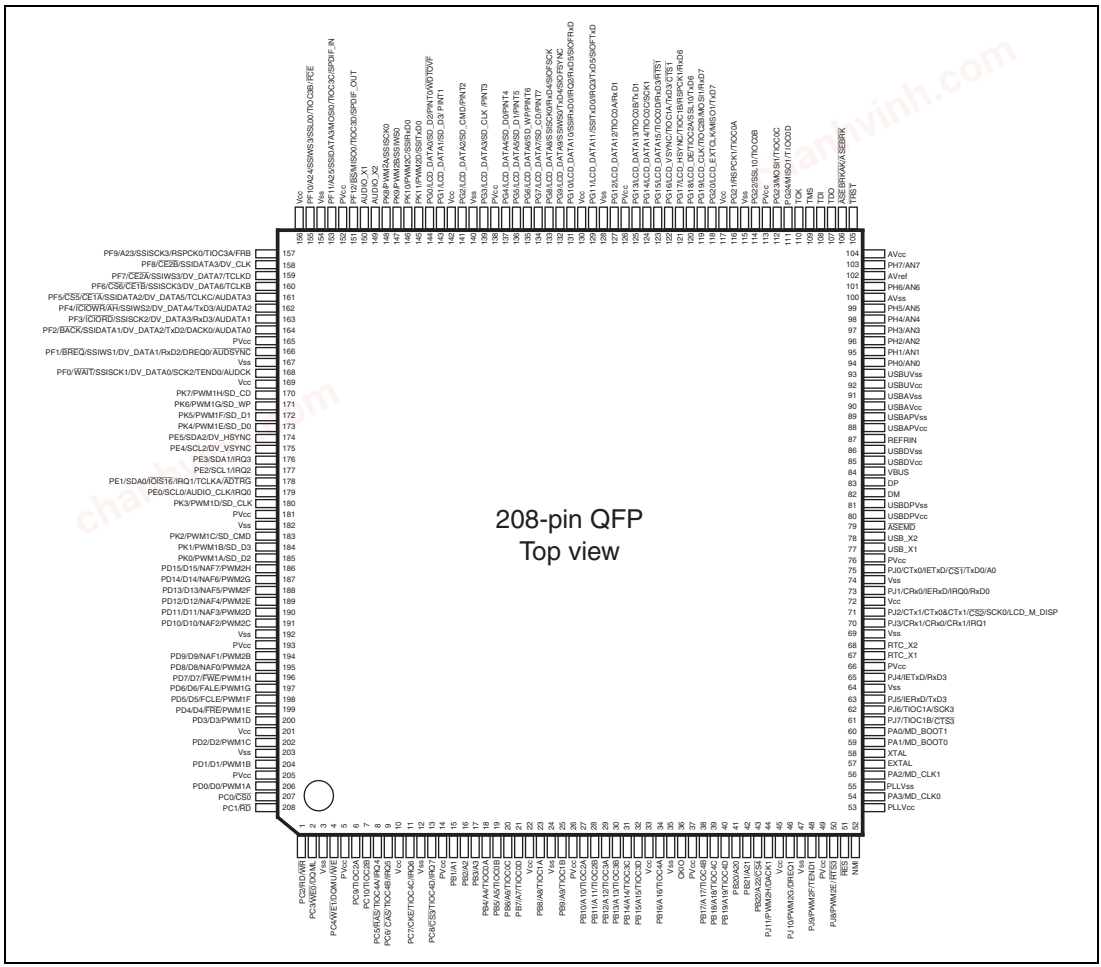


Figure 1.2 (3) Pin Assignment for the SH7264 Group (1-Mbyte Version)

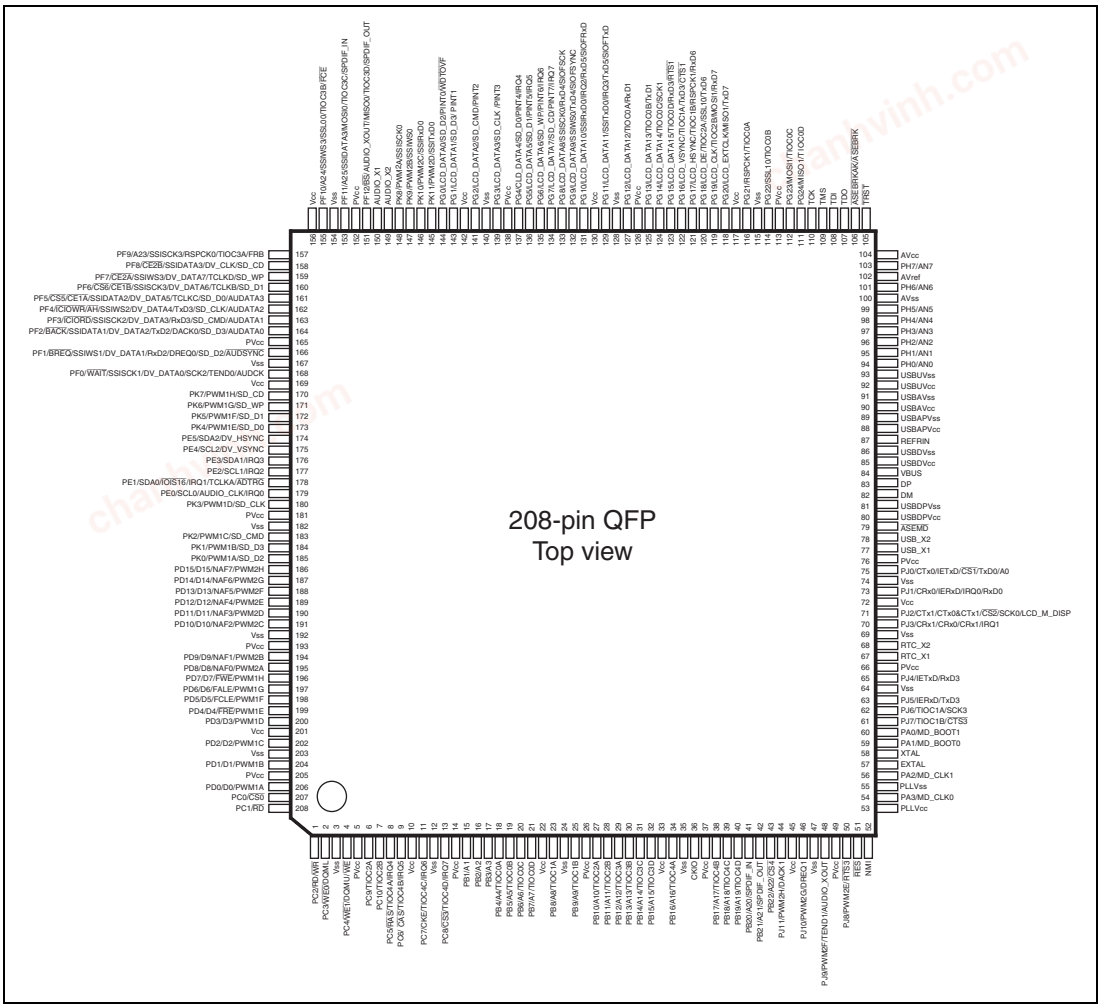


Figure 1.2 (4) Pin Assignment for the SH7264 Group (640-Kbyte Version)

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.
	PLLVss	I	Ground for PLL	Ground pin 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 I/O	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.

chanhvinh.com

Classification	Symbol	I/O	Name	Function
Clock	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 I/O	Output for the on-chip crystal oscillator on AUDIO_X1 or the external clock signal.
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_CLK1, MD_CLK0	I	Clock mode set	These pins set the clock 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.
	$\overline{\text{ASEMD}}$	I	ASE mode	If a low level is input at the $\overline{\text{ASEMD}}$ pin while the $\overline{\text{RES}}$ pin is asserted, ASE mode is entered; if a high level is input, product chip mode is entered. In ASE mode, the E10A-USB emulator function is enabled. When this function is not in use, fix it high.
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.
	$\overline{\text{BREQ}}$	I	Bus-mastership request	A low level is input to this pin when an external device requests the release of the bus mastership.
	$\overline{\text{BACK}}$	O	Bus-mastership request acknowledge	Indicates that the bus mastership has been released to an external device. Reception of the $\overline{\text{BACK}}$ signal informs the device which has output the $\overline{\text{BREQ}}$ signal that it has acquired the bus.

chanhvinh.com

Classification	Symbol	I/O	Name	Function
Interrupts	NMI	I	Non-maskable interrupt	Non-maskable interrupt request pin. 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.
	PINT7 to PINT0	I	Interrupt requests 7 to 0	Maskable interrupt request pins. Only level-input detection can be selected.
Address bus	A25 to A0	O	Address bus	Outputs addresses.
Data bus	D15 to D0	I/O	Data bus	Bidirectional data bus.
Bus control	$\overline{CS6}$ to $\overline{CS0}$	O	Chip select 6 to 0	Chip-select signals for external memory or devices.
	$\overline{RD}$	O	Read	Indicates that data is read from an external device.
	RD/$\overline{WR}$	O	Read/write	Read/write signal.
	$\overline{BS}$	O	Bus start	Bus-cycle start signal.
	$\overline{AH}$	O	Address hold	Address hold timing signal for the device that uses the address/data-multiplexed bus.
	$\overline{WAIT}$	I	Wait	Inserts a wait cycle into the bus cycles during access to the external space.
	$\overline{WE0}$	O	Byte select	Indicates a write access to bits 7 to 0 of data of external memory or device.
	$\overline{WE1}$	O	Byte select	Indicates a write access to bits 15 to 8 of data of external memory or device.
	DQML	O	Byte select	Selects bits D7 to D0 when SDRAM is connected.
	DQMU	O	Byte select	Selects bits D15 to D8 when SDRAM is connected.

chanhvinh.com

Classification	Symbol	I/O	Name	Function
Bus control	$\overline{RAS}$	O	RAS	Connected to the $\overline{RAS}$ pin when SDRAM is connected.
	$\overline{CAS}$	O	CAS	Connected to the $\overline{CAS}$ pin when SDRAM is connected.
	CKE	O	CK enable	Connected to the CKE pin when SDRAM is connected.
	$\overline{CE1A}$, $\overline{CE1B}$	O	Lower byte select for PCMCIA card	Connected to PCMCIA card select signals D7 to D0.
	$\overline{CE2A}$, $\overline{CE2B}$	O	Upper byte select for PCMCIA card	Connected to PCMCIA card select signals D15 to D8.
	$\overline{ICIOWR}$	O	Write strobe for PCMCIA	Connected to the PCMCIA I/O write strobe signal.
	$\overline{ICIORD}$	O	Read strobe for PCMCIA	Connected to the PCMCIA I/O read strobe signal.
	$\overline{WE}$	O	Write strobe for PCMCIA memory	Connected to the PCMCIA memory write strobe signal.
	$\overline{IOIS16}$	I	PCMCIA dynamic bus sizing	Indicates 16-bit I/O of the PCMCIA.
Direct memory access controller	DREQ1, DREQ0	I	DMA-transfer request	Input pins to receive external requests for DMA transfer. Only DREQ0 can be used in the SH7262 Group.
	DACK1, DACK0	O	DMA-transfer request accept	Output pins for signals indicating acceptance of external requests from external devices. Only DACK0 can be used in the SH7262 Group.
	TEND1, TEND0	O	DMA-transfer end output	Output pins for DMA transfer end. Only TEND0 can be used in the SH7262 Group.

chanhvinh.com

Classification	Symbol	I/O	Name	Function
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	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 and TGRB_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	TxD7 to TxD0	O	Transmit data	Data output pins.
	RxD7 to RxD0	I	Receive data	Data input pins.
	SCK3 to SCK0	I/O	Serial clock	Clock input/output pins.
	$\overline{\text{RTS3}}$, $\overline{\text{RTS1}}$	O	Transmit request	Modem control pin. Only $\overline{\text{RTS1}}$ can be used in the SH7262 Group.
	$\overline{\text{CTS3}}$, $\overline{\text{CTS1}}$	I	Transmit enable	Modem control pin. Only $\overline{\text{CTS1}}$ can be used in the SH7262 Group.

chanhvinh.com

Classification	Symbol	I/O	Name	Function
Renesas serial peripheral interface	MOSI1, MOSI0	I/O	Data	Data I/O pin.
	MISO1, MISO0	I/O	Data	Data I/O pin.
	RSPCK1, RSPCK0	I/O	Clock	Clock I/O pin.
	SSL10, SSL00	I/O	Slave select	Slave select I/O pin.
I^2C bus interface 3	SCL2 to SCL0	I/O	Serial clock pin	Serial clock I/O pin.
	SDA2 to SDA0	I/O	Serial data pin	Serial data I/O pin.
Serial sound interface	SSITxD0	O	Data output	Serial data output pin.
	SSIRxD0	I	Data input	Serial data input pin.
	SSIDATA3 to SSIDATA1	I/O	Data I/O	Serial data I/O pin.
	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.
Serial I/O with FIFO	SIOFTxD	O	Data output	Data output pin.
	SIOFRxD	I	Data input	Data input pin.
	SIOFSCK	I/O	I/O clock	Clock I/O pin.
	SIOFSYNC	I/O	I/O chip select	I/O pin for chip selection.
Controller area network	CTx0, CTx1	O	CAN bus transmit data	Output pin for transmit data on the CAN bus.
	CRx0, CRx1	I	CAN bus receive data	Output pin for receive data on the CAN bus.
IEBus™ controller	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.

chanhvinh.com

Classification	Symbol	I/O	Name	Function
NAND flash memory controller	FALE	O	Flash memory address latch enable	Asserted for address output and negated for data I/O. Negated at data input/output.
	$\overline{\text{FRE}}$	O	Flash memory read enable	Reads data at falling edge.
	$\overline{\text{FCE}}$	O	Flash memory chip enable	Enables the flash memory connected to this LSI.
	FCLE	O	Flash memory command latch enable	Asserted at command output.
	FRB	I	Flash memory ready/busy	High level indicates ready state and low level indicates busy state.
	$\overline{\text{FWE}}$	O	Flash memory write enable	Flash memory latches commands, addresses, and data at rising edge.
	NAF7 to NAF0	I/O	Flash memory data	Data I/O pins.
USB 2.0 host/function module	DP	I/O	USB 2.0 host/function module D+ data	D+ data pin for USB 2.0 host/function module bus.
	DM	I/O	USB 2.0 host/function module D– data	D– data pin for USB 2.0 host/function module bus.
	VBUS	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.
	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.

chanhvinh.com

Classification	Symbol	I/O	Name	Function
USB 2.0 host/function module	USBAPVss	I	Ground for transceiver analog pins	Ground for pins.
	USBDPVcc	I	Power supply for transceiver digital pins	Power supply for pins.
	USBDPVss	I	Ground for transceiver digital pins	Ground for pins.
	USBAVcc	I	Power supply for transceiver analog core	Power supply for core.
	USBAVss	I	Ground for transceiver analog core	Ground for core.
	USBDVcc	I	Power supply for transceiver digital core	Power supply for core.
	USBDVss	I	Ground for transceiver digital core	Ground for core.
	USBUVcc	I	480-MHz power supply for USB 2.0 host/function module	Power supply for 480-MHz sections
	USBUVss	I	480-MHz ground for USB 2.0 host/function module	Ground for 480-MHz sections

chanhvinh.com

Classification	Symbol	I/O	Name	Function
Video display controller 3	LCD_DATA15 to LCD_DATA0	O	Output data	Data output pin for panel.
	LCD_DE	O	Data enable	Data enable output pin for panel.
	LCD_VSYNC	O	VSYNC output	VSYNC output pin for panel.
	LCD_HSYNC	O	HSYNC output	HSYNC output pin for panel.
	LCD_CLK	O	Panel clock	Panel clock output pin.
	LCD_EXTCLK	I	Panel clock source	Panel clock source input pin.
	LCD_M_DISP	O	LCD current alternation	LCD current alternating signal pin.
	DV_DATA7 to DV_DATA0	I	Input data	Data input pin for BT.601 and BT.656.
	DV_VSYNC	I	VSYNC input	VSYNC input pin.
	DV_HSYNC	I	HSYNC input	HSYNC input pin.
	DV_CLK	I	Input clock	Clock input signal pin for BT.601 and BT.656.
SD host interface	SD_CLK	O	SD clock	Output pin for SD clock.
	SD_CMD	I/O	SD command	SD command output and response input signal.
	SD_D3 to SD_D0	I/O	SD data	SD data bus signal.
	SD_CD	I	SD card detection	SD card detection.
	SD_WP	I	SD write protection	SD write protection signal.
A/D converter	AN7 to AN0	I	Analog input pins	Analog input pins. Only AN3 to AN0 can be used in the SH7262 Group.
	$\overline{\text{ADTRG}}$	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.

chanhvinh.com

Classification	Symbol	I/O	Name	Function
General I/O ports	PA3 to PA0, PB22 to PB1, PC10 to PC0, PD15 to PD0, PF12 to PF0, PG24 to PG0, PJ11 to PJ0, PK11 to PK0	I/O	General port	89 general I/O port pins in the SH7262 Group. 115 general I/O port pins in the SH7264 Group. Only PA3 to PA0, PB22 to PB1, PC8 to PC0, PD15 to PD0, PF12 to PF0, PG20 to PG0, and PJ3 to PJ0 can be used in the SH7262 Group.
	PE5 to PE0	I/O	General port	6 input port pins with open-drain output.
	PH7 to PH0	I	General port	4 general input port pins in the SH7262 Group. 8 general input ports pin in the SH7264 Group. Only PH3 to PH0 can be used in the SH7262 Group.
Motor control PWM timer	PWM1H to PWM1A, PWM2H to PWM2A	O	Timer output	PWM output pins.
User debugging interface	TCK	I	Test clock	Test-clock input pin.
	TMS	I	Test mode select	Test-mode select signal input pin.
	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.
Emulator interface	AUDATA3 to AUDATA0	O	Data	Branch source or destination address output pins.
	AUDCK	O	Clock	Sync-clock output pin.
	$\overline{\text{AUDSYNC}}$	O	Sync signal	Data start-position acknowledge-signal output pin.
	$\overline{\text{ASEBRKAK}}$	O	Break mode acknowledge	Indicates that the E10A-USB emulator has entered its break mode.
	$\overline{\text{ASEBRK}}$	I	Break request	E10A-USB emulator break input pin.

chanhvinh.com

1.6 List of Pins

Table 1.4 List of Pins

SH7262 Pin No.	SH7264 Pin No.	Function 1 Symbol	Function 1 I/O	Function 2 Symbol	Function 2 I/O	Function 3 Symbol	Function 3 I/O	Function 4 Symbol	Function 4 I/O
1	1	PC2	I(s)/O	RD/$\overline{WR}$	O	—	—	—	—
2	2	PC3	I(s)/O	$\overline{WE0}$/DQML	O	—	—	—	—
—	3	Vss							
3	4	PC4	I(s)/O	$\overline{WE1}$/DQMU/$\overline{WE}$	O	—	—	—	—
—	5	PVcc							
—	6	PC9	I(s)/O	TIOC2A	I(s)/O	—	—	—	—
—	7	PC10	I(s)/O	TIOC2B	I(s)/O	—	—	—	—
4	8	PC5	I(s)/O	$\overline{RAS}$	O	TIOC4A	I(s)/O	IRQ4	I(s)
5	9	PC6	I(s)/O	$\overline{CAS}$	O	TIOC4B	I(s)/O	IRQ5	I(s)
6	10	Vcc							
7	11	PC7	I(s)/O	CKE	O	TIOC4C	I(s)/O	IRQ6	I(s)
8	12	Vss							
9	13	PC8	I(s)/O	$\overline{CS3}$	O	TIOC4D	I(s)/O	IRQ7	I(s)

SH7262 Pin No.	SH7264 Pin No.	Function 5 Symbol	Function 5 I/O	Function 6 Symbol	Function 6 I/O	Function 7 Symbol	Function 7 I/O	Figure 1.3
1	1	—	—	—	—	—	—	(7)
2	2	—	—	—	—	—	—	(7)
—	3							
3	4	—	—	—	—	—	—	(7)
—	5							
—	6	—	—	—	—	—	—	(7)
—	7	—	—	—	—	—	—	(7)
4	8	—	—	—	—	—	—	(7)
5	9	—	—	—	—	—	—	(7)
6	10							
7	11	—	—	—	—	—	—	(7)
8	12							
9	13	—	—	—	—	—	—	(7)

chanhvinh.com

SH7262 Pin No.	SH7264 Pin No.	Function 1 Symbol	Function 1 I/O	Function 2 Symbol	Function 2 I/O	Function 3 Symbol	Function 3 I/O	Function 4 Symbol	Function 4 I/O
10	14	PVcc							
11	15	PB1	I(s)/O	A1	O	—	—	—	—
12	16	PB2	I(s)/O	A2	O	—	—	—	—
13	17	PB3	I(s)/O	A3	O	—	—	—	—
14	18	PB4	I(s)/O	A4	O	TIOC0A	I(s)/O	—	—
15	19	PB5	I(s)/O	A5	O	TIOC0B	I(s)/O	—	—
16	20	PB6	I(s)/O	A6	O	TIOC0C	I(s)/O	—	—
17	21	PB7	I(s)/O	A7	O	TIOC0D	I(s)/O	—	—
18	22	Vcc							
19	23	PB8	I(s)/O	A8	O	TIOC1A	I(s)/O	—	—
20	24	Vss							
21	25	PB9	I(s)/O	A9	O	TIOC1B	I(s)/O	—	—
22	26	PVcc							
23	27	PB10	I(s)/O	A10	O	TIOC2A	I(s)/O	—	—

SH7262 Pin No.	SH7264 Pin No.	Function 5 Symbol	Function 5 I/O	Function 6 Symbol	Function 6 I/O	Function 7 Symbol	Function 7 I/O	Figure 1.3
10	14							
11	15	—	—	—	—	—	—	(7)
12	16	—	—	—	—	—	—	(7)
13	17	—	—	—	—	—	—	(7)
14	18	—	—	—	—	—	—	(7)
15	19	—	—	—	—	—	—	(7)
16	20	—	—	—	—	—	—	(7)
17	21	—	—	—	—	—	—	(7)
18	22							
19	23	—	—	—	—	—	—	(7)
20	24							
21	25	—	—	—	—	—	—	(7)
22	26							
23	27	—	—	—	—	—	—	(7)

chanhvinh.com

SH7262 Pin No.	SH7264 Pin No.	Function 1 Symbol	Function 1 I/O	Function 2 Symbol	Function 2 I/O	Function 3 Symbol	Function 3 I/O	Function 4 Symbol	Function 4 I/O
24	28	PB11	I(s)/O	A11	O	TIOC2B	I(s)/O	—	—
25	29	PB12	I(s)/O	A12	O	TIOC3A	I(s)/O	—	—
26	30	PB13	I(s)/O	A13	O	TIOC3B	I(s)/O	—	—
27	31	PB14	I(s)/O	A14	O	TIOC3C	I(s)/O	—	—
28	32	PB15	I(s)/O	A15	O	TIOC3D	I(s)/O	—	—
29	33	Vcc							
30	34	PB16	I(s)/O	A16	O	TIOC4A	I(s)/O	—	—
31	35	Vss							
32	36	CKIO	O	—	—	—	—	—	—
33	37	PVcc							
34	38	PB17	I(s)/O	A17	O	TIOC4B	I(s)/O	—	—
35	39	PB18	I(s)/O	A18	O	TIOC4C	I(s)/O	—	—
36	40	PB19	I(s)/O	A19	O	TIOC4D	I(s)/O	—	—
37	41	PB20	I(s)/O	A20	O	SPDIF_IN*	I(s)	—	—

SH7262 Pin No.	SH7264 Pin No.	Function 5 Symbol	Function 5 I/O	Function 6 Symbol	Function 6 I/O	Function 7 Symbol	Function 7 I/O	Figure 1.3
24	28	—	—	—	—	—	—	(7)
25	29	—	—	—	—	—	—	(7)
26	30	—	—	—	—	—	—	(7)
27	31	—	—	—	—	—	—	(7)
28	32	—	—	—	—	—	—	(7)
29	33							
30	34	—	—	—	—	—	—	(7)
31	35							
32	36	—	—	—	—	—	—	(5)
33	37							
34	38	—	—	—	—	—	—	(7)
35	39	—	—	—	—	—	—	(7)
36	40	—	—	—	—	—	—	(7)
37	41	—	—	—	—	—	—	(7)

chanhvinh.com

SH7262 Pin No.	SH7264 Pin No.	Function 1 Symbol	Function 1 I/O	Function 2 Symbol	Function 2 I/O	Function 3 Symbol	Function 3 I/O	Function 4 Symbol	Function 4 I/O
38	42	PB21	I(s)/O	A21	O	SPDIF_OUT*	O	—	—
39	43	PB22	I(s)/O	A22	O	$\overline{CS4}$	O	—	—
—	44	PJ11	I(s)/O	PWM2H	O	DACK1	O	—	—
40	45	Vcc							
—	46	PJ10	I(s)/O	PWM2G	O	DREQ1	I(s)	—	—
41	47	Vss							
—	48	PJ9	I(s)/O	PWM2F	O	TEND1	O	AUDIO_XOUT	O
42	49	PVcc							
—	50	PJ8	I(s)/O	PWM2E	O	$\overline{RTS3}$	I(s)/O		—
43	51	$\overline{RES}$	I(s)						
44	52	NMI	I(s)						
45	53	PLLVcc							
46	54	PA3	I(s)/O	MD_CLK0	I(s)	—	—	—	—
47	55	PLLVss							

SH7262 Pin No.	SH7264 Pin No.	Function 5 Symbol	Function 5 I/O	Function 6 Symbol	Function 6 I/O	Function 7 Symbol	Function 7 I/O	Figure 1.3
38	42	—	—	—	—	—	—	(7)
39	43	—	—	—	—	—	—	(7)
—	44	—	—	—	—	—	—	(7)
40	45							
—	46	—	—	—	—	—	—	(7)
41	47							
—	48	—	—	—	—	—	—	(7)
42	49							
—	50	—	—	—	—	—	—	(7)
43	51							(1)
44	52							(1)
45	53							
46	54	—	—	—	—	—	—	(7)
47	55							

chanhvinh.com

SH7262 Pin No.	SH7264 Pin No.	Function 1 Symbol	Function 1 I/O	Function 2 Symbol	Function 2 I/O	Function 3 Symbol	Function 3 I/O	Function 4 Symbol	Function 4 I/O
48	56	PA2	I(s)/O	MD_CLK1	I(s)	—	—	—	—
49	57	EXTAL	I						
50	58	XTAL	O						
51	59	PA1	I(s)/O	MD_BOOT0	I(s)	—	—	—	—
52	60	PA0	I(s)/O	MD_BOOT1	I(s)	—	—	—	—
—	61	PJ7	I(s)/O	TIOC1B	I(s)/O	$\overline{\text{CTS3}}$	I(s)/O	—	—
—	62	PJ6	I(s)/O	TIOC1A	I(s)/O	SCK3	I(s)/O	—	—
—	63	PJ5	I(s)/O	IERxD	I(s)	TxD3	O	—	—
53	64	Vss							
—	65	PJ4	I(s)/O	IETxD	O	RxD3	I(s)	—	—
54	66	PVcc							
55	67	RTC_X1	I	—	—	—	—	—	—
56	68	RTC_X2	O	—	—	—	—	—	—
57	69	Vss							

SH7262 Pin No.	SH7264 Pin No.	Function 5 Symbol	Function 5 I/O	Function 6 Symbol	Function 6 I/O	Function 7 Symbol	Function 7 I/O	Figure 1.3
48	56	—	—	—	—	—	—	(7)
49	57							(9)
50	58							
51	59	—	—	—	—	—	—	(7)
52	60	—	—	—	—	—	—	(7)
—	61	—	—	—	—	—	—	(7)
—	62	—	—	—	—	—	—	(7)
—	63	—	—	—	—	—	—	(7)
53	64							
—	65	—	—	—	—	—	—	(7)
54	66							
55	67	—	—	—	—	—	—	(10)
56	68	—	—	—	—	—	—	
57	69							

chanhvinh.com

SH7262 Pin No.	SH7264 Pin No.	Function 1 Symbol	Function 1 I/O	Function 2 Symbol	Function 2 I/O	Function 3 Symbol	Function 3 I/O	Function 4 Symbol	Function 4 I/O
58	70	PJ3	I(s)/O	CRx1	I(s)	CRx0/CRx1	I(s)	IRQ1	I(s)
59	71	PJ2	I(s)/O	CTx1	O	CTx0&CTx1	O	$\overline{CS2}$	O
60	72	Vcc							
61	73	PJ1	I(s)/O	CRx0	I(s)	IERxD	I(s)	IRQ0	I(s)
62	74	Vss							
63	75	PJ0	I(s)/O	CTx0	O	IETxD	O	$\overline{CS1}$	O
64	76	PVcc							
65	77	USB_X1	I	—	—	—	—	—	—
66	78	USB_X2	O	—	—	—	—	—	—
67	79	$\overline{ASEMD}$	I(s)	—	—	—	—	—	—
68	80	USBDPVcc							
69	81	USBDPVss							
70	82	DM	I/O	—	—	—	—	—	—
71	83	DP	I/O	—	—	—	—	—	—

SH7262 Pin No.	SH7264 Pin No.	Function 5 Symbol	Function 5 I/O	Function 6 Symbol	Function 6 I/O	Function 7 Symbol	Function 7 I/O	Figure 1.3
58	70	—	—	—	—	—	—	(7)
59	71	SCK0	I(s)/O	LCD_M_DISP	O	—	—	(7)
60	72							
61	73	RxD0	I(s)/O	—	—	—	—	(7)
62	74							
63	75	TxD0	O	A0	O	—	—	(7)
64	76							
65	77	—	—	—	—	—	—	(9)
66	78	—	—	—	—	—	—	
67	79	—	—	—	—	—	—	(1)
68	80							
69	81							
70	82	—	—	—	—	—	—	
71	83	—	—	—	—	—	—	

chanhvinh.com

SH7262 Pin No.	SH7264 Pin No.	Function 1 Symbol	Function 1 I/O	Function 2 Symbol	Function 2 I/O	Function 3 Symbol	Function 3 I/O	Function 4 Symbol	Function 4 I/O
72	84	VBUS	I	—	—	—	—	—	—
73	85	USBDVcc							
74	86	USBDVss							
75	87	REFRIN	I	—	—	—	—	—	—
76	88	USBAPVcc							
77	89	USBAPVss							
78	90	USBAVcc							
79	91	USBAVss							
80	92	USBUVcc							
81	93	USBUVss							
82	94	PH0	I	AN0	I(a)	—	—	—	—
83	95	PH1	I	AN1	I(a)	—	—	—	—
84	96	PH2	I	AN2	I(a)	—	—	—	—
85	97	PH3	I	AN3	I(a)	—	—	—	—

SH7262 Pin No.	SH7264 Pin No.	Function 5 Symbol	Function 5 I/O	Function 6 Symbol	Function 6 I/O	Function 7 Symbol	Function 7 I/O	Figure 1.3
72	84	—	—	—	—	—	—	
73	85							
74	86							
75	87	—	—	—	—	—	—	
76	88							
77	89							
78	90							
79	91							
80	92							
81	93							
82	94	—	—	—	—	—	—	(4)
83	95	—	—	—	—	—	—	(4)
84	96	—	—	—	—	—	—	(4)
85	97	—	—	—	—	—	—	(4)

chanhvinh.com

SH7262 Pin No.	SH7264 Pin No.	Function 1 Symbol	Function 1 I/O	Function 2 Symbol	Function 2 I/O	Function 3 Symbol	Function 3 I/O	Function 4 Symbol	Function 4 I/O
—	98	PH4	I	AN4	I(a)	—	—	—	—
—	99	PH5	I	AN5	I(a)	—	—	—	—
86	100	AVss							
—	101	PH6	I	AN6	I(a)	—	—	—	—
87	102	AVref							
—	103	PH7	I	AN7	I(a)	—	—	—	—
88	104	AVcc							
89	105	$\overline{\text{TRST}}$	I(s)	—	—	—	—	—	—
90	106	$\overline{\text{ASEBRKAK/ASEBRK}}$	I(s)/O	—	—	—	—	—	—
91	107	TDO	O	—	—	—	—	—	—
92	108	TDI	I	—	—	—	—	—	—
93	109	TMS	I	—	—	—	—	—	—
94	110	TCK	I	—	—	—	—	—	—
—	111	PG24	I(s)/O	MISO1	I(s)/O	TIOC0D	I(s)/O	—	—

SH7262 Pin No.	SH7264 Pin No.	Function 5 Symbol	Function 5 I/O	Function 6 Symbol	Function 6 I/O	Function 7 Symbol	Function 7 I/O	Figure 1.3
—	98	—	—	—	—	—	—	(4)
—	99	—	—	—	—	—	—	(4)
86	100							
—	101	—	—	—	—	—	—	(4)
87	102							
—	103	—	—	—	—	—	—	(4)
88	104							
89	105	—	—	—	—	—	—	(3)
90	106	—	—	—	—	—	—	(7)
91	107	—	—	—	—	—	—	(5)
92	108	—	—	—	—	—	—	(2)
93	109	—	—	—	—	—	—	(2)
94	110	—	—	—	—	—	—	(2)
—	111	—	—	—	—	—	—	(7)

chanhvinh.com

SH7262 Pin No.	SH7264 Pin No.	Function 1		Function 2		Function 3		Function 4	
		Symbol	I/O	Symbol	I/O	Symbol	I/O	Symbol	I/O
—	112	PG23	I(s)/O	MOSI1	I(s)/O	TIOC0C	I(s)/O	—	—
95	113	PVcc							
—	114	PG22	I(s)/O	SSL10	I(s)/O	TIOC0B	I(s)/O	—	—
96	115	Vss							
—	116	PG21	I(s)/O	RSPCK1	I(s)/O	TIOC0A	I(s)/O	—	—
97	117	Vcc							
98	118	PG20	I(s)/O	LCD_EXTCLK	I(s)	—	—	MISO1	I(s)/O
99	119	PG19	I(s)/O	LCD_CLK	O	TIOC2B	I(s)/O	MOSI1	I(s)/O
100	120	PG18	I(s)/O	LCD_DE	O	TIOC2A	I(s)/O	SSL10	I(s)/O
101	121	PG17	I(s)/O	LCD_HSYNC	O	TIOC1B	I(s)/O	RSPCK1	I(s)/O
102	122	PG16	I(s)/O	LCD_VSYNC	O	TIOC1A	I(s)/O	TxD3	O
103	123	PG15	I(s)/O	LCD_DATA15	O	TIOC0D	I(s)/O	RxD3	I(s)
104	124	PG14	I(s)/O	LCD_DATA14	O	TIOC0C	I(s)/O	—	—
105	125	PG13	I(s)/O	LCD_DATA13	O	TIOC0B	I(s)/O	—	—

SH7262 Pin No.	SH7264 Pin No.	Function 5		Function 6		Function 7		
		Symbol	I/O	Symbol	I/O	Symbol	I/O	Figure 1.3
—	112	—	—	—	—	—	—	(7)
95	113							
—	114	—	—	—	—	—	—	(7)
96	115							
—	116	—	—	—	—	—	—	(7)
97	117							
98	118	TxD7	O	—	—	—	—	(7)
99	119	RxD7	I(s)	—	—	—	—	(7)
100	120	TxD6	O	—	—	—	—	(7)
101	121	RxD6	I(s)	—	—	—	—	(7)
102	122	$\overline{\text{CTS1}}$	I(s)/O	—	—	—	—	(7)
103	123	$\overline{\text{RTS1}}$	I(s)/O	—	—	—	—	(7)
104	124	SCK1	I(s)/O	—	—	—	—	(7)
105	125	TxD1	O	—	—	—	—	(7)

chanhvinh.com

SH7262 Pin No.	SH7264 Pin No.	Function 1 Symbol	Function 1 I/O	Function 2 Symbol	Function 2 I/O	Function 3 Symbol	Function 3 I/O	Function 4 Symbol	Function 4 I/O
106	126	PVcc							
107	127	PG12	I(s)/O	LCD_DATA12	O	TIOC0A	I(s)/O	—	—
108	128	Vss							
109	129	PG11	I(s)/O	LCD_DATA11	O	SSITxD0	O	IRQ3	I(s)
110	130	Vcc							
111	131	PG10	I(s)/O	LCD_DATA10	O	SSIRxD0	I(s)	IRQ2	I(s)
112	132	PG9	I(s)/O	LCD_DATA9	O	SSIWS0	I(s)/O	—	—
113	133	PG8	I(s)/O	LCD_DATA8	O	SSISCK0	I(s)/O	—	—
114	134	PG7	I(s)/O	LCD_DATA7	O	SD_CD	I(s)	PINT7	I(s)
115	135	PG6	I(s)/O	LCD_DATA6	O	SD_WP	I(s)	PINT6	I(s)
116	136	PG5	I(s)/O	LCD_DATA5	O	SD_D1	I(s)/O	PINT5	I(s)
117	137	PG4	I(s)/O	LCD_DATA4	O	SD_D0	I(s)/O	PINT4	I(s)
118	138	PVcc							
119	139	PG3	I(s)/O	LCD_DATA3	O	SD_CLK	O	PINT3	I(s)

SH7262 Pin No.	SH7264 Pin No.	Function 5 Symbol	Function 5 I/O	Function 6 Symbol	Function 6 I/O	Function 7 Symbol	Function 7 I/O	Figure 1.3
106	126							
107	127	RxD1	I(s)	—	—	—	—	(7)
108	128							
109	129	TxD5	O	SIOFTxD	O	—	—	(7)
110	130							
111	131	RxD5	I(s)	SIOFRxD	I(s)	—	—	(7)
112	132	TxD4	O	SIOFSYNC	I(s)/O	—	—	(7)
113	133	RxD4	I(s)	SIOFSCK	I(s)/O	—	—	(7)
114	134	IRQ7*	I(s)	—	—	—	—	(7)
115	135	IRQ6*	I(s)	—	—	—	—	(7)
116	136	IRQ5*	I(s)	—	—	—	—	(7)
117	137	IRQ4*	I(s)	—	—	—	—	(7)
118	138							
119	139	—	—	—	—	—	—	(7)

chanhvinh.com

SH7262 Pin No.	SH7264 Pin No.	Function 1 Symbol	Function 1 I/O	Function 2 Symbol	Function 2 I/O	Function 3 Symbol	Function 3 I/O	Function 4 Symbol	Function 4 I/O
120	140	Vss							
121	141	PG2	I(s)/O	LCD_DATA2	O	SD_CMD	I(s)/O	PINT2	I(s)
122	142	Vcc							
123	143	PG1	I(s)/O	LCD_DATA1	O	SD_D3	I(s)/O	PINT1	I(s)
124	144	PG0	I(s)/O	LCD_DATA0	O	SD_D2	I(s)/O	PINT0	I(s)
—	145	PK11	I(s)/O	PWM2D	O	SSITxD0	O	—	—
—	146	PK10	I(s)/O	PWM2C	O	SSIRxD0	I(s)	—	—
—	147	PK9	I(s)/O	PWM2B	O	SSIWS0	I(s)/O	—	—
—	148	PK8	I(s)/O	PWM2A	O	SSISCK0	I(s)/O	—	—
125	149	AUDIO_X2	O	—	—	—	—	—	—
126	150	AUDIO_X1	I	—	—	—	—	—	—
127	151	PF12	I(s)/O	$\overline{BS}$	O	AUDIO_XOUT	O	MISO0	I(s)/O
128	152	PVcc							
129	153	PF11	I(s)/O	A25	O	SSIDATA3	I(s)/O	MOSI0	I(s)/O

SH7262 Pin No.	SH7264 Pin No.	Function 5 Symbol	Function 5 I/O	Function 6 Symbol	Function 6 I/O	Function 7 Symbol	Function 7 I/O	Figure 1.3
120	140							
121	141	—	—	—	—	—	—	(7)
122	142							
123	143	—	—	—	—	—	—	(7)
124	144	$\overline{WDTOVF}$	O	—	—	—	—	(7)
—	145	—	—	—	—	—	—	(7)
—	146	—	—	—	—	—	—	(7)
—	147	—	—	—	—	—	—	(7)
—	148	—	—	—	—	—	—	(7)
125	149	—	—	—	—	—	—	(9)
126	150	—	—	—	—	—	—	
127	151	TIOC3D	I(s)/O	SPDIF_OUT	O	—	—	(7)
128	152							
129	153	TIOC3C	I(s)/O	SPDIF_IN	I(s)	—	—	(7)

chanhvinh.com

SH7262 Pin No.	SH7264 Pin No.	Function 1 Symbol	Function 1 I/O	Function 2 Symbol	Function 2 I/O	Function 3 Symbol	Function 3 I/O	Function 4 Symbol	Function 4 I/O
130	154	Vss							
131	155	PF10	I(s)/O	A24	O	SSIWS3	I(s)/O	SSL00	I(s)/O
132	156	Vcc							
133	157	PF9	I(s)/O	A23	O	SSISCK3	I(s)/O	RSPCK0	I(s)/O
134	158	PF8	I(s)/O	$\overline{\text{CE2B}}$	O	SSIDATA3	I(s)/O	DV_CLK	I(s)
135	159	PF7	I(s)/O	$\overline{\text{CE2A}}$	O	SSIWS3	I(s)/O	DV_DATA7	I(s)
136	160	PF6	I(s)/O	$\overline{\text{CS6/CE1B}}$	O	SSISCK3	I(s)/O	DV_DATA6	I(s)
137	161	PF5	I(s)/O	$\overline{\text{CS5/CE1A}}$	O	SSIDATA2	I(s)/O	DV_DATA5	I(s)
138	162	PF4	I(s)/O	$\overline{\text{ICIOWR/AH}}$	O	SSIWS2	I(s)/O	DV_DATA4	I(s)
139	163	PF3	I(s)/O	$\overline{\text{ICIORD}}$	O	SSISCK2	I(s)/O	DV_DATA3	I(s)
140	164	PF2	I(s)/O	$\overline{\text{BACK}}$	O	SSIDATA1	I(s)/O	DV_DATA2	I(s)
141	165	PVcc							
142	166	PF1	I(s)/O	$\overline{\text{BREQ}}$	I(s)	SSIWS1	I(s)/O	DV_DATA1	I(s)
143	167	Vss							

SH7262 Pin No.	SH7264 Pin No.	Function 5 Symbol	Function 5 I/O	Function 6 Symbol	Function 6 I/O	Function 7 Symbol	Function 7 I/O	Function 8 Symbol	Function 8 I/O	Figure 1.3
130	154									
131	155	TIOC3B	I(s)/O	$\overline{\text{FCE}}$	O	—	—	—	—	(7)
132	156									
133	157	TIOC3A	I(s)/O	FRB	I(s)	—	—	—	—	(7)
134	158	—	—	—	—	SD_CD*	I(s)	—	—	(7)
135	159	TCLKD	I(s)	—	—	SD_WP*	I(s)	—	—	(7)
136	160	TCLKB	I(s)	—	—	SD_D1*	I(s)/O	—	—	(7)
137	161	TCLKC	I(s)	—	—	SD_D0*	I(s)/O	AUDATA3	O	(7)
138	162	TxD3	O	—	—	SD_CLK*	O	AUDATA2	O	(7)
139	163	RxD3	I(s)	—	—	SD_CMD*	I(s)/O	AUDATA1	O	(7)
140	164	TxD2	O	DACK0	O	SD_D3*	I(s)/O	AUDATA0	O	(7)
141	165									
142	166	RxD2	I(s)	DREQ0	I(s)	SD_D2*	I(s)/O	$\overline{\text{AUDSYNC}}$	O	(7)
143	167									

chanhvinh.com

SH7262 Pin No.	SH7264 Pin No.	Function 1 Symbol	Function 1 I/O	Function 2 Symbol	Function 2 I/O	Function 3 Symbol	Function 3 I/O	Function 4 Symbol	Function 4 I/O
144	168	PF0	I(s)/O	$\overline{\text{WAIT}}$	I(s)	SSISCK1	I(s)/O	DV_DATA0	I(s)
145	169	Vcc							
—	170	PK7	I(s)/O	PWM1H	O	SD_CD	I(s)	—	—
—	171	PK6	I(s)/O	PWM1G	O	SD_WP	I(s)	—	—
—	172	PK5	I(s)/O	PWM1F	O	SD_D1	I(s)/O	—	—
—	173	PK4	I(s)/O	PWM1E	O	SD_D0	I(s)/O	—	—
146	174	PE5	I(s)/O(o)	SDA2	I(s)/O(o)	—	—	DV_HSYNC	I(s)
147	175	PE4	I(s)/O(o)	SCL2	I(s)/O(o)	—	—	DV_VSYNC	I(s)
148	176	PE3	I(s)/O(o)	SDA1	I(s)/O(o)	—	—	IRQ3	I(s)
149	177	PE2	I(s)/O(o)	SCL1	I(s)/O(o)	—	—	IRQ2	I(s)
150	178	PE1	I(s)/O(o)	SDA0	I(s)/O(o)	$\overline{\text{IOIS16}}$	I(s)	IRQ1	I(s)
151	179	PE0	I(s)/O(o)	SCL0	I(s)/O(o)	AUDIO_CLK	I(s)	IRQ0	I(s)
—	180	PK3	I(s)/O	PWM1D	O	SD_CLK	O	—	—
152	181	PVcc							

SH7262 Pin No.	SH7264 Pin No.	Function 5 Symbol	Function 5 I/O	Function 6 Symbol	Function 6 I/O	Function 7 Symbol	Function 7 I/O	Figure 1.3
144	168	SCK2	I(s)/O	TEND0	O	AUDCK	O	(7)
145	169							
—	170	—	—	—	—	—	—	(7)
—	171	—	—	—	—	—	—	(7)
—	172	—	—	—	—	—	—	(7)
—	173	—	—	—	—	—	—	(7)
146	174	—	—	—	—	—	—	(8)
147	175	—	—	—	—	—	—	(8)
148	176	—	—	—	—	—	—	(8)
149	177	—	—	—	—	—	—	(8)
150	178	TCLKA	I(s)	$\overline{\text{ADTRG}}$	I(s)	—	—	(8)
151	179	—	—	—	—	—	—	(8)
—	180	—	—	—	—	—	—	(7)
152	181							

chanhvinh.com

SH7262 Pin No.	SH7264 Pin No.	Function 1 Symbol	Function 1 I/O	Function 2 Symbol	Function 2 I/O	Function 3 Symbol	Function 3 I/O	Function 4 Symbol	Function 4 I/O
153	182	Vss							
—	183	PK2	I(s)/O	PWM1C	O	SD_CMD	I(s)/O	—	—
—	184	PK1	I(s)/O	PWM1B	O	SD_D3	I(s)/O	—	—
—	185	PK0	I(s)/O	PWM1A	O	SD_D2	I(s)/O	—	—
154	186	PD15	I/O	D15	I/O	NAF7	I/O	PWM2H	O
155	187	PD14	I/O	D14	I/O	NAF6	I/O	PWM2G	O
156	188	PD13	I/O	D13	I/O	NAF5	I/O	PWM2F	O
157	189	PD12	I/O	D12	I/O	NAF4	I/O	PWM2E	O
158	190	PD11	I/O	D11	I/O	NAF3	I/O	PWM2D	O
159	191	PD10	I/O	D10	I/O	NAF2	I/O	PWM2C	O
160	192	Vss							
161	193	PVcc							
162	194	PD9	I/O	D9	I/O	NAF1	I/O	PWM2B	O
163	195	PD8	I/O	D8	I/O	NAF0	I/O	PWM2A	O

SH7262 Pin No.	SH7264 Pin No.	Function 5 Symbol	Function 5 I/O	Function 6 Symbol	Function 6 I/O	Function 7 Symbol	Function 7 I/O	Figure 1.3
153	182							
—	183	—	—	—	—	—	—	(7)
—	184	—	—	—	—	—	—	(7)
—	185	—	—	—	—	—	—	(7)
154	186	—	—	—	—	—	—	(6)
155	187	—	—	—	—	—	—	(6)
156	188	—	—	—	—	—	—	(6)
157	189	—	—	—	—	—	—	(6)
158	190	—	—	—	—	—	—	(6)
159	191	—	—	—	—	—	—	(6)
160	192							
161	193							
162	194	—	—	—	—	—	—	(6)
163	195	—	—	—	—	—	—	(6)

chanhvinh.com

SH7262 Pin No.	SH7264 Pin No.	Function 1 Symbol	Function 1 I/O	Function 2 Symbol	Function 2 I/O	Function 3 Symbol	Function 3 I/O	Function 4 Symbol	Function 4 I/O
164	196	PD7	I/O	D7	I/O	$\overline{FWE}$	O	PWM1H	O
165	197	PD6	I/O	D6	I/O	FALE	O	PWM1G	O
166	198	PD5	I/O	D5	I/O	FCLE	O	PWM1F	O
167	199	PD4	I/O	D4	I/O	$\overline{FRE}$	O	PWM1E	O
168	200	PD3	I/O	D3	I/O	—	—	PWM1D	O
169	201	Vcc							
170	202	PD2	I/O	D2	I/O	—	—	PWM1C	O
171	203	Vss							
172	204	PD1	I/O	D1	I/O	—	—	PWM1B	O
173	205	PVcc							
174	206	PD0	I/O	D0	I/O	—	—	PWM1A	O
175	207	PC0	I(s)/O	$\overline{CS0}$	O	—	—	—	—
176	208	PC1	I(s)/O	$\overline{RD}$	O	—	—	—	—

SH7262 Pin No.	SH7264 Pin No.	Function 5 Symbol	Function 5 I/O	Function 6 Symbol	Function 6 I/O	Function 7 Symbol	Function 7 I/O	Figure 1.3
164	196	—	—	—	—	—	—	(6)
165	197	—	—	—	—	—	—	(6)
166	198	—	—	—	—	—	—	(6)
167	199	—	—	—	—	—	—	(6)
168	200	—	—	—	—	—	—	(6)
169	201							
170	202	—	—	—	—	—	—	(6)
171	203							
172	204	—	—	—	—	—	—	(6)
173	205							
174	206	—	—	—	—	—	—	(6)
175	207	—	—	—	—	—	—	(7)
176	208	—	—	—	—	—	—	(7)

[Legend]
(s): Schmitt
(a): Analog
(o): Open drain
Note: * Can be used in 640K-byte version only.

chanhvinh.com

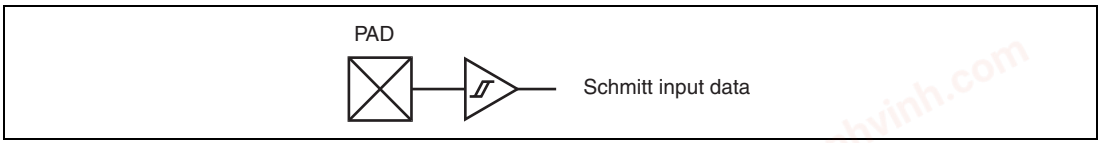


Figure 1.3 (1) Simplified Circuit Diagram (Schmitt Input Buffer)

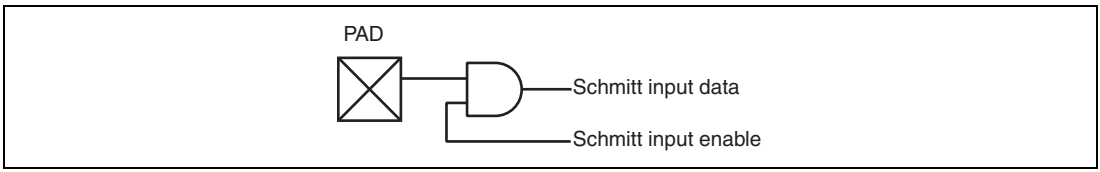


Figure 1.3 (2) Simplified Circuit Diagram (TTL AND Input Buffer)

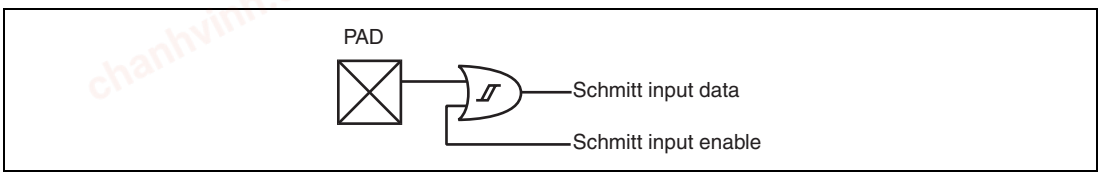


Figure 1.3 (3) Simplified Circuit Diagram (Schmitt OR Input Buffer)

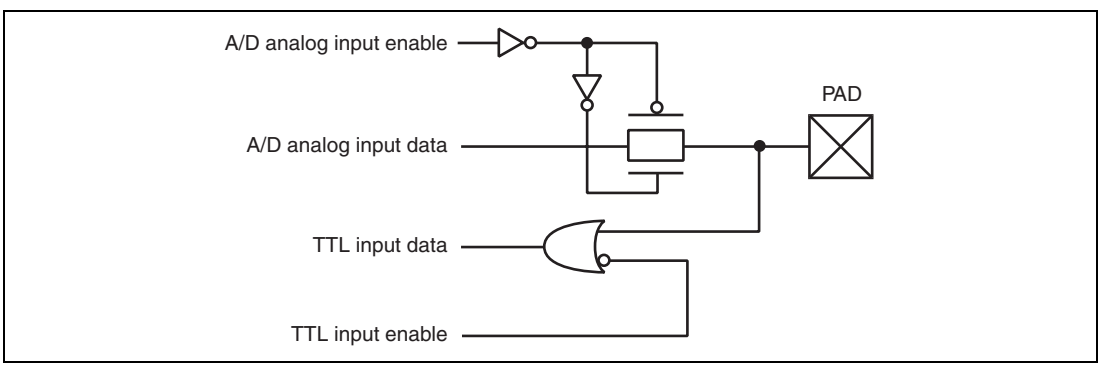


Figure 1.3 (4) Simplified Circuit Diagram (TTL OR Input and A/D Input Buffer)

chanhvinh.com

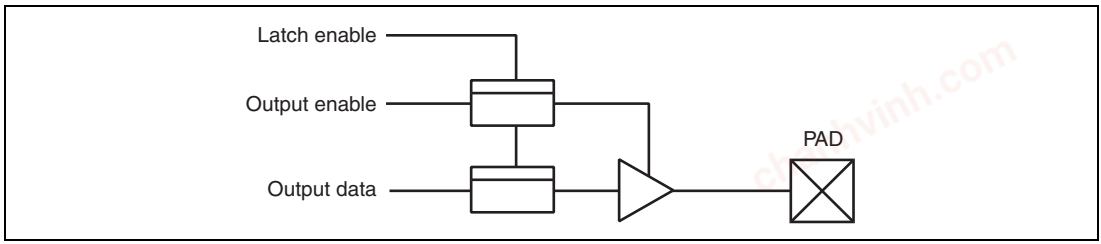


Figure 1.3 (5) Simplified Circuit Diagram (Output Buffer with Enable, with Latch)

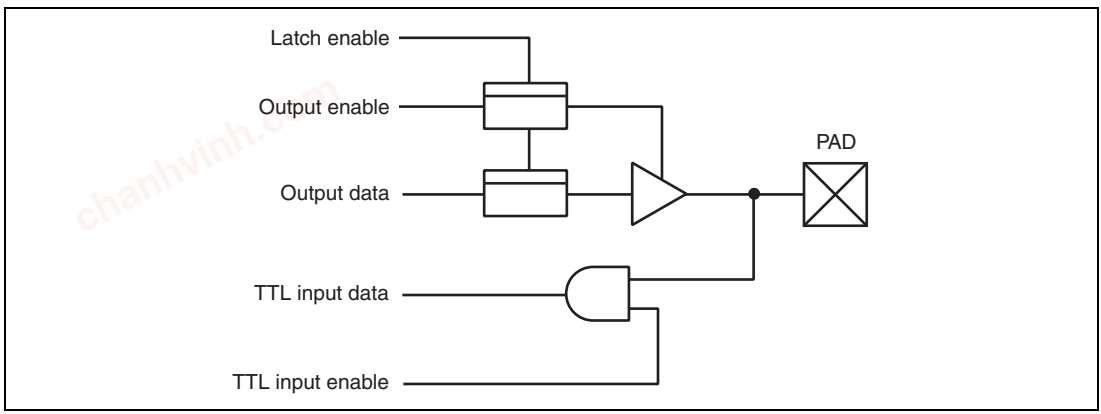


Figure 1.3 (6) Simplified Circuit Diagram (Bidirectional Buffer, TTL AND Input, with Latch)

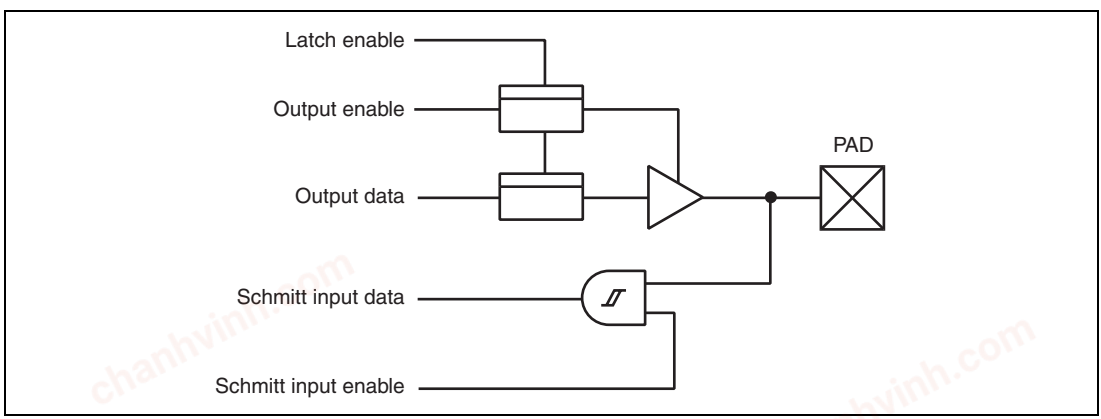


Figure 1.3 (7) Simplified Circuit Diagram (Bidirectional Buffer, Schmitt AND Input, with Latch)

chanhvinh.com

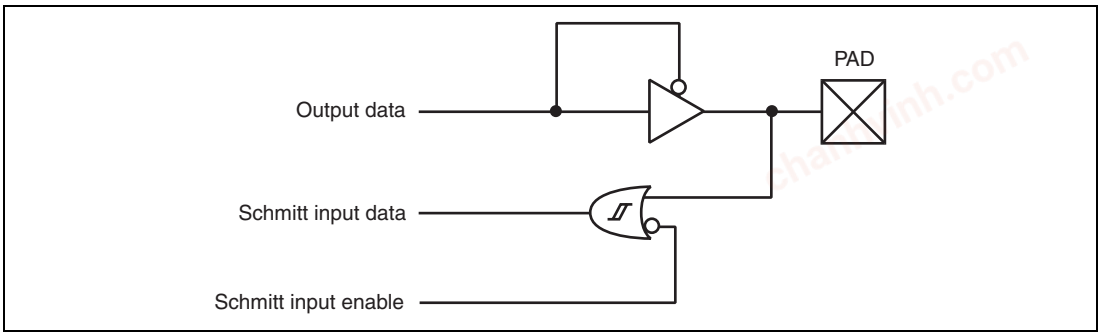


Figure 1.3 (8) Simplified Circuit Diagram (Open Drain Output and Schmitt OR Input Buffer)

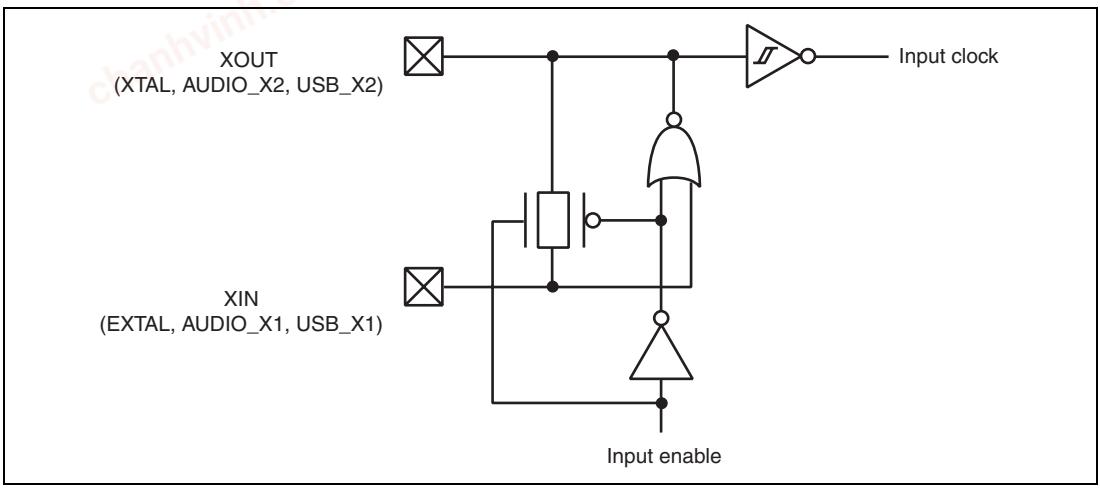


Figure 1.3 (9) Simplified Circuit Diagram (Oscillation Buffer 1)

chanhvinh.com

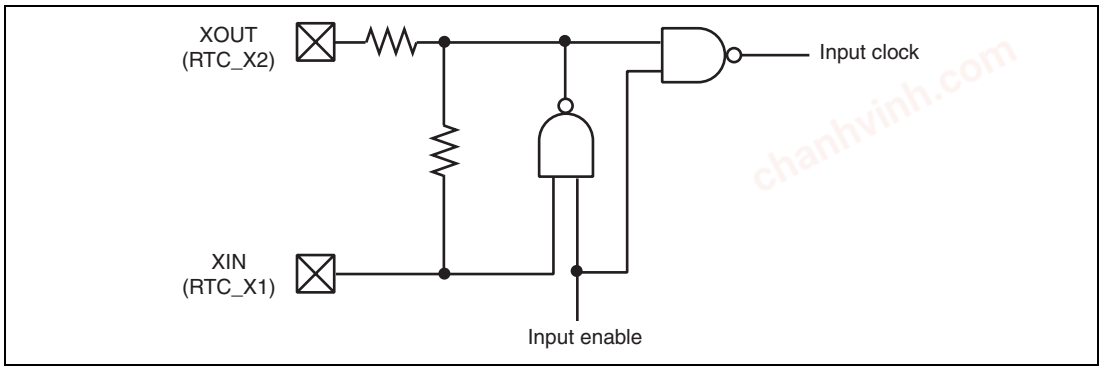


Figure 1.3 (10) Simplified Circuit Diagram (Oscillation Buffer 2)

chanhvinh.com

chanhvinh.com

Section 2 CPU

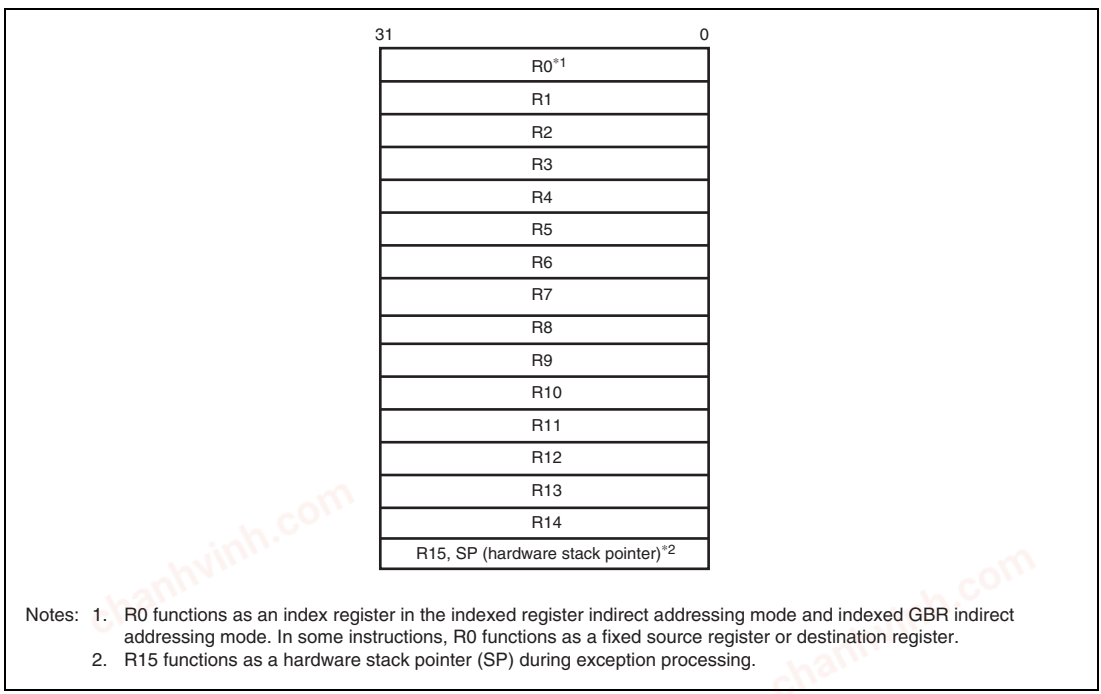
2.1 Register Configuration

The register set consists of sixteen 32-bit general registers, four 32-bit control registers, and four 32-bit system registers.

2.1.1 General Registers

Figure 2.1 shows the general registers.

The sixteen 32-bit general registers are numbered R0 to R15. General registers are used for data processing and address calculation. R0 is also used as an index register. Several instructions have R0 fixed as their only usable register. R15 is used as the hardware stack pointer (SP). Saving and restoring the status register (SR) and program counter (PC) in exception handling is accomplished by referencing the stack using R15.



Notes: 1. R0 functions as an index register in the indexed register indirect addressing mode and indexed GBR indirect addressing mode. In some instructions, R0 functions as a fixed source register or destination register.
2. R15 functions as a hardware stack pointer (SP) during exception processing.

Figure 2.1 General Registers

chanhvinh.com

2.1.2 Control Registers

The control registers consist of four 32-bit registers: the status register (SR), the global base register (GBR), the vector base register (VBR), and the jump table base register (TBR).

The status register indicates instruction processing states.

The global base register functions as a base address for the GBR indirect addressing mode to transfer data to the registers of on-chip peripheral modules.

The vector base register functions as the base address of the exception handling vector area (including interrupts).

The jump table base register functions as the base address of the function table area.

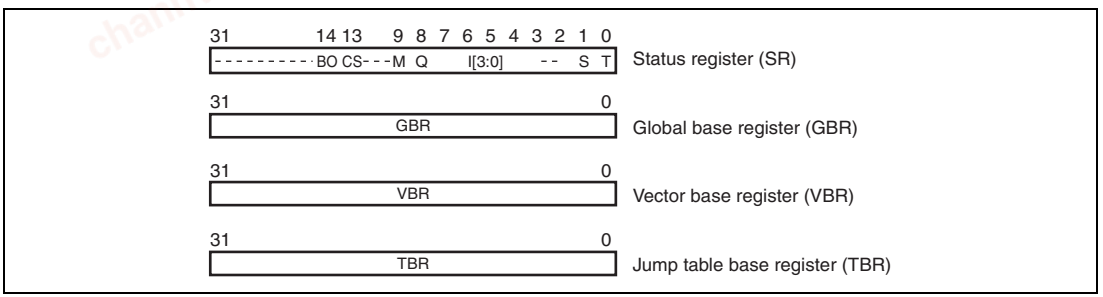


Figure 2.2 Control Registers

(1) Status Register (SR)

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
	-	BO	CS	-	-	-	M	Q	I[3:0]				-	-	S	T
Initial value:	0	0	0	0	0	0	-	-	1	1	1	1	0	0	-	-
R/W:	R	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W

chanhvinh.com

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	BO	0	R/W	BO Bit Indicates that a register bank has overflowed.
13	CS	0	R/W	CS Bit Indicates that, in CLIP instruction execution, the value has exceeded the saturation upper-limit value or fallen below the saturation lower-limit value.
12 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	M	—	R/W	M Bit
8	Q	—	R/W	Q Bit Used by the DIV0S, DIV0U, and DIV1 instructions.
7 to 4	I[3:0]	1111	R/W	Interrupt Mask Level
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	S	—	R/W	S Bit Specifies a saturation operation for a MAC instruction.
0	T	—	R/W	T Bit True/false condition or carry/borrow bit

(2) Global Base Register (GBR)

GBR is referenced as the base address in a GBR-referencing MOV instruction.

(3) Vector Base Register (VBR)

VBR is referenced as the branch destination base address in the event of an exception or an interrupt.

(4) Jump Table Base Register (TBR)

TBR is referenced as the start address of a function table located in memory in a JSR/N@@(disp8,TBR) table-referencing subroutine call instruction.

chanhvinh.com

2.1.3 System Registers

The system registers consist of four 32-bit registers: the high and low multiply and accumulate registers (MACH and MACL), the procedure register (PR), and the program counter (PC). MACH and MACL store the results of multiply or multiply and accumulate operations. PR stores the return address from a subroutine procedure. PC points four bytes ahead of the current instruction and controls the flow of the processing.

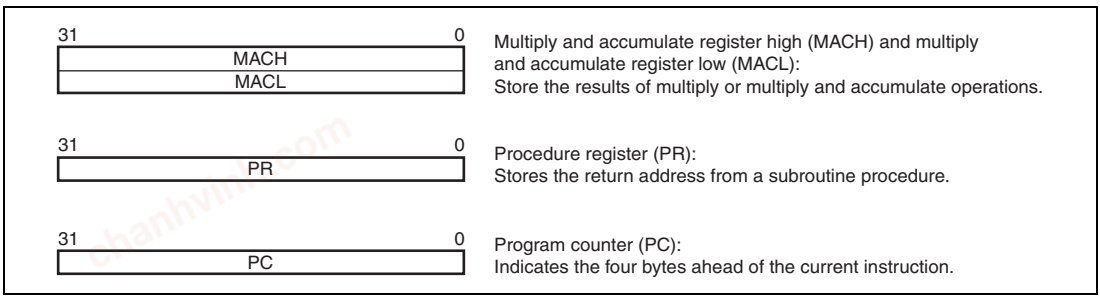


Figure 2.3 System Registers

(1) Multiply and Accumulate Register High (MACH) and Multiply and Accumulate Register Low (MACL)

MACH and MACL are used as the addition value in a MAC instruction, and store the result of a MAC or MUL instruction.

(2) Procedure Register (PR)

PR stores the return address of a subroutine call using a BSR, BSRF, or JSR instruction, and is referenced by a subroutine return instruction (RTS).

(3) Program Counter (PC)

PC points four bytes ahead of the instruction being executed.

chanhvinh.com

2.1.4 Register Banks

For the nineteen 32-bit registers comprising general registers R0 to R14, control register GBR, and system registers MACH, MACL, and PR, high-speed register saving and restoration can be carried out using a register bank. The register contents are automatically saved in the bank after the CPU accepts an interrupt that uses a register bank. Restoration from the bank is executed by issuing a RESBANK instruction in an interrupt processing routine.

This LSI has 15 banks. For details, see the SH-2A, SH2A-FPU Software Manual and section 7.8, Register Banks.

2.1.5 Initial Values of Registers

Table 2.1 lists the values of the registers after a reset.

Table 2.1 Initial Values of Registers

Classification	Register	Initial Value
General registers	R0 to R14	Undefined
	R15 (SP)	Value of the stack pointer in the vector address table
Control registers	SR	Bits I[3:0] are 1111 (H'F), BO and CS are 0, reserved bits are 0, and other bits are undefined
	GBR, TBR	Undefined
	VBR	H'00000000
System registers	MACH, MACL, PR	Undefined
	PC	Value of the program counter in the vector address table

chanhvinh.com

2.2 Data Formats

2.2.1 Data Format in Registers

Register operands are always longwords (32 bits). If the size of memory operand is a byte (8 bits) or a word (16 bits), it is changed into a longword by expanding the sign-part when loaded into a register.

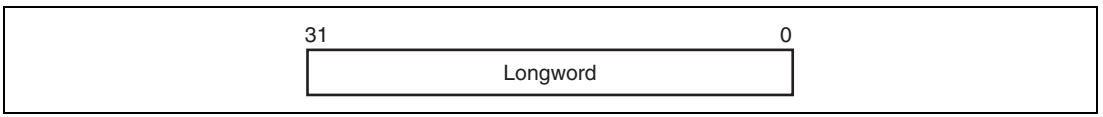


Figure 2.4 Data Format in Registers

2.2.2 Data Formats in Memory

Memory data formats are classified into bytes, words, and longwords. Memory can be accessed in 8-bit bytes, 16-bit words, or 32-bit longwords. A memory operand of fewer than 32 bits is stored in a register in sign-extended or zero-extended form.

A word operand should be accessed at a word boundary (an even address of multiple of two bytes: address 2n), and a longword operand at a longword boundary (an even address of multiple of four bytes: address 4n). Otherwise, an address error will occur. A byte operand can be accessed at any address.

Only big-endian byte order can be selected for the data format.

Data formats in memory are shown in figure 2.5.

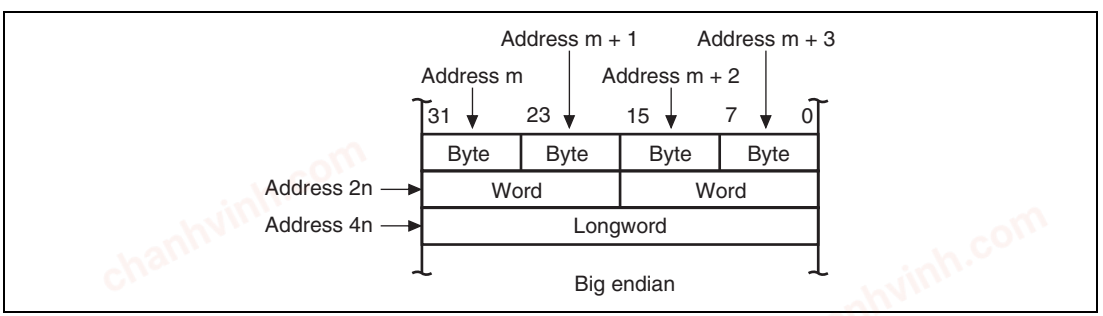


Figure 2.5 Data Formats in Memory

chanhvinh.com

2.2.3 Immediate Data Format

Byte (8-bit) immediate data is located in an instruction code. Immediate data accessed by the MOV, ADD, and CMP/EQ instructions is sign-extended and handled in registers as longword data. Immediate data accessed by the TST, AND, OR, and XOR instructions is zero-extended and handled as longword data. Consequently, AND instructions with immediate data always clear the upper 24 bits of the destination register.

20-bit immediate data is located in the code of a MOVI20 or MOVI20S 32-bit transfer instruction. The MOVI20 instruction stores immediate data in the destination register in sign-extended form. The MOVI20S instruction shifts immediate data by eight bits in the upper direction, and stores it in the destination register in sign-extended form.

Word or longword immediate data is not located in the instruction code, but rather is stored in a memory table. The memory table is accessed by an immediate data transfer instruction (MOV) using the PC relative addressing mode with displacement.

See examples given in section 2.3.1 (10), Immediate Data.

chanhvinh.com

2.3 Instruction Features

2.3.1 RISC-Type Instruction Set

Instructions are RISC type. This section details their functions.

(1) 16-Bit Fixed-Length Instructions

Basic instructions have a fixed length of 16 bits, improving program code efficiency.

(2) 32-Bit Fixed-Length Instructions

The SH-2A additionally features 32-bit fixed-length instructions, improving performance and ease of use.

(3) One Instruction per State

Each basic instruction can be executed in one cycle using the pipeline system.

(4) Data Length

Longword is the standard data length for all operations. Memory can be accessed in bytes, words, or longwords. Byte or word data in memory is sign-extended and handled as longword data. Immediate data is sign-extended for arithmetic operations or zero-extended for logic operations. It is also handled as longword data.

Table 2.2 Sign Extension of Word Data

SH2-A CPU		Description	Example of Other CPU
MOV.W	@(disp,PC),R1	Data is sign-extended to 32 bits, and R1 becomes H'00001234. It is next operated upon by an ADD instruction.	ADD.W #H'1234,R0
ADD	R1,R0		
	.....		
.DATA.W	H'1234		

Note: @(disp, PC) accesses the immediate data.

(5) Load-Store Architecture

Basic operations are executed between registers. For operations that involve memory access, data is loaded to the registers and executed (load-store architecture). Instructions such as AND that manipulate bits, however, are executed directly in memory.

chanhvinh.com

(6) Delayed Branch Instructions

With the exception of some instructions, unconditional branch instructions, etc., are executed as delayed branch instructions. With a delayed branch instruction, the branch is taken after execution of the instruction immediately following the delayed branch instruction. This reduces disturbance of the pipeline control when a branch is taken.

In a delayed branch, the actual branch operation occurs after execution of the slot instruction. However, instruction execution such as register updating excluding the actual branch operation, is performed in the order of delayed branch instruction → delay slot instruction. For example, even though the contents of the register holding the branch destination address are changed in the delay slot, the branch destination address remains as the register contents prior to the change.

Table 2.3 Delayed Branch Instructions

SH-2A CPU		Description	Example of Other CPU	
BRA	TRGET	Executes the ADD before branching to TRGET.	ADD.W	R1,R0
ADD	R1,R0		BRA	TRGET

(7) Unconditional Branch Instructions with No Delay Slot

The SH-2A additionally features unconditional branch instructions in which a delay slot instruction is not executed. This eliminates unnecessary NOP instructions, and so reduces the code size.

(8) Multiply/Multiply-and-Accumulate Operations

16-bit × 16-bit → 32-bit multiply operations are executed in one to two cycles. 16-bit × 16-bit + 64-bit → 64-bit multiply-and-accumulate operations are executed in two to three cycles. 32-bit × 32-bit → 64-bit multiply and 32-bit × 32-bit + 64-bit → 64-bit multiply-and-accumulate operations are executed in two to four cycles.

(9) T Bit

The T bit in the status register (SR) changes according to the result of the comparison. Whether a conditional branch is taken or not taken depends upon the T bit condition (true/false). The number of instructions that change the T bit is kept to a minimum to improve the processing speed.

chanhvinh.com

Table 2.4 T Bit

SH-2A CPU		Description	Example of Other CPU	
CMP/GE	R1,R0	T bit is set when R0 ≥ R1.	CMP.W	R1,R0
BT	TRGET0	The program branches to TRGET0 when R0 ≥ R1 and to TRGET1 when R0 < R1.	BGE	TRGET0
BF	TRGET1		BLT	TRGET1
ADD	#–1,R0	T bit is not changed by ADD.	SUB.W	#1,R0
CMP/EQ	#0,R0	T bit is set when R0 = 0.	BEQ	TRGET
BT	TRGET	The program branches if R0 = 0.		

(10) Immediate Data

Byte immediate data is located in an instruction code. Word or longword immediate data is not located in instruction codes but in a memory table. The memory table is accessed by an immediate data transfer instruction (MOV) using the PC relative addressing mode with displacement.

With the SH-2A, 17- to 28-bit immediate data can be located in an instruction code. However, for 21- to 28-bit immediate data, an OR instruction must be executed after the data is transferred to a register.

Table 2.5 Immediate Data Accessing

Classification	SH-2A CPU		Example of Other CPU	
8-bit immediate	MOV	#H'12,R0	MOV.B	#H'12,R0
16-bit immediate	MOVI20	#H'1234,R0	MOV.W	#H'1234,R0
20-bit immediate	MOVI20	#H'12345,R0	MOV.L	#H'12345,R0
28-bit immediate	MOVI20S	#H'12345,R0	MOV.L	#H'1234567,R0
	OR	#H'67,R0		
32-bit immediate	MOV.L	@(disp,PC),R0	MOV.L	#H'12345678,R0
		.....		
	.DATA.L	H'12345678		

Note: @(disp, PC) accesses the immediate data.

chanhvinh.com

(11) Absolute Address

When data is accessed by an absolute address, the absolute address value should be placed in the memory table in advance. That value is transferred to the register by loading the immediate data during the execution of the instruction, and the data is accessed in register indirect addressing mode.

With the SH-2A, when data is referenced using an absolute address not exceeding 28 bits, it is also possible to transfer immediate data located in the instruction code to a register and to reference the data in register indirect addressing mode. However, when referencing data using an absolute address of 21 to 28 bits, an OR instruction must be used after the data is transferred to a register.

Table 2.6 Absolute Address Accessing

Classification	SH-2A CPU		Example of Other CPU	
Up to 20 bits	MOVI20	#H'12345,R1	MOV.B	@H'12345,R0
	MOV.B	@R1,R0		
21 to 28 bits	MOVI20S	#H'12345,R1	MOV.B	@H'1234567,R0
	OR	#H'67,R1		
	MOV.B	@R1,R0		
29 bits or more	MOV.L	@(disp,PC),R1	MOV.B	@H'12345678,R0
	MOV.B	@R1,R0		
		.....		
	.DATA.L	H'12345678		

(12) 16-Bit/32-Bit Displacement

When data is accessed by 16-bit or 32-bit displacement, the displacement value should be placed in the memory table in advance. That value is transferred to the register by loading the immediate data during the execution of the instruction, and the data is accessed in the indexed indirect register addressing mode.

Table 2.7 Displacement Accessing

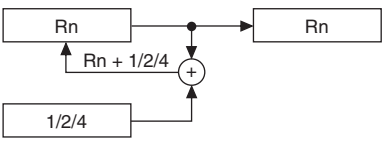
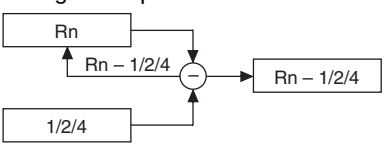
Classification	SH-2A CPU		Example of Other CPU	
16-bit displacement	MOV.W	@(disp,PC),R0	MOV.W	@(H'1234,R1),R2
	MOV.W	@(R0,R1),R2		
		.....		
	.DATA.W	H'1234		

chanhvinh.com

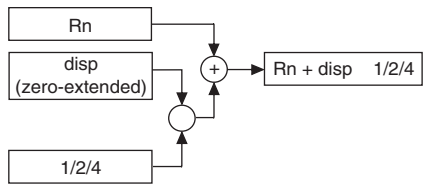
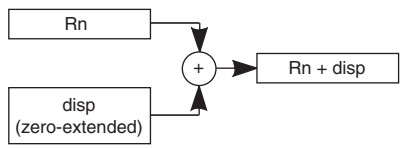
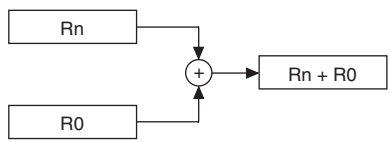
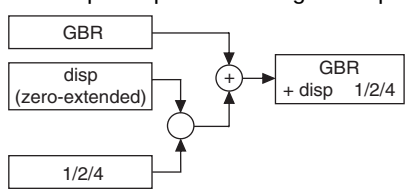
2.3.2 Addressing Modes

Addressing modes and effective address calculation are as follows:

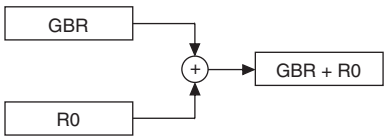
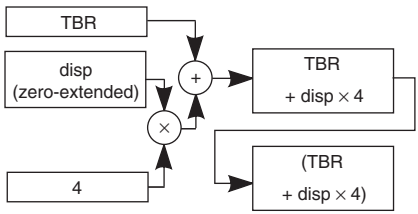
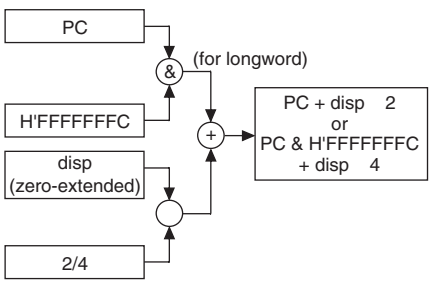
Table 2.8 Addressing Modes and Effective Addresses

Addressing Mode	Instruction Format	Effective Address Calculation	Equation
Register direct	Rn	The effective address is register Rn. (The operand is the contents of register Rn.)	—
Register indirect	@Rn	The effective address is the contents of register Rn. 	Rn
Register indirect with post-increment	@Rn+	The effective address is the contents of register Rn. A constant is added to the contents of Rn after the instruction is executed. 1 is added for a byte operation, 2 for a word operation, and 4 for a longword operation. 	Rn (After instruction execution) Byte: Rn + 1 → Rn Word: Rn + 2 → Rn Longword: Rn + 4 → Rn
Register indirect with pre-decrement	@-Rn	The effective address is the value obtained by subtracting a constant from Rn. 1 is subtracted for a byte operation, 2 for a word operation, and 4 for a longword operation. 	Byte: Rn – 1 → Rn Word: Rn – 2 → Rn Longword: Rn – 4 → Rn (Instruction is executed with Rn after this calculation)

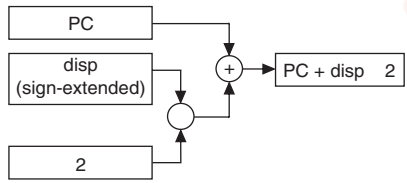
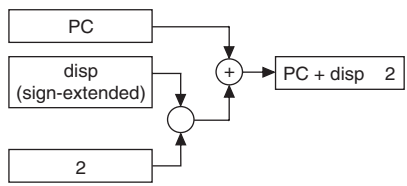
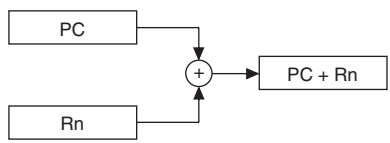
chanhvinh.com

Addressing Mode	Instruction Format	Effective Address Calculation	Equation
Register indirect with displacement	@(disp:4, Rn)	The effective address is the sum of Rn and a 4-bit displacement (disp). The value of disp is zero-extended, and remains unchanged for a byte operation, is doubled for a word operation, and is quadrupled for a longword operation. 	Byte: Rn + disp Word: Rn + disp × 2 Longword: Rn + disp × 4
Register indirect with displacement	@(disp:12, Rn)	The effective address is the sum of Rn and a 12-bit displacement (disp). The value of disp is zero-extended. 	Byte: Rn + disp Word: Rn + disp Longword: Rn + disp
Indexed register indirect	@(R0,Rn)	The effective address is the sum of Rn and R0. 	Rn + R0
GBR indirect with displacement	@(disp:8, GBR)	The effective address is the sum of GBR value and an 8-bit displacement (disp). The value of disp is zero-extended, and remains unchanged for a byte operation, is doubled for a word operation, and is quadrupled for a longword operation. 	Byte: GBR + disp Word: GBR + disp × 2 Longword: GBR + disp × 4

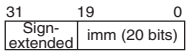
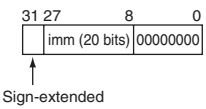
chanhvinh.com

Addressing Mode	Instruction Format	Effective Address Calculation	Equation
Indexed GBR indirect	@(R0, GBR)	The effective address is the sum of GBR value and R0. 	GBR + R0
TBR duplicate indirect with displacement	@@(disp:8, TBR)	The effective address is the sum of TBR value and an 8-bit displacement (disp). The value of disp is zero-extended, and is multiplied by 4. 	Contents of address (TBR + disp × 4)
PC indirect with displacement	@(disp:8, PC)	The effective address is the sum of PC value and an 8-bit displacement (disp). The value of disp is zero-extended, and is doubled for a word operation, and quadrupled for a longword operation. For a longword operation, the lowest two bits of the PC value are masked. 	Word: PC + disp × 2 Longword: PC & H'FFFFFFFC + disp × 4

chanhvinh.com

Addressing Mode	Instruction Format	Effective Address Calculation	Equation
PC relative	disp:8	The effective address is the sum of PC value and the value that is obtained by doubling the sign-extended 8-bit displacement (disp). 	PC + disp × 2
	disp:12	The effective address is the sum of PC value and the value that is obtained by doubling the sign-extended 12-bit displacement (disp). 	PC + disp × 2
	Rn	The effective address is the sum of PC value and Rn. 	PC + Rn

chanhvinh.com

Addressing Mode	Instruction Format	Effective Address Calculation	Equation
Immediate	#imm:20	The 20-bit immediate data (imm) for the MOVI20 instruction is sign-extended. 	—
		The 20-bit immediate data (imm) for the MOVI20S instruction is shifted by eight bits to the left, the upper bits are sign-extended, and the lower bits are padded with zero. 	—
	#imm:8	The 8-bit immediate data (imm) for the TST, AND, OR, and XOR instructions is zero-extended.	—
	#imm:8	The 8-bit immediate data (imm) for the MOV, ADD, and CMP/EQ instructions is sign-extended.	—
	#imm:8	The 8-bit immediate data (imm) for the TRAPA instruction is zero-extended and then quadrupled.	—
	#imm:3	The 3-bit immediate data (imm) for the BAND, BOR, BXOR, BST, BLD, BSET, and BCLR instructions indicates the target bit location.	—

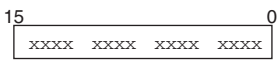
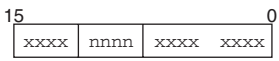
chanhvinh.com

2.3.3 Instruction Format

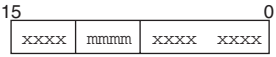
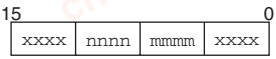
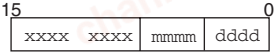
The instruction formats and the meaning of source and destination operands are described below. The meaning of the operand depends on the instruction code. The symbols used are as follows:

- xxxx: Instruction code
- mmmm: Source register
- nnnn: Destination register
- iiii: Immediate data
- dddd: Displacement

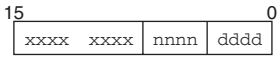
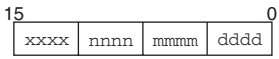
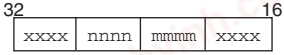
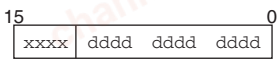
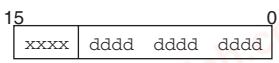
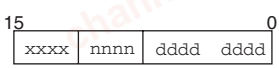
Table 2.9 Instruction Formats

Instruction Formats	Source Operand	Destination Operand	Example
0 format 	—	—	`NOP`
n format 	—	nnnn: Register direct	`MOVT Rn`
	Control register or system register	nnnn: Register direct	`STS MACH,Rn`
	R0 (Register direct)	nnnn: Register direct	`DIVU R0,Rn`
	Control register or system register	nnnn: Register indirect with pre-decrement	`STC.L SR,@-Rn`
	mmmm: Register direct	R15 (Register indirect with pre-decrement)	`MOVMU.L Rm,@-R15`
	R15 (Register indirect with post-increment)	nnnn: Register direct	`MOVMU.L @R15+,Rn`
	R0 (Register direct)	nnnn: (Register indirect with post-increment)	`MOV.L R0,@Rn+`

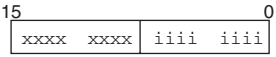
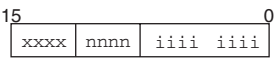
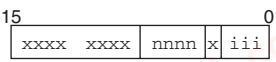
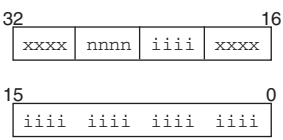
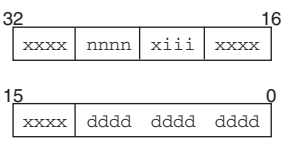
chanhvinh.com

Instruction Formats	Source Operand	Destination Operand	Example
m format  	mmmm: Register direct	Control register or system register	LDC Rm,SR
	mmmm: Register indirect with post-increment	Control register or system register	LDC.L @Rm+,SR
	mmmm: Register indirect	—	JMP @Rm
	mmmm: Register indirect with pre-decrement	R0 (Register direct)	MOV.L @-Rm,R0
	mmmm: PC relative using Rm	—	BRAF Rm
nm format  	mmmm: Register direct	nnnn: Register direct	ADD Rm,Rn
	mmmm: Register direct	nnnn: Register indirect	MOV.L Rm,@Rn
	mmmm: Register indirect with post-increment (multiply-and-accumulate) nnnn*: Register indirect with post-increment (multiply-and-accumulate)	MACH, MACL	MAC.W @Rm+,@Rn+
	mmmm: Register indirect with post-increment	nnnn: Register direct	MOV.L @Rm+,Rn
	mmmm: Register direct	nnnn: Register indirect with pre-decrement	MOV.L Rm,@-Rn
	mmmm: Register direct	nnnn: Indexed register indirect	MOV.L Rm,@(R0,Rn)
md format  	mmmmdddd: Register indirect with displacement	R0 (Register direct)	MOV.B @(disp,Rm),R0

chanhvinh.com

Instruction Formats	Source Operand	Destination Operand	Example
nd4 format 	R0 (Register direct)	nnnndddd: Register indirect with displacement	MOV.B R0,@(disp,Rn)
nmd format 	mmmm: Register direct	nnnndddd: Register indirect with displacement	MOV.L Rm,@(disp,Rn)
	mmmmdddd: Register indirect with displacement	nnnn: Register direct	MOV.L @(disp,Rm),Rn
nmd12 format  	mmmm: Register direct	nnnndddd: Register indirect with displacement	MOV.L Rm,@(disp12,Rn)
	mmmmdddd: Register indirect with displacement	nnnn: Register direct	MOV.L @(disp12,Rm),Rn
d format 	dddddddd: GBR indirect with displacement	R0 (Register direct)	MOV.L @(disp,GBR),R0
	R0 (Register direct)	dddddddd: GBR indirect with displacement	MOV.L R0,@(disp,GBR)
	dddddddd: PC relative with displacement	R0 (Register direct)	MOVA @(disp,PC),R0
	dddddddd: TBR duplicate indirect with displacement	—	JSR/N @@(disp8,TBR)
	dddddddd: PC relative	—	BF label
d12 format 	ddddddddddd: PC relative	—	BRA label (label = disp + PC)
nd8 format 	dddddddd: PC relative with displacement	nnnn: Register direct	MOV.L @(disp,PC),Rn

chanhvinh.com

Instruction Formats	Source Operand	Destination Operand	Example
i format 	iiiiiiii: Immediate	Indexed GBR indirect	AND.B #imm,@(R0,GBR)
	iiiiiiii: Immediate	R0 (Register direct)	AND #imm,R0
	iiiiiiii: Immediate	—	TRAPA #imm
ni format 	iiiiiiii: Immediate	nnnn: Register direct	ADD #imm,Rn
ni3 format 	nnnn: Register direct iii: Immediate	—	BLD #imm3,Rn
	—	nnnn: Register direct iii: Immediate	BST #imm3,Rn
ni20 format 	iiiiiiiiiiiiiiiiii: Immediate	nnnn: Register direct	MOVI20 #imm20, Rn
nid format 	nnnndddddddddddd: Register indirect with displacement iii: Immediate	—	BLD.B #imm3,@(disp12,Rn)
	—	nnnndddddddddddd: Register indirect with displacement iii: Immediate	BST.B #imm3,@(disp12,Rn)

Note: * In multiply-and-accumulate instructions, nnnn is the source register.

chanhvinh.com

2.4 Instruction Set

2.4.1 Instruction Set by Classification

Table 2.10 lists the instructions according to their classification.

Table 2.10 Classification of Instructions

Classification	Types	Operation Code	Function	No. of Instructions
Data transfer	13	MOV	Data transfer Immediate data transfer Peripheral module data transfer Structure data transfer Reverse stack transfer	62
		MOVA	Effective address transfer	
		MOVI20	20-bit immediate data transfer	
		MOVI20S	20-bit immediate data transfer 8-bit left-shit	
		MOVML	R0–Rn register save/restore	
		MOVMU	Rn–R14 and PR register save/restore	
		MOVRT	T bit inversion and transfer to Rn	
		MOVT	T bit transfer	
		MOVU	Unsigned data transfer	
		NOTT	T bit inversion	
		PREF	Prefetch to operand cache	
		SWAP	Swap of upper and lower bytes	
		XTRCT	Extraction of the middle of registers connected	

chanhvinh.com

Classification	Types	Operation Code	Function	No. of Instructions
Arithmetic operations	26	ADD	Binary addition	40
		ADDC	Binary addition with carry	
		ADDV	Binary addition with overflow check	
		CMP/cond	Comparison	
		CLIPS	Signed saturation value comparison	
		CLIPU	Unsigned saturation value comparison	
		DIVS	Signed division (32 ÷ 32)	
		DIVU	Unsigned division (32 ÷ 32)	
		DIV1	One-step division	
		DIV0S	Initialization of signed one-step division	
		DIV0U	Initialization of unsigned one-step division	
		DMULS	Signed double-precision multiplication	
		DMULU	Unsigned double-precision multiplication	
		DT	Decrement and test	
		EXTS	Sign extension	
		EXTU	Zero extension	
		MAC	Multiply-and-accumulate, double-precision multiply-and-accumulate operation	
		MUL	Double-precision multiply operation	
		MULR	Signed multiplication with result storage in Rn	
		MULS	Signed multiplication	
		MULU	Unsigned multiplication	
		NEG	Negation	
		NEGC	Negation with borrow	
		SUB	Binary subtraction	
		SUBC	Binary subtraction with borrow	
		SUBV	Binary subtraction with underflow	

chanhvinh.com

Classification	Types	Operation Code	Function	No. of Instructions
Logic operations	6	AND	Logical AND	14
		NOT	Bit inversion	
		OR	Logical OR	
		TAS	Memory test and bit set	
		TST	Logical AND and T bit set	
		XOR	Exclusive OR	
Shift	12	ROTL	One-bit left rotation	16
		ROTR	One-bit right rotation	
		ROTCL	One-bit left rotation with T bit	
		ROTCR	One-bit right rotation with T bit	
		SHAD	Dynamic arithmetic shift	
		SHAL	One-bit arithmetic left shift	
		SHAR	One-bit arithmetic right shift	
		SHLD	Dynamic logical shift	
		SHLL	One-bit logical left shift	
		SHLLn	n-bit logical left shift	
		SHLR	One-bit logical right shift	
		SHLRn	n-bit logical right shift	
Branch	10	BF	Conditional branch, conditional delayed branch (branch when T = 0)	15
		BT	Conditional branch, conditional delayed branch (branch when T = 1)	
		BRA	Unconditional delayed branch	
		BRAF	Unconditional delayed branch	
		BSR	Delayed branch to subroutine procedure	
		BSRF	Delayed branch to subroutine procedure	
		JMP	Unconditional delayed branch	
		JSR	Branch to subroutine procedure Delayed branch to subroutine procedure	
		RTS	Return from subroutine procedure Delayed return from subroutine procedure	
		RTV/N	Return from subroutine procedure with Rm → R0 transfer	

chanhvinh.com

Classification	Types	Operation Code	Function	No. of Instructions
System control	14	CLRT	T bit clear	36
		CLRMAC	MAC register clear	
		LDBANK	Register restoration from specified register bank entry	
		LDC	Load to control register	
		LDS	Load to system register	
		NOP	No operation	
		RESBANK	Register restoration from register bank	
		RTE	Return from exception handling	
		SETT	T bit set	
		SLEEP	Transition to power-down mode	
		STBANK	Register save to specified register bank entry	
		STC	Store control register data	
		STS	Store system register data	
		TRAPA	Trap exception handling	
Floating-point instructions	19	FABS	Floating-point absolute value	48
		FADD	Floating-point addition	
		FCMP	Floating-point comparison	
		FCNVDS	Conversion from double-precision to single-precision	
		FCNVSD	Conversion from single-precision to double -precision	
		FDIV	Floating-point division	
		FLDI0	Floating-point load immediate 0	
		FLDI1	Floating-point load immediate 1	
		FLDS	Floating-point load into system register FPUL	
		FLOAT	Conversion from integer to floating-point	
		FMAC	Floating-point multiply and accumulate operation	
		FMOV	Floating-point data transfer	
		FMUL	Floating-point multiplication	
		FNEG	Floating-point sign inversion	

chanhvinh.com

Classification	Types	Operation Code	Function	No. of Instructions
Floating-point instructions	19	FSCHG	SZ bit inversion	48
		FSQRT	Floating-point square root	
		FSTS	Floating-point store from system register FPUL	
		FSUB	Floating-point subtraction	
		FTRC	Floating-point conversion with rounding to integer	
FPU-related CPU instructions	2	LDS	Load into floating-point system register	8
		STS	Store from floating-point system register	
Bit manipulation	10	BAND	Bit AND	14
		BCLR	Bit clear	
		BLD	Bit load	
		BOR	Bit OR	
		BSET	Bit set	
		BST	Bit store	
		BXOR	Bit exclusive OR	
		BANDNOT	Bit NOT AND	
		BORNOT	Bit NOT OR	
		BLDNOT	Bit NOT load	
Total:	112			253

chanhvinh.com

The table below shows the format of instruction codes, operation, and execution states. They are described by using this format according to their classification.

Instruction	Instruction Code	Operation	Execution States	T Bit
Indicated by mnemonic.	Indicated in MSB ↔ LSB order.	Indicates summary of operation.	Value when no wait states are inserted.*1	Value of T bit after instruction is executed.
[Legend] Rm: Source register Rn: Destination register imm: Immediate data disp: Displacement*2	[Legend] mmmm: Source register nnnn: Destination register 0000: R0 0001: R1 ..... 1111: R15 iiii: Immediate data dddd: Displacement	[Legend] →, ←: Transfer direction (xx): Memory operand M/Q/T: Flag bits in SR &: Logical AND of each bit |: Logical OR of each bit ^: Exclusive logical OR of each bit ~: Logical NOT of each bit <<n: n-bit left shift >>n: n-bit right shift		Explanation of Symbols —: No change

Notes: 1. Instruction execution cycles: The execution cycles shown in the table are minimums. In practice, the number of instruction execution states will be increased in cases such as the following:

a. When there is a conflict between an instruction fetch and a data access

b. When the destination register of a load instruction (memory → register) is the same as the register used by the next instruction.

2. Depending on the operand size, displacement is scaled by ×1, ×2, or ×4. For details, refer to the SH-2A, SH2A-FPU Software Manual.

chanhvinh.com

2.4.2 Data Transfer Instructions

Table 2.11 Data Transfer Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
MOV	#imm,Rn	1110nnnniiiiiiii	imm → sign extension → Rn	1	—	Yes	Yes	Yes
MOV.W	@(disp,PC),Rn	1001nnnndddddddd	(disp × 2 + PC) → sign extension → Rn	1	—	Yes	Yes	Yes
MOV.L	@(disp,PC),Rn	1101nnnndddddddd	(disp × 4 + PC) → Rn	1	—	Yes	Yes	Yes
MOV	Rm,Rn	0110nnnnmmmm0011	Rm → Rn	1	—	Yes	Yes	Yes
MOV.B	Rm,@Rn	0010nnnnmmmm0000	Rm → (Rn)	1	—	Yes	Yes	Yes
MOV.W	Rm,@Rn	0010nnnnmmmm0001	Rm → (Rn)	1	—	Yes	Yes	Yes
MOV.L	Rm,@Rn	0010nnnnmmmm0010	Rm → (Rn)	1	—	Yes	Yes	Yes
MOV.B	@Rm,Rn	0110nnnnmmmm0000	(Rm) → sign extension → Rn	1	—	Yes	Yes	Yes
MOV.W	@Rm,Rn	0110nnnnmmmm0001	(Rm) → sign extension → Rn	1	—	Yes	Yes	Yes
MOV.L	@Rm,Rn	0110nnnnmmmm0010	(Rm) → Rn	1	—	Yes	Yes	Yes
MOV.B	Rm,@-Rn	0010nnnnmmmm0100	Rn-1 → Rn, Rm → (Rn)	1	—	Yes	Yes	Yes
MOV.W	Rm,@-Rn	0010nnnnmmmm0101	Rn-2 → Rn, Rm → (Rn)	1	—	Yes	Yes	Yes
MOV.L	Rm,@-Rn	0010nnnnmmmm0110	Rn-4 → Rn, Rm → (Rn)	1	—	Yes	Yes	Yes
MOV.B	@Rm+,Rn	0110nnnnmmmm0100	(Rm) → sign extension → Rn, Rm + 1 → Rm	1	—	Yes	Yes	Yes
MOV.W	@Rm+,Rn	0110nnnnmmmm0101	(Rm) → sign extension → Rn, Rm + 2 → Rm	1	—	Yes	Yes	Yes
MOV.L	@Rm+,Rn	0110nnnnmmmm0110	(Rm) → Rn, Rm + 4 → Rm	1	—	Yes	Yes	Yes
MOV.B	R0,@(disp,Rn)	10000000nnnndddd	R0 → (disp + Rn)	1	—	Yes	Yes	Yes
MOV.W	R0,@(disp,Rn)	10000001nnnndddd	R0 → (disp × 2 + Rn)	1	—	Yes	Yes	Yes
MOV.L	Rm,@(disp,Rn)	0001nnnnmmmmdddd	Rm → (disp × 4 + Rn)	1	—	Yes	Yes	Yes
MOV.B	@(disp,Rm),R0	10000100mmmmdddd	(disp + Rm) → sign extension → R0	1	—	Yes	Yes	Yes
MOV.W	@(disp,Rm),R0	10000101mmmmdddd	(disp × 2 + Rm) → sign extension → R0	1	—	Yes	Yes	Yes
MOV.L	@(disp,Rm),Rn	0101nnnnmmmmdddd	(disp × 4 + Rm) → Rn	1	—	Yes	Yes	Yes
MOV.B	Rm,@(R0,Rn)	0000nnnnmmmm0100	Rm → (R0 + Rn)	1	—	Yes	Yes	Yes
MOV.W	Rm,@(R0,Rn)	0000nnnnmmmm0101	Rm → (R0 + Rn)	1	—	Yes	Yes	Yes

chanhvinh.com

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2, SH2E	SH4	SH-2A
MOV.L	Rm,@(R0,Rn)	0000nnnnmmmm0110	Rm → (R0 + Rn)	1	—	Yes	Yes	Yes
MOV.B	@(R0,Rm),Rn	0000nnnnmmmm1100	(R0 + Rm) → sign extension → Rn	1	—	Yes	Yes	Yes
MOV.W	@(R0,Rm),Rn	0000nnnnmmmm1101	(R0 + Rm) → sign extension → Rn	1	—	Yes	Yes	Yes
MOV.L	@(R0,Rm),Rn	0000nnnnmmmm1110	(R0 + Rm) → Rn	1	—	Yes	Yes	Yes
MOV.B	R0,@(disp,GBR)	11000000dddddddd	R0 → (disp + GBR)	1	—	Yes	Yes	Yes
MOV.W	R0,@(disp,GBR)	11000001dddddddd	R0 → (disp × 2 + GBR)	1	—	Yes	Yes	Yes
MOV.L	R0,@(disp,GBR)	11000010dddddddd	R0 → (disp × 4 + GBR)	1	—	Yes	Yes	Yes
MOV.B	@(disp,GBR),R0	11000100dddddddd	(disp + GBR) → sign extension → R0	1	—	Yes	Yes	Yes
MOV.W	@(disp,GBR),R0	11000101dddddddd	(disp × 2 + GBR) → sign extension → R0	1	—	Yes	Yes	Yes
MOV.L	@(disp,GBR),R0	11000110dddddddd	(disp × 4 + GBR) → R0	1	—	Yes	Yes	Yes
MOV.B	R0,@Rn+	0100nnnn10001011	R0 → (Rn), Rn + 1 → Rn	1	—			Yes
MOV.W	R0,@Rn+	0100nnnn10011011	R0 → (Rn), Rn + 2 → Rn	1	—			Yes
MOV.L	R0,@Rn+	0100nnnn10101011	R0 → Rn), Rn + 4 → Rn	1	—			Yes
MOV.B	@-Rm,R0	0100mmmm11001011	Rm-1 → Rm, (Rm) → sign extension → R0	1	—			Yes
MOV.W	@-Rm,R0	0100mmmm11011011	Rm-2 → Rm, (Rm) → sign extension → R0	1	—			Yes
MOV.L	@-Rm,R0	0100mmmm11101011	Rm-4 → Rm, (Rm) → R0	1	—			Yes
MOV.B	Rm,@(disp12,Rn)	0011nnnnmmmm0001 0000dddddddddddd	Rm → (disp + Rn)	1	—			Yes
MOV.W	Rm,@(disp12,Rn)	0011nnnnmmmm0001 0001dddddddddddd	Rm → (disp × 2 + Rn)	1	—			Yes
MOV.L	Rm,@(disp12,Rn)	0011nnnnmmmm0001 0010dddddddddddd	Rm → (disp × 4 + Rn)	1	—			Yes
MOV.B	@(disp12,Rm),Rn	0011nnnnmmmm0001 0100dddddddddddd	(disp + Rm) → sign extension → Rn	1	—			Yes
MOV.W	@(disp12,Rm),Rn	0011nnnnmmmm0001 0101dddddddddddd	(disp × 2 + Rm) → sign extension → Rn	1	—			Yes

chanhvinh.com

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2, SH2E	SH4	SH-2A
MOV.L	@(disp12,Rm),Rn	0011nnnnmmmm0001 0110dddddddddddd	(disp × 4 + Rm) → Rn	1	—			Yes
MOVA	@(disp,PC),R0	11000111dddddddd	disp × 4 + PC → R0	1	—	Yes	Yes	Yes
MOVI20	#imm20,Rn	0000nnnniiii0000 iiiiiiiiiiiiiiii	imm → sign extension → Rn	1	—			Yes
MOVI20S	#imm20,Rn	0000nnnniiii0001 iiiiiiiiiiiiiiii	imm << 8 → sign extension → Rn	1	—			Yes
MOVML.L	Rm,@-R15	0100mmmm11110001	R15-4 → R15, Rm → (R15) R15-4 → R15, Rm-1 → (R15) : R15-4 → R15, R0 → (R15) Note: When Rm = R15, read Rm as PR	1 to 16	—			Yes
MOVML.L	@R15+,Rn	0100nnnn11110101	(R15) → R0, R15 + 4 → R15 (R15) → R1, R15 + 4 → R15 : (R15) → Rn Note: When Rn = R15, read Rn as PR	1 to 16	—			Yes
MOVMU.L	Rm,@-R15	0100mmmm11110000	R15-4 → R15, PR → (R15) R15-4 → R15, R14 → (R15) : R15-4 → R15, Rm → (R15) Note: When Rm = R15, read Rm as PR	1 to 16	—			Yes
MOVMU.L	@R15+,Rn	0100nnnn11110100	(R15) → Rn, R15 + 4 → R15 (R15) → Rn + 1, R15 + 4 → R15 : (R15) → R14, R15 + 4 → R15 (R15) → PR Note: When Rn = R15, read Rn as PR	1 to 16	—			Yes
MOVRT	Rn	0000nnnn00111001	~T → Rn	1	—			Yes
MOVT	Rn	0000nnnn00101001	T → Rn	1	—	Yes	Yes	Yes

chanhvinh.com

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2, SH2E	SH4	SH-2A
MOVU.B	@(disp12,Rm),Rn	0011nnnnmmmm0001 1000dddddddddddd	(disp + Rm) → zero extension → Rn	1	—			Yes
MOVU.W	@(disp12,Rm),Rn	0011nnnnmmmm0001 1001dddddddddddd	(disp × 2 + Rm) → zero extension → Rn	1	—			Yes
NOTT		0000000001101000	~T → T	1	Operation result			Yes
PREF	@Rn	0000nnnn10000011	(Rn) → operand cache	1	—		Yes	Yes
SWAP.B	Rm,Rn	0110nnnnmmmm1000	Rm → swap lower 2 bytes → Rn	1	—	Yes	Yes	Yes
SWAP.W	Rm,Rn	0110nnnnmmmm1001	Rm → swap upper and lower words → Rn	1	—	Yes	Yes	Yes
XTRCT	Rm,Rn	0010nnnnmmmm1101	Middle 32 bits of Rm:Rn → Rn	1	—	Yes	Yes	Yes

chanhvinh.com

2.4.3 Arithmetic Operation Instructions

Table 2.12 Arithmetic Operation Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2, SH2E	SH4	SH-2A
ADD	Rm,Rn	0011nnnnmmmm1100	Rn + Rm → Rn	1	—	Yes	Yes	Yes
ADD	#imm,Rn	0111nnnniiiiiiii	Rn + imm → Rn	1	—	Yes	Yes	Yes
ADDC	Rm,Rn	0011nnnnmmmm1110	Rn + Rm + T → Rn, carry → T	1	Carry	Yes	Yes	Yes
ADDV	Rm,Rn	0011nnnnmmmm1111	Rn + Rm → Rn, overflow → T	1	Over-flow	Yes	Yes	Yes
CMP/EQ	#imm,R0	10001000iiiiiiii	When R0 = imm, 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes
CMP/EQ	Rm,Rn	0011nnnnmmmm0000	When Rn = Rm, 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes
CMP/HS	Rm,Rn	0011nnnnmmmm0010	When Rn ≥ Rm (unsigned), 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes
CMP/GE	Rm,Rn	0011nnnnmmmm0011	When Rn ≥ Rm (signed), 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes
CMP/HI	Rm,Rn	0011nnnnmmmm0110	When Rn > Rm (unsigned), 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes
CMP/GT	Rm,Rn	0011nnnnmmmm0111	When Rn > Rm (signed), 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes
CMP/PL	Rn	0100nnnn00010101	When Rn > 0, 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes
CMP/PZ	Rn	0100nnnn00010001	When Rn ≥ 0, 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes
CMP/STR	Rm,Rn	0010nnnnmmmm1100	When any bytes are equal, 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes

chanhvinh.com

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2, SH2E	SH4	SH-2A
CLIPS.B	Rn	0100nnnn10010001	When Rn > (H'0000007F), (H'0000007F) → Rn, 1 → CS when Rn < (H'FFFFFF80), (H'FFFFFF80) → Rn, 1 → CS	1	—			Yes
CLIPS.W	Rn	0100nnnn10010101	When Rn > (H'00007FFF), (H'00007FFF) → Rn, 1 → CS When Rn < (H'FFFF8000), (H'FFFF8000) → Rn, 1 → CS	1	—			Yes
CLIPU.B	Rn	0100nnnn10000001	When Rn > (H'000000FF), (H'000000FF) → Rn, 1 → CS	1	—			Yes
CLIPU.W	Rn	0100nnnn10000101	When Rn > (H'0000FFFF), (H'0000FFFF) → Rn, 1 → CS	1	—			Yes
DIV1	Rm,Rn	0011nnnnmmmm0100	1-step division (Rn ÷ Rm)	1	Calcu-lation result	Yes	Yes	Yes
DIV0S	Rm,Rn	0010nnnnmmmm0111	MSB of Rn → Q, MSB of Rm → M, M ^ Q → T	1	Calcu-lation result	Yes	Yes	Yes
DIV0U		0000000000011001	0 → M/Q/T	1	0	Yes	Yes	Yes
DIVS	R0,Rn	0100nnnn10010100	Signed operation of Rn ÷ R0 → Rn 32 ÷ 32 → 32 bits	36	—			Yes
DIVU	R0,Rn	0100nnnn10000100	Unsigned operation of Rn ÷ R0 → Rn 32 ÷ 32 → 32 bits	34	—			Yes
DMULS.L	Rm,Rn	0011nnnnmmmm1101	Signed operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	2	—	Yes	Yes	Yes
DMULU.L	Rm,Rn	0011nnnnmmmm0101	Unsigned operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	2	—	Yes	Yes	Yes
DT	Rn	0100nnnn00010000	Rn – 1 → Rn When Rn is 0, 1 → T When Rn is not 0, 0 → T	1	Compa-rison result	Yes	Yes	Yes
EXTS.B	Rm,Rn	0110nnnnmmmm1110	Byte in Rm is sign-extended → Rn	1	—	Yes	Yes	Yes
EXTS.W	Rm,Rn	0110nnnnmmmm1111	Word in Rm is sign-extended → Rn	1	—	Yes	Yes	Yes

chanhvinh.com

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2, SH2E	SH4	SH-2A
EXTU.B	Rm,Rn	0110nnnnmmmm1100	Byte in Rm is zero-extended → Rn	1	—	Yes	Yes	Yes
EXTU.W	Rm,Rn	0110nnnnmmmm1101	Word in Rm is zero-extended → Rn	1	—	Yes	Yes	Yes
MAC.L	@Rm+,@Rn+	0000nnnnmmmm1111	Signed operation of (Rn) × (Rm) + MAC → MAC 32 × 32 + 64 → 64 bits	4	—	Yes	Yes	Yes
MAC.W	@Rm+,@Rn+	0100nnnnmmmm1111	Signed operation of (Rn) × (Rm) + MAC → MAC 16 × 16 + 64 → 64 bits	3	—	Yes	Yes	Yes
MUL.L	Rm,Rn	0000nnnnmmmm0111	Rn × Rm → MACL 32 × 32 → 32 bits	2	—	Yes	Yes	Yes
MULR	R0,Rn	0100nnnn10000000	R0 × Rn → Rn 32 × 32 → 32 bits	2				Yes
MULS.W	Rm,Rn	0010nnnnmmmm1111	Signed operation of Rn × Rm → MACL 16 × 16 → 32 bits	1	—	Yes	Yes	Yes
MULU.W	Rm,Rn	0010nnnnmmmm1110	Unsigned operation of Rn × Rm → MACL 16 × 16 → 32 bits	1	—	Yes	Yes	Yes
NEG	Rm,Rn	0110nnnnmmmm1011	0-Rm → Rn	1	—	Yes	Yes	Yes
NEGC	Rm,Rn	0110nnnnmmmm1010	0-Rm-T → Rn, borrow → T	1	Borrow	Yes	Yes	Yes
SUB	Rm,Rn	0011nnnnmmmm1000	Rn-Rm → Rn	1	—	Yes	Yes	Yes
SUBC	Rm,Rn	0011nnnnmmmm1010	Rn-Rm-T → Rn, borrow → T	1	Borrow	Yes	Yes	Yes
SUBV	Rm,Rn	0011nnnnmmmm1011	Rn-Rm → Rn, underflow → T	1	Over-flow	Yes	Yes	Yes

chanhvinh.com

2.4.4 Logic Operation Instructions

Table 2.13 Logic Operation Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
AND	Rm,Rn	0010nnnnmmmm1001	Rn & Rm → Rn	1	—	Yes	Yes	Yes
AND	#imm,R0	11001001iiiiiiii	R0 & imm → R0	1	—	Yes	Yes	Yes
AND.B	#imm,@(R0,GBR)	11001101iiiiiiii	(R0 + GBR) & imm → (R0 + GBR)	3	—	Yes	Yes	Yes
NOT	Rm,Rn	0110nnnnmmmm0111	~Rm → Rn	1	—	Yes	Yes	Yes
OR	Rm,Rn	0010nnnnmmmm1011	Rn | Rm → Rn	1	—	Yes	Yes	Yes
OR	#imm,R0	11001011iiiiiiii	R0 | imm → R0	1	—	Yes	Yes	Yes
OR.B	#imm,@(R0,GBR)	11001111iiiiiiii	(R0 + GBR) | imm → (R0 + GBR)	3	—	Yes	Yes	Yes
TAS.B	@Rn	0100nnnn00011011	When (Rn) is 0, 1 → T Otherwise, 0 → T, 1 → MSB of(Rn)	3	Test result	Yes	Yes	Yes
TST	Rm,Rn	0010nnnnmmmm1000	Rn & Rm When the result is 0, 1 → T Otherwise, 0 → T	1	Test result	Yes	Yes	Yes
TST	#imm,R0	11001000iiiiiiii	R0 & imm When the result is 0, 1 → T Otherwise, 0 → T	1	Test result	Yes	Yes	Yes
TST.B	#imm,@(R0,GBR)	11001100iiiiiiii	(R0 + GBR) & imm When the result is 0, 1 → T Otherwise, 0 → T	3	Test result	Yes	Yes	Yes
XOR	Rm,Rn	0010nnnnmmmm1010	Rn ^ Rm → Rn	1	—	Yes	Yes	Yes
XOR	#imm,R0	11001010iiiiiiii	R0 ^ imm → R0	1	—	Yes	Yes	Yes
XOR.B	#imm,@(R0,GBR)	11001110iiiiiiii	(R0 + GBR) ^ imm → (R0 + GBR)	3	—	Yes	Yes	Yes

chanhvinh.com

2.4.5 Shift Instructions

Table 2.14 Shift Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
ROTL	Rn	0100nnnn00000100	T ← Rn ← MSB	1	MSB	Yes	Yes	Yes
ROTR	Rn	0100nnnn00000101	LSB → Rn → T	1	LSB	Yes	Yes	Yes
ROTCL	Rn	0100nnnn00100100	T ← Rn ← T	1	MSB	Yes	Yes	Yes
ROTCR	Rn	0100nnnn00100101	T → Rn → T	1	LSB	Yes	Yes	Yes
SHAD	Rm,Rn	0100nnnnmmmm1100	When Rm ≥ 0, Rn << Rm → Rn When Rm < 0, Rn >> |Rm| → [MSB → Rn]	1	—		Yes	Yes
SHAL	Rn	0100nnnn00100000	T ← Rn ← 0	1	MSB	Yes	Yes	Yes
SHAR	Rn	0100nnnn00100001	MSB → Rn → T	1	LSB	Yes	Yes	Yes
SHLD	Rm,Rn	0100nnnnmmmm1101	When Rm ≥ 0, Rn << Rm → Rn When Rm < 0, Rn >> |Rm| → [0 → Rn]	1	—		Yes	Yes
SHLL	Rn	0100nnnn00000000	T ← Rn ← 0	1	MSB	Yes	Yes	Yes
SHLR	Rn	0100nnnn00000001	0 → Rn → T	1	LSB	Yes	Yes	Yes
SHLL2	Rn	0100nnnn00001000	Rn << 2 → Rn	1	—	Yes	Yes	Yes
SHLR2	Rn	0100nnnn00001001	Rn >> 2 → Rn	1	—	Yes	Yes	Yes
SHLL8	Rn	0100nnnn00011000	Rn << 8 → Rn	1	—	Yes	Yes	Yes
SHLR8	Rn	0100nnnn00011001	Rn >> 8 → Rn	1	—	Yes	Yes	Yes
SHLL16	Rn	0100nnnn00101000	Rn << 16 → Rn	1	—	Yes	Yes	Yes
SHLR16	Rn	0100nnnn00101001	Rn >> 16 → Rn	1	—	Yes	Yes	Yes

chanhvinh.com

2.4.6 Branch Instructions

Table 2.15 Branch Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
BF	label	10001011dddddddd	When T = 0, disp × 2 + PC → PC, When T = 1, nop	3/1*	—	Yes	Yes	Yes
BF/S	label	10001111dddddddd	Delayed branch When T = 0, disp × 2 + PC → PC, When T = 1, nop	2/1*	—	Yes	Yes	Yes
BT	label	10001001dddddddd	When T = 1, disp × 2 + PC → PC, When T = 0, nop	3/1*	—	Yes	Yes	Yes
BT/S	label	10001101dddddddd	Delayed branch When T = 1, disp × 2 + PC → PC, When T = 0, nop	2/1*	—	Yes	Yes	Yes
BRA	label	1010dddddddddddd	Delayed branch, disp × 2 + PC → PC	2	—	Yes	Yes	Yes
BRAF	Rm	0000mmmm00100011	Delayed branch, Rm + PC → PC	2	—	Yes	Yes	Yes
BSR	label	1011dddddddddddd	Delayed branch, PC → PR, disp × 2 + PC → PC	2	—	Yes	Yes	Yes
BSRF	Rm	0000mmmm00000011	Delayed branch, PC → PR, Rm + PC → PC	2	—	Yes	Yes	Yes
JMP	@Rm	0100mmmm00101011	Delayed branch, Rm → PC	2	—	Yes	Yes	Yes
JSR	@Rm	0100mmmm00001011	Delayed branch, PC → PR, Rm → PC	2	—	Yes	Yes	Yes
JSR/N	@Rm	0100mmmm01001011	PC-2 → PR, Rm → PC	3	—			Yes
JSR/N	@@(disp8,TBR)	10000011dddddddd	PC-2 → PR, (disp × 4 + TBR) → PC	5	—			Yes
RTS		0000000000001011	Delayed branch, PR → PC	2	—	Yes	Yes	Yes
RTS/N		0000000001101011	PR → PC	3	—			Yes
RTV/N	Rm	0000mmmm01111011	Rm → R0, PR → PC	3	—			Yes

Note: * One cycle when the program does not branch.

chanhvinh.com

2.4.7 System Control Instructions

Table 2.16 System Control Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2, SH2E	SH4	SH-2A
CLRT		0000000000001000	0 → T	1	0	Yes	Yes	Yes
CLRMAC		0000000000101000	0 → MACH,MACL	1	—	Yes	Yes	Yes
LDBANK	@Rm,R0	0100mmmm11100101	(Specified register bank entry) → R0	6	—			Yes
LDC	Rm,SR	0100mmmm00001110	Rm → SR	3	LSB	Yes	Yes	Yes
LDC	Rm,TBR	0100mmmm01001010	Rm → TBR	1	—			Yes
LDC	Rm,GBR	0100mmmm00011110	Rm → GBR	1	—	Yes	Yes	Yes
LDC	Rm,VBR	0100mmmm00101110	Rm → VBR	1	—	Yes	Yes	Yes
LDC.L	@Rm+,SR	0100mmmm00000111	(Rm) → SR, Rm + 4 → Rm	5	LSB	Yes	Yes	Yes
LDC.L	@Rm+,GBR	0100mmmm00010111	(Rm) → GBR, Rm + 4 → Rm	1	—	Yes	Yes	Yes
LDC.L	@Rm+,VBR	0100mmmm00100111	(Rm) → VBR, Rm + 4 → Rm	1	—	Yes	Yes	Yes
LDS	Rm,MACH	0100mmmm00001010	Rm → MACH	1	—	Yes	Yes	Yes
LDS	Rm,MACL	0100mmmm00011010	Rm → MACL	1	—	Yes	Yes	Yes
LDS	Rm,PR	0100mmmm00101010	Rm → PR	1	—	Yes	Yes	Yes
LDS.L	@Rm+,MACH	0100mmmm00000110	(Rm) → MACH, Rm + 4 → Rm	1	—	Yes	Yes	Yes
LDS.L	@Rm+,MACL	0100mmmm00010110	(Rm) → MACL, Rm + 4 → Rm	1	—	Yes	Yes	Yes
LDS.L	@Rm+,PR	0100mmmm00100110	(Rm) → PR, Rm + 4 → Rm	1	—	Yes	Yes	Yes
NOP		0000000000001001	No operation	1	—	Yes	Yes	Yes
RESBANK		0000000001011011	Bank → R0 to R14, GBR, MACH, MACL, PR	9*	—			Yes
RTE		0000000000101011	Delayed branch, stack area → PC/SR	6	—	Yes	Yes	Yes
SETT		0000000000011000	1 → T	1	1	Yes	Yes	Yes
SLEEP		0000000000011011	Sleep	5	—	Yes	Yes	Yes
STBANK	R0,@Rn	0100nnnn11100001	R0 → (specified register bank entry)	7	—			Yes
STC	SR,Rn	0000nnnn00000010	SR → Rn	2	—	Yes	Yes	Yes
STC	TBR,Rn	0000nnnn01001010	TBR → Rn	1	—			Yes

chanhvinh.com

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
STC	GBR,Rn	0000nnnn00010010	GBR → Rn	1	—	Yes	Yes	Yes
STC	VBR,Rn	0000nnnn00100010	VBR → Rn	1	—	Yes	Yes	Yes
STC.L	SR,@-Rn	0100nnnn00000011	Rn-4 → Rn, SR → (Rn)	2	—	Yes	Yes	Yes
STC.L	GBR,@-Rn	0100nnnn00010011	Rn-4 → Rn, GBR → (Rn)	1	—	Yes	Yes	Yes
STC.L	VBR,@-Rn	0100nnnn00100011	Rn-4 → Rn, VBR → (Rn)	1	—	Yes	Yes	Yes
STS	MACH,Rn	0000nnnn00001010	MACH → Rn	1	—	Yes	Yes	Yes
STS	MACL,Rn	0000nnnn00011010	MACL → Rn	1	—	Yes	Yes	Yes
STS	PR,Rn	0000nnnn00101010	PR → Rn	1	—	Yes	Yes	Yes
STS.L	MACH,@-Rn	0100nnnn00000010	Rn-4 → Rn, MACH → (Rn)	1	—	Yes	Yes	Yes
STS.L	MACL,@-Rn	0100nnnn00010010	Rn-4 → Rn, MACL → (Rn)	1	—	Yes	Yes	Yes
STS.L	PR,@-Rn	0100nnnn00100010	Rn-4 → Rn, PR → (Rn)	1	—	Yes	Yes	Yes
TRAPA	#imm	11000011iiiiiiii	PC/SR → stack area, (imm × 4 + VBR) → PC	5	—	Yes	Yes	Yes

Notes: 1. Instruction execution cycles: The execution cycles shown in the table are minimums. In practice, the number of instruction execution states in cases such as the following:

a. When there is a conflict between an instruction fetch and a data access

b. When the destination register of a load instruction (memory → register) is the same as the register used by the next instruction.

* In the event of bank overflow, the number of cycles is 19.

chanhvinh.com

2.4.8 Floating-Point Operation Instructions

Table 2.17 Floating-Point Operation Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2E	SH4	SH-2A/ SH2A-FPU
FABS	FRn	1111nnnn01011101	|FRn| → FRn	1	—	Yes	Yes	Yes
FABS	DRn	1111nnn001011101	|DRn| → DRn	1	—		Yes	Yes
FADD	FRm, FRn	1111nnnnmmmm0000	FRn + FRm → FRn	1	—	Yes	Yes	Yes
FADD	DRm, DRn	1111nnn0mmm00000	DRn + DRm → DRn	6	—		Yes	Yes
FCMP/EQ	FRm, FRn	1111nnnnmmmm0100	(FRn = FRm)? 1:0 → T	1	Comparison result	Yes	Yes	Yes
FCMP/EQ	DRm, DRn	1111nnn0mmm00100	(DRn = DRm)? 1:0 → T	2	Comparison result		Yes	Yes
FCMP/GT	FRm, FRn	1111nnnnmmmm0101	(FRn > FRm)? 1:0 → T	1	Comparison result	Yes	Yes	Yes
FCMP/GT	DRm, DRn	1111nnn0mmm00101	(DRn > DRm)? 1:0 → T	2	Comparison result		Yes	Yes
FCNVDS	DRm, FPUL	1111mmm010111101	(float) DRm → FPUL	2	—		Yes	Yes
FCNVSD	FPUL, DRn	1111nnn010101101	(double) FPUL → DRn	2	—		Yes	Yes
FDIV	FRm, FRn	1111nnnnmmmm0011	FRn/FRm → FRn	10	—	Yes	Yes	Yes
FDIV	DRm, DRn	1111nnn0mmm00011	DRn/DRm → DRn	23	—		Yes	Yes
FLDI0	FRn	1111nnnn10001101	0 × 00000000 → FRn	1	—	Yes	Yes	Yes
FLDI1	FRn	1111nnnn10011101	0 × 3F800000 → FRn	1	—	Yes	Yes	Yes
FLDS	FRm, FPUL	1111mmmm00011101	FRm → FPUL	1	—	Yes	Yes	Yes
FLOAT	FPUL,FRn	1111nnnn00101101	(float)FPUL → FRn	1	—	Yes	Yes	Yes
FLOAT	FPUL,DRn	1111nnn000101101	(double)FPUL → DRn	2	—		Yes	Yes
FMAC	FR0,FRm,FRn	1111nnnnmmmm1110	FR0 × FRm+FRn → FRn	1	—	Yes	Yes	Yes
FMOV	FRm, FRn	1111nnnnmmmm1100	FRm → FRn	1	—	Yes	Yes	Yes
FMOV	DRm, DRn	1111nnn0mmm01100	DRm → DRn	2	—		Yes	Yes

chanhvinh.com

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	SH4	SH-2A/ SH2A-FPU
FMOV.S	@(R0, Rm), FRn	1111nnnnmmmm0110	(R0 + Rm) → FRn	1	—	Yes	Yes	Yes
FMOV.D	@(R0, Rm), DRn	1111nnn0mmmm0110	(R0 + Rm) → DRn	2	—		Yes	Yes
FMOV.S	@Rm+, FRn	1111nnnnmmmm1001	(Rm) → FRn, Rm+=4	1	—	Yes	Yes	Yes
FMOV.D	@Rm+, DRn	1111nnn0mmmm1001	(Rm) → DRn, Rm += 8	2	—		Yes	Yes
FMOV.S	@Rm, FRn	1111nnnnmmmm1000	(Rm) → FRn	1	—	Yes	Yes	Yes
FMOV.D	@Rm, DRn	1111nnn0mmmm1000	(Rm) → DRn	2	—		Yes	Yes
FMOV.S	@(disp12,Rm),FRn	0011nnnnmmmm0001 0111dddddddddddd	(disp × 4 + Rm) → FRn	1	—			Yes
FMOV.D	@(disp12,Rm),DRn	0011nnn0mmmm0001 0111dddddddddddd	(disp × 8 + Rm) → DRn	2	—			Yes
FMOV.S	FRm, @(R0,Rn)	1111nnnnmmmm0111	FRm → (R0 + Rn)	1	—	Yes	Yes	Yes
FMOV.D	DRm, @(R0,Rn)	1111nnnnmmm00111	DRm → (R0 + Rn)	2	—		Yes	Yes
FMOV.S	FRm, @-Rn	1111nnnnmmmm1011	Rn-=4, FRm → (Rn)	1	—	Yes	Yes	Yes
FMOV.D	DRm, @-Rn	1111nnnnmmm01011	Rn-=8, DRm → (Rn)	2	—		Yes	Yes
FMOV.S	FRm, @Rn	1111nnnnmmmm1010	FRm → (Rn)	1	—	Yes	Yes	Yes
FMOV.D	DRm, @Rn	1111nnnnmmm01010	DRm → (Rn)	2	—		Yes	Yes
FMOV.S	FRm, @(disp12,Rn)	0011nnnnmmmm0001 0011ddddddddddd	FRm → (disp × 4 + Rn)	1	—			Yes
FMOV.D	DRm, @(disp12,Rn)	0011nnnnmmm00001 0011ddddddddddd	DRm → (disp × 8 + Rn)	2	—			Yes
FMUL	FRm, FRn	1111nnnnmmmm0010	FRn × FRm → FRn	1	—	Yes	Yes	Yes
FMUL	DRm, DRn	1111nnn0mmm00010	DRn × DRm → DRn	6	—		Yes	Yes
FNEG	FRn	1111nnnn01001101	-FRn → FRn	1	—	Yes	Yes	Yes
FNEG	DRn	1111nnn001001101	-DRn → DRn	1	—		Yes	Yes
FSCHG		1111001111111101	FPSCR.SZ=~FPSCR.SZ	1	—		Yes	Yes
FSQRT	FRn	1111nnnn01101101	√FRn → FRn	9	—		Yes	Yes
FSQRT	DRn	1111nnn001101101	√DRn → DRn	22	—		Yes	Yes
FSTS	FPUL,FRn	1111nnnn00001101	FPUL → FRn	1	—	Yes	Yes	Yes
FSUB	FRm, FRn	1111nnnnmmmm0001	FRn-FRm → FRn	1	—	Yes	Yes	Yes

chanhvinh.com

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	SH4	SH-2A/ SH2A-FPU
FSUB	DRm, DRn	1111nnn0mmm00001	DRn-DRm → DRn	6	—		Yes	Yes
FTRC	FRm, FPUL	1111mmmm00111101	(long)FRm → FPUL	1	—	Yes	Yes	Yes
FTRC	DRm, FPUL	1111mmm000111101	(long)DRm → FPUL	2	—		Yes	Yes

2.4.9 FPU-Related CPU Instructions

Table 2.18 FPU-Related CPU Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	SH4	SH-2A/ SH2A-FPU
LDS	Rm,FPSCR	0100mmmm01101010	Rm → FPSCR	1	—	Yes	Yes	Yes
LDS	Rm,FPUL	0100mmmm01011010	Rm → FPUL	1	—	Yes	Yes	Yes
LDS.L	@Rm+, FPSCR	0100mmmm01100110	(Rm) → FPSCR, Rm+=4	1	—	Yes	Yes	Yes
LDS.L	@Rm+, FPUL	0100mmmm01010110	(Rm) → FPUL, Rm+=4	1	—	Yes	Yes	Yes
STS	FPSCR, Rn	0000nnnn01101010	FPSCR → Rn	1	—	Yes	Yes	Yes
STS	FPUL,Rn	0000nnnn01011010	FPUL → Rn	1	—	Yes	Yes	Yes
STS.L	FPSCR,@-Rn	0100nnnn01100010	Rn-=4, FPCSR → (Rn)	1	—	Yes	Yes	Yes
STS.L	FPUL,@-Rn	0100nnnn01010010	Rn-=4, FPUL → (Rn)	1	—	Yes	Yes	Yes

chanhvinh.com

2.4.10 Bit Manipulation Instructions

Table 2.19 Bit Manipulation Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E SH4	SH-2A
BAND.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0100dddddddddddd	(imm of (disp + Rn)) & T →	3	Operation result		Yes
BANDNOT.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 1100dddddddddddd	~(imm of (disp + Rn)) & T → T	3	Operation result		Yes
BCLR.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0000dddddddddddd	0 → (imm of (disp + Rn))	3	—		Yes
BCLR	#imm3,Rn	10000110nnnn0iii	0 → imm of Rn	1	—		Yes
BLD.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0011dddddddddddd	(imm of (disp + Rn)) →	3	Operation result		Yes
BLD	#imm3,Rn	10000111nnnn1iii	imm of Rn → T	1	Operation result		Yes
BLDNOT.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 1011dddddddddddd	~(imm of (disp + Rn)) → T	3	Operation result		Yes
BOR.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0101dddddddddddd	(imm of (disp + Rn)) | T → T	3	Operation result		Yes
BORNOT.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 1101dddddddddddd	~(imm of (disp + Rn)) | T → T	3	Operation result		Yes
BSET.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0001dddddddddddd	1 → (imm of (disp + Rn))	3	—		Yes
BSET	#imm3,Rn	10000110nnnn1iii	1 → imm of Rn	1	—		Yes
BST.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0010dddddddddddd	T → (imm of (disp + Rn))	3	—		Yes
BST	#imm3,Rn	10000111nnnn0iii	T → imm of Rn	1	—		Yes

chanhvinh.com

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
BXOR.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0110ddddddddddd	(imm of (disp + Rn)) ^ T → T	3	Operation result			Yes

chanhvinh.com

2.5 Processing States

The CPU has five processing states: reset, exception handling, bus-released, program execution, and power-down. Figure 2.6 shows the transitions between the states.

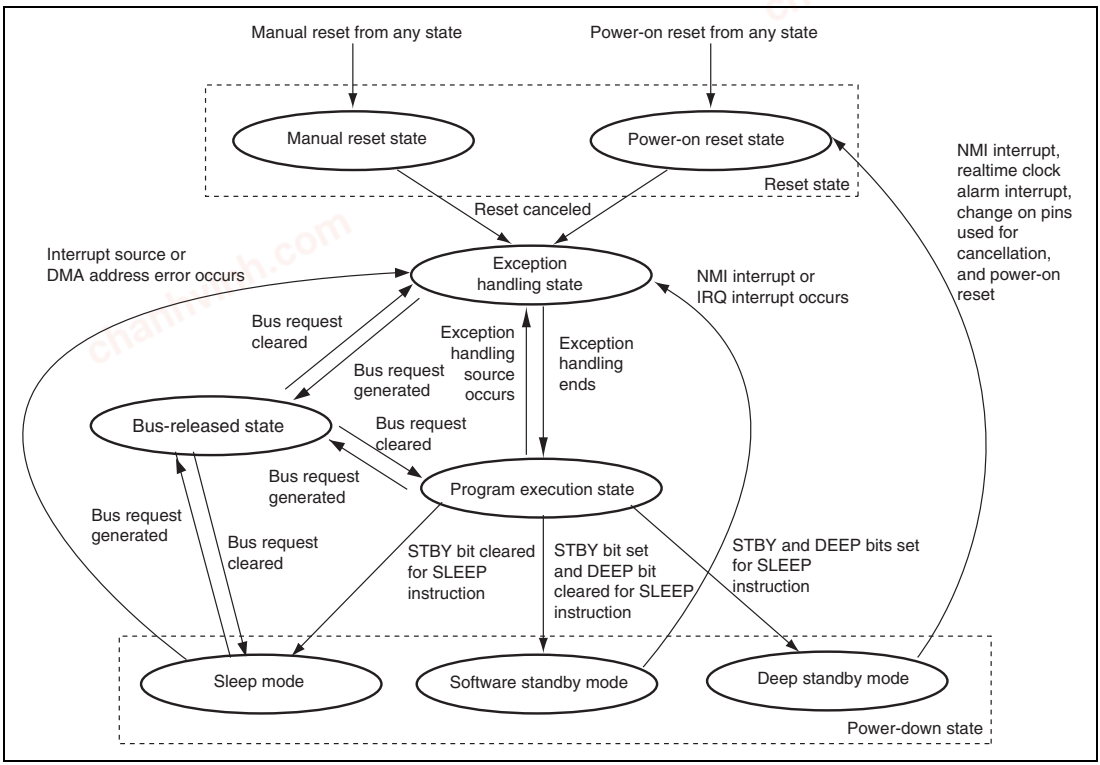


Figure 2.6 Transitions between Processing States

chanhvinh.com

(1) Reset State

In the reset state, the CPU is reset. There are two kinds of reset, power-on reset and manual reset.

(2) Exception Handling State

The exception handling state is a transient state that occurs when exception handling sources such as resets or interrupts alter the CPU's processing state flow.

For a reset, the initial values of the program counter (PC) (execution start address) and stack pointer (SP) are fetched from the exception handling vector table and stored; the CPU then branches to the execution start address and execution of the program begins.

For an interrupt, the stack pointer (SP) is accessed and the program counter (PC) and status register (SR) are saved to the stack area. The exception service routine start address is fetched from the exception handling vector table; the CPU then branches to that address and the program starts executing, thereby entering the program execution state.

(3) Program Execution State

In the program execution state, the CPU sequentially executes the program.

(4) Power-Down State

In the power-down state, the CPU stops operating to reduce power consumption. The SLEEP instruction places the CPU in sleep mode, software standby mode, or deep standby mode.

(5) Bus-Released State

In the bus-released state, the CPU releases bus to a device that has requested it.

chanhvinh.com

chanhvinh.com
chanhvinh.com
chanhvinh.com
chanhvinh.com

chanhvinh.com

Section 3 Floating-Point Unit (FPU)

3.1 Features

The FPU has the following features.

- Conforms to IEEE754 standard
- 16 single-precision floating-point registers (can also be referenced as eight double-precision registers)
- Two rounding modes: Round to nearest and round to zero
- Denormalization modes: Flush to zero
- Five exception sources: Invalid operation, divide by zero, overflow, underflow, and inexact
- Comprehensive instructions: Single-precision, double-precision, and system control

chanhvinh.com

3.2 Data Formats

3.2.1 Floating-Point Format

A floating-point number consists of the following three fields:

- Sign (s)
- Exponent (e)
- Fraction (f)

This LSI can handle single-precision and double-precision floating-point numbers, using the formats shown in figures 3.1 and 3.2.

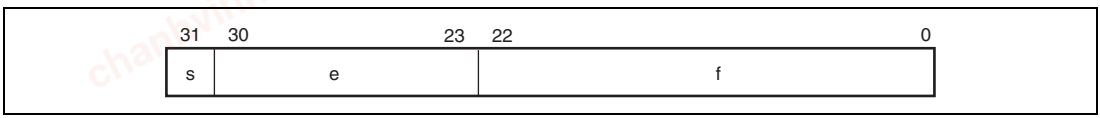


Figure 3.1 Format of Single-Precision Floating-Point Number

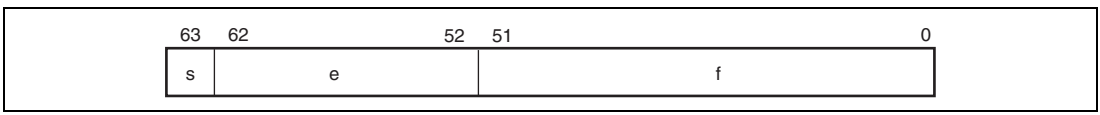


Figure 3.2 Format of Double-Precision Floating-Point Number

The exponent is expressed in biased form, as follows:

$$e = E + bias$$

The range of unbiased exponent E is $E_{min} - 1$ to $E_{max} + 1$. The two values $E_{min} - 1$ and $E_{max} + 1$ are distinguished as follows. $E_{min} - 1$ indicates zero (both positive and negative sign) and a denormalized number, and $E_{max} + 1$ indicates positive or negative infinity or a non-number (NaN). Table 3.1 shows E_{min} and E_{max} values.

chanhvinh.com

Table 3.1 Floating-Point Number Formats and Parameters

Parameter	Single-Precision	Double-Precision
Total bit width	32 bits	64 bits
Sign bit	1 bit	1 bit
Exponent field	8 bits	11 bits
Fraction field	23 bits	52 bits
Precision	24 bits	53 bits
Bias	+127	+1023
E_{max}	+127	+1023
E_{min}	–126	–1022

Floating-point number value v is determined as follows:

If $E = E_{max} + 1$ and $f \neq 0$, v is a non-number (NaN) irrespective of sign s
If $E = E_{max} + 1$ and $f = 0$, $v = (-1)^s$ (infinity) [positive or negative infinity]
If $E_{min} \leq E \leq E_{max}$, $v = (-1)^s 2^E$ (1.f) [normalized number]
If $E = E_{min} - 1$ and $f \neq 0$, $v = (-1)^s 2^{Emin}$ (0.f) [denormalized number]
If $E = E_{min} - 1$ and $f = 0$, $v = (-1)^s 0$ [positive or negative zero]

chanhvinh.com

Table 3.2 shows the ranges of the various numbers in hexadecimal notation.

Table 3.2 Floating-Point Ranges

Type	Single-Precision	Double-Precision
Signaling non-number	H'7FFF FFFF to H'7FC0 0000	H'7FFF FFFF FFFF FFFF to H'7FF8 0000 0000 0000
Quiet non-number	H'7FBF FFFF to H'7F80 0001	H'7FF7 FFFF FFFF FFFF to H'7FF0 0000 0000 0001
Positive infinity	H'7F80 0000	H'7FF0 0000 0000 0000
Positive normalized number	H'7F7F FFFF to H'0080 0000	H'7FEF FFFF FFFF FFFF to H'0010 0000 0000 0000
Positive denormalized number	H'007F FFFF to H'0000 0001	H'000F FFFF FFFF FFFF to H'0000 0000 0000 0001
Positive zero	H'0000 0000	H'0000 0000 0000 0000
Negative zero	H'8000 0000	H'8000 0000 0000 0000
Negative denormalized number	H'8000 0001 to H'807F FFFF	H'8000 0000 0000 0001 to H'800F FFFF FFFF FFFF
Negative normalized number	H'8080 0000 to H'FF7F FFFF	H'8010 0000 0000 0000 to H'FFEF FFFF FFFF FFFF
Negative infinity	H'FF80 0000	H'FFF0 0000 0000 0000
Quiet non-number	H'FF80 0001 to H'FFBF FFFF	H'FFF0 0000 0000 0001 to H'FFF7 FFFF FFFF FFFF
Signaling non-number	H'FFC0 0000 to H'FFFF FFFF	H'FFF8 0000 0000 0000 to H'FFFF FFFF FFFF FFFF

chanhvinh.com

3.2.2 Non-Numbers (NaN)

Figure 3.3 shows the bit pattern of a non-number (NaN). A value is NaN in the following case:

- Sign bit: Don't care
- Exponent field: All bits are 1
- Fraction field: At least one bit is 1

The NaN is a signaling NaN (sNaN) if the MSB of the fraction field is 1, and a quiet NaN (qNaN) if the MSB is 0.

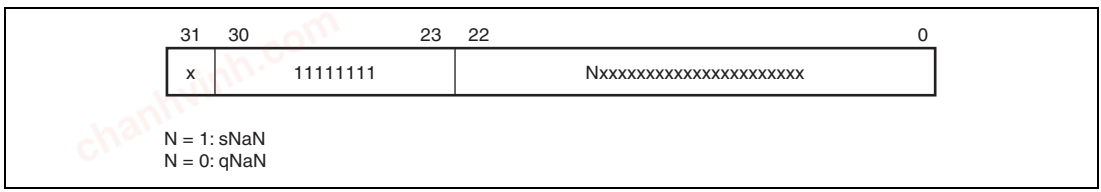


Figure 3.3 Single-Precision NaN Bit Pattern

An sNaN is input in an operation, except copy, FABS, and FNEG, that generates a floating-point value.

- When the EN.V bit in FPSCR is 0, the operation result (output) is a qNaN.
- When the EN.V bit in FPSCR is 1, an invalid operation exception will generate FPU exception processing. In this case, the contents of the operation destination register are unchanged.

If a qNaN is input in an operation that generates a floating-point value, and an sNaN has not been input in that operation, the output will always be a qNaN irrespective of the setting of the EN.V bit in FPSCR. An exception will not be generated in this case.

The qNAN values as operation results are as follows:

- Single-precision qNaN: H'7FBF FFFF
- Double-precision qNaN: H'7FF7 FFFF FFFF FFFF

See the individual instruction descriptions for details of floating-point operations when a non-number (NaN) is input.

chanhvinh.com

3.2.3 Denormalized Numbers

For a denormalized number floating-point value, the exponent field is expressed as 0, and the fraction field as a non-zero value.

In the SH2A-FPU, the DN bit in the status register FPSCR is always set to 1, therefore a denormalized number (source operand or operation result) is always flushed to 0 in a floating-point operation that generates a value (an operation other than copy, FNEG, or FABS).

When the DN bit in FPSCR is 0, a denormalized number (source operand or operation result) is processed as it is. See the individual instruction descriptions for details of floating-point operations when a denormalized number is input.

chanhvinh.com

3.3 Register Descriptions

3.3.1 Floating-Point Registers

Figure 3.4 shows the floating-point register configuration. There are sixteen 32-bit floating-point registers FPR0 to FPR15, referenced by specifying FR0 to FR15, DR0/2/4/6/8/10/12/14. The correspondence between FRPn and the reference name is determined by the PR and SZ bits in FPSCR. Refer figure 3.4.

1. Floating-point registers, FPRi (16 registers)
   FPR0 to FPR15
2. Single-precision floating-point registers, FRi (16 registers)
   FR0 to FR15 indicate FPR0 to FPR15
3. Double-precision floating-point registers or single-precision floating-point vector registers in pairs, DRi (8 registers)
   A DR register comprises two FR registers.
   DR0 = {FR0, FR1}, DR2 = {FR2, FR3}, DR4 = {FR4, FR5}, DR6 = {FR6, FR7},
   DR8 = {FR8, FR9}, DR10 = {FR10, FR11}, DR12 = {FR12, FR13}, DR14 = {FR14, FR15}

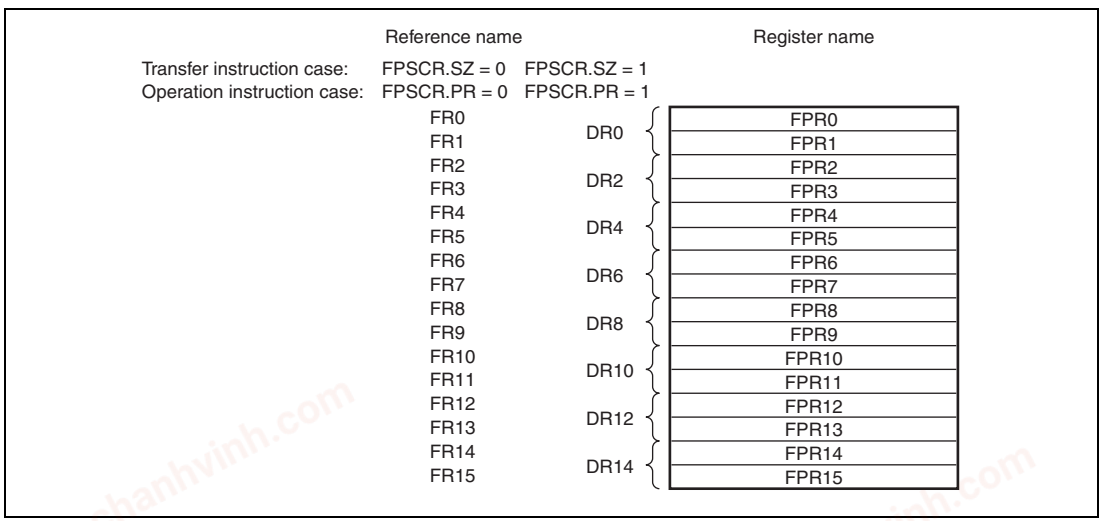


Figure 3.4 Floating-Point Registers

chanhvinh.com

3.3.2 Floating-Point Status/Control Register (FPSCR)

FPSCR is a 32-bit register that controls floating-point instructions, sets FPU exceptions, and selects the rounding mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	QIS	-	SZ	PR	DN	Cause	
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	R	R/W	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
	Cause				Enable					Flag					RM1	RM0
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

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	QIS	0	R/W	Nonnunerical Processing Mode 0: Processes qNaN or ±∞ as such 1: Treats qNaN or ±∞ as the same as sNaN (valid only when FPSCR.Enable.V = 1)
21	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
20	SZ	0	R/W	Transfer Size Mode 0: Data size of FMOV instruction is 32-bits 1: Data size of FMOV instruction is a 32-bit register pair (64 bits)
19	PR	0	R/W	Precision Mode 0: Floating-point instructions are executed as single-precision operations 1: Floating-point instructions are executed as double-precision operations (graphics support instructions are undefined)
18	DN	1	R	Denormalization Mode (Always fixed to 1 in SH2A-FPU) 1: Denormalized number is treated as zero

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
17 to 12	Cause	H'00	R/W	FPU Exception Cause Field
11 to 7	Enable	H'00	R/W	FPU Exception Enable Field
6 to 2	Flag	H'00	R/W	FPU Exception Flag Field Each time floating-point operation instruction is executed, the FPU exception cause field is cleared to 0 first. When an FPU exception on floating-point operation occurs, the bits corresponding to the FPU exception cause field and FPU exception flag field are set to 1. The FPU exception flag field remains set to 1 until it is cleared to 0 by software. As the bits corresponding to FPU exception enable filed are sets to 1, FPU exception processing occurs. For bit allocations of each field, see table 3.3.
1	RM1	0	R/W	Rounding Mode
0	RM0	1	R/W	These bits select the rounding mode. 00: Round to Nearest 01: Round to Zero 10: Reserved 11: Reserved

Table 3.3 Bit Allocation for FPU Exception Handling

Field Name		FPU Error (E)	Invalid Operation (V)	Division by Zero (Z)	Overflow (O)	Underflow (U)	Inexact (I)
Cause	FPU exception cause field	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12
Enable	FPU exception enable field	None	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7
Flag	FPU exception flag field	None	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2

Note: No FPU error occurs in the SH2A-FPU.

chanhvinh.com

3.3.3 Floating-Point Communication Register (FPUL)

Information is transferred between the FPU and CPU via FPUL. FPUL is a 32-bit system register that is accessed from the CPU side by means of LDS and STS instructions. For example, to convert the integer stored in general register R1 to a single-precision floating-point number, the processing flow is as follows:

R1 → (LDS instruction) → FPUL → (single-precision FLOAT instruction) → FR1

chanhvinh.com

3.4 Rounding

In a floating-point instruction, rounding is performed when generating the final operation result from the intermediate result. Therefore, the result of combination instructions such as FMAC will differ from the result when using a basic instruction such as FADD, FSUB, or FMUL. Rounding is performed once in FMAC, but twice in FADD, FSUB, and FMUL.

Which of the two rounding methods is to be used is determined by the RM bits in FPSCR.

FPSCR.RM[1:0] = 00: Round to Nearest
FPSCR.RM[1:0] = 01: Round to Zero

(1) Round to Nearest

The operation result is rounded to the nearest expressible value. If there are two nearest expressible values, the one with an LSB of 0 is selected.

If the unrounded value is $2^{Emax} (2 - 2^{-P})$ or more, the result will be infinity with the same sign as the unrounded value. The values of Emax and P, respectively, are 127 and 24 for single-precision, and 1023 and 53 for double-precision.

(2) Round to Zero

The digits below the round bit of the unrounded value are discarded.

If the unrounded value is larger than the maximum expressible absolute value, the value will become the maximum expressible absolute value.

chanhvinh.com

3.5 FPU Exceptions

3.5.1 FPU Exception Sources

FPU exceptions may occur on floating-point operation instruction and the exception sources are as follows:

- FPU error (E): When FPSCR.DN = 0 and a denormalized number is input (No error occurs in the SH2A-FPU)
- Invalid operation (V): In case of an invalid operation, such as NaN input
- Division by zero (Z): Division with a zero divisor
- Overflow (O): When the operation result overflows
- Underflow (U): When the operation result underflows
- Inexact exception (I): When overflow, underflow, or rounding occurs

The FPU exception cause field in FPSCR contains bits corresponding to all of above sources E, V, Z, O, U, and I, and the FPU exception flag and enable fields in FPSCR contain bits corresponding to sources V, Z, O, U, and I, but not E. Thus, FPU errors cannot be disabled.

When an FPU exception occurs, the corresponding bit in the FPU exception cause field is set to 1, and 1 is added to the corresponding bit in the FPU exception flag field. When an FPU exception does not occur, the corresponding bit in the FPU exception cause field is cleared to 0, but the corresponding bit in the FPU exception flag field remains unchanged.

3.5.2 FPU Exception Handling

FPU exception handling is initiated in the following cases:

- FPU error (E): FPSCR.DN = 0 and a denormalized number is input (No error occurs in the SH2A-FPU)
- Invalid operation (V): FPSCR.Enable.V = 1 and invalid operation
- Division by zero (Z): FPSCR.Enable.Z = 1 and division with a zero divisor
- Overflow (O): FPSCR.Enable.O = 1 and instruction with possibility of operation result overflow
- Underflow (U): FPSCR.Enable.U = 1 and instruction with possibility of operation result underflow
- Inexact exception (I): FPSCR.Enable.I = 1 and instruction with possibility of inexact operation result

chanhvinh.com

These possibilities of each exceptional handling on floating-point operation are shown in the individual instruction descriptions. All exception events that originate in the floating-point operation are assigned as the same FPU exceptional handling event. The meaning of an exception generated by floating-point operation is determined by software by reading from FPSCR and interpreting the information it contains. Also, the destination register is not changed when FPU exception handling operation occurs.

Except for the above, the FPU disables exception handling. In every processing, the bit corresponding to source V, Z, O, U, or I is set to 1, and a default value is generated as the operation result.

- Invalid operation (V): qNaN is generated as the result.
- Division by zero (Z): Infinity with the same sign as the unrounded value is generated.
- Overflow (O):
  When rounding mode = RZ, the maximum normalized number, with the same sign as the unrounded value, is generated.
  When rounding mode = RN, infinity with the same sign as the unrounded value is generated.
- Underflow (U):
  Zero with the same sign as the unrounded value is generated.
- Inexact exception (I): An inexact result is generated.

chanhvinh.com

chanhvinh.com
chanhvinh.com
chanhvinh.com
chanhvinh.com

chanhvinh.com

Section 4 Boot Mode

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

4.1 Features

- Four boot modes
  Boot mode 0: Boots the LSI from the memory connected to the CS0 space
  Boot mode 1: Boots the LSI from the serial flash memory (high-speed communication)
  Boot mode 2: Boots the LSI from the NAND flash memory
  Boot mode 3: Boots the LSI from the serial flash memory (low-speed communication)

4.2 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 4.1.

Table 4.1 External Pin (MD_BOOT1 and MD_BOOT0) Settings and Corresponding Boot Modes

MD_BOOT1	MD_BOOT0	Boot Mode
0	0	Boots the LSI from the memory connected to the CS0 space (boot mode 0).
0	1	Boots the LSI, through high-speed communication, from the serial flash memory connected to channel 0 of the Renesas serial peripheral interface (boot mode 1).
1	0	Boots the LSI from the NAND flash memory connected to the NAND flash memory controller (boot mode 2).
1	1	Boots the LSI, through low-speed communication, from the serial flash memory connected to channel 0 of the Renesas serial peripheral interface (boot mode 3).

chanhvinh.com

4.3 Operation

4.3.1 Boot Mode 0

In boot mode 0, this LSI is booted from the memory connected to the CS0 space. In this mode, this LSI operates as follows:

After the power-on reset is canceled, the initial value (execution start address) of the program counter (PC) and the initial value of the stack pointer (SP) are fetched from the exception handling vector table located in the memory connected to the CS0 space, then program execution is started.

4.3.2 Boot Modes 1 and 3

In boot modes 1 and 3, booting up is from serial flash memory, which is connected to channel 0 of the Renesas serial peripheral interface. The flow of initiation in boot mode is as described below.

(1) Execution from on-Chip ROM of the Program for Boot Initiation

After release from the power-on reset state, the CPU executes the boot initiation program that has been stored in on-chip ROM (and is not publicly disclosed).

(2) Transfer of the Loader Program

Starting with transfer from the respective first locations, the 8-KB loader program is transferred from serial flash memory, which is connected to channel 0 of the Renesas serial peripheral interface, to high-speed on-chip RAM.

Transfer proceeds at 1/2 of the rate of the bus clock (Bϕ) in boot mode 1 and at 1/4 of the rate of the bus clock in boot mode 3. Decide on the boot mode in accord with the connected serial flash memory.

Once transfer of the loader program has been completed, execution by the CPU jumps to high-speed on-chip RAM so that it can start executing the transferred loader program.

(3) Transfer of an Application Program (as Desired)

The loader program employs the Renesas serial peripheral interface to transfer the data to be deployed from serial flash memory to on-chip RAM or external RAM.

chanhvinh.com

Figure 4.1 is a schematic view of the specification for boot modes 1 and 3.

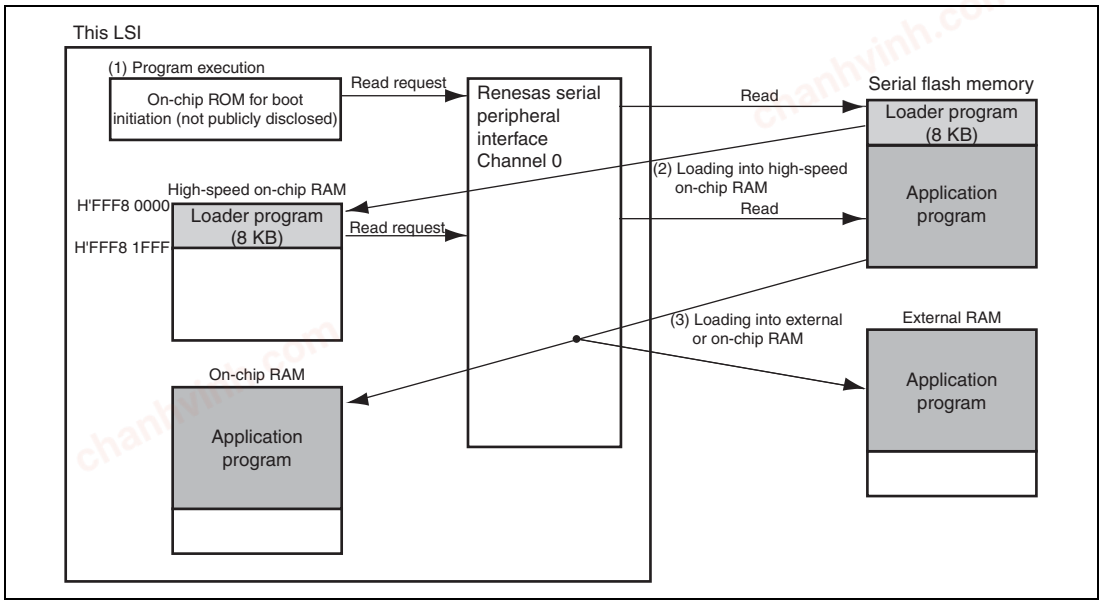


Figure 4.1 Schematic View of Specification for Boot Modes 1 and 3

4.3.3 Boot Mode 2

In boot mode 2, booting up is from NAND flash memory, which is connected to the NAND flash memory controller. Suitable NAND flash memory has a large block size (2048 + 64) and takes five-byte addresses (has a capacity of 2 GB or greater). Furthermore, errors in up to four locations are correctable. The flow of initiation for a NAND-flash boot is as described below.

(1) Execution from on-Chip ROM of the Program for Boot Initiation

After release from the power-on reset state, the CPU executes the boot initiation program that has been stored in on-chip ROM (and is not publicly disclosed).

(2) Transfer of the Loader Program

Starting with transfer from the respective first locations, the 8-KB loader program is transferred from NAND flash memory, which is connected to the NAND flash memory controller, to high-speed on-chip RAM. Transfer and checking by the loader program proceed as follows.

chanhvinh.com

(a) A search is conducted to find the block which holds the loader program. Block addresses 0 to 1023 (max.)

(b) The 8-KB (16-sector) loader program is read out and transferred to high-speed on-chip RAM.

(c) ECC checking

(d) Error correction (up to four locations)

Once transfer of the loader program has been completed, execution by the CPU jumps to high-speed on-chip RAM so that it can start executing the transferred loader program.

(3) Transfer of an Application Program (as Desired)

The loader program employs the NAND flash memory controller to transfer the data to be deployed from NAND flash memory to on-chip RAM or external RAM.

Figure 4.2 is a schematic view of the specifications for boot mode 2.

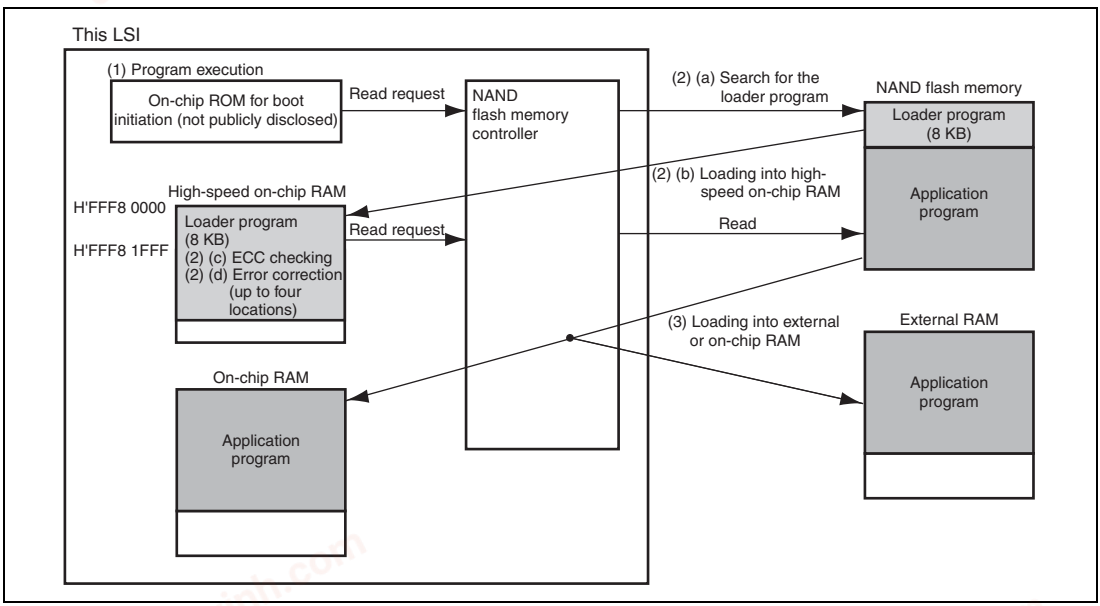


Figure 4.2 Schematic View of Specification for Boot Mode 2

chanhvinh.com

Figure 4.3 shows the configuration of the sector data that are read out as the loader program.

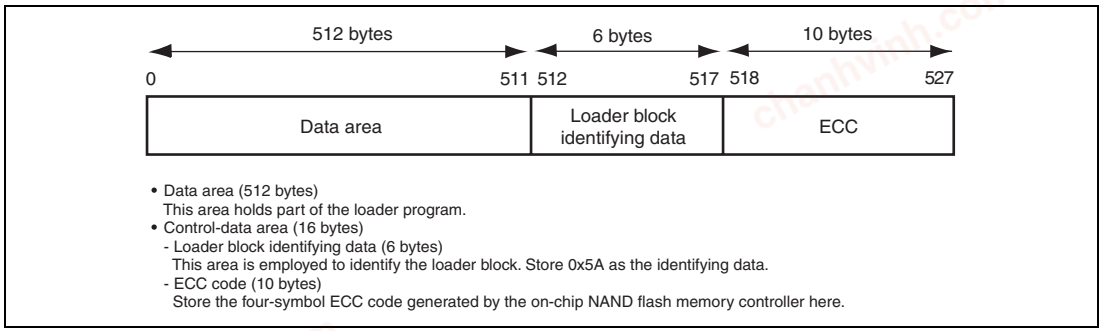


Figure 4.3 Sector Configuration for the Loader Program

Figure 4.4 shows the locations where the loader program is stored.

Store the loader program in sectors 0 to 15 of the loader block. Read out the loader program in sectors.

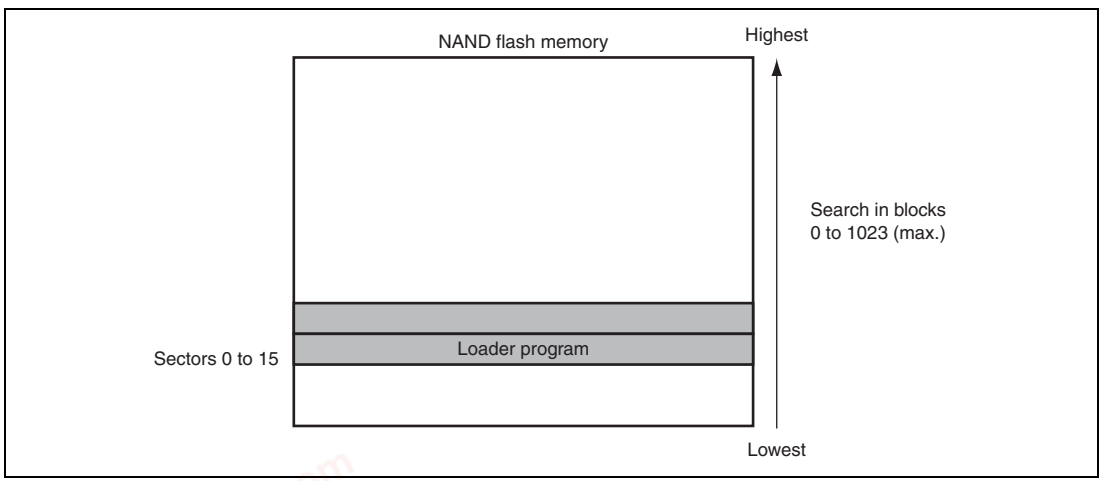


Figure 4.4 Locations where the Loader Program is Stored

chanhvinh.com

4.4 Notes

4.4.1 Boot Related Pins

The initial states and output states in deep standby mode of the pins related to CS0 space memory read, NAND flash memory controller, and channel 0 of the Renesas serial peripheral interface are different in each boot mode.

For details, refer to section 9, Bus State Controller, section 32, General Purpose I/O Ports, and section 33, Power-Down Modes.

chanhvinh.com

Section 5 Clock Pulse Generator

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

5.1 Features

- Four clock operating modes

  The mode is selected from among the four clock operating modes based on the frequency range to be used and the input clock type: the clock from crystal resonator or the clock for USB 2.0 host/function module.
- Three clocks generated independently

  A CPU clock (Iϕ) for the CPU and cache; a peripheral clock (Pϕ) for the on-chip peripheral modules; a bus clock (Bϕ = CKIO) for the external bus interface
- Frequency change function

  CPU and peripheral clock frequencies can be changed independently using the PLL (phase locked loop) circuits and divider circuits within this module. Frequencies are changed by software using frequency control register (FRQCR) settings.
- 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 33, Power-Down Modes.

chanhvinh.com

Figure 5.1 shows a block diagram of the clock pulse generator.

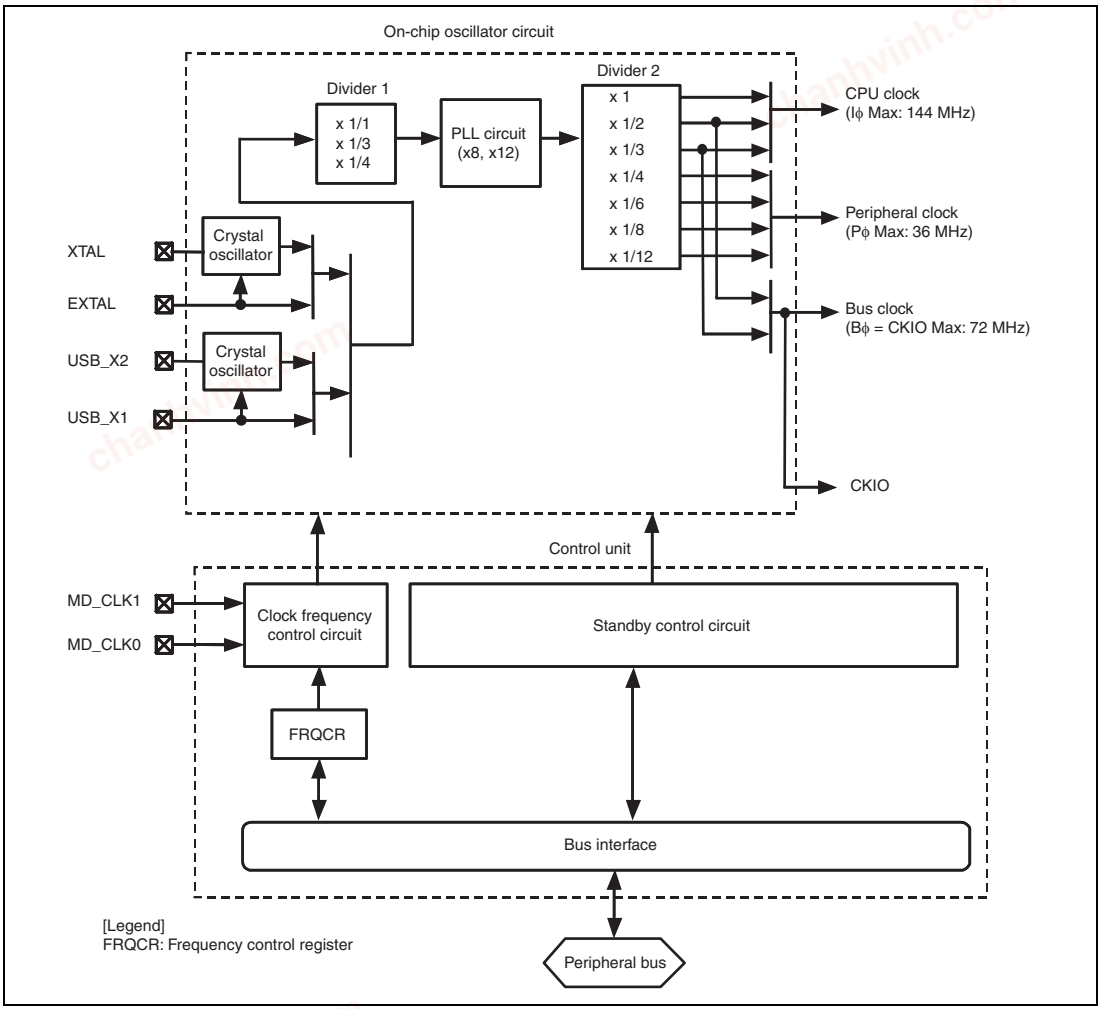


Figure 5.1 Block Diagram

chanhvinh.com

The blocks of this module function as follows:

(1) Crystal Oscillator

The crystal oscillator is used in which the crystal resonator is connected to the XTAL/EXTAL pin or USB_X1/USB_X2 pin. One of them is selected according to the clock operating mode.

(2) Divider 1

Divider 1 divides the output from the crystal oscillator or the external clock input. The division ratio depends on the clock operating mode.

(3) PLL Circuit

PLL circuit multiplies the frequency of the output from the divider 1. The multiplication ratio depends on the clock operating mode.

(4) Divider 2

Divider 2 generates a clock signal whose operating frequency can be used for the CPU clock, the peripheral clock, and the bus clock. The division ratio of the bus clock depends on the clock operating mode. The division ratio of the CPU clock and the peripheral clock is set by the frequency control register.

(5) Clock Frequency Control Circuit

The clock frequency control circuit controls the clock frequency using the MD_CLK0 and MD_CLK1 pins and the frequency control register (FRQCR).

(6) Standby Control Circuit

The standby control circuit controls the states of the on-chip oscillation circuit and other modules during clock switching, or sleep, 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 33, Power-Down Modes.

(7) 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 and the frequency division ratio of the CPU clock and the peripheral clock (Pϕ).

chanhvinh.com

5.2 Input/Output Pins

Table 5.1 lists the clock pulse generator pins and their functions.

Table 5.1 Pin Configuration and Functions of the Clock Pulse Generator

Pin Name	Symbol	I/O	Function (Clock Operating Modes 0 and 2)	Function (Clock Operating Modes 1 and 3)
Mode control pins	MD_CLK0	Input	Sets the clock operating mode.	
	MD_CLK1	Input	Sets the clock operating mode.	
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.)	Leave this pin open.
	EXTAL	Input	Connected to the crystal resonator or used to input external clock.	Fix this pin (Connect it to pull-up/down resistor, power supply, or ground.)
Clock output pin	CKIO	Output	Clock output pin.	Clock output pin
Crystal input/output pins for USB 2.0 host/function module (clock input pins)	USB_X1	Input	Connected to the crystal resonator to input the clock for the USB 2.0 host/function module only, or used to input external clock. When the USB 2.0 host/function module is not used, this pin should be fixed (connected to pull-up/down resistor, power supply or ground).	Connected to the crystal resonator to input the clock for both the USB 2.0 host/function module and the LSI, or used to input external clock.
	USB_X2	Output	Connected to the crystal resonator for the USB 2.0 host/function module. (Leave this pin open when the crystal resonator is not in use.)	Connected to the crystal resonator for both the USB 2.0 host/function module and the LSI. (Leave this pin open when the crystal resonator is not in use.)

chanhvinh.com

5.3 Clock Operating Modes

Table 5.2 shows the relationship between the combinations of the mode control pins (MD_CLK1 and MD_CLK0) and the clock operating modes. Table 5.3 shows the usable frequency ranges in the clock operating modes.

Table 5.2 Clock Operating Modes

	Pin Values		Clock I/O				
Mode	MD_CLK1	MD_CLK0	Source	Output	Divider 1	PLL Circuit On/Off	CKIO Frequency
0	0	0	EXTAL or crystal resonator	CKIO	1	ON (×12)	(EXTAL or crystal resonator) × 4
1	0	1	USB_X1 or crystal resonator	CKIO	1/4	ON (×12)	(USB_X1 or crystal resonator)
2	1	0	EXTAL or crystal resonator	CKIO	1	ON (×8)	(EXTAL or crystal resonator) × 4
3	1	1	USB_X1 or crystal resonator	CKIO	1/3	ON (×8)	(USB_X1 or crystal resonator) × 4/3

- Mode 0

  In mode 0, clock is 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 12 MHz. The frequency range of CKIO is from 40 to 48 MHz. To reduce current supply, fix the USB_X1 pin (connect it to a pull-up/down resistor, the power supply, or the ground) and open the USB_X2 pin when the USB 2.0 host/function module is not used.

- Mode 1

  In mode 1, clock is input from the USB_X1 pin or the crystal oscillator. The external clock is input through this pin and waveform is shaped in the PLL circuit. Then the frequency is multiplied according to the frequency control register setting before the clock is supplied to the LSI. The frequency of CKIO is the same as that of the input clock (USB_X1/crystal resonator) (48 MHz). To reduce current supply, fix the EXTAL pin (connect it to a pull-up/down resistor, the power supply, or the ground) and open the XTAL pin when the LSI is used in mode 1.

chanhvinh.com

- Mode 2

  In mode 2, clock is 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 18 MHz. The frequency range of CKIO is from 40 to 72 MHz. To reduce current supply, fix the USB_X1 pin (connect it to a pull-up/down resistor, the power supply, or the ground) and open the USB_X2 pin when the USB 2.0 host/function module is not used.

- Mode 3

  In mode 3, clock is input from the USB_X1 pin or the crystal oscillator. The external clock is input through this pin and waveform is shaped in the PLL circuit. Then the frequency is multiplied according to the frequency control register setting before the clock is supplied to the LSI. The frequency of CKIO is the same as that of the input clock (USB_X1/crystal resonator) multiplied by 4/3, (64 MHz). To reduce current supply, fix the EXTAL pin (connect it to a pull-up/down resistor, the power supply, or the ground) and open the XTAL pin when the LSI is used in mode 3.

chanhvinh.com

Table 5.3 Relationship between Clock Operating Mode and Frequency Range

Clock Operating Mode	FRQCR Setting*1	PLL Frequency Multiplier PLL Circuit	Ratio of Internal Clock Frequencies (I:B:P)*2	Selectable Frequency Range (MHz) Input Clock*3	Output Clock (CKIO Pin)	CPU Clock (Iϕ)	Bus Clock (Bϕ)	Peripheral Clock (Pϕ)
0	H'x104	ON (× 12)	12:4:2	10 to 12	40 to 48	120 to 144	40 to 48	20 to 24
	H'x106	ON (× 12)	12:4:1	10 to 12	40 to 48	120 to 144	40 to 48	10 to 12
	H'x124	ON (× 12)	4:4:2	10 to 12	40 to 48	40 to 48	40 to 48	20 to 24
	H'x126	ON (× 12)	4:4:1	10 to 12	40 to 48	40 to 48	40 to 48	10 to 12
1	H'x104	ON (× 12)	3:1:1/2	48	48	144	48	24
	H'x106	ON (× 12)	3:1:1/4	48	48	144	48	12
	H'x124	ON (× 12)	1:1:1/2	48	48	48	48	24
	H'x126	ON (× 12)	1:1:1/4	48	48	48	48	12
2	H'x003	ON (× 8)	8:4:2	10 to 18	40 to 72	80 to 144	40 to 72	20 to 36
	H'x004	ON (× 8)	8:4:4/3	10 to 18	40 to 72	80 to 144	40 to 72	13.3 to 24
	H'x005	ON (× 8)	8:4:1	10 to 18	40 to 72	80 to 144	40 to 72	10 to 18
	H'x006	ON (× 8)	8:4:2/3	10 to 18	40 to 72	80 to 144	40 to 72	6.7 to 12
	H'x013	ON (× 8)	4:4:2	10 to 18	40 to 72	40 to 72	40 to 72	20 to 36
	H'x014	ON (× 8)	4:4:4/3	10 to 18	40 to 72	40 to 72	40 to 72	13.3 to 24
	H'x015	ON (× 8)	4:4:1	10 to 18	40 to 72	40 to 72	40 to 72	10 to 18
	H'x016	ON (× 8)	4:4:2/3	10 to 18	40 to 72	40 to 72	40 to 72	6.7 to 12
3	H'x003	ON (× 8)	8/3:4/3:2/3	48	64	128	64	32
	H'x004	ON (× 8)	8/3:4/3:4/9	48	64	128	64	21.3
	H'x005	ON (× 8)	8/3:4/3:1/3	48	64	128	64	16
	H'x006	ON (× 8)	8/3:4/3:2/9	48	64	128	64	10.7
	H'x013	ON (× 8)	4/3:4/3:2/3	48	64	64	64	32
	H'x014	ON (× 8)	4/3:4/3:4/9	48	64	64	64	21.3
	H'x015	ON (× 8)	4/3:4/3:1/3	48	64	64	64	16
	H'x016	ON (× 8)	4/3:4/3:2/9	48	64	64	64	10.7

Notes:
1. x in the FRQCR register setting depends on the set value in bits 12, 13, and 14.
2. The ratio of clock frequencies, where the input clock frequency is assumed to be 1.
3. In modes 0 and 2, the frequency of the EXTAL pin input clock or the crystal resonator
   In modes 1 and 3, 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 5.3.

chanhvinh.com

5.4 Register Descriptions

Table 5.4 shows the register configuration of the clock pulse generator.

Table 5.4 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Frequency control register	FRQCR	R/W	Modes 0 and 1: H'0124 Modes 2 and 3: H'0013	H'FFFE0010	16

5.4.1 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, release of bus mastership, change of gain of crystal oscillator for the XTAL pin, software standby mode, and standby mode cancellation. The register specifies the frequency division ratio for the CPU clock and peripheral clock (Pϕ). FRQCR is accessed by word.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	CKO EN2	CKOEN[1:0]		-	-	-	STC	-	-	IFC[1:0]		-	PFC[2:0]		
Initial value:	0	0	0	0	0	0	0	0/1*	0	0	0/1*	1/0*	0	1/0*	0/1*	0/1*
R/W:	R	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W

Note: * The initial value changes depending on the clock mode.

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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 Specifies whether the CKIO pin outputs clock signals, or is set to a fixed level or high impedance (Hi-Z) during normal operation mode, release of bus mastership, standby mode, or cancellation of standby mode. If these bits are set to 01, the CKIO pin is fixed at low during 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.
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	STC	0/1*	R	Frequency-Multiplier of PLL Circuit 0: 8 times (in modes 2 and 3) 1: 12 times (in modes 0 and 1) The fixed values are always read out depending on clock operating mode. The written value should always be the same one as above.
7, 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, 4	IFC[1:0]	01/10*	R/W	CPU Clock Frequency Division Ratio This bit specifies the frequency division ratio of the CPU clock with respect to the output frequency of PLL circuit. 00: 1 time 01: 1/2 times 10: 1/3 times 11: Reserved (setting prohibited)
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PFC[2:0]	100/011*	R/W	Peripheral Clock Frequency Division Ratio These bits specify the frequency division ratio of the peripheral clock with respect to the output frequency of PLL circuit. 000: Reserved (setting prohibited) 001: Reserved (setting prohibited) 010: Reserved (setting prohibited) 011: 1/4 times 100: 1/6 times 101: 1/8 times 110: 1/12 times 111: Reserved (setting prohibited)

Note: * The initial value changes depending on the clock mode.

chanhvinh.com

Table 5.5 CKOEN[1:0] Settings

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

5.5 Changing the Frequency

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

5.5.1 Changing the Division Ratio

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

1. In the initial state, IFC1 to IFC0 = B'01 or B'10 and PFC2 to PFC0 = B'100 or B'011.
2. Set the desired value in the IFC1 to IFC0 and PFC2 to PFC0 bits. The values that can be set are limited by the clock operating mode and the multiplication rate of PLL circuit. Note that if the wrong value is set, this LSI will malfunction.
3. After the register bits (IFC1 to IFC0 and PFC2 to PFC0) have been set, the clock is supplied of the new division ratio.

Note: When executing the SLEEP instruction after the frequency has been changed, be sure to read the frequency control register (FRQCR) three times before executing the SLEEP instruction.

chanhvinh.com

5.6 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 5.6. 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 5.6 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

5.6.1 In the Case of Inputting an External Clock

An example of the connection of an external clock is shown in figure 5.2. In cases where the Xout pin is left open state, take the parasitic capacitance as less than 10 pF.

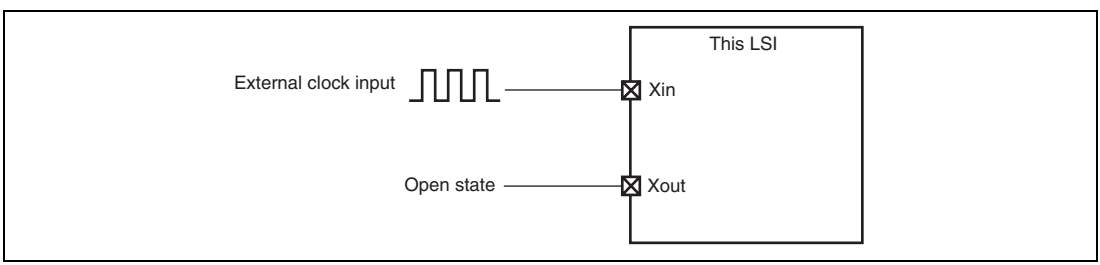


Figure 5.2 Example of the Connection of an External Clock

chanhvinh.com

5.6.2 In the Case of Using a Crystal Resonator

An example of the connection of crystal resonator is shown in figure 5.3.

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.

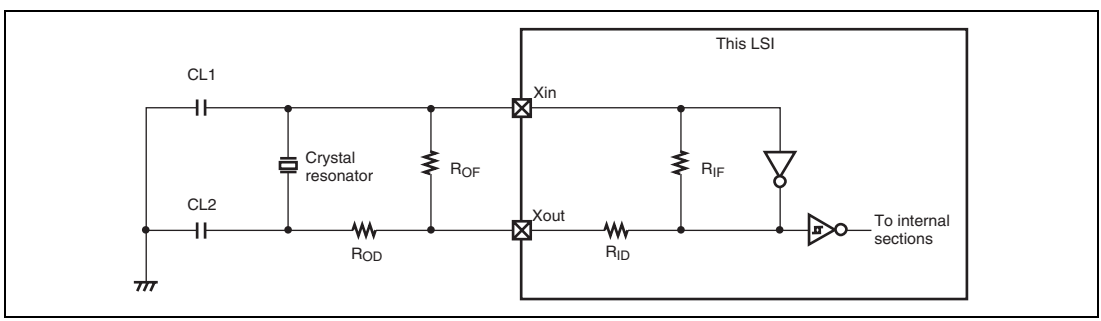


Figure 5.3 Example of the Connection of a Crystal Resonator

5.6.3 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

5.7 Oscillation Stabilizing Time

5.7.1 Oscillation Stabilizing Time of the On-chip Crystal Oscillator

In the case of using a crystal resonator, please wait longer than the oscillation stabilizing time 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, RTC_X1)
- Changing the gain of the on-chip crystal oscillator by $\overline{\text{RES}}$ pin (EXTAL)

5.7.2 Oscillation Stabilizing Time of the PLL circuit

The clock from EXTAL in the clock mode 0 and 2 or USB_X1 in the clock mode 1 and 3 is supplied to the PLL circuit. So, regardless of whether using a crystal resonator or inputting an external clock from EXTAL (clock mode 0 and 2) or USB_X1 (clock mode 1 and 3), please wait longer than 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

5.8 Notes on Board Design

5.8.1 Note on Using a PLL Oscillation Circuit

In the PLLVcc and PLLVss 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.

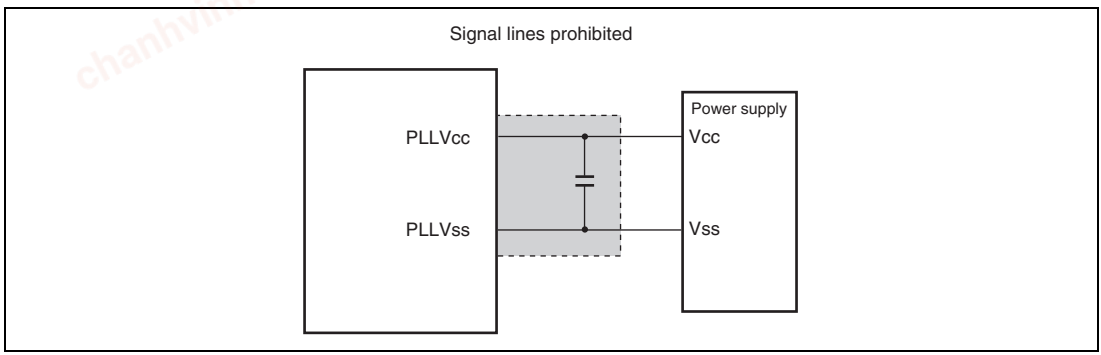


Figure 5.4 Note on Using a PLL Oscillation Circuit

chanhvinh.com

chanhvinh.com
chanhvinh.com
chanhvinh.com
chanhvinh.com

chanhvinh.com

Section 6 Exception Handling

6.1 Overview

6.1.1 Types of Exception Handling and Priority

Exception handling is started by sources, such as resets, address errors, register bank errors, interrupts, and instructions. Table 6.1 shows their priorities. When several exception handling sources occur at once, they are processed according to the priority shown.

Table 6.1 Types of Exception Handling and Priority Order

Type	Exception Handling	Priority
Reset	Power-on reset	High
	Manual reset	
Address error	CPU address error	
	DMA address error	
Instruction	FPU exception	
	Integer division exception (division by zero)	
	Integer division exception (overflow)	
Register bank error	Bank underflow	
	Bank overflow	
Interrupt	NMI	
	User debugging interface	
	IRQ	
	PINT	Low

chanhvinh.com

Type	Exception Handling	Priority
Instruction	Trap instruction (TRAPA instruction)	High
	General illegal instructions (undefined code)	
	Slot illegal instructions (undefined code placed directly after a delayed branch instruction*[1] (including FPU instructions and FPU-related CPU instructions in FPU module standby state), instructions that rewrite the PC*[2], 32-bit instructions*[3], RESBANK instruction, DIVS instruction, and DIVU instruction)	Low

Notes: 1. Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF.
2. Instructions that rewrite the PC: JMP, JSR, BRA, BSR, RTS, RTE, BT, BF, TRAPA, BF/S, BT/S, BSRF, BRAF, JSR/N, RTV/N.
3. 32-bit instructions: BAND.B, BANDNOT.B, BCLR.B, BLD.B, BLDNOT.B, BOR.B, BORNOT.B, BSET.B, BST.B, BXOR.B, MOV.B@disp12, MOV.W@disp12, MOV.L@disp12, MOVI20, MOVI20S, MOVU.B, MOVU.W.

6.1.2 Exception Handling Operations

The exception handling sources are detected and start processing according to the timing shown in table 6.2.

Table 6.2 Timing of Exception Source Detection and Start of Exception Handling

Exception	Source	Timing of Source Detection and Start of Handling
Reset	Power-on reset	Starts when the $\overline{\text{RES}}$ pin changes from low to high, when the user debugging interface reset negate command is set after the user debugging interface reset assert command has been set, or when the watchdog timer overflows.
	Manual reset	Starts when the watchdog timer overflows.
Address error		Detected when instruction is decoded and starts when the previous executing instruction finishes executing.
Interrupts		
Register bank error	Bank underflow	Starts upon attempted execution of a RESBANK instruction when saving has not been performed to register banks.
	Bank overflow	In the state where saving has been performed to all register bank areas, starts when acceptance of register bank overflow exception has been set by the interrupt controller (the BOVE bit in IBNR of the interrupt controller is 1) and an interrupt that uses a register bank has occurred and been accepted by the CPU.

chanhvinh.com

Exception	Source	Timing of Source Detection and Start of Handling
Instructions	Trap instruction	Starts from the execution of a TRAPA instruction.
	General illegal instructions	Starts from the decoding of undefined code anytime except immediately after a delayed branch instruction (delay slot) (including FPU instructions and FPU-related CPU instructions in FPU module standby state).
	Slot illegal instructions	Starts from the decoding of undefined code placed directly after a delayed branch instruction (delay slot) (including FPU instructions and FPU-related CPU instructions in FPU module standby state), of instructions that rewrite the PC, of 32-bit instructions, of the RESBANK instruction, of the DIVS instruction, or of the DIVU instruction.
	Integer division exceptions	Starts when detecting division-by-zero exception or overflow exception caused by division of the negative maximum value (H'80000000) by –1.
Instructions	FPU exceptions	Starts when detecting invalid floating point operation exception defined by IEEE standard 754, division-by-zero exception, overflow, underflow, or inexact exception. Also starts when qNaN or ±∞ is input to the source for a floating point operation instruction when the QIS bit in FPSCR is set.

When exception handling starts, the CPU operates as follows:

(1) Exception Handling Triggered by Reset

The initial values of the program counter (PC) and stack pointer (SP) are fetched from the exception handling vector table (PC and SP are respectively the H'00000000 and H'00000004 addresses for power-on resets and the H'00000008 and H'0000000C addresses for manual resets). See section 6.1.3, Exception Handling Vector Table, for more information. The vector base register (VBR) is then initialized to H'00000000, the interrupt mask level bits (I3 to I0) of the status register (SR) are initialized to H'F (B'1111), and the BO and CS bits are initialized to 0. The BN bit in IBNR of the interrupt controller is also initialized to 0. The floating point status/control register (FPSCR) is initialized to H'00040001 by a power-on reset. The program begins running from the PC address fetched from the exception handling vector table.

chanhvinh.com

(2) Exception Handling Triggered by Address Errors, Register Bank Errors, Interrupts, and Instructions

SR and PC are saved to the stack indicated by R15. In the case of interrupt exception handling other than NMI with usage of the register banks enabled, general registers R0 to R14, control register GBR, system registers MACH, MACL, and PR, and the vector table address offset of the interrupt exception handling to be executed are saved to the register banks. In the case of exception handling due to an address error, register bank error, NMI interrupt, or instruction, saving to a register bank is not performed. When saving is performed to all register banks, automatic saving to the stack is performed instead of register bank saving. In this case, an interrupt controller setting must have been made so that register bank overflow exceptions are not accepted (the BOVE bit in IBNR of the interrupt controller is 0). If a setting to accept register bank overflow exceptions has been made (the BOVE bit in IBNR of the interrupt controller is 1), register bank overflow exception will be generated. In the case of interrupt exception handling, the interrupt priority level is written to the I3 to I0 bits in SR. In the case of exception handling due to an address error or instruction, the I3 to I0 bits are not affected. The exception service routine start address is then fetched from the exception handling vector table and the program begins running from that address.

6.1.3 Exception Handling Vector Table

Before exception handling begins running, the exception handling vector table must be set in memory. The exception handling vector table stores the start addresses of exception service routines. (The reset exception handling table holds the initial values of PC and SP.)

All exception sources are given different vector numbers and vector table address offsets, from which the vector table addresses are calculated. During exception handling, the start addresses of the exception service routines are fetched from the exception handling vector table, which is indicated by this vector table address.

chanhvinh.com

Table 6.3 shows the vector numbers and vector table address offsets. Table 6.4 shows how vector table addresses are calculated.

Table 6.3 Exception Handling Vector Table

Exception Sources		Vector Numbers	Vector Table Address Offset
Power-on reset	PC	0	H'00000000 to H'00000003
	SP	1	H'00000004 to H'00000007
Manual reset	PC	2	H'00000008 to H'0000000B
	SP	3	H'0000000C to H'0000000F
General illegal instruction		4	H'00000010 to H'00000013
(Reserved by system)		5	H'00000014 to H'00000017
Slot illegal instruction		6	H'00000018 to H'0000001B
(Reserved by system)		7	H'0000001C to H'0000001F
		8	H'00000020 to H'00000023
CPU address error		9	H'00000024 to H'00000027
DMA address error		10	H'00000028 to H'0000002B
Interrupts	NMI	11	H'0000002C to H'0000002F
	(Reserved by system)	12	H'00000030 to H'00000033
FPU exception		13	H'00000034 to H'00000037
User debugging interface		14	H'00000038 to H'0000003B
Bank overflow		15	H'0000003C to H'0000003F
Bank underflow		16	H'00000040 to H'00000043
Integer division exception (division by zero)		17	H'00000044 to H'00000047
Integer division exception (overflow)		18	H'00000048 to H'0000004B
(Reserved by system)		19	H'0000004C to H'0000004F
		:	:
		31	H'0000007C to H'0000007F
Trap instruction (user vector)		32	H'00000080 to H'00000083
		:	:
		63	H'000000FC to H'000000FF

chanhvinh.com

Exception Sources	Vector Numbers	Vector Table Address Offset
External interrupts (IRQ, PINT), on-chip peripheral module interrupts*	64	H'00000100 to H'00000103
	:	:
	511	H'000007FC to H'000007FF

Note: * The vector numbers and vector table address offsets for each external interrupt and on-chip peripheral module interrupt are given in table 7.4 in section 7, Interrupt Controller.

Table 6.4 Calculating Exception Handling Vector Table Addresses

Exception Source	Vector Table Address Calculation
Resets	Vector table address = (vector table address offset) = (vector number) × 4
Address errors, register bank errors, interrupts, instructions	Vector table address = VBR + (vector table address offset) = VBR + (vector number) × 4

Notes: 1. Vector table address offset: See table 6.3.
2. Vector number: See table 6.3.

chanhvinh.com

6.2 Resets

6.2.1 Input/Output Pins

Table 6.5 shows the pin configuration.

Table 6.5 Pin Configuration

Pin Name	Symbol	I/O	Function
Power-on reset	$\overline{\text{RES}}$	Input	When this pin is driven low, this LSI shifts to the power-on reset processing

6.2.2 Types of Reset

A reset is the highest-priority exception handling source. There are two kinds of reset, power-on and manual. As shown in table 6.6, the CPU state is initialized in both a power-on reset and a manual reset. The FPU state is initialized by a power-on reset, but not by a manual reset. On-chip peripheral module registers except a few registers are also initialized by a power-on reset, but not by a manual reset.

Table 6.6 Reset States

Type	Conditions for Transition to Reset State			Internal States				
	$\overline{\text{RES}}$	User Debugging Interface Command	Watchdog Timer Overflow	CPU	Other Modules	On-Chip High-Speed RAM	On-Chip Large-Capacity RAM (Excluding On-Chip Data Retention RAM)	On-Chip Data Retention RAM
Power-on reset	Low	—	—	Initialized	Initialized	Initialized or Retained contents*2	Initialized or Retained contents*3	Initialized or Retained contents*4, *5
	High	User debugging interface reset assert command is set	—	Initialized	Initialized	Initialized or Retained contents*2	Initialized or Retained contents*3	Initialized or Retained contents*4
	High	Command other than user debugging interface reset assert is set	Power-on reset	Initialized	*1	Initialized or Retained contents*2	Initialized or Retained contents*3	Initialized or Retained contents*4

chanhvinh.com

Type	Conditions for Transition to Reset State					Internal States		
	$\overline{\text{RES}}$	User Debugging Interface Command	Watchdog Timer Overflow	CPU	Other Modules	On-Chip High-Speed RAM	On-Chip Large-Capacity RAM (Excluding On-Chip Data Retention RAM)	On-Chip Data Retention RAM
Manual reset	High	Command other than user debugging interface reset assert is set	Manual reset	Initialized	*1	Retained contents	Retained contents	Retained contents

Notes: 1. See section 36.3, Register States in Each Operating Mode.
2. Data are retained when the setting of either the RAME or RAMWE bit is disabled.
3. Data are retained when the setting of either the VRAME or VRAMWE bit is disabled.
4. Data are retained when the setting of any of the VRAME, VRAMWE, or RRAMWE bits is disabled.
5. When the deep standby mode is canceled by a power-on reset, the data cannot be retained.

chanhvinh.com

6.2.3 Power-On Reset

(1) Power-On Reset by Means of $\overline{\text{RES}}$ Pin

When the $\overline{\text{RES}}$ pin is driven low, this LSI enters the power-on reset state. To reliably reset this LSI, the $\overline{\text{RES}}$ pin should be kept at the low level for the duration of the oscillation settling time at power-on or when in software standby mode (when the clock is halted), or at least 20-tcyc when the clock is running. In the power-on reset state, the internal state of the CPU and all the on-chip peripheral module registers are initialized. See appendix A, Pin States, for the status of individual pins during the power-on reset state.

In the power-on reset state, power-on reset exception handling starts when the $\overline{\text{RES}}$ pin is first driven low for a fixed period and then returned to high. The CPU operates as follows:

1. The initial value (execution start address) of the program counter (PC) is fetched from the exception handling vector table.
2. The initial value of the stack pointer (SP) is fetched from the exception handling vector table.
3. The vector base register (VBR) is cleared to H'00000000, the interrupt mask level bits (I3 to I0) of the status register (SR) are initialized to H'F (B'1111), and the BO and CS bits are initialized to 0. The BN bit in IBNR of the interrupt controller is also initialized to 0. FPSCR is initialized to H'00040001
4. The values fetched from the exception handling vector table are set in the PC and SP, and the program begins executing.

Be certain to always perform power-on reset processing when turning the system power on.

(2) Power-On Reset by Means of User Debugging Interface Reset Assert Command

When the user debugging interface reset assert command is set, this LSI enters the power-on reset state. Power-on reset by means of the user debugging interface reset assert command is equivalent to power-on reset by means of the $\overline{\text{RES}}$ pin. Setting the user debugging interface reset negate command cancels the power-on reset state. The time required between the user debugging interface reset assert command and the user debugging interface reset negate command is the same as the time to keep the $\overline{\text{RES}}$ pin low to initiate a power-on reset. In the power-on reset state generated by the user debugging interface reset assert command, setting the user debugging interface reset negate command starts power-on reset exception handling. The CPU operates in the same way as when a power-on reset was caused by the $\overline{\text{RES}}$ pin.

chanhvinh.com

(3) Power-On Reset Initiated by Watchdog Timer

When a setting is made for a power-on reset to be generated in watchdog timer mode of the watchdog timer, and WTCNT of the watchdog timer overflows, this LSI enters the power-on reset state.

In this case, WRCSR of the watchdog timer and FRQCR of the clock pulse generator are not initialized by the reset signal generated by the watchdog timer.

If a reset caused by the $\overline{\text{RES}}$ pin or the user debugging interface reset assert command occurs simultaneously with a reset caused by watchdog timer overflow, the reset caused by the $\overline{\text{RES}}$ pin or the user debugging interface reset assert command has priority, and the WOVF bit in WRCSR is cleared to 0. When power-on reset exception processing is started by the watchdog timer, the CPU operates in the same way as when a power-on reset was caused by the $\overline{\text{RES}}$ pin.

6.2.4 Manual Reset

(1) Manual Reset Initiated by Watchdog Timer

When a setting is made for a manual reset to be generated in watchdog timer mode of the watchdog timer, and WTCNT of the watchdog timer overflows, this LSI enters the manual reset state.

When manual reset exception processing is started by the watchdog timer, the CPU operates as follows:

1. The initial value (execution start address) of the program counter (PC) is fetched from the exception handling vector table.
2. The initial value of the stack pointer (SP) is fetched from the exception handling vector table.
3. The vector base register (VBR) is cleared to H'00000000, the interrupt mask level bits (I3 to I0) of the status register (SR) are initialized to H'F (B'1111), and the BO and CS bits are initialized to 0. The BN bit in IBNR of interrupt controller is also initialized to 0.
4. The values fetched from the exception handling vector table are set in the PC and SP, and the program begins executing.

chanhvinh.com

(2) Note in Manual Reset

When a manual reset is generated, the bus cycle is retained, but if a manual reset occurs while the bus is released or during burst transfer by the direct memory access controller, manual reset exception handling will be deferred until the CPU acquires the bus. The CPU and the BN bit in IBNR of the interrupt controller are initialized by a manual reset. The FPU and other modules are not initialized.

chanhvinh.com

6.3 Address Errors

6.3.1 Address Error Sources

Address errors occur when instructions are fetched or data read or written, as shown in table 6.7.

Table 6.7 Bus Cycles and Address Errors

Bus Cycle Type	Bus Master	Bus Cycle Description	Address Errors
Instruction fetch	CPU	Instruction fetched from even address	None (normal)
		Instruction fetched from odd address	Address error occurs
		Instruction fetched from other than on-chip peripheral module space* or H'F0000000 to H'F5FFFFFF in on-chip RAM space*	None (normal)
		Instruction fetched from on-chip peripheral module space* or H'F0000000 to H'F5FFFFFF in on-chip RAM space*	Address error occurs
Data read/write	CPU or direct memory access controller	Word data accessed from even address	None (normal)
		Word data accessed from odd address	Address error occurs
		Longword data accessed from a longword boundary	None (normal)
		Longword data accessed from other than a long-word boundary	Address error occurs
		Double longword data accessed from double longword boundary	None (normal)
		Double longword data accessed from other than double longword boundary	Address error occurs
		Byte or word data accessed in on-chip peripheral module space*	None (normal)
		Longword data accessed in 16-bit on-chip peripheral module space*	None (normal)
		Longword data accessed in 8-bit on-chip peripheral module space*	None (normal)

Note: * See section 9, Bus State Controller, for details of the on-chip peripheral module space and on-chip RAM space.

chanhvinh.com

6.3.2 Address Error Exception Handling

When an address error occurs, the bus cycle in which the address error occurred ends. When the executing instruction then finishes, address error exception handling starts. The CPU operates as follows:

1. The exception service routine start address which corresponds to the address error that occurred is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the last executed instruction.
4. After jumping to the exception service routine start address fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch.

6.4 Register Bank Errors

6.4.1 Register Bank Error Sources

(1) Bank Overflow

In the state where saving has already been performed to all register bank areas, bank overflow occurs when acceptance of register bank overflow exception has been set by the interrupt controller (the BOVE bit in IBNR of the interrupt controller is set to 1) and an interrupt that uses a register bank has occurred and been accepted by the CPU.

(2) Bank Underflow

Bank underflow occurs when an attempt is made to execute a RESBANK instruction while saving has not been performed to register banks.

chanhvinh.com

6.4.2 Register Bank Error Exception Handling

When a register bank error occurs, register bank error exception handling starts. The CPU operates as follows:

1. The exception service routine start address which corresponds to the register bank error that occurred is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the last executed instruction for a bank overflow, and the start address of the executed RESBANK instruction for a bank underflow.

   To prevent multiple interrupts from occurring at a bank overflow, the priority level of the interrupt that caused the bank overflow is written to the interrupt mask level bits (I3 to I0) of the status register (SR).
4. After jumping to the exception service routine start address fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch.

6.5 Interrupts

6.5.1 Interrupt Sources

The sources that start interrupt exception handling are divided into NMI, user debugging interface, IRQ, PINT, and on-chip peripheral modules.

Each interrupt source is allocated a different vector number and vector table offset. See table 7.4 in section 7, Interrupt Controller, for more information on vector numbers and vector table address offsets.

chanhvinh.com

6.5.2 Interrupt Priority Level

The interrupt priority order is predetermined. When multiple interrupts occur simultaneously (overlap), the interrupt controller determines their relative priorities and starts exception handling according to the results.

The priority order of interrupts is expressed as priority levels 0 to 16, with priority 0 the lowest and priority 16 the highest. The NMI interrupt has priority 16 and cannot be masked, so it is always accepted. The priority level of user debugging interface interrupts is 15. Priority levels of IRQ interrupts, PINT interrupts, and on-chip peripheral module interrupts can be set freely using the interrupt priority registers 01, 02, and 05 to 22 (IPR01, IPR02, and IPR05 to IPR22) of the interrupt controller as shown in table 6.9. The priority levels that can be set are 0 to 15. Level 16 cannot be set. See section 7.3.1, Interrupt Priority Registers 01, 02, 05 to 22 (IPR01, IPR02, IPR05 to IPR22), for details of IPR01, IPR02, and IPR05 to IPR22.

Table 6.9 Interrupt Priority Order

Type	Priority Level	Comment
NMI	16	Fixed priority level. Cannot be masked.
User debugging interface	15	Fixed priority level.
IRQ	0 to 15	Set with interrupt priority registers 01, 02, and 05 to 22 (IPR01, IPR02, and IPR05 to IPR22).
PINT		
On-chip peripheral module		

chanhvinh.com

6.5.3 Interrupt Exception Handling

When an interrupt occurs, its priority level is ascertained by the interrupt controller. NMI is always accepted, but other interrupts are only accepted if they have a priority level higher than the priority level set in the interrupt mask level bits (I3 to I0) of the status register (SR).

When an interrupt is accepted, interrupt exception handling begins. In interrupt exception handling, the CPU fetches the exception service routine start address which corresponds to the accepted interrupt from the exception handling vector table, and saves SR and the program counter (PC) to the stack. In the case of interrupt exception handling other than NMI with usage of the register banks enabled, general registers R0 to R14, control register GBR, system registers MACH, MACL, and PR, and the vector table address offset of the interrupt exception handling to be executed are saved in the register banks. In the case of exception handling due to an address error, NMI interrupt, or instruction, saving is not performed to the register banks. If saving has been performed to all register banks (0 to 14), automatic saving to the stack is performed instead of register bank saving. In this case, an interrupt controller setting must have been made so that register bank overflow exceptions are not accepted (the BOVE bit in IBNR of the interrupt controller is 0). If a setting to accept register bank overflow exceptions has been made (the BOVE bit in IBNR of the interrupt controller is 1), register bank overflow exception occurs. Next, the priority level value of the accepted interrupt is written to the I3 to I0 bits in SR. For NMI, however, the priority level is 16, but the value set in the I3 to I0 bits is H'F (level 15). Then, after jumping to the start address fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch. See section 7.6, Operation, for further details of interrupt exception handling.

chanhvinh.com

6.6 Exceptions Triggered by Instructions

6.6.1 Types of Exceptions Triggered by Instructions

Exception handling can be triggered by trap instructions, general illegal instructions, slot illegal instructions, integer division exceptions, and FPU exceptions, as shown in table 6.10.

Table 6.10 Types of Exceptions Triggered by Instructions

Type	Source Instruction	Comment
Trap instruction	TRAPA	
Slot illegal instructions	Undefined code placed immediately after a delayed branch instruction (delay slot) (including FPU instructions and FPU-related CPU instructions in FPU module standby state), instructions that rewrite the PC, 32-bit instructions, RESBANK instruction, DIVS instruction, and DIVU instruction	Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF Instructions that rewrite the PC: JMP, JSR, BRA, BSR, RTS, RTE, BT, BF, TRAPA, BF/S, BT/S, BSRF, BRAF, JSR/N, RTV/N 32-bit instructions: BAND.B, BANDNOT.B, BCLR.B, BLD.B, BLDNOT.B, BOR.B, BORNOT.B, BSET.B, BST.B, BXOR.B, MOV.B@disp12, MOV.W@disp12, MOV.L@disp12, MOVI20, MOVI20S, MOVU.B, MOVU.W.
General illegal instructions	Undefined code anywhere besides in a delay slot (including FPU instructions and FPU-related CPU instructions in FPU module standby state)	
Integer division exceptions	Division by zero	DIVU, DIVS
	Negative maximum value ÷ (–1)	DIVS
FPU exceptions	Starts when detecting invalid operation exception defined by IEEE754, division-by-zero exception, overflow, underflow, or inexact exception.	FADD, FSUB, FMUL, FDIV, FMAC, FCMP/EQ, FCMP/GT, FLOAT, FTRC, FCNVDS, FCNVSD, FSQRT

chanhvinh.com

6.6.2 Trap Instructions

When a TRAPA instruction is executed, trap instruction exception handling starts. The CPU operates as follows:

1. The exception service routine start address which corresponds to the vector number specified in the TRAPA instruction is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the TRAPA instruction.
4. After jumping to the exception service routine start address fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch.

6.6.3 Slot Illegal Instructions

An instruction placed immediately after a delayed branch instruction is called the “instruction placed in a delay slot”. When the instruction placed in the delay slot is undefined code (including FPU instructions and FPU-related CPU instructions in FPU module standby state), an instruction that rewrites the PC, a 32-bit instruction, an RESBANK instruction, a DIVS instruction, or a DIVU instruction, slot illegal exception handling starts when such kind of instruction is decoded. When the FPU has entered a module standby state, the floating point operation instruction and FPU-related CPU instructions are handled as undefined codes. If these instructions are placed in a delay slot and then decoded, a slot illegal instruction exception handling starts.

The CPU operates as follows:

1. The exception service routine start address is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the jump address of the delayed branch instruction immediately before the undefined code, the instruction that rewrites the PC, the 32-bit instruction, the RESBANK instruction, the DIVS instruction, or the DIVU instruction.
4. After jumping to the exception service routine start address fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch.

chanhvinh.com

6.6.4 General Illegal Instructions

When an undefined code, including FPU instructions and FPU-related CPU instructions in FPU module standby state, placed anywhere other than immediately after a delayed branch instruction, i.e., in a delay slot, is decoded, general illegal instruction exception handling starts. When the FPU has entered a module standby state, the floating point instruction and FPU-related CPU instructions are handled as undefined codes. If these instructions are placed anywhere other than immediately after a delayed branch instruction (i.e., in a delay slot) and then decoded, general illegal instruction exception handling starts.

In general illegal instruction exception handling, the CPU handles general illegal instructions in the same way as slot illegal instructions. Unlike processing of slot illegal instructions, however, the program counter value stored is the start address of the undefined code.

6.6.5 Integer Division Exceptions

When an integer division instruction performs division by zero or the result of integer division overflows, integer division instruction exception handling starts. The instructions that may become the source of division-by-zero exception are DIVU and DIVS. The only source instruction of overflow exception is DIVS, and overflow exception occurs only when the negative maximum value is divided by –1. The CPU operates as follows:

1. The exception service routine start address which corresponds to the integer division exception that occurred is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start address of the integer division instruction at which the exception occurred.
4. After jumping to the exception service routine start address fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch.

chanhvinh.com

6.6.6 FPU Exceptions

An FPU exception handling is generated when the V, Z, O, U or I bit in the FPU exception enable field (Enable) of the floating point status/control register (FPSCR) is set. This indicates the occurrence of an invalid operation exception defined by the IEEE standard 754, a division-by-zero exception, overflow (in the case of an instruction for which this is possible), underflow (in the case of an instruction for which this is possible), or inexact exception (in the case of an instruction for which this is possible).

The floating point operation instructions that may cause an FPU exception handling are FADD, FSUB, FMUL, FDIV, FMAC, FCMP/EQ, FCMP/GT, FLOAT, FTRC, FCNVDS, FCNVSD, and FSQRT.

An FPU exception handling is generated only when the corresponding FPU exception enable bit (Enable) is set. When the FPU detects an exception source in floating point operation, FPU operation is halted and generation of an FPU exception handling is reported to the CPU. When exception handling is started, the CPU operations are as follows.

1. The start address of the exception service routine which corresponds to the FPU exception handling that occurred is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the last executed instruction.
4. After jumping to the exception service routine start address fetched from the exception handling vector table, program execution starts. This jump is not a delayed branch.

The FPU exception flag field (Flag) of FPSCR is always updated regardless of whether or not an FPU exception handling has been accepted, and remains set until explicitly cleared by the user through an instruction. The FPU exception source field (Cause) of FPSCR changes each time a floating point operation instruction is executed.

When the V bit in the FPU exception enable field (Enable) of FPSCR is set and the QIS bit in FPSCR is also set, FPU exception handling is generated when qNAN or ±∞ is input to a floating point operation instruction source.

chanhvinh.com

6.7 When Exception Sources Are Not Accepted

When an address error, FPU exception, register bank error (overflow), or interrupt is generated immediately after a delayed branch instruction, it is sometimes not accepted immediately but stored instead, as shown in table 6.11. When this happens, it will be accepted when an instruction that can accept the exception is decoded.

Table 6.11 Exception Source Generation Immediately after Delayed Branch Instruction

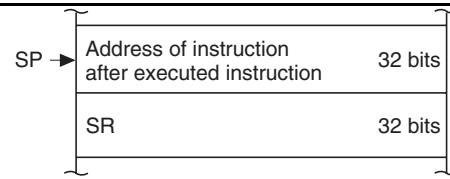
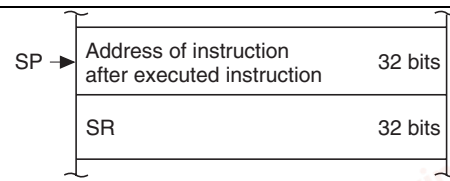
	Exception Source			
Point of Occurrence	**Address Error**	**Floating-Point Unit Exception**	**Register Bank Error (Overflow)**	**Interrupt**
Immediately after a delayed branch instruction*	Not accepted	Not accepted	Not accepted	Not accepted

Note: * Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF

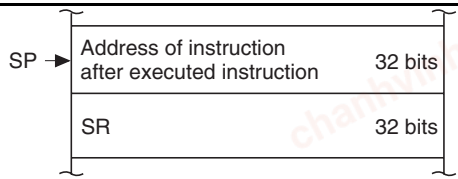
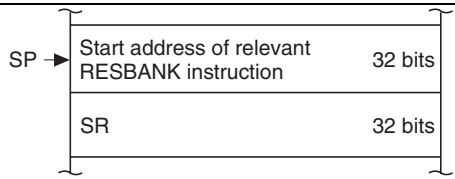
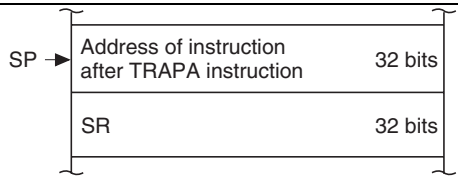
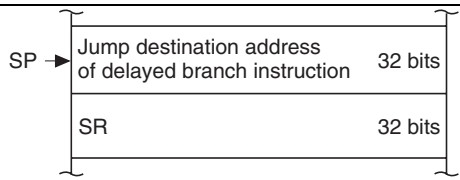
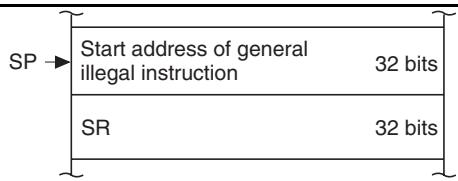
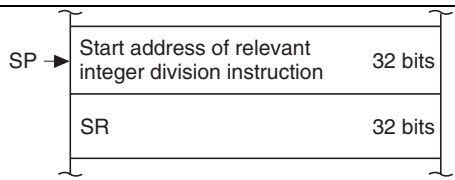
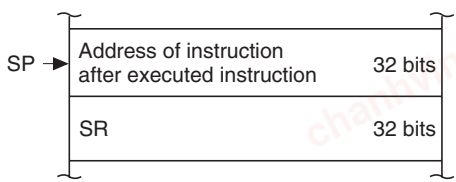
6.8 Stack Status after Exception Handling Ends

The status of the stack after exception handling ends is as shown in table 6.12.

Table 6.12 Stack Status After Exception Handling Ends

Exception Type	Stack Status
Address error	
Interrupt	

chanhvinh.com

Exception Type	Stack Status
Register bank error (overflow)	
Register bank error (underflow)	
Trap instruction	
Slot illegal instruction	
General illegal instruction	
Integer division exception	
FPU exception	

chanhvinh.com

6.9 Usage Notes

6.9.1 Value of Stack Pointer (SP)

The value of the stack pointer must always be a multiple of four. If it is not, an address error will occur when the stack is accessed during exception handling.

6.9.2 Value of Vector Base Register (VBR)

The value of the vector base register must always be a multiple of four. If it is not, an address error will occur when the stack is accessed during exception handling.

6.9.3 Address Errors Caused by Stacking of Address Error Exception Handling

When the stack pointer is not a multiple of four, an address error will occur during stacking of the exception handling (interrupts, etc.) and address error exception handling will start up as soon as the first exception handling is ended. Address errors will then also occur in the stacking for this address error exception handling. To ensure that address error exception handling does not go into an endless loop, no address errors are accepted at that point. This allows program control to be shifted to the address error exception service routine and enables error processing.

When an address error occurs during exception handling stacking, the stacking bus cycle (write) is executed. During stacking of the status register (SR) and program counter (PC), the SP is decremented by 4 for both, so the value of SP will not be a multiple of four after the stacking either. The address value output during stacking is the SP value, so the address where the error occurred is itself output. This means the write data stacked will be undefined.

6.9.4 Interrupt Control via Modification of Interrupt Mask Bits

When enabling interrupts by changing the Interrupt Mask bits (I3-I0) of the Status Register (SR) using the LDC or LDC.L instructions, interrupts might not be accepted during the execution of the 5 instructions immediately after the LDC/LDC.L instruction.

Therefore, when enabling/disabling interrupts by changing the Interrupt Mask bits (I3-I0) of the Status Register (SR) using LDC/LDC.L instructions, please place at least 5 instructions between the interrupt-enable instruction and the interrupt-disable instruction.

chanhvinh.com

6.9.5 Note before Exception Handling Begins Running

Before exception handling begins running, the exception handling vector table must be stored in a memory, and the CPU must be able to access the memory. So, if the exception handling is generated

- Ex. 1: when the exception handling vector table is stored in an external address space, but the settings of bus state controller and general I/O ports to access the external address space have been not completed yet, or
- Ex. 2: when the exception handling vector table is stored in the on-chip RAM, but the vector base register (VBR) has been not changed to the on-chip RAM address yet,

the CPU fetches an unintended value as the execution start address, and starts executing programs from unintended address.

(1) Manual Reset

Before the settings necessary to access the external CS0 space are completed, the manual reset should not be generated. When a manual reset is issued, the CPU fetches the program execution start address from the manual reset vector table address offset (H'00000008), which is to say always from the external CS0 space. Additionally, in the case that no memory is connected to the external CS0 space in boot mode 1 to 3, the manual reset should not be generated.

(2) NMI Interrupt

Before the exception handling vector table is stored in a memory and the settings necessary to access the memory are completed, the NMI interrupt should not be generated. (Do not enable interrupts on the 640 KB version.)

Specially in boot mode 1 to 3, the VBR is kept as the initial value H'00000000 in the period of the boot operation (before the transfer of the loader program is completed and the CPU jumps to the on-chip high-speed RAM). Before the VBR is changed or the settings necessary to access the external address space are completed in the loader program, the NMI interrupt should not be generated. (Do not enable interrupts on the 640 KB version.)

(3) Interrupts Other Than NMI

Before the exception handling vector table is stored in a memory and the settings necessary to access the memory are completed, the settings to permit the interrupts should not be done.

chanhvinh.com

(4) The Other Exceptions

Before the exception handling vector table is stored in a memory and the settings necessary to access the memory are completed, the exception handling should not be generated.

chanhvinh.com

chanhvinh.com
chanhvinh.com
chanhvinh.com
chanhvinh.com

chanhvinh.com

Section 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

- 16 levels of interrupt priority can be set.

  By setting the twenty interrupt priority registers, the priorities of IRQ interrupts, PINT interrupts, and on-chip peripheral module interrupts can be selected from 16 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.
- Register banks

  This LSI has register banks that enable register saving and restoration required in the interrupt processing to be performed at high speed.

chanhvinh.com

Figure 7.1 shows a block diagram.

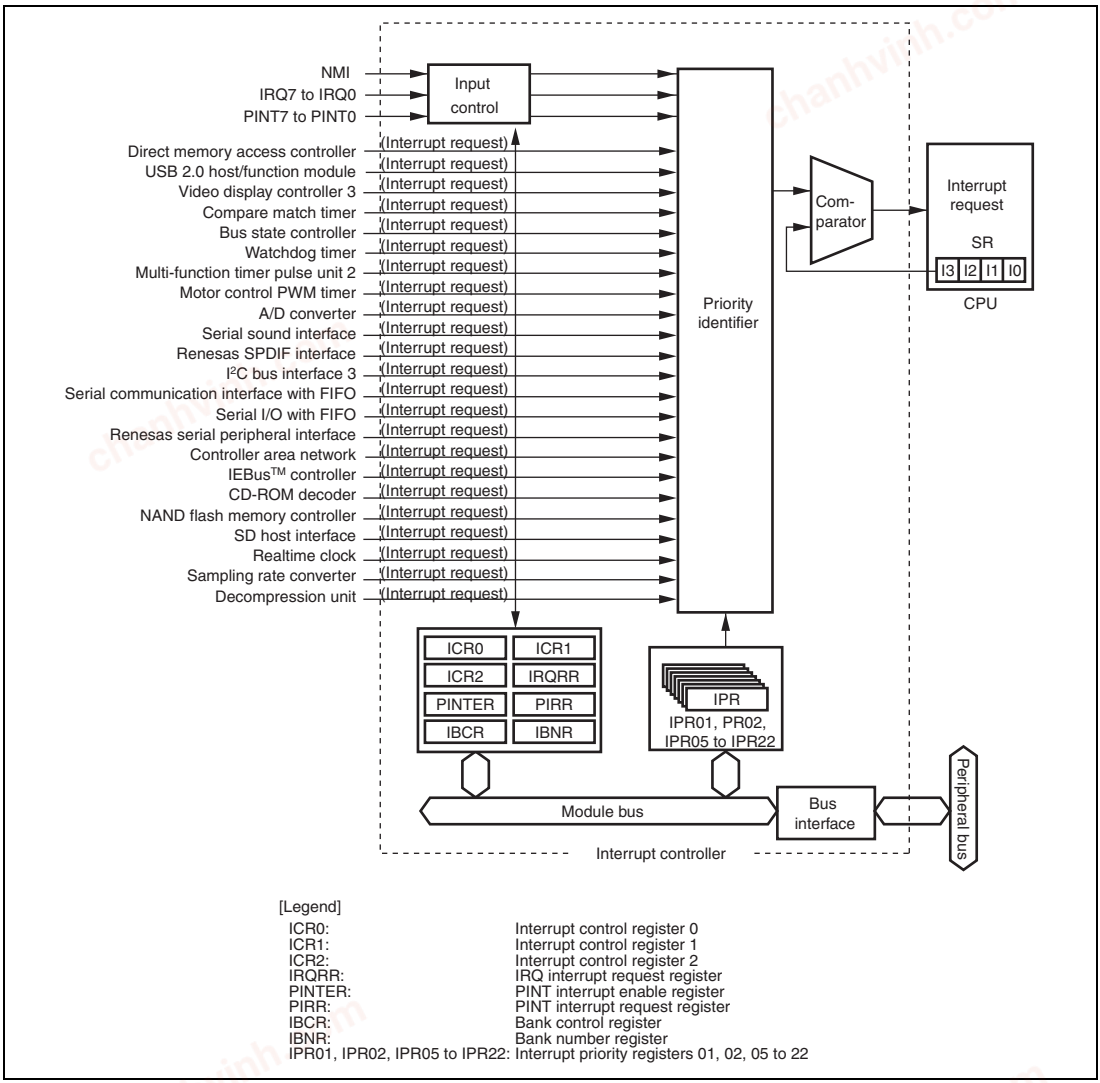


Figure 7.1 Block Diagram

chanhvinh.com

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
	PINT7 to PINT0	Input	

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.

Table 7.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Interrupt control register 0	ICR0	R/W	*[1]	H'FFFE0800	16, 32
Interrupt control register 1	ICR1	R/W	H'0000	H'FFFE0802	16, 32
Interrupt control register 2	ICR2	R/W	H'0000	H'FFFE0804	16, 32
IRQ interrupt request register	IRQRR	R/(W)*[2]	H'0000	H'FFFE0806	16, 32
PINT interrupt enable register	PINTER	R/W	H'0000	H'FFFE0808	16, 32
PINT interrupt request register	PIRR	R	H'0000	H'FFFE080A	16, 32
Bank control register	IBCR	R/W	H'0000	H'FFFE080C	16, 32
Bank number register	IBNR	R/W	H'0000	H'FFFE080E	16, 32
Interrupt priority register 01	IPR01	R/W	H'0000	H'FFFE0818	16, 32
Interrupt priority register 02	IPR02	R/W	H'0000	H'FFFE081A	16, 32
Interrupt priority register 05	IPR05	R/W	H'0000	H'FFFE0820	16, 32
Interrupt priority register 06	IPR06	R/W	H'0000	H'FFFE0C00	16, 32
Interrupt priority register 07	IPR07	R/W	H'0000	H'FFFE0C02	16, 32
Interrupt priority register 08	IPR08	R/W	H'0000	H'FFFE0C04	16, 32
Interrupt priority register 09	IPR09	R/W	H'0000	H'FFFE0C06	16, 32
Interrupt priority register 10	IPR10	R/W	H'0000	H'FFFE0C08	16, 32
Interrupt priority register 11	IPR11	R/W	H'0000	H'FFFE0C0A	16, 32
Interrupt priority register 12	IPR12	R/W	H'0000	H'FFFE0C0C	16, 32
Interrupt priority register 13	IPR13	R/W	H'0000	H'FFFE0C0E	16, 32
Interrupt priority register 14	IPR14	R/W	H'0000	H'FFFE0C10	16, 32
Interrupt priority register 15	IPR15	R/W	H'0000	H'FFFE0C12	16, 32
Interrupt priority register 16	IPR16	R/W	H'0000	H'FFFE0C14	16, 32
Interrupt priority register 17	IPR17	R/W	H'0000	H'FFFE0C16	16, 32

chanhvinh.com

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Interrupt priority register 18	IPR18	R/W	H'0000	H'FFFE0C18	16, 32
Interrupt priority register 19	IPR19	R/W	H'0000	H'FFFE0C1A	16, 32
Interrupt priority register 20	IPR20	R/W	H'0000	H'FFFE0C1C	16, 32
Interrupt priority register 21	IPR21	R/W	H'0000	H'FFFE0C1E	16, 32
Interrupt priority register 22	IPR22	R/W	H'0000	H'FFFE0C20	16, 32

Notes: 1. For 1-Mbyte version, when the NMI pin is high, becomes H'8000; when low, becomes H'0000. For 640-Kbyte version, when the pin is high, becomes H'8001; when low, becomes H'0001.

2. Only 0 can be written after reading 1, to clear the flag.

chanhvinh.com

7.3.1 Interrupt Priority Registers 01, 02, 05 to 22 (IPR01, IPR02, IPR05 to IPR22)

IPR01, IPR02, and IPR05 to IPR22 are 16-bit readable/writable registers in which priority levels from 0 to 15 are set for IRQ interrupts, PINT interrupts, and on-chip peripheral module interrupts. Table 7.3 shows the correspondence between the interrupt request sources and the bits in IPR01, IPR02, and IPR05 to IPR22.

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

Table 7.3 Interrupt Request Sources and IPR01, IPR02, and IPR05 to IPR22

Register Name	Bits 15 to 12	Bits 11 to 8	Bits 7 to 4	Bits 3 to 0
IPR01	IRQ0	IRQ1	IRQ2	IRQ3
IPR02	IRQ4	IRQ5	IRQ6	IRQ7
IPR05	PINT7 to PINT0	Reserved	Reserved	Reserved
IPR06	Direct memory access controller channel 0	Direct memory access controller channel 1	Direct memory access controller channel 2	Direct memory access controller channel 3
IPR07	Direct memory access controller channel 4	Direct memory access controller channel 5	Direct memory access controller channel 6	Direct memory access controller channel 7
IPR08	Direct memory access controller channel 8	Direct memory access controller channel 9	Direct memory access controller channel 10	Direct memory access controller channel 11
IPR09	Direct memory access controller channel 12	Direct memory access controller channel 13	Direct memory access controller channel 14	Direct memory access controller channel 15
IPR10	USB 2.0 host/function module	Video display controller 3	Compare match timer channel 0	Compare match timer channel 1
IPR11	Bus state controller	Watchdog timer	Multi-function timer pulse unit 2 channel 0 (TGI0A to TGI0D)	Multi-function timer pulse unit 2 channel 0 (TCI0V, TGI0E, TGI0F)

chanhvinh.com

Register Name	Bits 15 to 12	Bits 11 to 8	Bits 7 to 4	Bits 3 to 0
IPR12	Multi-function timer pulse unit 2 channel 1 (TGI1A, TGI1B)	Multi-function timer pulse unit 2 channel 1 (TCI1V, TCI1U)	Multi-function timer pulse unit 2 channel 2 (TGI2A, TGI2B)	Multi-function timer pulse unit 2 channel 2 (TCI2V, TCI2U)
IPR13	Multi-function timer pulse unit 2 channel 3 (TGI3A to TGI3D)	Multi-function timer pulse unit 2 channel 3 (TCI3V)	Multi-function timer pulse unit 2 channel 4 (TGI4A to TGI4D)	Multi-function timer pulse unit 2 channel 4 (TCI4V)
IPR14	Motor control PWM timer channel 1	Motor control PWM timer channel 2	A/D converter	Reserved
IPR15	Serial sound interface channel 0	Serial1sound interface channel 0	Serial sound interface channel 2	Serial sound interface channel 3
IPR16	Renesas SPDIF interface	I^2C bus interface 3 channel 0	I^2C bus interface 3 channel 1	I^2C bus interface 3 channel 2
IPR17	Channel 0 for serial communication interface with FIFO	Channel 1 for serial communication interface with FIFO	Channel 2 for serial communication interface with FIFO	Channel 3 for serial communication interface with FIFO
IPR18	Channel 4 for serial communication interface with FIFO	Channel 5 for serial communication interface with FIFO	Channel 6 for serial communication interface with FIFO	Channel 7 for serial communication interface with FIFO
IPR19	Serial I/O with FIFO	Reserved	Renesas serial peripheral interface channel 0	Renesas serial peripheral interface channel 1
IPR20	Controller area network channel 0	Controller area network channel 1	IEBus™ controller	CD-ROM decoder
IPR21	NAND flash memory controller	SD host interface	Realtime clock	Reserved
IPR22	Sampling rate converter channel 0	Sampling rate converter channel 1	Reserved	Decompression unit

chanhvinh.com

As shown in table 7.3, by setting the 4-bit groups (bits 15 to 12, bits 11 to 8, bits 7 to 4, and bits 3 to 0) with values from H'0 (0000) to H'F (1111), the priority of each corresponding interrupt is set. Setting of H'0 means priority level 0 (the lowest level) and H'F means priority level 15 (the highest level).

7.3.2 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	NMIM
Initial value:	*1	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/W	R	R	R	R	R	R	R*3	R/(W)*2*3

Notes: 1. 1 when the NMI pin is high, and 0 when the NMI pin is low.
2. Only 0 can be written to this bit.
3. Only 640-Kbyte version is valid.

Bit	Bit Name	Initial Value	R/W	Description
15	NMIL	*	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 Note: Only 640-Kbyte version is valid. For 1-Mbyte version, these bits are reserved and always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
1	NMIF	0	R	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 executing NMI interrupt exception handling 1: NMI interrupt request is detected [Setting condition] • Edge corresponding to NMIE of ICR0 has occurred at NMI pin Note: Only 640-Kbyte version is valid. For 1-Mbyte version, this bit is reserved and always read as 0. The write value should always be 0.
0	NMIM	1	R/(W)*2	NMI Mask Selects whether to enable interrupt request input to external interrupt input pin NMI. 0: NMI input interrupt request is enabled 1: NMI input interrupt request is masked Note: Only 640-Kbyte version is valid. For 1-Mbyte version, this bit is reserved and always read as 0. The write value should always be 0.

chanhvinh.com

7.3.3 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
14	IRQ70S	0	R/W	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.
13	IRQ61S	0	R/W	
12	IRQ60S	0	R/W	00: Interrupt request is detected on low level of IRQn input
11	IRQ51S	0	R/W	
10	IRQ50S	0	R/W	01: Interrupt request is detected on falling edge of IRQn input
9	IRQ41S	0	R/W	
8	IRQ40S	0	R/W	10: Interrupt request is detected on rising edge of IRQn input
7	IRQ31S	0	R/W	11: Interrupt request is detected on both edges of IRQn input
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

7.3.4 Interrupt Control Register 2 (ICR2)

ICR2 is a 16-bit register that specifies the detection mode for external interrupt input pins PINT7 to PINT0 individually: low level or high level.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PINT7S	PINT6S	PINT5S	PINT4S	PINT3S	PINT2S	PINT1S	PINT0S
Initial value:	0	0	0	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
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	PINT7S	0	R/W	PINT Sense Select These bits select whether interrupt signals corresponding to pins PINT7 to PINT0 are detected by a low level or high level. 0: Interrupt request is detected on low level of PINTn input 1: Interrupt request is detected on high level of PINTn input
6	PINT6S	0	R/W	
5	PINT5S	0	R/W	
4	PINT4S	0	R/W	
3	PINT3S	0	R/W	
2	PINT2S	0	R/W	
1	PINT1S	0	R/W	
0	PINT0S	0	R/W	

[Legend]
n = 7 to 0

chanhvinh.com

7.3.5 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 conditions] • Cleared by reading IRQnF while IRQnF = 1, then writing 0 to IRQnF • Cleared by executing IRQn interrupt exception handling 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

7.3.6 PINT Interrupt Enable Register (PINTER)

PINTER is a 16-bit register that enables interrupt request inputs to external interrupt input pins PINT7 to PINT0.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PINT7E	PINT6E	PINT5E	PINT4E	PINT3E	PINT2E	PINT1E	PINT0E
Initial value:	0	0	0	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
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	PINT7E	0	R/W	PINT Enable These bits select whether to enable interrupt request inputs to external interrupt input pins PINT7 to PINT0. 0: PINTn input interrupt request is disabled 1: PINTn input interrupt request is enabled
6	PINT6E	0	R/W	
5	PINT5E	0	R/W	
4	PINT4E	0	R/W	
3	PINT3E	0	R/W	
2	PINT2E	0	R/W	
1	PINT1E	0	R/W	
0	PINT0E	0	R/W	

[Legend]
n = 7 to 0

chanhvinh.com

7.3.7 PINT Interrupt Request Register (PIRR)

PIRR is a 16-bit register that indicates interrupt requests from external input pins PINT7 to PINT0.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PINT7R	PINT6R	PINT5R	PINT4R	PINT3R	PINT2R	PINT1R	PINT0R
Initial value:	0	0	0	0	0	0	0	0	0	0	0	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	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	PINT7R	0	R	PINT Interrupt Request These bits indicate the status of the PINT7 to PINT0 interrupt requests. 0: No interrupt request at PINTn pin 1: Interrupt request at PINTn pin
6	PINT6R	0	R	
5	PINT5R	0	R	
4	PINT4R	0	R	
3	PINT3R	0	R	
2	PINT2R	0	R	
1	PINT1R	0	R	
0	PINT0R	0	R	

[Legend]

n = 7 to 0

chanhvinh.com

7.3.8 Bank Control Register (IBCR)

IBCR is a 16-bit register that enables or disables use of register banks for each interrupt priority level.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	-
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

Bit	Bit Name	Initial Value	R/W	Description
15	E15	0	R/W	Enable These bits enable or disable use of register banks for interrupt priority levels 15 to 1. 0: Use of register banks is disabled 1: Use of register banks is enabled
14	E14	0	R/W	
13	E13	0	R/W	
12	E12	0	R/W	
11	E11	0	R/W	
10	E10	0	R/W	
9	E9	0	R/W	
8	E8	0	R/W	
7	E7	0	R/W	
6	E6	0	R/W	
5	E5	0	R/W	
4	E4	0	R/W	
3	E3	0	R/W	
2	E2	0	R/W	
1	E1	0	R/W	
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

chanhvinh.com

7.3.9 Bank Number Register (IBNR)

IBNR is a 16-bit register that enables or disables use of register banks and register bank overflow exception. IBNR also indicates the bank number to which saving is performed next through the bits BN3 to BN0.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BE[1:0]		BOVE	-	-	-	-	-	-	-	-	-	BN[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	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15, 14	BE[1:0]	00	R/W	Register Bank Enable These bits enable or disable use of register banks. 00: Use of register banks is disabled for all interrupts. The setting of IBCR is ignored. 01: Use of register banks is enabled for all interrupts except NMI. The setting of IBCR is ignored. 10: Reserved (setting prohibited) 11: Use of register banks is controlled by the setting of IBCR.
13	BOVE	0	R/W	Register Bank Overflow Enable Enables of disables register bank overflow exception. 0: Generation of register bank overflow exception is disabled 1: Generation of register bank overflow exception is enabled
12 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	BN[3:0]	0000	R	Bank Number These bits indicate the bank number to which saving is performed next. When an interrupt using register banks is accepted, saving is performed to the register bank indicated by these bits, and BN is incremented by 1. After BN is decremented by 1 due to execution of a RESBANK (restore from register bank) instruction, restoration from the register bank is performed.

chanhvinh.com

7.4 Interrupt Sources

There are five types of interrupt sources: NMI, user debugging interface, IRQ, PINT, and on-chip peripheral modules. Each interrupt has a priority level (0 to 16), with 0 the lowest and 16 the highest. When set to level 0, that interrupt is masked at all times.

7.4.1 NMI Interrupt

- 1-Mbyte version

  The NMI interrupt has a priority level of 16 and is accepted at all times. NMI interrupt requests are edge-detected, and the NMI edge select bit (NMIE) in interrupt control register 0 (ICR0) selects whether the rising edge or falling edge is detected.

  Though the priority level of the NMI interrupt is 16, the NMI interrupt exception handling sets the interrupt mask level bits (I3 to I0) in the status register (SR) to level 15.

- 640-Kbyte version

  The NMI interrupt has a priority level of 16 and is accepted at all times when the NMI mask bit (NMIM) in interrupt control register 0 (ICR0) is enabled. 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.

  Though the priority level of the NMI interrupt is 16, the NMI interrupt exception handling sets the interrupt mask level bits (I3 to I0) in the status register (SR) to level 15.

  When the NMIM bit in ICR0 is set to 1 (NMI interrupt request is masked), the NMI interrupt is not generated, however the NMI edge corresponding to NMIE bit of ICR0 is detected and the NMI interrupt request is retained until the interrupt request is accepted. The status of the interrupt request can be checked by reading the NMI interrupt request bit (NMIF) in the ICR0. If 0 is written to the NMIM bit (NMI interrupt request is enabled) when the NMIF bit is set to 1, the NMI interrupt request that is retained is accepted. Once the NMIM bit is set to 0 (NMI interrupt request is enabled), the NMIM bit cannot be set to 1 again, because only 0 can be written to the NMIM bit. When the NME bit is changed, the NMI interrupt request that is retained is cleared.

chanhvinh.com

When canceling software standby mode by the NMI interrupt, set the NMIM bit to 0 to enable the NMI interrupt request after confirming that the NMI interrupt request has been cleared in the NMIF. If software standby mode is entered when the NMIM bit is 1 (NMI interrupt request is masked), the NMI interrupt cannot cancel software standby mode. In this case, the NMI edge cannot be detected in software standby mode and the NMI interrupt is not generated even if software standby mode is canceled by cancel source other than NMI. When the NMI pin keeps level (low level after the falling edge or high level after the rising edge) in software standby mode until software standby mode is canceled by cancel source other than NMI (until the clock is initiated after the oscillation settling), that edge of the NMI in software standby mode can be detected.

When deep standby mode is entered, deep standby mode is canceled by the NMI interrupt regardless of the NMI mask bit setting. NMIM bit is initialized by a power-on reset after canceling deep standby mode.

When a sleep instruction is to be executed after 0 has been written to the NMIM bit (enabling the NMI), read the value of the NMIM bit before executing the sleep instruction.

7.4.2 User Debugging Interface Interrupt

The user debugging interface interrupt has a priority level of 15, and occurs at serial input of a user debugging interface interrupt instruction. User debugging interface interrupt requests are edge-detected and retained until they are accepted. The user debugging interface interrupt exception handling sets the I3 to I0 bits in SR to level 15. For user debugging interface interrupts, see section 34, User Debugging Interface.

7.4.3 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). The priority level can be set individually in a range from 0 to 15 for each pin by interrupt priority registers 01 and 02 (IPR01 and IPR02).

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.

chanhvinh.com

Whether IRQ interrupt requests have been detected or not can be checked by reading the IRQ7F to IRQ0F bits in IRQRR. Writing 0 to these bits after reading them as 1 clears the result of IRQ interrupt request detection.

The IRQ interrupt exception handling sets the I3 to I0 bits in SR to the priority level of the accepted IRQ interrupt.

When returning from IRQ interrupt exception service routine, execute the RTE 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.

7.4.4 PINT Interrupts

PINT interrupts are input from pins PINT7 to PINT0. Input of the interrupt requests is enabled by the PINT enable bits (PINT7E to PINT0E) in the PINT interrupt enable register (PINTER). For the PINT7 to PINT0 interrupts, low-level or high-level detection can be selected individually for each pin by the PINT sense select bits (PINT7S to PINT0S) in interrupt control register 2 (ICR2). A single priority level in a range from 0 to 15 can be set for all PINT7 to PINT0 interrupts by bits 15 to 12 in interrupt priority register 05 (IPR05).

When using low-level sensing for the PINT7 to PINT0 interrupts, an interrupt request signal is sent to the interrupt controller while the PINT7 to PINT0 pins are low. An interrupt request signal is stopped being sent to the interrupt controller when the PINT7 to PINT0 pins are driven high. The status of the interrupt requests can be checked by reading the PINT interrupt request bits (PINT7R to PINT0R) in the PINT interrupt request register (PIRR). The above description also applies to when using high-level sensing, except for the polarity being reversed. The PINT interrupt exception handling sets the I3 to I0 bits in SR to the priority level of the PINT interrupt.

When returning from IRQ interrupt exception service routine, execute the RTE instruction after confirming that the interrupt request has been cleared by the PINT interrupt request register (PIRR) so as not to accidentally receive the interrupt request again.

chanhvinh.com

7.4.5 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 3
- Compare match timer
- Bus state controller
- Watchdog timer
- Multi-function timer pulse unit 2
- Motor control PWM timer
- A/D converter
- Serial sound interface
- Renesas SPDIF interface
- I^2C bus interface 3
- Serial communication interface with FIFO
- Serial I/O with FIFO
- Renesas serial peripheral interface
- Controller area network
- IEBus™ controller
- CD-ROM decoder
- NAND flash memory controller
- SD host interface
- Realtime clock
- Sampling rate converter
- Decompression unit

As every source is assigned a different interrupt vector, the source does not need to be identified in the exception service routine. A priority level in a range from 0 to 15 can be set for each module by interrupt priority registers 05 to 22 (IPR05 to IPR22). The on-chip peripheral module interrupt exception handling sets the I3 to I0 bits in SR to the priority level of the accepted on-chip peripheral module interrupt.

chanhvinh.com

7.5 Interrupt Exception Handling Vector Table and Priority

Table 7.4 lists interrupt sources and their vector numbers, vector table address offsets, and interrupt priorities.

Each interrupt source is allocated a different vector number and vector table address offset. Vector table addresses are calculated from the vector numbers and vector table address offsets. In interrupt exception handling, the interrupt exception service routine start address is fetched from the vector table indicated by the vector table address. For details of calculation of the vector table address, see table 6.4 in section 6, Exception Handling.

The priorities of IRQ interrupts, PINT interrupts, and on-chip peripheral module interrupts can be set freely between 0 and 15 for each pin or module by setting interrupt priority registers 01, 02, and 05 to 22 (IPR01, IPR02, and IPR05 to IPR22). However, if two or more interrupts specified by the same IPR among IPR05 to IPR22 occur, the priorities are defined as shown in the IPR setting unit internal priority of table 7.4, and the priorities cannot be changed. A power-on reset assigns priority level 0 to IRQ interrupts, PINT interrupts, and on-chip peripheral module interrupts. If the same priority level is assigned to two or more interrupt sources and interrupts from those sources occur simultaneously, they are processed by the default priorities indicated in table 7.4.

chanhvinh.com

Table 7.4 Interrupt Exception Handling Vectors and Priorities

Interrupt Source		Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
NMI		11	H'0000002C to H'0000002F	16	—	—	High
User debug interface		14	H'00000038 to H'0000003B	15	—	—	
IRQ	IRQ0	64	H'00000100 to H'00000103	0 to 15 (0)	IPR01 (15 to 12)	—	
	IRQ1	65	H'00000104 to H'00000107	0 to 15 (0)	IPR01 (11 to 8)	—	
	IRQ2	66	H'00000108 to H'0000010B	0 to 15 (0)	IPR01 (7 to 4)	—	
	IRQ3	67	H'0000010C to H'0000010F	0 to 15 (0)	IPR01 (3 to 0)	—	
	IRQ4	68	H'00000110 to H'00000113	0 to 15 (0)	IPR02 (15 to 12)	—	
	IRQ5	69	H'00000114 to H'00000117	0 to 15 (0)	IPR02 (11 to 8)	—	
	IRQ6	70	H'00000118 to H'0000011B	0 to 15 (0)	IPR02 (7 to 4)	—	
	IRQ7	71	H'0000011C to H'0000011F	0 to 15 (0)	IPR02 (3 to 0)	—	
PINT	PINT0	80	H'00000140 to H'00000143	0 to 15 (0)	IPR05 (15 to 12)	1	
	PINT1	81	H'00000144 to H'00000147			2	
	PINT2	82	H'00000148 to H'0000014B			3	
	PINT3	83	H'0000014C to H'0000014F			4	
	PINT4	84	H'00000150 to H'00000153			5	Low

chanhvinh.com

Interrupt Source			Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
PINT	PINT5		85	H'00000154 to H'00000157	0 to 15 (0)	IPR05 (15 to 12)	6	High
	PINT6		86	H'00000158 to H'0000015B			7	
	PINT7		87	H'0000015C to H'0000015F			8	
Direct memory access controller	Channel 0	DEI0	108	H'000001B0 to H'000001B3	0 to 15 (0)	IPR06 (15 to 12)	1	
		HEI0	109	H'000001B4 to H'000001B7			2	
	Channel 1	DEI1	112	H'000001C0 to H'000001C3	0 to 15 (0)	IPR06 (11 to 8)	1	
		HEI1	113	H'000001C4 to H'000001C7			2	
	Channel 2	DEI2	116	H'000001D0 to H'000001D3	0 to 15 (0)	IPR06 (7 to 4)	1	
		HEI2	117	H'000001D4 to H'000001D7			2	
	Channel 3	DEI3	120	H'000001E0 to H'000001E3	0 to 15 (0)	IPR06 (3 to 0)	1	
		HEI3	121	H'000001E4 to H'000001E7			2	
	Channel 4	DEI4	124	H'000001F0 to H'000001F3	0 to 15 (0)	IPR07 (15 to 12)	1	
		HEI4	125	H'000001F4 to H'000001F7			2	
	Channel 5	DEI5	128	H'00000200 to H'00000203	0 to 15 (0)	IPR07 (11 to 8)	1	
		HEI5	129	H'00000204 to H'00000207			2	
	Channel 6	DEI6	132	H'00000210 to H'00000213	0 to 15 (0)	IPR07 (7 to 4)	1	
		HEI6	133	H'00000214 to H'00000217			2	Low

chanhvinh.com

Interrupt Source			Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
Direct memory access controller	Channel 7	DEI7	136	H'00000220 to H'00000223	0 to 15 (0)	IPR07 (3 to 0)	1	High
		HEI7	137	H'00000224 to H'00000227			2	
	Channel 8	DEI8	140	H'00000230 to H'00000233	0 to 15 (0)	IPR08 (15 to 12)	1	
		HEI8	141	H'00000234 to H'00000237			2	
	Channel 9	DEI9	144	H'00000240 to H'00000243	0 to 15 (0)	IPR08 (11 to 8)	1	
		HEI9	145	H'00000244 to H'00000247			2	
	Channel 10	DEI10	148	H'00000250 to H'00000253	0 to 15 (0)	IPR08 (7 to 4)	1	
		HEI10	149	H'00000254 to H'00000257			2	
	Channel 11	DEI11	152	H'00000260 to H'00000263	0 to 15 (0)	IPR08 (3 to 0)	1	
		HEI11	153	H'00000264 to H'00000267			2	
	Channel 12	DEI12	156	H'00000270 to H'00000273	0 to 15 (0)	IPR09 (15 to 12)	1	
		HEI12	157	H'00000274 to H'00000277			2	
	Channel 13	DEI13	160	H'00000280 to H'00000283	0 to 15 (0)	IPR09 (11 to 8)	1	
		HEI13	161	H'00000284 to H'00000287			2	
	Channel 14	DEI14	164	H'00000290 to H'00000293	0 to 15 (0)	IPR09 (7 to 4)	1	
		HEI14	165	H'00000294 to H'00000297			2	Low

chanhvinh.com

Interrupt Source			Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit: Internal Priority	Default Priority
Direct memory access controller	Channel 15	DEI15	168	H'000002A0 to H'000002A3	0 to 15 (0)	IPR09 (3 to 0)	1	High
		HEI15	169	H'000002A4 to H'000002A7			2	
USB 2.0 host/ function module	USBI		170	H'000002A8 to H'000002AB	0 to 15 (0)	IPR10 (15 to 12)	—	
Video display controller 3	VIVSYNCJ		171	H'000002AC to H'000002AF	0 to 15 (0)	IPR10 (11 to 8)	1	
	VBUFERR		172	H'000002B0 to H'000002B3			2	
	VIFIELDE		173	H'000002B4 to H'000002B7			3	
	VOLINE		174	H'000002B8 to H'000002BB			4	
Compare match timer	Channel 0	CMI0	175	H'000002BC to H'000002BF	0 to 15 (0)	IPR10 (7 to 4)		
	Channel 1	CMI1	176	H'000002C0 to H'000002C3	0 to 15 (0)	IPR10 (3 to 0)	—	
Bus state controller	CMI		177	H'000002C4 to H'000002C7	0 to 15 (0)	IPR11 (15 to 12)	—	
Watchdog timer	ITI		178	H'000002C8 to H'000002CB	0 to 15 (0)	IPR11 (11 to 8)	—	Low

chanhvinh.com

Interrupt Source			Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
Multi-function timer pulse unit 2	Channel 0	TGI0A	179	H'000002CC to H'000002CF	0 to 15 (0)	IPR11 (7 to 4)	1	High
		TGI0B	180	H'000002D0 to H'000002D3			2	
		TGI0C	181	H'000002D4 to H'000002D7			3	
		TGI0D	182	H'000002D8 to H'000002DB			4	
		TCI0V	183	H'000002DC to H'000002DF	0 to 15 (0)	IPR11 (3 to 0)	1	
		TGI0E	184	H'000002E0 to H'000002E3			2	
		TGI0F	185	H'000002E4 to H'000002E7			3	
	Channel 1	TGI1A	186	H'000002E8 to H'000002EB	0 to 15 (0)	IPR12 (15 to 12)	1	
		TGI1B	187	H'000002EC to H'000002EF			2	
		TCI1V	188	H'000002F0 to H'000002F3	0 to 15 (0)	IPR12 (11 to 8)	1	
		TCI1U	189	H'000002F4 to H'000002F7			2	
	Channel 2	TGI2A	190	H'000002F8 to H'000002FB	0 to 15 (0)	IPR12 (7 to 4)	1	
		TGI2B	191	H'000002FC to H'000002FF			2	
		TCI2V	192	H'00000300 to H'00000303	0 to 15 (0)	IPR12 (3 to 0)	1	
		TCI2U	193	H'00000304 to H'00000307			2	Low

chanhvinh.com

Interrupt Source			Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
Multi-function timer pulse unit 2	Channel 3	TGI3A	194	H'00000308 to H'0000030B	0 to 15 (0)	IPR13 (15 to 12)	1	High
		TGI3B	195	H'0000030C to H'0000030F			2	
		TGI3C	196	H'00000310 to H'00000313			3	
		TGI3D	197	H'00000314 to H'00000317			4	
		TCI3V	198	H'00000318 to H'0000031B	0 to 15 (0)	IPR13 (11 to 8)	—	
	Channel 4	TGI4A	199	H'0000031C to H'0000031F	0 to 15 (0)	IPR13 (7 to 4)	1	
		TGI4B	200	H'00000320 to H'00000323			2	
		TGI4C	201	H'00000324 to H'00000327			3	
		TGI4D	202	H'00000328 to H'0000032B			4	
		TCI4V	203	H'0000032C to H'0000032F	0 to 15 (0)	IPR13 (3 to 0)	—	
Motor control PWM timer	Channel 1		204	H'00000330 to H'00000333	0 to 15 (0)	IPR14 (15 to 12)	—	
	Channel 2		205	H'00000334 to H'00000337	0 to 15 (0)	IPR14 (11 to 8)	—	
A/D con-verter	ADI		206	H'00000338 to H'0000033B	0 to 15 (0)	IPR14 (7 to 4)	—	
Serial sound interface	Channel 0	SSIF0	207	H'0000033C to H'0000033F	0 to 15 (0)	IPR15 (15 to 12)	1	
		SSIRXI0	208	H'00000340 to H'00000343			2	
		SSITXI0	209	H'00000344 to H'00000347			3	Low

chanhvinh.com

Interrupt Source			Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
Serial sound interface	Channel 1	SSII1	210	H'00000348 to H'0000034B	0 to 15 (0)	IPR15 (11 to 8)	1	High
		SSIRTI1	211	H'0000034C to H'0000034F			2	
	Channel 2	SSII2	212	H'00000350 to H'00000353	0 to 15 (0)	IPR15 (7 to 4)	1	
		SSIRTI2	213	H'00000354 to H'00000357			2	
	Channel 3	SSII3	214	H'00000358 to H'0000035B	0 to 15 (0)	IPR15 (3 to 0)	1	
		SSIRTI3	215	H'0000035C to H'0000035F			2	
Renesas SPDIF interface	SPDIFI		216	H'00000360 to H'00000363	0 to 15 (0)	IPR16 (15 to 12)	—	
I²C bus interface 3	Channel 0	STPI0	217	H'00000364 to H'00000367	0 to 15 (0)	IPR16 (11 to 8)	1	
		NAKI0	218	H'00000368 to H'0000036B			2	
		RXI0	219	H'0000036C to H'0000036F			3	
		TXI0	220	H'00000370 to H'00000373			4	
		TEI0	221	H'00000374 to H'00000377			5	Low

chanhvinh.com

Interrupt Source			Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit: Internal Priority	Default Priority
I²C bus interface 3	Channel 1	STPI1	222	H'00000378 to H'0000037B	0 to 15 (0)	IPR16 (7 to 4)	1	High
		NAKI1	223	H'0000037C to H'0000037F			2	
		RXI1	224	H'00000380 to H'00000383			3	
		TXI1	225	H'00000384 to H'00000387			4	
		TEI1	226	H'00000388 to H'0000038B			5	
	Channel 2	STPI2	227	H'0000038C to H'0000038F	0 to 15 (0)	IPR16 (3 to 0)	1	
		NAKI2	228	H'00000390 to H'00000393			2	
		RXI2	229	H'00000394 to H'00000397			3	
		TXI2	230	H'00000398 to H'0000039B			4	
		TEI2	231	H'0000039C to H'0000039F			5	
Serial communi-cation interface with FIFO	Channel 0	BRI0	232	H'000003A0 to H'000003A3	0 to 15 (0)	IPR17 (15 to 12)	1	
		ERI0	233	H'000003A4 to H'000003A7			2	
		RXI0	234	H'000003A8 to H'000003AB			3	
		TXI0	235	H'000003AC to H'0000030AF			4	Low

chanhvinh.com

Interrupt Source			Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
Serial communi-cation interface with FIFO	Channel 1	BRI1	236	H'000003B0 to H'000003B3	0 to 15 (0)	IPR17 (11 to 8)	1	High
		ERI1	237	H'000003B4 to H'000003B7			2	
		RXI1	238	H'000003B8 to H'000003BB			3	
		TXI1	239	H'000003BC to H'000003BF			4	
	Channel 2	BRI2	240	H'000003C0 to H'000003C3	0 to 15 (0)	IPR17 (7 to 4)	1	
		ERI2	241	H'000003C4 to H'000003C7			2	
		RXI2	242	H'000003C8 to H'000003CB			3	
		TXI2	243	H'000003CC to H'000003CF			4	
	Channel 3	BRI3	244	H'000003D0 to H'000003D3	0 to 15 (0)	IPR17 (3 to 0)	1	
		ERI3	245	H'000003D4 to H'000003D7			2	
		RXI3	246	H'000003D8 to H'000003DB			3	
		TXI3	247	H'000003DC to H'000003DF			4	
	Channel 4	BRI4	248	H'000003E0 to H'000003E3	0 to 15 (0)	IPR18 (15 to 12)	1	
		ERI4	249	H'000003E4 to H'000003E7			2	
		RXI4	250	H'000003E8 to H'000003EB			3	
		TXI4	251	H'000003EC to H'000003EF			4	Low

chanhvinh.com

Interrupt Source			Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit: Internal Priority	Default Priority
Serial communi-cation interface with FIFO	Channel 5	BRI5	252	H'000003F0 to H'000003F3	0 to 15 (0)	IPR18 (11 to 8)	1	High
		ERI5	253	H'000003F4 to H'000003F7			2	
		RXI5	254	H'000003F8 to H'000003FB			3	
		TXI5	255	H'000003FC to H'000003FF			4	
	Channel 6	BRI6	256	H'00000400 to H'00000403	0 to 15 (0)	IPR18 (7 to 4)	1	
		ERI6	257	H'00000404 to H'00000407			2	
		RXI6	258	H'00000408 to H'0000040B			3	
		TXI6	259	H'0000040C to H'0000040F			4	
	Channel 7	BRI7	260	H'00000410 to H'00000413	0 to 15 (0)	IPR18 (3 to 0)	1	
		ERI7	261	H'00000414 to H'00000417			2	
		RXI7	262	H'00000418 to H'0000041B			3	
		TXI7	263	H'0000041C to H'0000041F			4	
Serial I/O with FIFO	SIOFI		264	H'00000420 to H'00000423	0 to 15 (0)	IPR19 (15 to 12)	—	Low

chanhvinh.com

Interrupt Source			Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
Renesas serial peripheral interface	Channel 0	SPEI0	265	H'00000424 to H'00000427	0 to 15 (0)	IPR19 (7 to 4)	1	High
		SPRI0	266	H'00000428 to H'0000042B			2	
		SPTI0	267	H'0000042C to H'0000042F			3	
	Channel 1	SPEI1	268	H'00000430 to H'00000433	0 to 15 (0)	IPR19 (3 to 0)	1	
		SPRI1	269	H'00000434 to H'00000437			2	
		SPTI1	270	H'00000438 to H'0000043B			3	
Controller area network	Channel 0	ERS0	271	H'0000043C to H'0000043F	0 to 15 (0)	IPR20 (15 to 12)	1	
		OVR0	272	H'00000440 to H'00000443			2	
		RM00	273	H'00000444 to H'00000447			3	
		RM10	274	H'00000448 to H'0000044B			4	
		SLE0	275	H'0000044C to H'0000044F			5	
	Channel 1	ERS1	276	H'00000450 to H'00000453	0 to 15 (0)	IPR20 (11 to 8)	1	
		OVR1	277	H'00000454 to H'00000457			2	
		RM01	278	H'00000458 to H'0000045B			3	
		RM11	279	H'0000045C to H'0000045F			4	
		SLE1	280	H'00000460 to H'00000463			5	Low

chanhvinh.com

Interrupt Source		Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
IEBus™ controller	IEB	281	H'00000464 to H'00000467	0 to 15 (0)	IPR20 (7 to 4)	—	High
CD-ROM decoder	ISY	282	H'00000468 to H'0000046B	0 to 15 (0)	IPR20 (3 to 0)	1	
	IERR	283	H'0000046C to H'0000046F			2	
	ITARG	284	H'00000470 to H'00000473			3	
	ISEC	285	H'00000474 to H'00000477			4	
	IBUF	286	H'00000478 to H'0000047B			5	
	IREADY	287	H'0000047C to H'0000047F			6	
NAND flash memory controller	FLSTEI	288	H'00000480 to H'00000483	0 to 15 (0)	IPR21 (15 to 12)	1	
	FLTENDI	289	H'00000484 to H'00000487			2	
	FLTREQ0I	290	H'00000488 to H'0000048B			3	
	FLTREQ1I	291	H'0000048C to H'0000048F			4	
SD host interface	SDHI3	292	H'00000490 to H'00000493	0 to 15 (0)	IPR21 (11 to 8)	1	
	SDHI0	293	H'00000494 to H'00000497			2	
	SDHI1	294	H'00000498 to H'0000049B			3	Low

chanhvinh.com

Interrupt Source			Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
Realtime clock	ARM		296	H'000004A0 to H'000004A3	0 to 15 (0)	IPR21 (7 to 4)	1	High
	PRD		297	H'000004A4 to H'000004A7			2	
	CUP		298	H'000004A8 to H'000004AB			3	
Sampling rate converter	Channel 0	OVF0	299	H'000004AC to H'000004AF	0 to 15 (0)	IPR22 (15 to 12)	1	
		UDF0	300	H'000004B0 to H'000004B3			2	
		CEF0	301	H'000004B4 to H'000004B7			3	
		ODFI0	302	H'000004B8 to H'000004BB			4	
		IDEI0	303	H'000004BC to H'000004BF			5	
	Channel 1	OVF1	304	H'000004C0 to H'000004C3	0 to 15 (0)	IPR22 (11 to 8)	1	
		UDF1	305	H'000004C4 to H'000004C7			2	
		CEF1	306	H'000004C8 to H'000004CB			3	
		ODFI1	307	H'000004CC to H'000004CF			4	
		IDEI1	308	H'000004D0 to H'000004D3			5	
Decom-pression unit	DCUEI		310	H'000004D8 to H'000004DB	0 to 15 (0)	IPR22 (3 to 0)	1	
	OFFI		311	H'000004DC to H'000004DF			2	
	IFEI		312	H'000004E0 to H'000004E3			3	Low

chanhvinh.com

7.6 Operation

7.6.1 Interrupt Operation Sequence

The sequence of interrupt operations is described below. Figure 7.2 shows the operation flow.

1. The interrupt request sources send interrupt request signals to the interrupt controller.
2. The interrupt controller selects the highest-priority interrupt from the interrupt requests sent, following the priority levels set in interrupt priority registers 01, 02, and 05 to 22 (IPR01, IPR02, and IPR05 to IPR22). Lower priority interrupts are ignored*. If two of these interrupts have the same priority level or if multiple interrupts occur within a single IPR, the interrupt with the highest priority is selected, according to the default priority and IPR setting unit internal priority shown in table 7.4.
3. The priority level of the interrupt selected by the interrupt controller is compared with the interrupt level mask bits (I3 to I0) in the status register (SR) of the CPU. If the interrupt request priority level is equal to or less than the level set in bits I3 to I0, the interrupt request is ignored. If the interrupt request priority level is higher than the level in bits I3 to I0, the interrupt controller accepts the interrupt and sends an interrupt request signal to the CPU.
4. The CPU detects the interrupt request sent from the interrupt controller when the CPU decodes the instruction to be executed. Instead of executing the decoded instruction, the CPU starts interrupt exception handling (figure 7.4).
5. The interrupt exception service routine start address is fetched from the exception handling vector table corresponding to the accepted interrupt.
6. The status register (SR) is saved onto the stack, and the priority level of the accepted interrupt is copied to bits I3 to I0 in SR.
7. The program counter (PC) is saved onto the stack.
8. The CPU jumps to the fetched interrupt exception service routine start address and starts executing the program. The jump that occurs is not a delayed branch.

chanhvinh.com

Notes: The interrupt source flag should be cleared in the interrupt handler. After clearing the interrupt source flag, "time from occurrence of interrupt request until interrupt controller identifies priority, compares it with mask bits in SR, and sends interrupt request signal to CPU" shown in table 7.5 is required before the interrupt source sent to the CPU is actually cancelled. To ensure that an interrupt request that should have been cleared is not inadvertently accepted again, read the interrupt source flag after it has been cleared, and then execute an RTE instruction.

* Interrupt requests that are designated as edge-sensing are held pending until the interrupt requests are accepted. IRQ interrupts, however, can be cancelled by accessing the IRQ interrupt request register (IRQRR). For details, see section 7.4.3, IRQ Interrupts.
  Interrupts held pending due to edge-sensing are cleared by a power-on reset.

chanhvinh.com

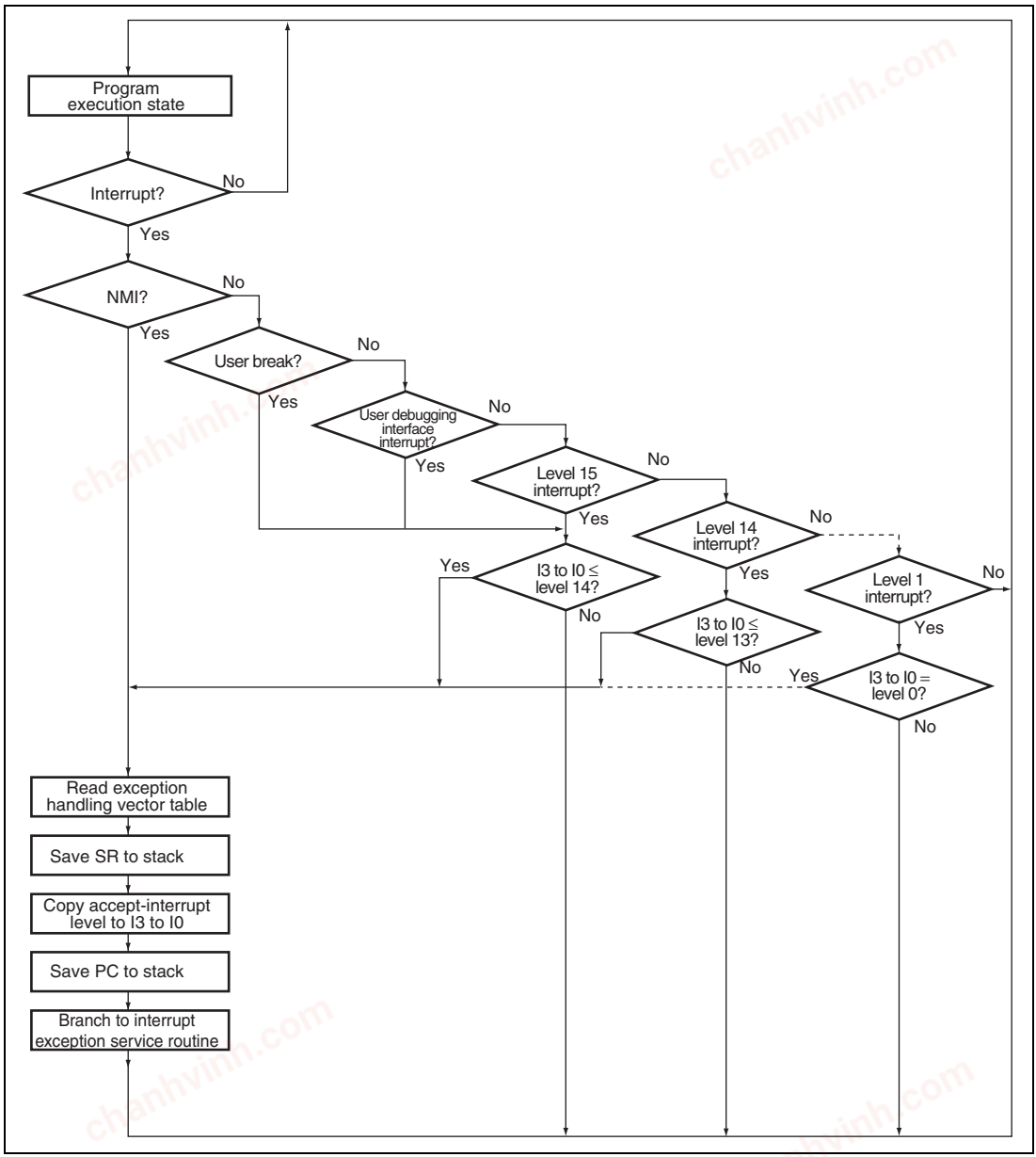


Figure 7.2 Interrupt Operation Flow

chanhvinh.com

7.6.2 Stack after Interrupt Exception Handling

Figure 7.3 shows the stack after interrupt exception handling.

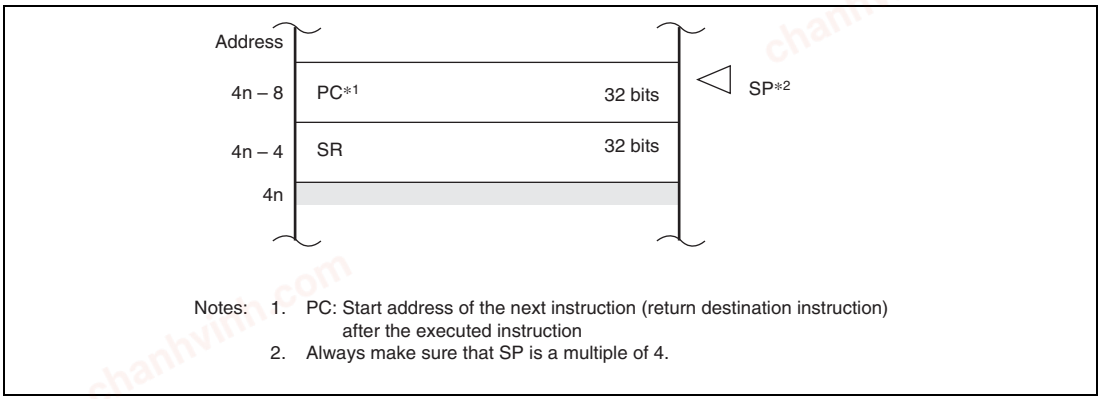


Figure 7.3 Stack after Interrupt Exception Handling

chanhvinh.com

7.7 Interrupt Response Time

Table 7.5 lists the interrupt response time, which is the time from the occurrence of an interrupt request until the interrupt exception handling starts and fetching of the first instruction in the exception service routine begins. The interrupt processing operations differ in the cases when banking is disabled, when banking is enabled without register bank overflow, and when banking is enabled with register bank overflow. Figures 7.4 and 7.5 show examples of pipeline operation when banking is disabled. Figures 7.6 and 7.7 show examples of pipeline operation when banking is enabled without register bank overflow. Figures 7.8 and 7.9 show examples of pipeline operation when banking is enabled with register bank overflow.

Table 7.5 Interrupt Response Time

			Number of States					
Item			**NMI**	**User Debugging Interface**	**IRQ, PINT**	**USB 2.0 host/ function module**	**Peripheral Module (Other than USB 2.0 host/ function module)**	**Remarks**
Time from occurrence of interrupt request until interrupt controller identifies priority, compares it with mask bits in SR, and sends interrupt request signal to CPU			2 Icyc + 2 Bcyc + 1 Pcyc	2 Icyc + 1 Pcyc	2 Icyc + 3 Bcyc + 1 Pcyc	2 Icyc + 4 Bcyc	2 Icyc + 2 Bcyc	
Time from input of interrupt request signal to CPU until sequence currently being executed is completed, interrupt exception handling starts, and first instruction in interrupt exception service routine is fetched	No register banking	Min.	3 Icyc + m1 + m2					Min. is when the interrupt wait time is zero. Max. is when a higher-priority interrupt request has occurred during interrupt exception handling.
		Max.	4 Icyc + 2(m1 + m2) + m3					
	Register banking without register bank overflow	Min.	—	3 Icyc + m1 + m2				Min. is when the interrupt wait time is zero. Max. is when an interrupt request has occurred during execution of the RESBANK instruction.
		Max.	—	12 Icyc + m1 + m2				
	Register banking with register bank overflow	Min.	—	3 Icyc + m1 + m2				Min. is when the interrupt wait time is zero. Max. is when an interrupt request has occurred during execution of the RESBANK instruction.
		Max.	—	3 Icyc + m1 + m2 + 19(m4)				

chanhvinh.com

Item			Number of States: NMI	User Debugging Interface	IRQ, PINT	USB 2.0 host/ function module	Peripheral Module (Other than USB 2.0 host/ function module)	Remarks
Interrupt response time	No register banking	Min.	5 Icyc + 2 Bcyc + 1 Pcyc + m1 + m2	5 Icyc + 1 Pcyc + m1 + m2	5 Icyc + 3 Bcyc + 1 Pcyc + m1 + m2	5 Icyc + 4 Bcyc + m1 + m2	5 Icyc + 2 Bcyc + m1 + m2	144-MHz operation*1*2: 0.076 to 0.118 µs
		Max.	6 Icyc + 2 Bcyc + 1 Pcyc + 2(m1 + m2) + m3	6 Icyc + 1 Pcyc + 2(m1 + m2) + m3	6 Icyc + 3 Bcyc + 1 Pcyc + 2(m1 + m2) + m3	6 Icyc + 4 Bcyc + 2(m1 + m2) + m3	6 Icyc + 2 Bcyc + 2(m1 + m2) + m3	144-MHz operation*1*2: 0.104 to 0.145 µs
	Register banking without register bank overflow	Min.	—	5 Icyc + 1 Pcyc + m1 + m2	5 Icyc + 3 Bcyc + 1 Pcyc + m1 + m2	5 Icyc + 4 Bcyc + m1 + m2	5 Icyc + 2 Bcyc + m1 + m2	144-MHz operation*1*2: 0.076 to 0.118 µs
		Max.	—	14 Icyc + 1 Pcyc + m1 + m2	14 Icyc + 3 Bcyc + 1 Pcyc + m1 + m2	14 Icyc + 4 Bcyc + m1 + m2	14 Icyc + 2 Bcyc + m1 + m2	144-MHz operation*1*2: 0.138 to 0.180 µs
	Register banking with register bank overflow	Min.	—	5 Icyc + 1 Pcyc + m1 + m2	5 Icyc + 3 Bcyc + 1 Pcyc + m1 + m2	5 Icyc + 4 Bcyc + m1 + m2	5 Icyc + 2 Bcyc + m1 + m2	144-MHz operation*1*2: 0.076 to 0.118 µs
		Max.	—	5 Icyc + 1 Pcyc + m1 + m2 + 19(m4)	5 Icyc + 3 Bcyc + 1 Pcyc + m1 + m2 + 19(m4)	5 Icyc + 4 Bcyc + m1 + m2 + 19(m4)	5 Icyc + 2 Bcyc + m1 + m2 + 19(m4)	144-MHz operation*1*2: 0.208 to 0.250 µs

Notes: m1 to m4 are the number of states needed for the following memory accesses.

m1: Vector address read (longword read)

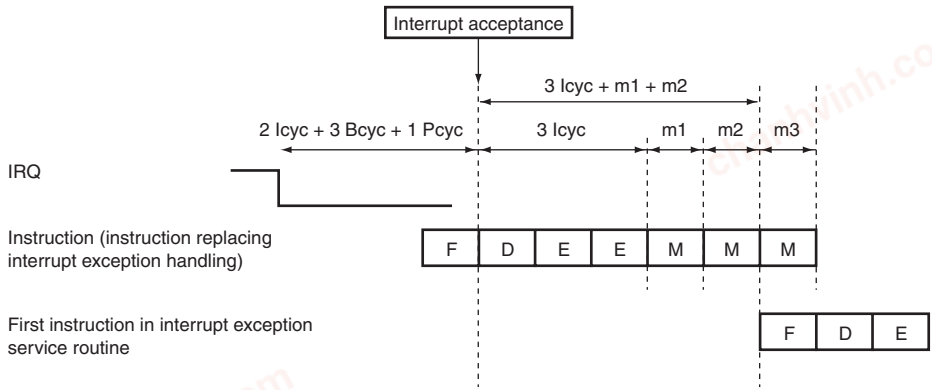
m2: SR save (longword write)

m3: PC save (longword write)

m4: Banked registers (R0 to R14, GBR, MACH, MACL, and PR) are restored from the stack.

1. In the case that m1 = m2 = m3 = m4 = 1 Icyc.
2. In the case that (Iϕ, Bϕ, Pϕ) = (144 MHz, 72 MHz, 36 MHz).

chanhvinh.com



[Legend]
m1: Vector address read
m2: Saving of SR (stack)
m3: Saving of PC (stack)
F: Instruction fetch. Instruction is fetched from memory in which program is stored.
D: Instruction decoding. Fetched instruction is decoded.
E: Instruction execution. Data operation or address calculation is performed in accordance with the result of decoding.
M: Memory access. Memory data access is performed.

Figure 7.4 Example of Pipeline Operation when IRQ Interrupt is Accepted (No Register Banking)

chanhvinh.com

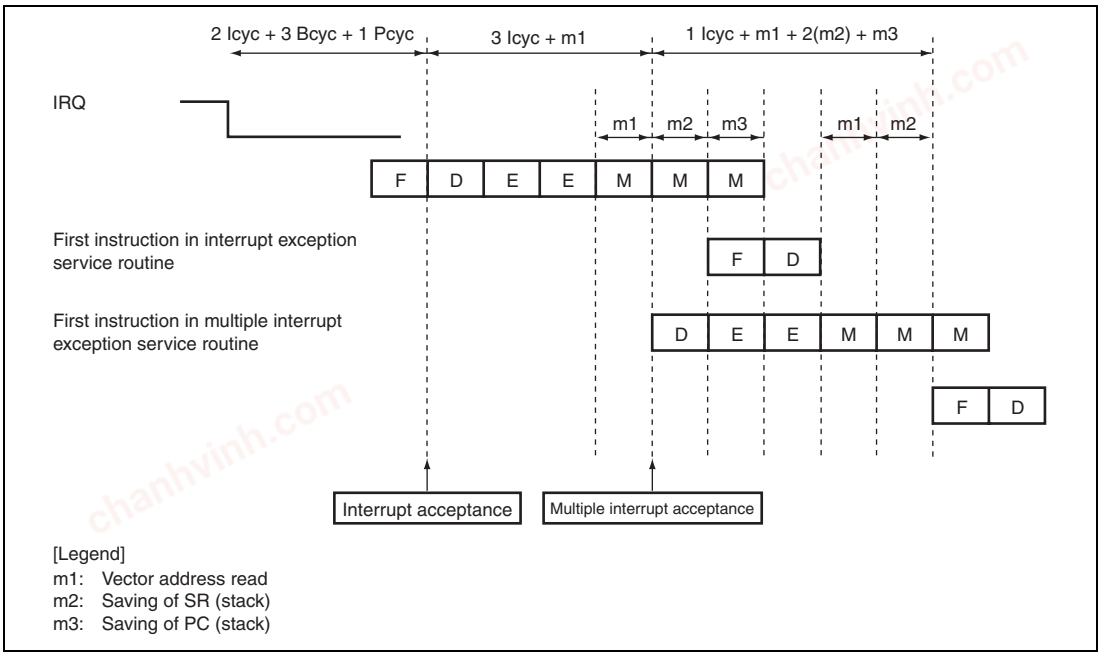


Figure 7.5 Example of Pipeline Operation for Multiple Interrupts (No Register Banking)

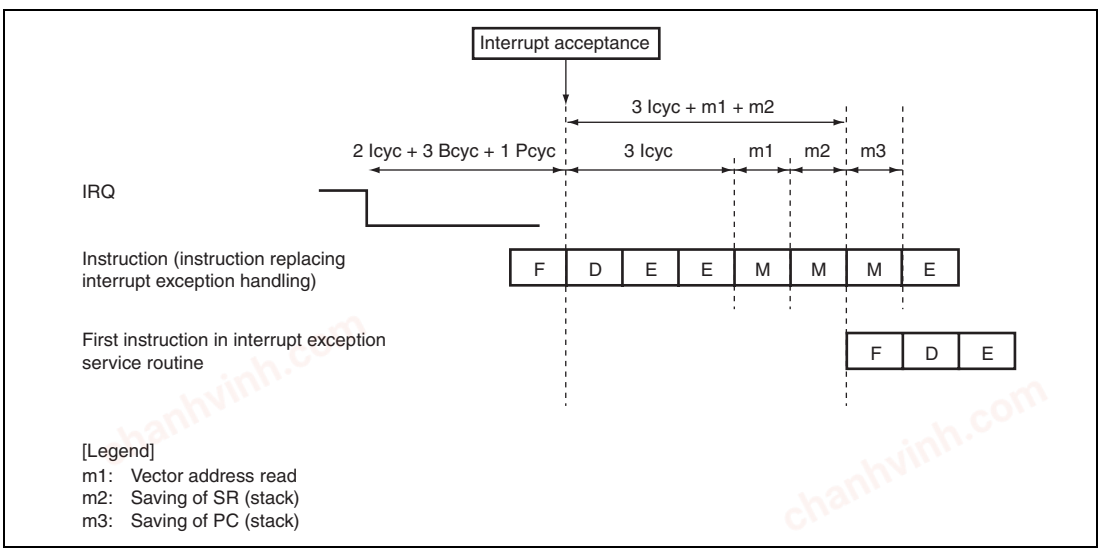


Figure 7.6 Example of Pipeline Operation when IRQ Interrupt is Accepted (Register Banking without Register Bank Overflow)

chanhvinh.com

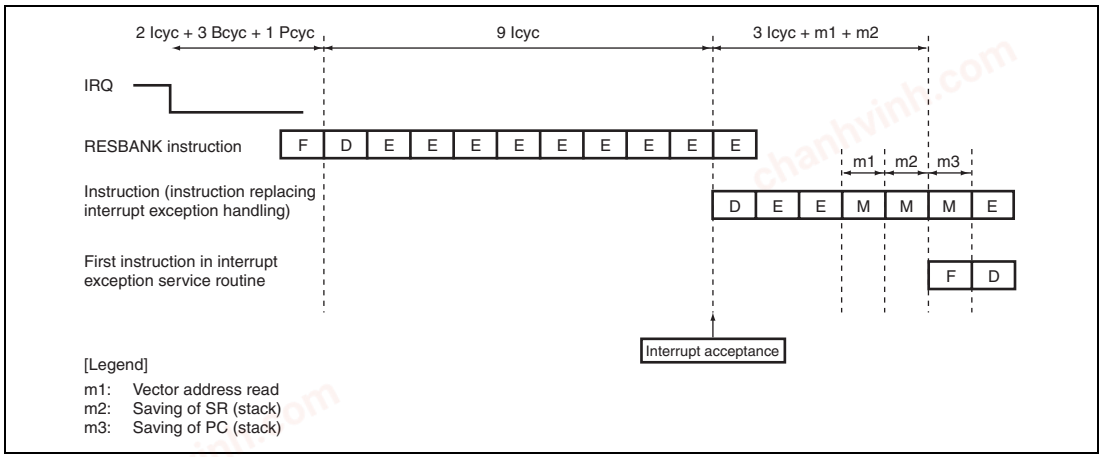


Figure 7.7 Example of Pipeline Operation when Interrupt is Accepted during RESBANK Instruction Execution (Register Banking without Register Bank Overflow)

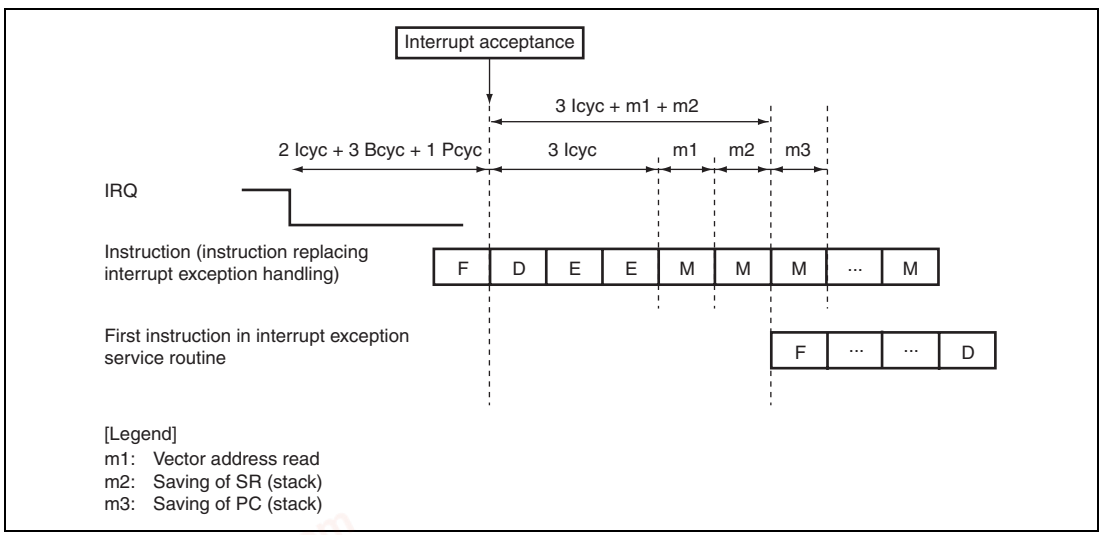


Figure 7.8 Example of Pipeline Operation when IRQ Interrupt is Accepted (Register Banking with Register Bank Overflow)

chanhvinh.com

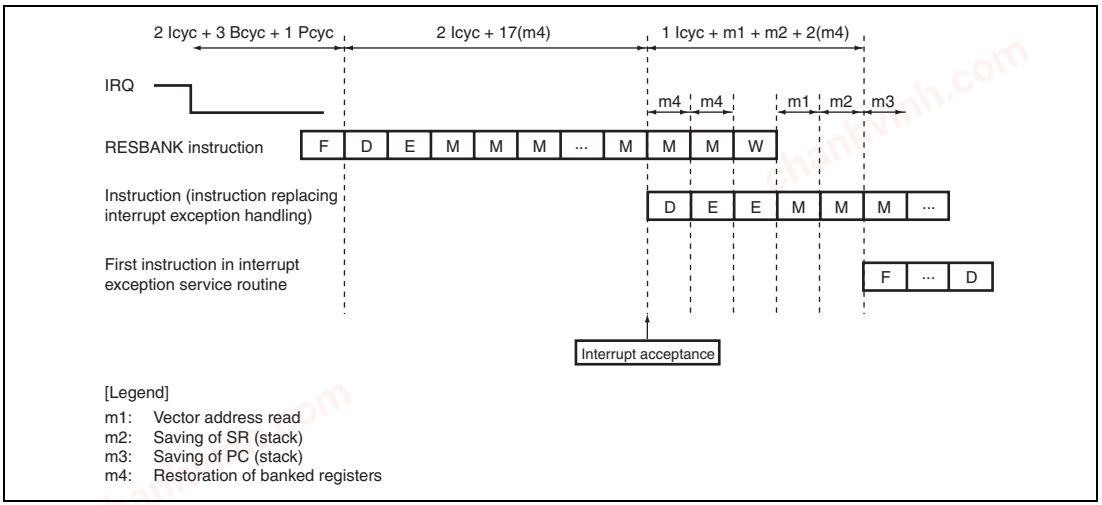


Figure 7.9 Example of Pipeline Operation when Interrupt is Accepted during RESBANK Instruction Execution (Register Banking with Register Bank Overflow)

chanhvinh.com

7.8 Register Banks

This LSI has fifteen register banks used to perform register saving and restoration required in the interrupt processing at high speed. Figure 7.10 shows the register bank configuration.

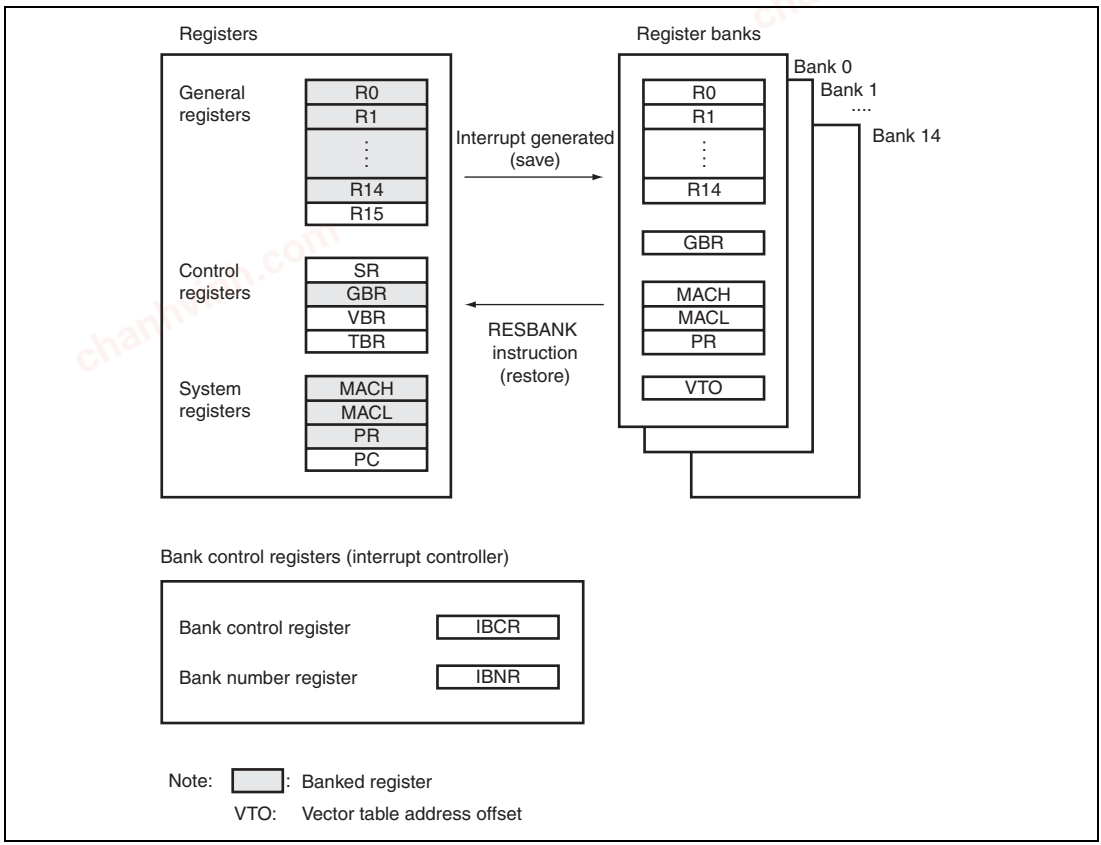


Figure 7.10 Overview of Register Bank Configuration

chanhvinh.com

7.8.1 Banked Register and Input/Output of Banks

(1) Banked Register

The contents of the general registers (R0 to R14), global base register (GBR), multiply and accumulate registers (MACH and MACL), and procedure register (PR), and the vector table address offset are banked.

(2) Input/Output of Banks

This LSI has fifteen register banks, bank 0 to bank 14. Register banks are stacked in first-in last-out (FILO) sequence. Saving takes place in order, beginning from bank 0, and restoration takes place in the reverse order, beginning from the last bank saved to.

7.8.2 Bank Save and Restore Operations

(1) Saving to Bank

Figure 7.11 shows register bank save operations. The following operations are performed when an interrupt for which usage of register banks is allowed is accepted by the CPU:

a. Assume that the bank number bit value in the bank number register (IBNR), BN, is i before the interrupt is generated.
b. The contents of registers R0 to R14, GBR, MACH, MACL, and PR, and the interrupt vector table address offset (VTO) of the accepted interrupt are saved in the bank indicated by BN, bank i.
c. The BN value is incremented by 1.

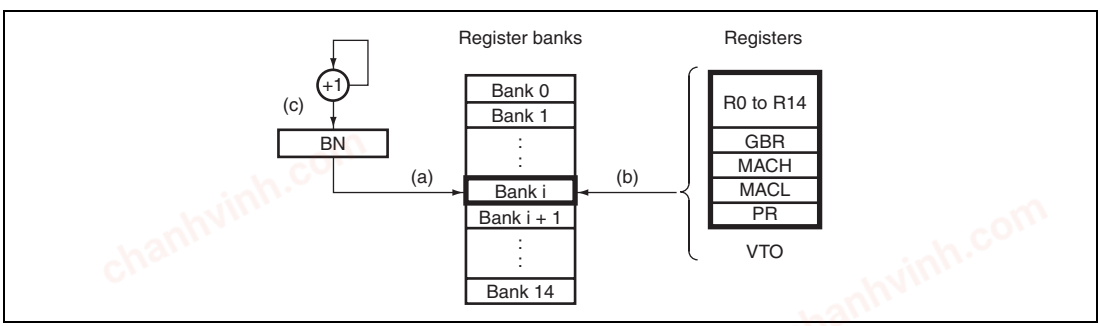


Figure 7.11 Bank Save Operations

chanhvinh.com

Figure 7.12 shows the timing for saving to a register bank. Saving to a register bank takes place between the start of interrupt exception handling and the start of fetching the first instruction in the interrupt exception service routine.

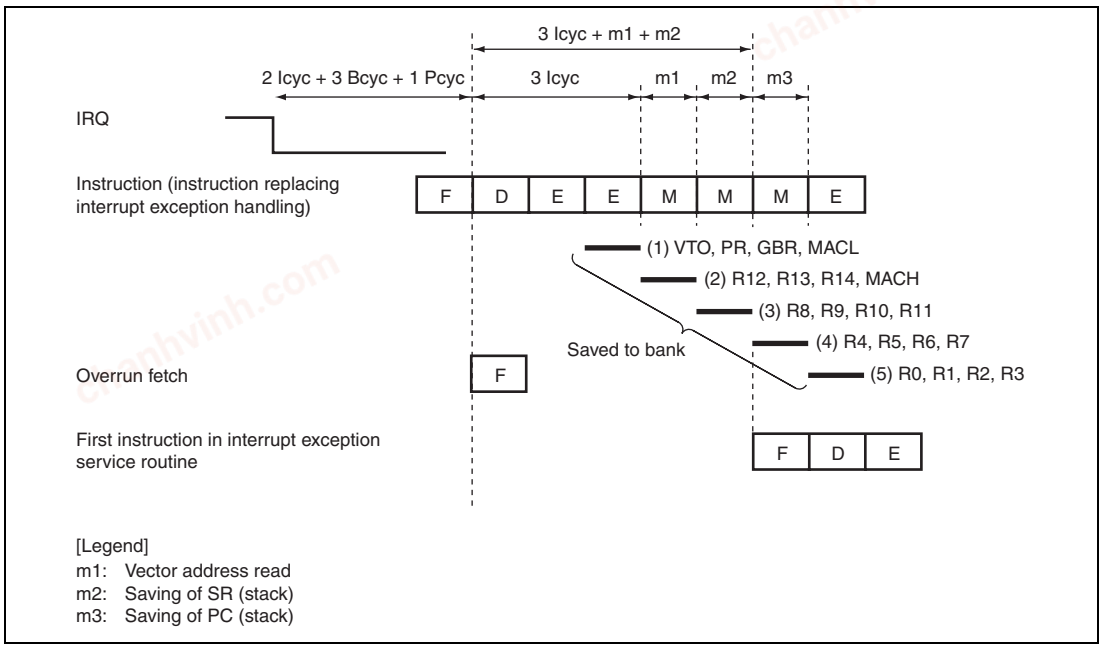


Figure 7.12 Bank Save Timing

(2) Restoration from Bank

The RESBANK (restore from register bank) instruction is used to restore data saved in a register bank. After restoring data from the register banks with the RESBANK instruction at the end of the interrupt exception service routine, execute the RTE instruction to return from interrupt exception service routine.

chanhvinh.com

7.8.3 Save and Restore Operations after Saving to All Banks

If an interrupt occurs and usage of the register banks is enabled for the interrupt accepted by the CPU in a state where saving has been performed to all register banks, automatic saving to the stack is performed instead of register bank saving if the BOVE bit in the bank number register (IBNR) is cleared to 0. If the BOVE bit in IBNR is set to 1, register bank overflow exception occurs and data is not saved to the stack.

Save and restore operations when using the stack are as follows:

(1) Saving to Stack

1. The status register (SR) and program counter (PC) are saved to the stack during interrupt exception handling.
2. The contents of the banked registers (R0 to R14, GBR, MACH, MACL, and PR) are saved to the stack. The registers are saved to the stack in the order of MACL, MACH, GBR, PR, R14, R13, …, R1, and R0.
3. The register bank overflow bit (BO) in SR is set to 1.
4. The bank number bit (BN) value in the bank number register (IBNR) remains set to the maximum value of 15.

(2) Restoration from Stack

When the RESBANK (restore from register bank) instruction is executed with the register bank overflow bit (BO) in SR set to 1, the CPU operates as follows:

1. The contents of the banked registers (R0 to R14, GBR, MACH, MACL, and PR) are restored from the stack. The registers are restored from the stack in the order of R0, R1, …, R13, R14, PR, GBR, MACH, and MACL.
2. The bank number bit (BN) value in the bank number register (IBNR) remains set to the maximum value of 15.

chanhvinh.com

7.8.4 Register Bank Exception

There are two register bank exceptions (register bank errors): register bank overflow and register bank underflow.

(1) Register Bank Overflow

This exception occurs if, after data has been saved to all of the register banks, an interrupt for which register bank use is allowed is accepted by the CPU, and the BOVE bit in the bank number register (IBNR) is set to 1. In this case, the bank number bit (BN) value in the bank number register (IBNR) remains set to the bank count of 15 and saving is not performed to the register bank.

(2) Register Bank Underflow

This exception occurs if the RESBANK (restore from register bank) instruction is executed when no data has been saved to the register banks. In this case, the values of R0 to R14, GBR, MACH, MACL, and PR do not change. In addition, the bank number bit (BN) value in the bank number register (IBNR) remains set to 0.

7.8.5 Register Bank Error Exception Handling

When a register bank error occurs, register bank error exception handling starts. When this happens, the CPU operates as follows:

1. The exception service routine start address which corresponds to the register bank error that occurred is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the last executed instruction for a register bank overflow, and the start address of the executed RESBANK instruction for a register bank underflow. To prevent multiple interrupts from occurring at a register bank overflow, the interrupt priority level that caused the register bank overflow is written to the interrupt mask level bits (I3 to I0) of the status register (SR).
4. Program execution starts from the exception service routine start address.

chanhvinh.com

7.9 Data Transfer with Interrupt Request Signals

Interrupt request signals can be used to activate the direct memory access controller and transfer data.

Interrupt sources that are designated to activate the direct memory access controller are masked without being input to the interrupt controller. The mask condition is as follows:

Mask condition = DME • (DE0 • interrupt source select 0 + DE1 • interrupt source select 1 + DE2 • interrupt source select 2 + DE3 • interrupt source select 3 + DE4 • interrupt source select 4 + DE5 • interrupt source select 5 + DE6 • interrupt source select 6 + DE7 • interrupt source select 7 + DE8 • interrupt source select 8 + DE9 • interrupt source select 9 + DE10 • interrupt source select 10 + DE11 • interrupt source select 11 + DE12 • interrupt source select 12 + DE13 • interrupt source select 13 + DE14 • interrupt source select 14 + DE15 • interrupt source select 15)

Figure 7.13 shows a block diagram of interrupt control.

Here, DME is bit 0 in DMAOR of the direct memory access controller, and DEn (n = 0 to 15) is bit 0 in CHCR_0 to CHCR_15 of the direct memory access controller. For details, see section 10, Direct Memory Access Controller.

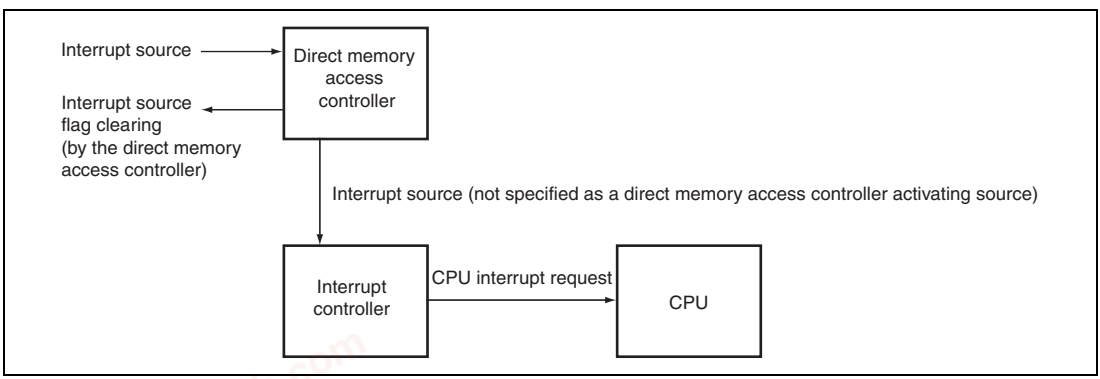


Figure 7.13 Interrupt Control Block Diagram

chanhvinh.com

7.9.1 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 or clear the DME bit to 0. If, direct memory access controller activating sources are selected, clear the DE bit to 0 for the relevant channel of the direct memory access controller.
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.

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

1. Select direct memory access controller activating sources and set both the DE and DME bits to 1. This masks CPU interrupt sources regardless of the interrupt priority register settings.
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

7.10 Usage Note

7.10.1 Timing to Clear an Interrupt Source

The interrupt source flags should be cleared in the interrupt exception service routine. After clearing the interrupt source flag, "time from occurrence of interrupt request until interrupt controller identifies priority, compares it with mask bits in SR, and sends interrupt request signal to CPU" shown in table 7.5 is required before the interrupt source sent to the CPU is actually cancelled. To ensure that an interrupt request that should have been cleared is not inadvertently accepted again, read* the interrupt source flag after it has been cleared, and then execute an RTE instruction.

Note: * When clearing the USB 2.0 host/function module interrupt source flag, read the flag three times after clearing it.

chanhvinh.com

Section 8 Cache

8.1 Features

- Capacity
  Instruction cache: 8 Kbytes
  Operand cache: 8 Kbytes
- Structure: Instructions/data separated, 4-way set associative
- Way lock function (only for operand cache): Way 2 and way 3 are lockable
- Line size: 16 bytes
- Number of entries: 128 entries/way
- Write system: Write-back/write-through selectable
- Replacement method: Least-recently-used (LRU) algorithm

8.1.1 Cache Structure

The cache separates data and instructions and uses a 4-way set associative system. It is composed of four ways (banks), each of which is divided into an address section and a data section.

In each way, each of the address and data sections is divided into 128 entries. The data section of the entry is called a line. Each line consists of 16 bytes (4 bytes × 4). The data capacity per way is 2 Kbytes (16 bytes × 128 entries), with a total of 8 Kbytes in the cache as a whole (4 ways).

Figure 8.1 shows the operand cache structure. The instruction cache structure is the same as the operand cache structure except for not having the U bit.

chanhvinh.com

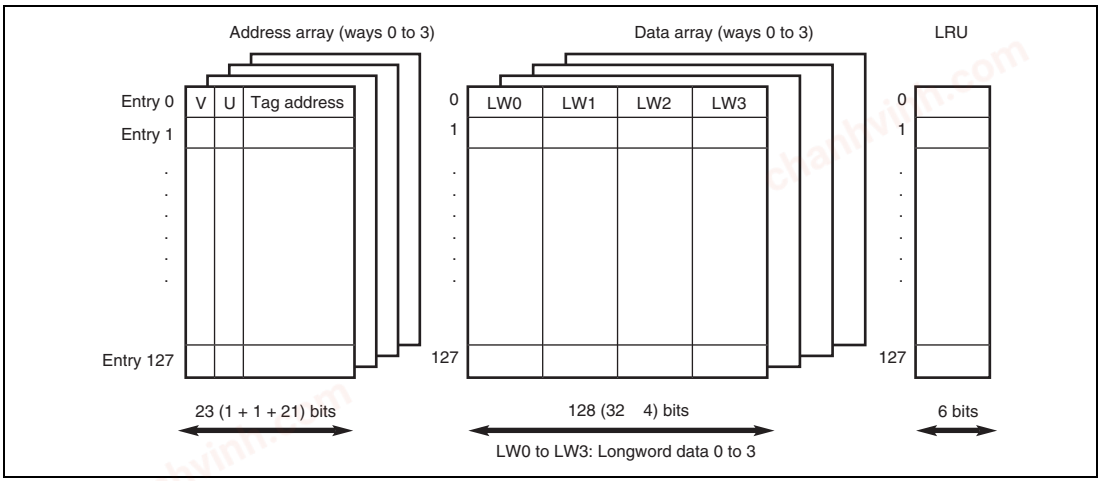


Figure 8.1 Operand Cache Structure

(1) Address Array

The V bit indicates whether the entry data is valid. When the V bit is 1, data is valid; when 0, data is not valid.

The U bit (only for operand cache) indicates whether the entry has been written to in write-back mode. When the U bit is 1, the entry has been written to; when 0, it has not.

The tag address holds the physical address used in the access to external memory or large-capacity on-chip RAM. It consists of 21 bits (address bits 31 to 11) used for comparison during cache searches. In this LSI, the addresses of the cache-enabled space are H'00000000 to H'1FFFFFFF (see section 9, Bus State Controller), and therefore the upper three bits of the tag address are cleared to 0.

The V and U bits are initialized to 0 by a power-on reset but not initialized by a manual reset or in software standby mode. The tag address is not initialized by a power-on reset or manual reset or in software standby mode.

(2) Data Array

Holds a 16-byte instruction or data. Entries are registered in the cache in line units (16 bytes).

The data array is not initialized by a power-on reset or manual reset or in software standby mode.

chanhvinh.com

(3) LRU

With the 4-way set associative system, up to four instructions or data with the same entry address can be registered in the cache. When an entry is registered, LRU shows which of the four ways it is recorded in. There are six LRU bits, controlled by hardware. A least-recently-used (LRU) algorithm is used to select the way that has been least recently accessed.

Six LRU bits indicate the way to be replaced in case of a cache miss. The relationship between LRU and way replacement is shown in table 8.1 when the cache lock function (only for operand cache) is not used (concerning the case where the cache lock function is used, see section 8.2.2, Cache Control Register 2 (CCR2)). If a bit pattern other than those listed in table 8.1 is set in the LRU bits by software, the cache will not function correctly. When modifying the LRU bits by software, set one of the patterns listed in table 8.1.

The LRU bits are initialized to B'000000 by a power-on reset but not initialized by a manual reset or in software standby mode.

Table 8.1 LRU and Way Replacement (Cache Lock Function Not Used)

LRU (Bits 5 to 0)	Way to be Replaced
000000, 000100, 010100, 100000, 110000, 110100	3
000001, 000011, 001011, 100001, 101001, 101011	2
000110, 000111, 001111, 010110, 011110, 011111	1
111000, 111001, 111011, 111100, 111110, 111111	0

chanhvinh.com

8.2 Register Descriptions

Table 8.2 shows the register configuration of the cache.

Table 8.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Cache control register 1	CCR1	R/W	H'00000000	H'FFFC1000	32
Cache control register 2	CCR2	R/W	H'00000000	H'FFFC1004	32

8.2.1 Cache Control Register 1 (CCR1)

The instruction cache is enabled or disabled using the ICE bit. The ICF bit controls disabling of all instruction cache entries. The operand cache is enabled or disabled using the OCE bit. The OCF bit controls disabling of all operand cache entries. The WT bit selects either write-through mode or write-back mode for operand cache.

Programs that change the contents of CCR1 should be placed in a cache-disabled space, and a cache-enabled space should be accessed after reading the contents of CCR1.

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
	-	-	-	-	ICF	-	-	ICE	-	-	-	-	OCF	-	WT	OCE
Initial 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	R	R/W	R	R	R	R	R/W	R	R/W	R/W

chanhvinh.com

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	ICF	0	R/W	Instruction Cache Flush Writing 1 flushes all instruction cache entries (clears the V and LRU bits of all instruction cache entries to 0). Always reads 0. Write-back to the external memory or the large-capacity on-chip RAM is not performed when the instruction cache is flushed.
10, 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	ICE	0	R/W	Instruction Cache Enable Indicates whether the instruction cache function is enabled/disabled. 0: Instruction cache disable 1: Instruction cache enable
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	OCF	0	R/W	Operand Cache Flush Writing 1 flushes all operand cache entries (clears the V, U, and LRU bits of all operand cache entries to 0). Always reads 0. Write-back to the external memory or the large-capacity on-chip RAM is not performed when the operand cache is flushed.
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	WT	0	R/W	Write Through Selects write-back mode or write-through mode. 0: Write-back mode 1: Write-through mode
0	OCE	0	R/W	Operand Cache Enable Indicates whether the operand cache function is enabled/disabled. 0: Operand cache disable 1: Operand cache enable

chanhvinh.com

8.2.2 Cache Control Register 2 (CCR2)

CCR2 is used to enable or disable the cache locking function for operand cache and is valid in cache locking mode only. In cache locking mode, the lock enable bit (the LE bit) in CCR2 is set to 1. In non-cache-locking mode, the cache locking function is invalid.

When a cache miss occurs in cache locking mode by executing the prefetch instruction (PREF @Rn), the line of data pointed to by Rn is loaded into the cache according to bits 9 and 8 (the W3LOAD and W3LOCK bits) and bits 1 and 0 (the W2LOAD and W2LOCK bits) in CCR2. The relationship between the setting of each bit and a way, to be replaced when the prefetch instruction is executed, are listed in table 8.3. On the other hand, when the prefetch instruction is executed and a cache hit occurs, new data is not fetched and the entry which is already enabled is held. For example, when the prefetch instruction is executed with W3LOAD = 1 and W3LOCK = 1 specified in cache locking mode while one-line data already exists in way 0 which is specified by Rn, a cache hit occurs and data is not fetched to way 3.

In the cache access other than the prefetch instruction in cache locking mode, ways to be replaced by bits W3LOCK and W2LOCK are restricted. The relationship between the setting of each bit in CCR2 and ways to be replaced are listed in table 8.4.

Programs that change the contents of CCR2 should be placed in a cache-disabled space, and a cache-enabled space should be accessed after reading the contents of CCR2.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	LE
Initial value:	0	0	0	0	0	0	0	0	0	0	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
	-	-	-	-	-	-	W3 LOAD*	W3 LOCK	-	-	-	-	-	-	W2 LOAD*	W2 LOCK
Initial value:	0	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/W	R/W

Note: * The W3LOAD and W2LOAD bits should not be set to 1 at the same time.

chanhvinh.com

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	LE	0	R/W	Lock Enable Controls the cache locking function. 0: Not cache locking mode 1: Cache locking mode
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	W3LOAD*	0	R/W	Way 3 Load
8	W3LOCK	0	R/W	Way 3 Lock When a cache miss occurs by the prefetch instruction while W3LOAD = 1 and W3LOCK = 1 in cache locking mode, the data is always loaded into way 3. Under any other condition, the cache miss data is loaded into the way to which LRU points.
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	W2LOAD*	0	R/W	Way 2 Load
0	W2LOCK	0	R/W	Way 2 Lock When a cache miss occurs by the prefetch instruction while W2LOAD = 1 and W2LOCK =1 in cache locking mode, the data is always loaded into way 2. Under any other condition, the cache miss data is loaded into the way to which LRU points.

Note: * The W3LOAD and W2LOAD bits should not be set to 1 at the same time.

chanhvinh.com

Table 8.3 Way to be Replaced when a Cache Miss Occurs in PREF Instruction

LE	W3LOAD*	W3LOCK	W2LOAD*	W2LOCK	Way to be Replaced
0	x	x	x	x	Decided by LRU (table 8.1)
1	x	0	x	0	Decided by LRU (table 8.1)
1	x	0	0	1	Decided by LRU (table 8.5)
1	0	1	x	0	Decided by LRU (table 8.6)
1	0	1	0	1	Decided by LRU (table 8.7)
1	0	x	1	1	Way 2
1	1	1	0	x	Way 3

[Legend]

x: Don't care

Note: * The W3LOAD and W2LOAD bits should not be set to 1 at the same time.

Table 8.4 Way to be Replaced when a Cache Miss Occurs in Other than PREF Instruction

LE	W3LOAD*	W3LOCK	W2LOAD*	W2LOCK	Way to be Replaced
0	x	x	x	x	Decided by LRU (table 8.1)
1	x	0	x	0	Decided by LRU (table 8.1)
1	x	0	x	1	Decided by LRU (table 8.5)
1	x	1	x	0	Decided by LRU (table 8.6)
1	x	1	x	1	Decided by LRU (table 8.7)

[Legend]

x: Don't care

Note: * The W3LOAD and W2LOAD bits should not be set to 1 at the same time.

Table 8.5 LRU and Way Replacement (when W2LOCK=1 and W3LOCK=0)

LRU (Bits 5 to 0)	Way to be Replaced
000000, 000001, 000100, 010100, 100000, 100001, 110000, 110100	3
000011, 000110, 000111, 001011, 001111, 010110, 011110, 011111	1
101001, 101011, 111000, 111001, 111011, 111100, 111110, 111111	0

chanhvinh.com

Table 8.6 LRU and Way Replacement (when W2LOCK=0 and W3LOCK=1)

LRU (Bits 5 to 0)	Way to be Replaced
000000, 000001, 000011, 001011, 100000, 100001, 101001, 101011	2
000100, 000110, 000111, 001111, 010100, 010110, 011110, 011111	1
110000, 110100, 111000, 111001, 111011, 111100, 111110, 111111	0

Table 8.7 LRU and Way Replacement (when W2LOCK=1 and W3LOCK=1)

LRU (Bits 5 to 0)	Way to be Replaced
000000, 000001, 000011, 000100, 000110, 000111, 001011, 001111, 010100, 010110, 011110, 011111	1
100000, 100001, 101001, 101011, 110000, 110100, 111000, 111001, 111011, 111100, 111110, 111111	0

chanhvinh.com

8.3 Operation

Operations for the operand cache are described here. Operations for the instruction cache are similar to those for the operand cache except for the address array not having the U bit, and there being no prefetch operation or write operation, or a write-back buffer.

8.3.1 Searching Cache

If the operand cache is enabled (OCE bit in CCR1 is 1), whenever data in a cache-enabled area is accessed, the cache will be searched to see if the desired data is in the cache. Figure 8.2 illustrates the method by which the cache is searched.

Entries are selected using bits 10 to 4 of the address used to access memory and the tag address of that entry is read. At this time, the upper three bits of the tag address are always cleared to 0. Bits 31 to 11 of the address used to access memory are compared with the read tag address. The address comparison uses all four ways. When the comparison shows a match and the selected entry is valid (V = 1), a cache hit occurs. When the comparison does not show a match or the selected entry is not valid (V = 0), a cache miss occurs. Figure 8.2 shows a hit on way 1.

chanhvinh.com

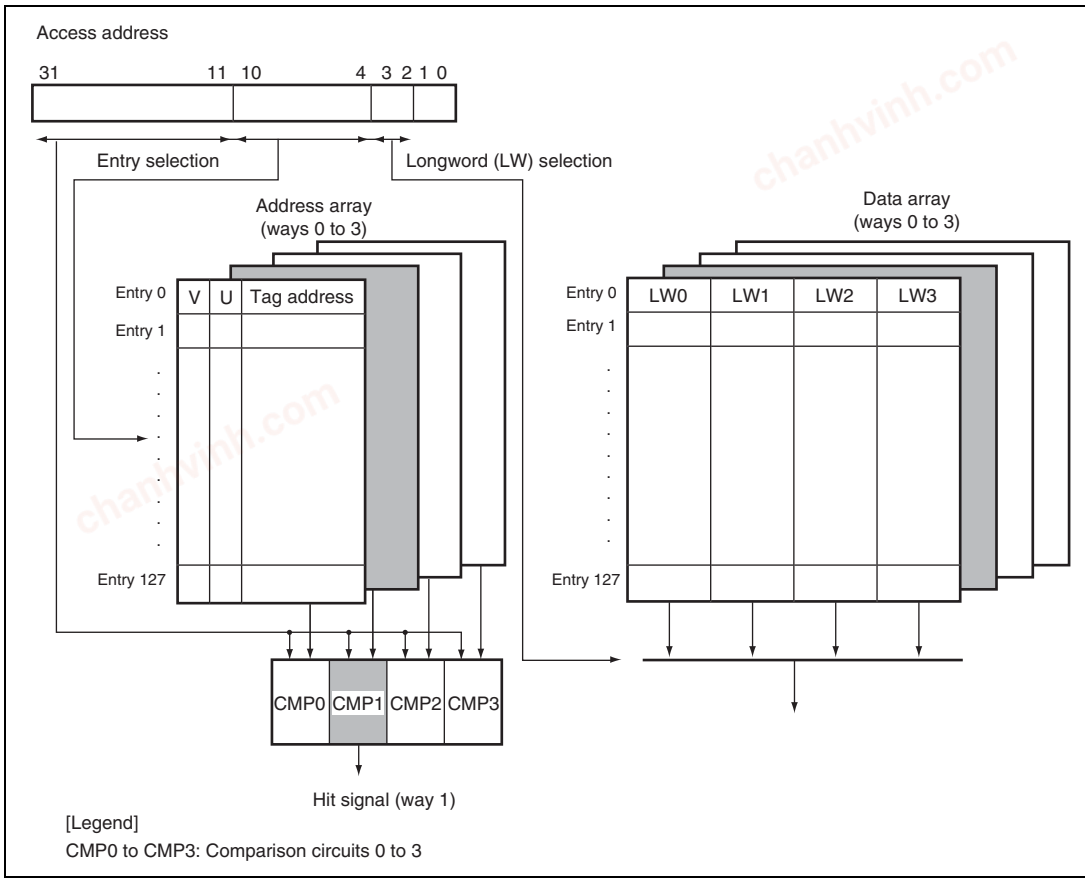


Figure 8.2 Cache Search Scheme

chanhvinh.com

8.3.2 Read Access

(1) Read Hit

In a read access, data is transferred from the cache to the CPU. LRU is updated so that the hit way is the latest.

(2) Read Miss

An internal bus cycle starts and the entry is updated. The way replaced follows table 8.4. Entries are updated in 16-byte units. When the desired data that caused the miss is loaded from the external memory or the large-capacity on-chip RAM to the cache, the data is transferred to the CPU in parallel with being loaded to the cache. When it is loaded in the cache, the V bit is set to 1, and LRU is updated so that the replaced way becomes the latest. In operand cache, the U bit is additionally cleared to 0. When the U bit of the entry to be replaced by updating the entry in write-back mode is 1, the cache update cycle starts after the entry is transferred to the write-back buffer. After the cache completes its update cycle, the write-back buffer writes the entry back to the memory. The write-back unit is 16 bytes. Cache update operation and write-back operation to the memory are performed in wrap-around mode. When the lower four bits of the address of read-miss data are H'4, for example, cache update operation and write-back operation to the memory are performed in the following order of the lower 4-bit value of address: H'4 → H'8 → H'C → H'0.

8.3.3 Prefetch Operation (Only for Operand Cache)

(1) Prefetch Hit

LRU is updated so that the hit way becomes the latest. The contents in other caches are not modified. No data is transferred to the CPU.

(2) Prefetch Miss

No data is transferred to the CPU. The way to be replaced follows table 8.3. Other operations are the same as those in the case of read miss.

chanhvinh.com

8.3.4 Write Operation (Only for Operand Cache)

(1) Write Hit

In a write access in write-back mode, the data is written to the cache and no write cycle to the external memory or the large-capacity on-chip RAM is issued. The U bit of the entry written is set to 1 and LRU is updated so that the hit way becomes the latest.

In write-through mode, the data is written to the cache and a write cycle to the external memory or the large-capacity on-chip RAM is issued. The U bit of the written entry is not updated and LRU is updated so that the replaced way becomes the latest.

(2) Write Miss

In write-back mode, an internal bus cycle starts when a write miss occurs, and the entry is updated. The way to be replaced follows table 8.4. When the U bit of the entry to be replaced is 1, the cache update cycle starts after the entry is transferred to the write-back buffer. Data is written to the cache, the U bit is set to 1, and the V bit is set to 1. LRU is updated so that the replaced way becomes the latest. After the cache completes its update cycle, the write-back buffer writes the entry back to the memory. The write-back unit is 16 bytes. Cache update operation and write-back operation to the memory are performed in wrap-around mode. When the lower four bits of the address of write-miss data are H'4, for example, cache update operation and write-back operation to the memory are performed in the following order of the lower 4-bit value of address: H'4 → H'8 → H'C → H'0.

In write-through mode, no write to cache occurs in a write miss; the write is only to the external memory or the large-capacity on-chip RAM.

8.3.5 Write-Back Buffer (Only for Operand Cache)

When the U bit of the entry to be replaced in the write-back mode is 1, it must be written back to the external memory or the large-capacity on-chip RAM. To increase performance, the entry to be replaced is first transferred to the write-back buffer and fetching of new entries to the cache takes priority over writing back to the external memory. After the cache completes to fetch the new entry, the write-back buffer writes the entry back to the external memory or the large-capacity on-chip RAM. During the write-back cycles, the cache can be accessed. The write-back buffer can hold one line of cache data (16 bytes) and its physical address. Figure 8.3 shows the configuration of the write-back buffer.

chanhvinh.com

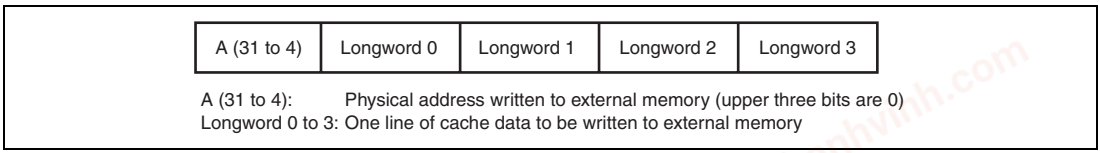


Figure 8.3 Write-Back Buffer Configuration

Operations in sections 8.3.2 to 8.3.5 are summarized in table 8.8.

Table 8.8 Cache Operations

Cache	CPU Cycle	Hit/ Miss	Write-Back Mode/ Write-Through Mode	U Bit	Access to External Memory or Large-Capacity On-Chip RAM (Through Internal Bus)	Cache Contents
Instruction cache	Instruction fetch	Hit	—	—	Not generated	Not updated
		Miss	—	—	Cache update cycle is generated	Updated to new values by cache update cycle
Operand cache	Prefetch/ read	Hit	Either mode is available	x	Not generated	Not updated
		Miss	Write-through mode	—	Cache update cycle is generated	Updated to new values by cache update cycle
			Write-back mode	0	Cache update cycle is generated	Updated to new values by cache update cycle
				1	Cache update cycle is generated. Then write-back cycle in write-back buffer is generated.	Updated to new values by cache update cycle
	Write	Hit	Write-through mode	—	Write cycle CPU issues is generated.	Updated to new values by write cycle the CPU issues
			Write-back mode	x	Not generated	Updated to new values by write cycle the CPU issues
		Miss	Write-through mode	—	Write cycle CPU issues is generated.	Not updated*
			Write-back mode	0	Cache update cycle is generated	Updated to new values by cache update cycle. Subsequently updated again to new values in write cycle CPU issues.
				1	Cache update cycle is generated. Then write-back cycle in write-back buffer is generated.	Updated to new values by cache update cycle. Subsequently updated again to new values in write cycle CPU issues.

chanhvinh.com

[Legend]

x: Don't care.

Note: Cache update cycle: 16-byte read access
Write-back cycle in write-back buffer: 16-byte write access
* Neither LRU updated. LRU is updated in all other cases.

8.3.6 Coherency of Cache and External Memory or Large-Capacity On-Chip RAM

Use software to ensure coherency between the cache and the external memory or the large-capacity on-chip RAM. When memory shared by this LSI and another device is mapped in the cache-enabled space, operate the memory-mapped cache to invalidate and write back as required. The same operation should be performed for the memory shared by the CPU and the direct memory access controller in this LSI.

chanhvinh.com

8.4 Memory-Mapped Cache

To allow software management of the cache, cache contents can be read and written by means of MOV instructions. The instruction cache address array is mapped onto addresses H'F0000000 to H'F07FFFFF, and the data array onto addresses H'F1000000 to H'F17FFFFF. The operand cache address array is mapped onto addresses H'F0800000 to H'F0FFFFFF, and the data array onto addresses H'F1800000 to H'F1FFFFFF. Only longword can be used as the access size for the address array and data array, and instruction fetches cannot be performed.

8.4.1 Address Array

To access an address array, the 32-bit address field (for read/write accesses) and 32-bit data field (for write accesses) must be specified.

In the address field, specify the entry address for selecting the entry, the W bit for selecting the way, and the A bit for specifying the existence of associative operation. In the W bit, B'00 is way 0, B'01 is way 1, B'10 is way 2, and B'11 is way 3. Since the access size of the address array is fixed at longword, specify B'00 for bits 1 and 0 of the address.

The tag address, LRU bits, U bit (only for operand cache), and V bit are specified as data. Always specify 0 for the upper three bits (bits 31 to 29) of the tag address.

For the address and data formats, see figure 8.4.

The following three operations are possible for the address array.

(1) Address Array Read

The tag address, LRU bits, U bit (only for operand cache), and V bit are read from the entry address specified by the address and the entry corresponding to the way. For the read operation, associative operation is not performed regardless of whether the associative bit (A bit) specified by the address is 1 or 0.

(2) Address-Array Write (Non-Associative Operation)

When the associative bit (A bit) in the address field is cleared to 0, write the tag address, LRU bits, U bit (only for operand cache), and V bit, specified by the data field, to the entry address specified by the address and the entry corresponding to the way. When writing to a cache line for which the U bit = 1 and the V bit =1 in the operand cache address array, write the contents of the cache line back to memory, then write the tag address, LRU bits, U bit, and V bit specified by the data field. When 0 is written to the V bit, 0 must also be written to the U bit of that entry. Write-

chanhvinh.com

back operation to the memory is performed in the following order of the lower 4-bit value of address: H'0 → H'4 → H'8 → H'C.

(3) Address-Array Write (Associative Operation)

When writing with the associative bit (A bit) of the address field set to 1, the addresses in the four ways for the entry specified by the address field are compared with the tag address that is specified by the data field. Write the U bit (only for operand cache) and the V bit specified by the data field to the entry of the way that has a hit. However, the tag address and LRU bits remain unchanged. When there is no way that has a hit, nothing is written and there is no operation. This function is used to invalidate a specific entry in the cache.

When the U bit of the entry that has had a hit is 1 in the operand cache, writing back should be performed. However, when 0 is written to the V bit, 0 must also be written to the U bit of that entry. Write-back operation to the memory is performed in the following order of the lower 4-bit value of address: H'0 → H'4 → H'8 → H'C.

8.4.2 Data Array

To access a data array, the 32-bit address field (for read/write accesses) and 32-bit data field (for write accesses) must be specified. The address field specifies information for selecting the entry to be accessed; the data field specifies the longword data to be written to the data array.

Specify the entry address for selecting the entry, the L bit indicating the longword position within the (16-byte) line, and the W bit for selecting the way. In the L bit, B'00 is longword 0, B'01 is longword 1, B'10 is longword 2, and B'11 is longword 3. In the W bit, B'00 is way 0, B'01 is way 1, B'10 is way 2, and B'11 is way 3. Since the access size of the data array is fixed at longword, specify B'00 for bits 1 and 0 of the address.

For the address and data formats, see figure 8.4.

The following two operations are possible for the data array. Information in the address array is not modified by this operation.

(1) Data Array Read

The data specified by the L bit in the address is read from the entry address specified by the address and the entry corresponding to the way.

chanhvinh.com

(2) Data Array Write

The longword data specified by the data is written to the position specified by the L bit in the address from the entry address specified by the address and the entry corresponding to the way.

1. Instruction cache

1.1 Address array access

(a) Address specification

Read access

31 to 23	22 to 13	12	11	10 to 4	3	2	1	0
111100000	*---------*	W		Entry address	0	*	0	0

Write access

31 to 23	22 to 13	12	11	10 to 4	3	2	1	0
111100000	*---------*	W		Entry address	A	*	0	0

(b) Data specification (both read and write accesses)

31 to 29	28 to 11	10	9 to 4	3	2	1	0
0 0 0	Tag address (28 to 11)	E	LRU	X	X	X	V

1.2 Data array access (both read and write accesses)

(a) Address specification

31 to 23	22 to 13	12	11	10 to 4	3	2	1	0
111100010	*---------*	W		Entry address		L	0	0

(b) Data specification

31 to 0
Longword data

2. Operand cache

2.1 Address array access

(a) Address specification

Read access

31 to 23	22 to 13	12	11	10 to 4	3	2	1	0
111100001	*---------*	W		Entry address	0	*	0	0

Write access

31 to 23	22 to 13	12	11	10 to 4	3	2	1	0
111100001	*---------*	W		Entry address	A	*	0	0

(b) Data specification (both read and write accesses)

31 to 29	28 to 11	10	9 to 4	3	2	1	0
0 0 0	Tag address (28 to 11)	E	LRU	X	X	U	V

2.2 Data array access (both read and write accesses)

(a) Address specification

31 to 23	22 to 13	12	11	10 to 4	3	2	1	0
111100011	*---------*	W		Entry address		L	0	0

(b) Data specification

31 to 0
Longword data

[Legend]
*: Don't care
E: Bit 10 of entry address for read, don't care for write
X: 0 for read, don't care for write

Figure 8.4 Specifying Address and Data for Memory-Mapped Cache Access

chanhvinh.com

8.4.3 Usage Examples

(1) Invalidating Specific Entries

Specific cache entries can be invalidated by writing 0 to the entry's V bit in the memory mapping cache access. When the A bit is 1, the tag address specified by the write data is compared to the tag address within the cache selected by the entry address, and data is written to the bits V and U specified by the write data when a match is found. If no match is found, there is no operation. When the V bit of an entry in the address array is set to 0, the entry is written back if the entry's U bit is 1.

An example when a write data is specified in R0 and an address is specified in R1 is shown below.

```
; R0=H'0110 0010; tag address(28-11)=B'0 0001 0001 0000 0000 0, U=0, V=0
; R1=H'F080 0088; operand cache address array access, entry=B'000 1000, A=1
;
 MOV.L  R0,@R1
```

(2) Reading the Data of a Specific Entry

The data section of a specific cache entry can be read by the memory mapping cache access. The longword indicated in the data field of the data array in figure 8.4 is read into the register.

An example when an address is specified in R0 and data is read in R1 is shown below.

```
; R0=H'F100 004C; instruction cache data array access, entry=B'000 0100,
; Way=0, longword address=3
;
 MOV.L  @R0,R1
```

chanhvinh.com

8.4.4 Usage Notes

1. Programs that access memory-mapped cache of the operand cache should be placed in a cache-disabled space. Programs that access memory-mapped cache of the instruction cache should be placed in a cache-disabled space, and in each of the beginning and the end of that, two or more read accesses to on-chip peripheral modules or external address space (cache-disabled address) should be executed.
2. Rewriting the address array contents so that two or more ways are hit simultaneously is prohibited. Operation is not guaranteed if the address array contents are changed so that two or more ways are hit simultaneously.
3. Registers and memory-mapped cache can be accessed only by the CPU and not by the direct memory access controller.

chanhvinh.com

Section 9 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.

9.1 Features

1. External address space
   - A maximum of 64 Mbytes for each of areas CS0 to CS6.
   - Can specify the normal space interface, SRAM interface with byte selection, burst ROM (clocked synchronous or asynchronous), MPX-I/O, SDRAM memory type, and PCMCIA interface for each address space.
   - Data bus width for CS0 space is 16 bits. Can select the data bus width (8 or 16 bits) for each of address spaces CS1 to CS6.
   - 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 low-frequency and power-down modes.
   - Issues MRS and EMRS commands.

chanhvinh.com

6. PCMCIA direct interface
   - Supports the IC memory card and I/O card interface defined in JEIDA specifications Ver. 4.2 (PCMCIA2.1 Rev. 2.1).
   - Wait-cycle insertion controllable by program.
7. SRAM interface with byte selection
   - Can connect directly to a SRAM with byte selection.
8. Burst ROM interface (clocked synchronous)
   - Can connect directly to a burst ROM of the clocked synchronous type.
9. Bus arbitration
   - Shares all of the resources with other CPU and outputs the bus enable after receiving the bus request from external devices.
10. 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).
11. Usage as interval timer for refresh counter
    - Generates an interrupt request at compare match.

chanhvinh.com

Figure 9.1 shows a block diagram of this module.

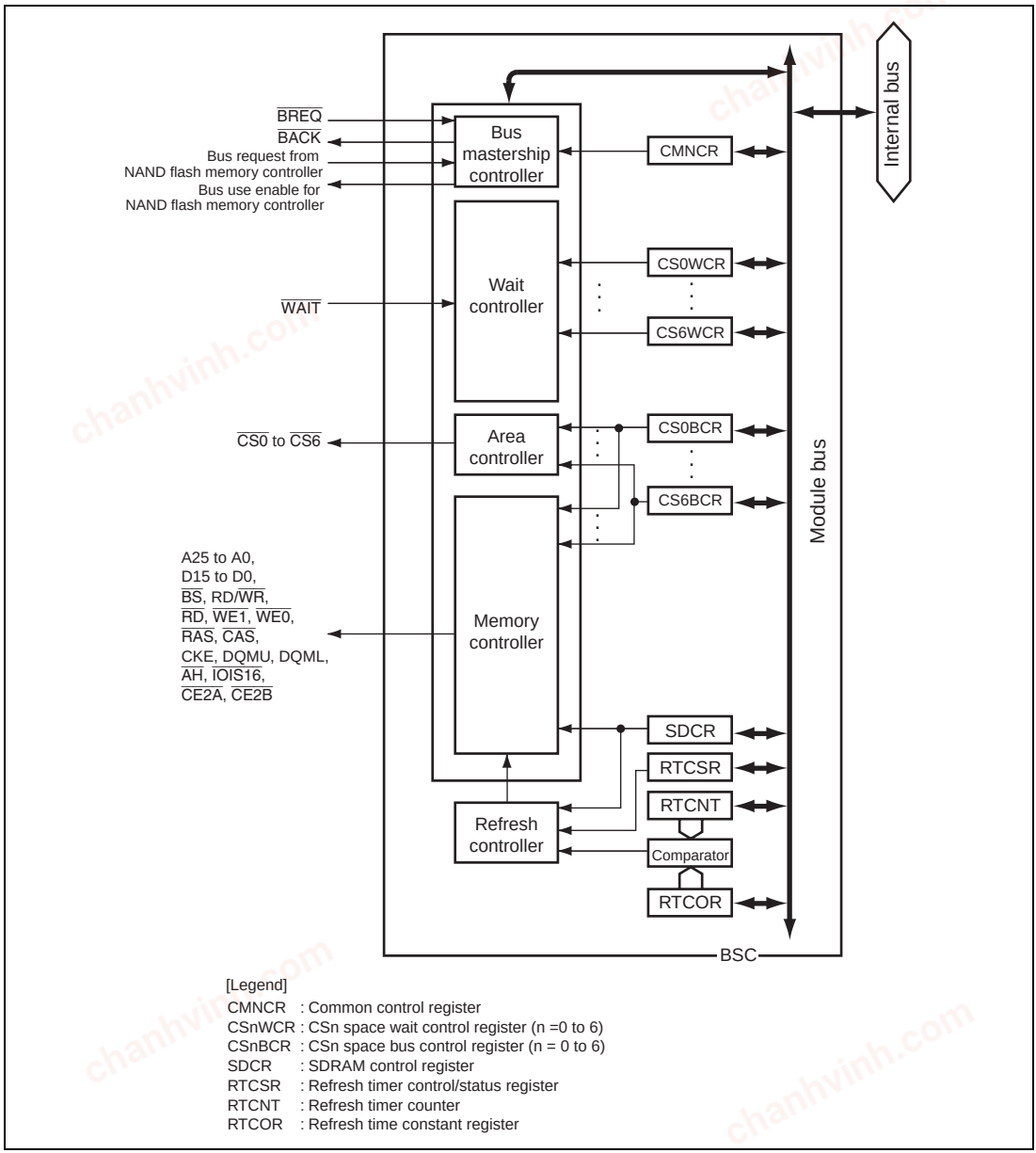


Figure 9.1 Block Diagram of Bus State Controller

chanhvinh.com

9.2 Input/Output Pins

Table 9.1 shows the pin configuration.

Table 9.1 Pin Configuration

Name	I/O	Function
A25 to A0	Output	Address bus
D15 to D0	I/O	Data bus
$\overline{BS}$	Output	Bus cycle start
$\overline{CS0}$ to $\overline{CS4}$	Output	Chip select
$\overline{CS5}$/$\overline{CE1A}$, $\overline{CS6}$/$\overline{CE1B}$	Output	Chip select Function as PCMCIA card select signals for D7 to D0 when PCMCIA is used.
$\overline{CE2A}$, $\overline{CE2B}$	Output	Function as PCMCIA card select signals for D15 to D8.
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) Functions as a strobe signal for indicating memory read cycles when PCMCIA is used.
$\overline{ICIOWR}$/$\overline{AH}$	Output	Functions as a strobe signal for indicating I/O write cycles when PCMCIA is used. Functions as the address hold signal when the MPX-I/O is used.
$\overline{ICIORD}$	Output	Functions as a strobe signal for indicating I/O read cycles when PCMCIA is used.
$\overline{WE1}$/DQMU/$\overline{WE}$	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. Functions as a strobe signal for indicating memory write cycles when PCMCIA is used.

chanhvinh.com

Name	I/O	Function
$\overline{WE0}$/DQML	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
$\overline{BREQ}$	Input	Bus request input
$\overline{BACK}$	Output	Bus enable output
$\overline{IOIS16}$	Input	Indicates 16-bit I/O of PCMIA. Enabled only in little endian mode. The pin should be driven low in big endian mode.

chanhvinh.com

9.3 Area Overview

9.3.1 Address Map

In the architecture, this LSI has a 32-bit address space, which is divided into cache-enabled, cache-disabled, and on-chip spaces (on-chip RAM, on-chip peripheral modules, and reserved areas) according to the upper bits of the address.

External address spaces CS0 to CS6 are cache-enabled when internal address A29 = 0 or cache-disabled when A29 = 1.

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 9.2 Address Map

Internal Address	Space	Memory to be Connected	Cache
H'00000000 to H'03FFFFFF	CS0	Normal space, SRAM with byte selection, burst ROM (asynchronous or synchronous)	Cache-enabled
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, PCMCIA	
H'18000000 to H'1BFFFFFF	CS6	Normal space, SRAM with byte selection, PCMCIA	
H'1C000000 to H'1FFFFFFF	Other	On-chip RAM	
H'20000000 to H'23FFFFFF	CS0	Normal space, SRAM with byte selection, burst ROM (asynchronous or synchronous)	Cache-disabled
H'24000000 to H'27FFFFFF	CS1	Normal space, SRAM with byte selection	
H'28000000 to H'2BFFFFFF	CS2	Normal space, SRAM with byte selection, SDRAM	
H'2C000000 to H'2FFFFFFF	CS3	Normal space, SRAM with byte selection, SDRAM	
H'30000000 to H'33FFFFFF	CS4	Normal space, SRAM with byte selection, burst ROM (asynchronous)	
H'34000000 to H'37FFFFFF	CS5	Normal space, SRAM with byte selection, MPX-I/O, PCMCIA	

chanhvinh.com

Internal Address	Space	Memory to be Connected	Cache
H'38000000 to H'3BFFFFFF	CS6	Normal space, SRAM with byte selection, PCMCIA	Cache-disabled
H'3C000000 to H'3FFFFFFF	Other	On-chip RAM	
H'40000000 to H'FFFBFFFF	Other	On-chip RAM, reserved area*	—
H'FFFC0000 to H'FFFFFFFF	Other	On-chip peripheral modules, reserved area*	—

Note: * For the on-chip RAM space, access the addresses shown in section 31, On-Chip RAM. For the on-chip peripheral module space, access the addresses shown in section 36, List of Registers. Do not access addresses which are not described in these sections. Otherwise, the correct operation cannot be guaranteed.

9.3.2 Data Bus Width and Endian Specification of Each Area and Related Pin Settings Depending on Boot Mode

The initial state of data bus, endian specification, and settings of the pins related to this module depends on boot mode. For boot mode, refer to section 4, Boot Mode.

In boot mode 0, the state of area 0 is fixed to the state with bus width of 16 bits and big endian, because this LSI is started up by the program stored in the ROM connected to area 0. The initial state of areas 1 to 6 is the same as that of area 0, but the bus width and endian specification can be changed by a program.

In this mode, settings required to read the ROM in area 0, such as settings for some of the addresses, the data bus, and the pin functions of $\overline{CS0}$ and $\overline{RD}$ only, are selected automatically immediately after a power-on reset, but the initial function settings of the other pins are as general ports. These pins cannot be used until their functions are set by a program. Do not perform other than read access to area 0 until the pin settings are completed.

In boot modes 1 to 3, the state of areas 0 to 6 can be changed from the initial state by the program, because in these modes the LSI is started by the program stored in the NAND flash memory or the serial flash memory. Since pin functions related to this module are not set automatically, they need to be set by the user. Do not access the external address space until the pin settings are completed.

Table 9.3 shows the initial state of areas 0 to 6 in boot mode 0, 2, and 3.

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

For details on pin function settings, see section 32, General Purpose I/O Ports.

chanhvinh.com

Table 9.3 Initial States of Areas in Boot Modes 0, 2, and 3

Boot Mode	Item	Area 0	Areas 1 to 6
0	Data bus width	Fixed to 16 bits. Not changeable.	The initial state is 16 bits. Can be changed by program.
	Endian specification	Fixed to big endian. Not changeable.	The initial state is big endian. Can be changed by program.
	Settings of pins related to this module	Pin functions for A20 to A1, D15 to D0, $\overline{CS0}$, and $\overline{RD}$ are set automatically. Other pins need to be set by program.	
1 to 3	Data bus width	The initial state is 16 bits. Can be changed by program.	
	Endian specification	The initial state is big endian. 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.	

Notes:
1. In order to connect boot ROM using address lines A21 and above in boot mode 0, pull-down processing must be provided on the board for address lines A21 and above.
2. Depending on the type of memory used, there may be limitations on the data bus width. For details, see 9.4.2, CSn Space Bus Control Register.
3. Since the $\overline{CS1}$ and A0 functions are assigned to the same pin, it is not possible to use area 1 and a device with an 8-bit bus connection at the same time.
4. Since the $\overline{CS4}$ and A22 functions are assigned to the same pin, it is not possible to use area 4 and a device using address lines A22 and above at the same time.

chanhvinh.com

9.4 Register Descriptions

Table 9.4 shows the register configuration of this module.

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

Table 9.4 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Common control register	CMNCR	R/W	H'00001010	H'FFFC0000	32
CS0 space bus control register	CS0BCR	R/W	H'36DB0400	H'FFFC0004	32
CS1 space bus control register	CS1BCR	R/W	H'36DB0400	H'FFFC0008	32
CS2 space bus control register	CS2BCR	R/W	H'36DB0400	H'FFFC000C	32
CS3 space bus control register	CS3BCR	R/W	H'36DB0400	H'FFFC0010	32
CS4 space bus control register	CS4BCR	R/W	H'36DB0400	H'FFFC0014	32
CS5 space bus control register	CS5BCR	R/W	H'36DB0400	H'FFFC0018	32
CS6 space bus control register	CS6BCR	R/W	H'36DB0400	H'FFFC001C	32
CS0 space wait control register	CS0WCR	R/W	H'00000500	H'FFFC0028	32
CS1 space wait control register	CS1WCR	R/W	H'00000500	H'FFFC002C	32
CS2 space wait control register	CS2WCR	R/W	H'00000500	H'FFFC0030	32
CS3 space wait control register	CS3WCR	R/W	H'00000500	H'FFFC0034	32
CS4 space wait control register	CS4WCR	R/W	H'00000500	H'FFFC0038	32
CS5 space wait control register	CS5WCR	R/W	H'00000500	H'FFFC003C	32
CS6 space wait control register	CS6WCR	R/W	H'00000500	H'FFFC0040	32

chanhvinh.com

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
SDRAM control register	SDCR	R/W	H'00000000	H'FFFC004C	32
Refresh timer control/status register	RTCSR	R/W	H'00000000	H'FFFC0050	32
Refresh timer counter	RTCNT	R/W	H'00000000	H'FFFC0054	32
Refresh time constant register	RTCOR	R/W	H'00000000	H'FFFC0058	32
AC characteristics switching register	ACSWR	R/W*[1]	H'00000000	H'FFFC180C	32
AC characteristics switching key register	ACKEYR	W*[2]	—	H'FFFC1BFC	8

Notes: 1. A special sequence using the AC characteristics switching key register is necessary for writing to this register.
2. Write-only register. The write value is arbitrary.

chanhvinh.com

9.4.1 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
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	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
	-	-	-	-	BLOCK	DPRTY[1:0]		DMAIW[2:0]			DMA IWA	-	-	-	HIZ MEM	HIZ CNT*
Initial value:	0	0	0	1	0	0	0	0	0	0	0	1	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	R	R	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	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
11	BLOCK	0	R/W	Bus Lock Specifies whether or not the $\overline{\text{BREQ}}$ signal is received. 0: Receives $\overline{\text{BREQ}}$. 1: Does not receive $\overline{\text{BREQ}}$.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
10, 9	DPRTY[1:0]	00	R/W	DMA Burst Transfer Priority Specify the priority for a refresh request/bus mastership request during DMA burst transfer. 00: Accepts a refresh request and bus mastership request during DMA burst transfer. 01: Accepts a refresh request but does not accept a bus mastership request during DMA burst transfer. 10: Accepts neither a refresh request nor a bus mastership request during DMA burst transfer. 11: Reserved (setting prohibited)
8 to 6	DMAIW[2:0]	000	R/W	Wait states between access cycles when DMA single address transfer is performed. Specify the number of idle cycles to be inserted after an access to an external device with DACK when DMA single address transfer is performed. The method of inserting idle cycles depends on the contents of DMAIWA. 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
5	DMAIWA	0	R/W	Method of inserting wait states between access cycles when DMA single address transfer is performed. Specifies the method of inserting the idle cycles specified by the DMAIW[2:0] bit. Clearing this bit will make this LSI insert the idle cycles when another device, which includes this LSI, drives the data bus after an external device with DACK drove it. However, when the external device with DACK drives the data bus continuously, idle cycles are not inserted. Setting this bit will make this LSI insert the idle cycles after an access to an external device with DACK, even when the continuous access cycles to an external device with DACK are performed. 0: Idle cycles inserted when another device drives the data bus after an external device with DACK drove it. 1: Idle cycles always inserted after an access to an external device with DACK
4	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
3, 2	—	All 0	R	Reserved These bits are 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, $\overline{BS}$, $\overline{CSn}$, $\overline{CS2x}$, RD/$\overline{WR}$, $\overline{WEn}$/DQMx/$\overline{AH}$, and $\overline{RD}$. At bus-released state, these pin are high-impedance states regardless of the setting value of the HIZMEM bit. 0: High impedance in software standby mode or deep standby mode. 1: Driven in software standby mode or deep standby mode

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
0	HIZCNT*	0	R/W	High-Z Control Specifies the state in software standby mode, deep standby mode, or bus-released state for CKE, $\overline{RAS}$, and $\overline{CAS}$. 0: High impedance in software standby mode, deep standby mode, or bus-released state for CKE, $\overline{RAS}$, and $\overline{CAS}$. 1: Driven in software standby mode, deep standby mode, or bus-released state for CKE, $\overline{RAS}$, and $\overline{CAS}$.

Note: * For High-Z control of CKIO, see section 5, Clock Pulse Generator.

9.4.2 CSn Space Bus Control Register (CSnBCR) (n = 0 to 6)

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 9.5.11, 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]			ENDIAN	BSZ[1:0]		-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	1	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	R	R	R	R	R	R	R

chanhvinh.com

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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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
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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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: PCMCIA 110: Reserved (setting prohibited) 111: Burst ROM (clock synchronous) For details for memory type in each area, see table 9.2. 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 mode 2 and 3, memory access should be performed after setting CS0BCR and CS0WCR.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
11	ENDIAN	0	R/W	Endian Setting Specifies the arrangement of data in a space. 0: Arranged in big endian 1: Arranged in little endian Note: Little endian cannot be set for area 0 in boot mode 0. In this case, this bit of CS0BCR is always read as 0. The write value should always be 0.
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: Bus width selected by address when the MPX-I/O is used. When the MPX-I/O is not used, setting prohibited. 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 mode 0, the BSZ[1:0] bits settings in CS0BCR are ignored. 3. If area 5 or area 6 is specified as PCMCIA space, the bus width can be specified as either 8 bits or 16 bits. 4. If area 2 or area 3 is specified as SDRAM space, the bus width can be specified as 16 bits. 5. If area 0 is specified as clocked synchronous burst ROM space, the bus width can be specified as 16 bits.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
8 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

9.4.3 CSn Space Wait Control Register (CSnWCR) (n = 0 to 6)

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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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/write timing and asserts the RD/$\overline{WR}$ signal during the write access cycle. 1: Asserts the $\overline{WEn}$ signal during the read/write access cycle and asserts the RD/ $\overline{WR}$ 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, $\overline{CS0}$ Assertion to $\overline{RD}$, $\overline{WEn}$ Assertion Specify the number of delay cycles from address and $\overline{CS0}$ 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	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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
1, 0	HW[1:0]	00	R/W	Delay Cycles from RD, $\overline{WEn}$ Negation to Address, $\overline{CS0}$ Negation Specify the number of delay cycles from 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: * 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.

- 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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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/write timing and asserts the RD/$\overline{WR}$ signal during the write access cycle. 1: Asserts the $\overline{WEn}$ signal during the read/write 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, $\overline{CSn}$ Assertion to $\overline{RD}$, $\overline{WEn}$ Assertion Specify the number of delay cycles from address and $\overline{CSn}$ 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	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 RD, $\overline{\text{WE}}$n Negation to Address, $\overline{\text{CS}}$n Negation Specify the number of delay cycles from RD and $\overline{\text{WE}}$n negation to address and $\overline{\text{CS}}$n 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 $\overline{WEn}$ 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 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 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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 RD, $\overline{\text{WEn}}$ Negation to Address, $\overline{\text{CS4}}$ Negation Specify the number of delay cycles from RD and $\overline{\text{WEn}}$ negation to address and $\overline{\text{CS4}}$ 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

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
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.
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 $\overline{WE}$ 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 RD, $\overline{\text{WEn}}$ Negation to Address, $\overline{\text{CS5}}$ Negation Specify the number of delay cycles from RD and $\overline{\text{WEn}}$ negation to address and $\overline{\text{CS5}}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

chanhvinh.com

- CS6WCR

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
	-	-	-	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/write access cycle and asserts the RD/$\overline{WR}$ signal at the write timing.
19 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, $\overline{CS6}$ Assertion to $\overline{RD}$, $\overline{WEn}$ Assertion Specify the number of delay cycles from address, $\overline{CS6}$ 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	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	WN	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification of this bit is valid even when the number of access wait cycles is 0. 0: The external wait input is valid 1: The 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	Number of Delay Cycles from $\overline{RD}$, $\overline{WEn}$ Negation to Address, $\overline{CS6}$ Negation Specify the number of delay cycles from $\overline{RD}$, $\overline{WEn}$ negation to address, and $\overline{CS6}$ negation. 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 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.

Bus Width	BST[1:0]	Burst count
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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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)

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.

- 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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
21, 20	BST[1:0]	00	R/W	Burst Count Specification Specify the burst count for 16-byte access. These bits must not be set to B'11, because B'11 setting is reserved.

Bus Width	BST[1:0]	Burst count
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

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 $\overline{RD}$, $\overline{WE}$ 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	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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 WRITE 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 WRITA 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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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.

(4) PCMCIA

- CS5WCR, CS6WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	SA[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	R

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

chanhvinh.com

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	SA[1:0]	00	R/W	Space Attribute Specification Select memory card interface or I/O card interface when PCMCIA interface is selected. SA1: 0: Selects memory card interface for the space for A25 = 1. 1: Selects I/O card interface for the space for A25 = 1. SA0: 0: Selects memory card interface for the space for A25 = 0. 1: Selects I/O card interface for the space for A25 = 0.
19 to 15	—	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
14 to 11	TED[3:0]	0000	R/W	Number of Delay Cycles from Address Output to $\overline{RD}$/$\overline{WE}$ Assertion Specify the number of delay cycles from address output to $\overline{RD}$/$\overline{WE}$ assertion for the memory card or to $\overline{ICIORD}$/$\overline{ICIOWR}$ assertion for the I/O card in PCMCIA interface. 0000: 0.5 cycle 0001: 1.5 cycles 0010: 2.5 cycles 0011: 3.5 cycles 0100: 4.5 cycles 0101: 5.5 cycles 0110: 6.5 cycles 0111: 7.5 cycles 1000: 8.5 cycles 1001: 9.5 cycles 1010: 10.5 cycles 1011: 11.5 cycles 1100: 12.5 cycles 1101: 13.5 cycles 1110: 14.5 cycles 1111: 15.5 cycles

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
10 to 7	PCW[3:0]	1010	R/W	Number of Access Wait Cycles Specify the number of wait cycles to be inserted. 0000: 3 cycles 0001: 6 cycles 0010: 9 cycles 0011: 12 cycles 0100: 15 cycles 0101: 18 cycles 0110: 22 cycles 0111: 26 cycles 1000: 30 cycles 1001: 33 cycles 1010: 36 cycles 1011: 38 cycles 1100: 52 cycles 1101: 60 cycles 1110: 64 cycles 1111: 80 cycles
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 cycles is 0. 0: External wait input is valid 1: External wait input is ignored
5, 4	—	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
3 to 0	TEH[3:0]	0000	R/W	Delay Cycles from $\overline{RD}$/$\overline{WE}$ Negation to Address Specify the number of address hold cycles from $\overline{RD}$/$\overline{WE}$ negation for the memory card or those from $\overline{ICIORD}$/$\overline{ICIOWR}$ negation for the I/O card in PCMCIA interface. 0000: 0.5 cycle 0001: 1.5 cycles 0010: 2.5 cycles 0011: 3.5 cycles 0100: 4.5 cycles 0101: 5.5 cycles 0110: 6.5 cycles 0111: 7.5 cycles 1000: 8.5 cycles 1001: 9.5 cycles 1010: 10.5 cycles 1011: 11.5 cycles 1100: 12.5 cycles 1101: 13.5 cycles 1110: 14.5 cycles 1111: 15.5 cycles

chanhvinh.com

(5) 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.

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 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

9.4.4 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	SLOW	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/W	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)

chanhvinh.com

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	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	SLOW	0	R/W	Low-Frequency Mode Specifies the output timing of command, address, and write data for SDRAM and the latch timing of read data from SDRAM. Setting this bit makes the hold time for command, address, write and read data extended for half cycle (output or read at the falling edge of CKIO). This mode is suitable for SDRAM with low-frequency clock. 0: Command, address, and write data for SDRAM is output at the rising edge of CKIO. Read data from SDRAM is latched at the rising edge of CKIO. 1: Command, address, and write data for SDRAM is output at the falling edge of CKIO. Read data from SDRAM is latched at the falling edge of CKIO.
11	RFSH	0	R/W	Refresh Control Specifies whether or not the refresh operation of the SDRAM is performed. 0: No refresh 1: Refresh

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
10	RMODE	0	R/W	Refresh Control 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
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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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

9.4.5 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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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: Bϕ/4 010: Bϕ/16 011: Bϕ/64 100: Bϕ/256 101: Bϕ/1024 110: Bϕ/2048 111: Bϕ/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

9.4.6 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

chanhvinh.com

9.4.7 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

9.4.8 AC Characteristics Switching Register (ACSWR)

To use the SDRAM, set the AC characteristics switching register (ACSWR) and AC characteristics key switching register (ACKEYR).

Only a special sequence can write to this register to prevent accidental erroneous write. The setting procedure is shown in section 9.4.10, Sequence to Write to ACSWR. Read is done by the normal longword.

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

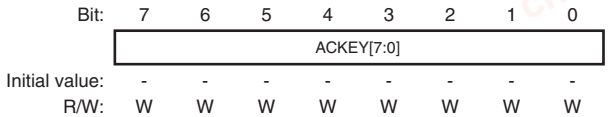
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	ACOSW[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

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	ACOSW[3:0]	0000	R/W	AC Characteristics Switch Specifies AC characteristics switching 0000: Not extend the delay time 0010: Switches characteristics and extends the delay time Others: Setting prohibited

chanhvinh.com

9.4.9 AC Characteristics Switching Key Register (ACKEYR)

ACKEYR is an 8-bit write-only register to access the AC characteristics switching register (ACSWR). The write value is ignored and the read value is undefined.



Bit	Bit Name	Initial Value	R/W	Description
7 to 0	ACKEY[7:0]	—	W	AC Key Writing to these bits is required to write to ACSWR. The write value is arbitrary.

chanhvinh.com

9.4.10 Sequence to Write to ACSWR

Figure 9.2 shows the sequence to write to ACSWR. Write must be executed in the high-speed on-chip RAM.

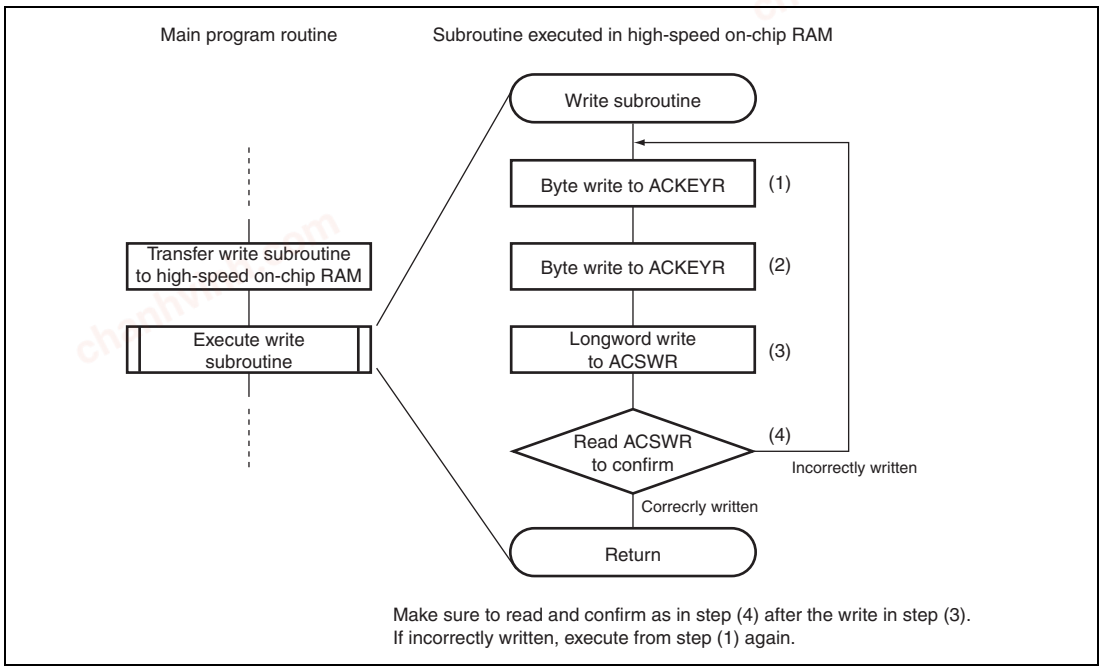


Figure 9.2 Recommended Sequence to Write to ACSWR

chanhvinh.com

9.5 Operation

9.5.1 Endian/Access Size and Data Alignment

This LSI supports both big endian, in which the most significant byte (MSB) of data is that in the direction of the 0th address, and little endian, in which the least significant byte (LSB) is that in the direction of the 0th address. In the initial state after a power-on reset, all areas will be in big endian mode. Endian mode can be changed by setting the CSnBCR register as long as the target space is not being accessed.

Data bus width can be selected from 8 bits and 16 bits for the normal memory and SRAM with byte selection. It is fixed to 16 bits for SDRAM. Two data bus widths (8 bits and 16 bits) are available for the PCMCIA interface. 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.

Endian specification and data bus width varies depending on boot mode. For details, refer to section 9.3.2, Data Bus Width and Endian Specification 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 longword data from a byte-width device. In this LSI, data alignment and conversion of data length is performed automatically between the respective interfaces.

Tables 9.5 to 9.8 show the relationship between device data width and access unit. Note that the correspondence between addresses and strobe signals for the 16-bit bus width depends on the endian setting. For example, with big endian and a 16-bit bus width, $\overline{WE1}$ corresponds to the 0th address, which is represented by $\overline{WE0}$ when little endian has been selected.

Since instructions are fetched with both 32- and 16-bit accesses, their alignment in the little-endian area is difficult. Execute instructions from big-endian area.

chanhvinh.com

Table 9.5 16-Bit External Device Access and Data Alignment in Big Endian

Operation		Data Bus		Strobe Signals	
		D15 to D8	D7 to D0	$\overline{WE1}$, DQMU	$\overline{WE0}$, DQML
Byte access at address 0		Data 7 to 0	—	Assert	—
Byte access at address 1		—	Data 7 to 0	—	Assert
Byte access at address 2		Data 7 to 0	—	Assert	—
Byte access at address 3		—	Data 7 to 0	—	Assert
Word access at address 0		Data 15 to 8	Data 7 to 0	Assert	Assert
Word access at address 2		Data 15 to 8	Data 7 to 0	Assert	Assert
Longword access at address 0	1st access at address 0	Data 31 to 24	Data 23 to 16	Assert	Assert
	2nd access at address 2	Data 15 to 8	Data 7 to 0	Assert	Assert

Table 9.6 8-Bit External Device Access and Data Alignment in Big Endian

Operation		Data Bus		Strobe Signals	
		D15 to D8	D7 to D0	$\overline{WE1}$, DQMU	$\overline{WE0}$, DQML
Byte access at address 0		—	Data 7 to 0	—	Assert
Byte access at address 1		—	Data 7 to 0	—	Assert
Byte access at address 2		—	Data 7 to 0	—	Assert
Byte access at address 3		—	Data 7 to 0	—	Assert
Word access at address 0	1st access at address 0	—	Data 15 to 8	—	Assert
	2nd access at address 1	—	Data 7 to 0	—	Assert
Word access at address 2	1st access at address 2	—	Data 15 to 8	—	Assert
	2nd access at address 3	—	Data 7 to 0	—	Assert
Longword access at address 0	1st access at address 0	—	Data 31 to 24	—	Assert
	2nd access at address 1	—	Data 23 to 16	—	Assert
	3rd access at address 2	—	Data 15 to 8	—	Assert
	4th access at address 3	—	Data 7 to 0	—	Assert

chanhvinh.com

Table 9.7 16-Bit External Device Access and Data Alignment in Little Endian

Operation		Data Bus		Strobe Signals	
		D15 to D8	D7 to D0	$\overline{WE1}$, DQMU	$\overline{WE0}$, DQML
Byte access at address 0		—	Data 7 to 0	—	Assert
Byte access at address 1		Data 7 to 0	—	Assert	—
Byte access at address 2		—	Data 7 to 0	—	Assert
Byte access at address 3		Data 7 to 0	—	Assert	—
Word access at address 0		Data 15 to 8	Data 7 to 0	Assert	Assert
Word access at address 2		Data 15 to 8	Data 7 to 0	Assert	Assert
Longword 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 9.8 8-Bit External Device Access and Data Alignment in Little Endian

Operation		Data Bus		Strobe Signals	
		D15 to D8	D7 to D0	$\overline{WE1}$, DQMU	$\overline{WE0}$, DQML
Byte access at address 0		—	Data 7 to 0	—	Assert
Byte access at address 1		—	Data 7 to 0	—	Assert
Byte access at address 2		—	Data 7 to 0	—	Assert
Byte access at address 3		—	Data 7 to 0	—	Assert
Word access at address 0	1st access at address 0	—	Data 7 to 0	—	Assert
	2nd access at address 1	—	Data 15 to 8	—	Assert
Word access at address 2	1st access at address 2	—	Data 7 to 0	—	Assert
	2nd access at address 3	—	Data 15 to 8	—	Assert
Longword 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

9.5.2 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 9.5.8, SRAM Interface with Byte Selection. Figure 9.3 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.

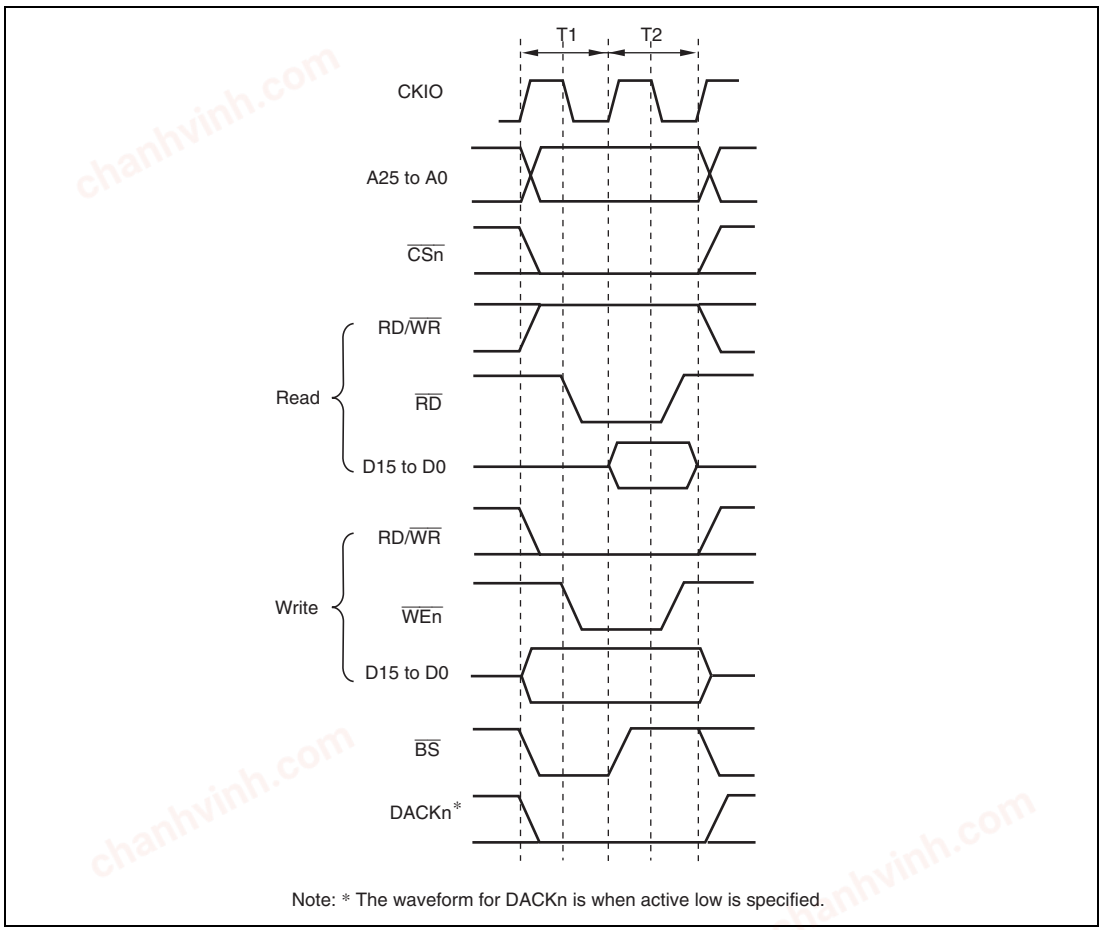


Figure 9.3 Normal Space Basic Access Timing (Access Wait 0, Word Access)

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, 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.

Figures 9.4 and 9.5 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 9.4). If the WM bit in CSnWCR is set to 1, external waits are ignored and no Tnop cycle is inserted (figure 9.5).

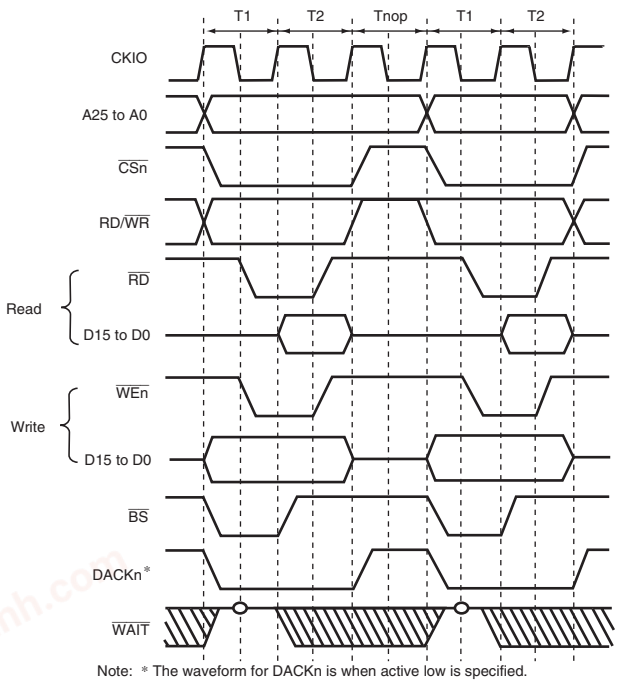


Figure 9.4 Continuous Access to Normal Space (1)
Bus Width = 16 Bits, Longword Access, CSnWCR.WM Bit = 0
(Access Wait = 0, Cycle Wait = 0)

chanhvinh.com

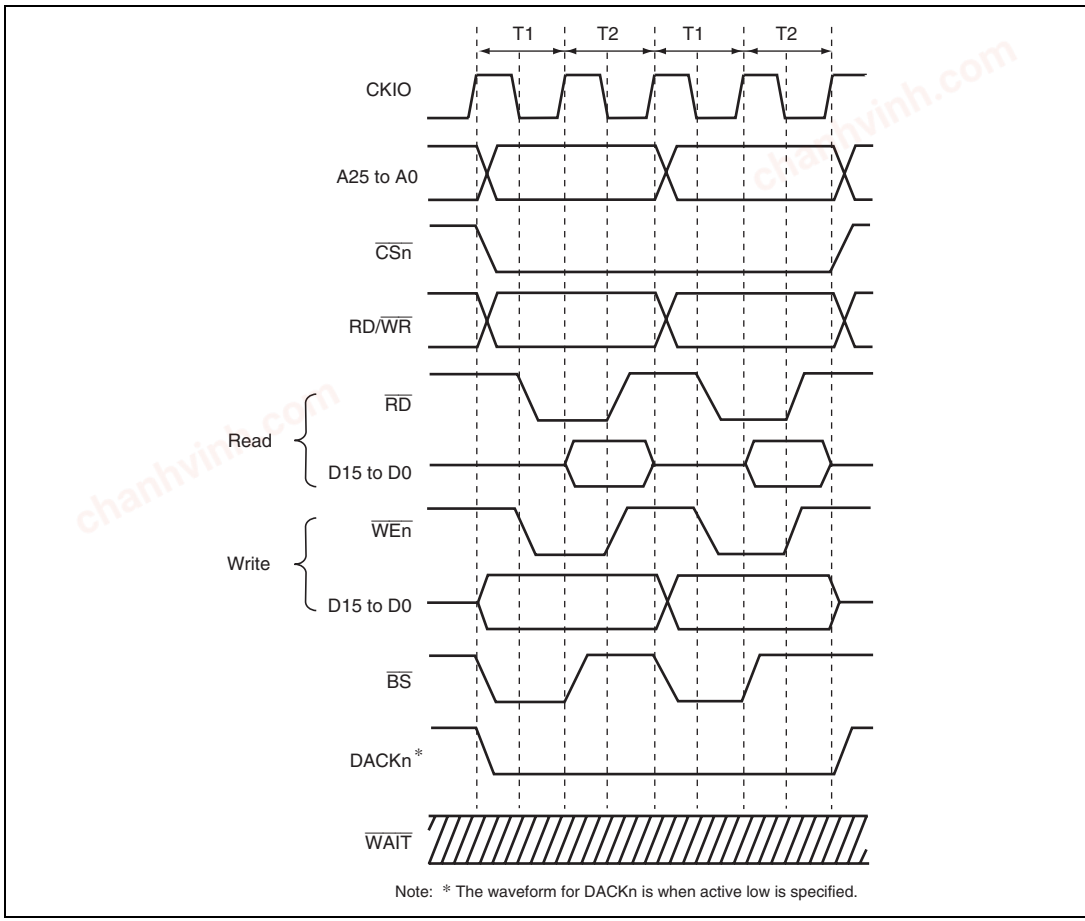


Figure 9.5 Continuous Access to Normal Space (2)
Bus Width = 16 Bits, Longword Access, CSnWCR.WM Bit = 1
(Access Wait = 0, Cycle Wait = 0)

chanhvinh.com

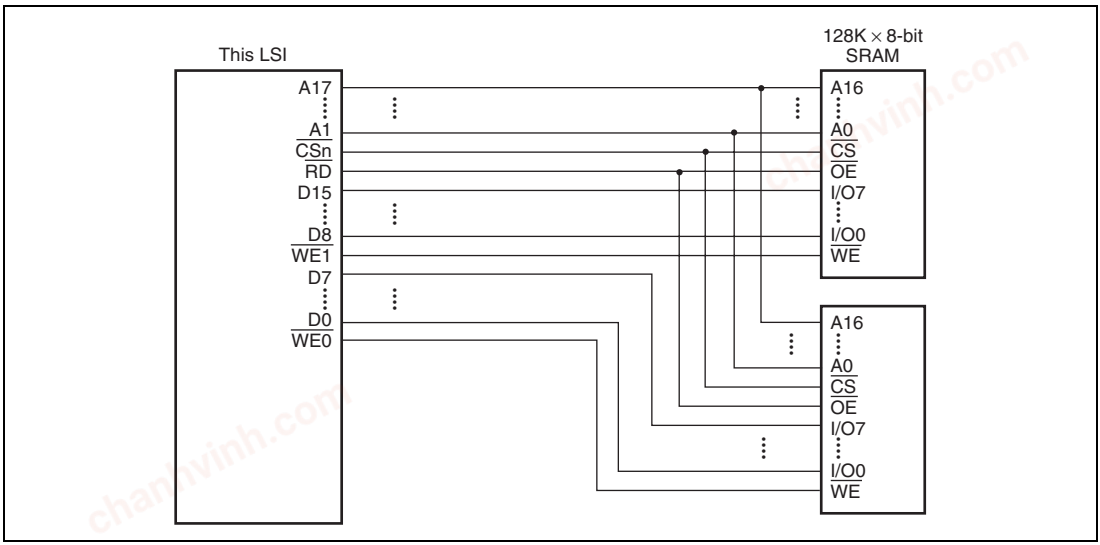


Figure 9.6 Example of 16-Bit Data-Width SRAM Connection

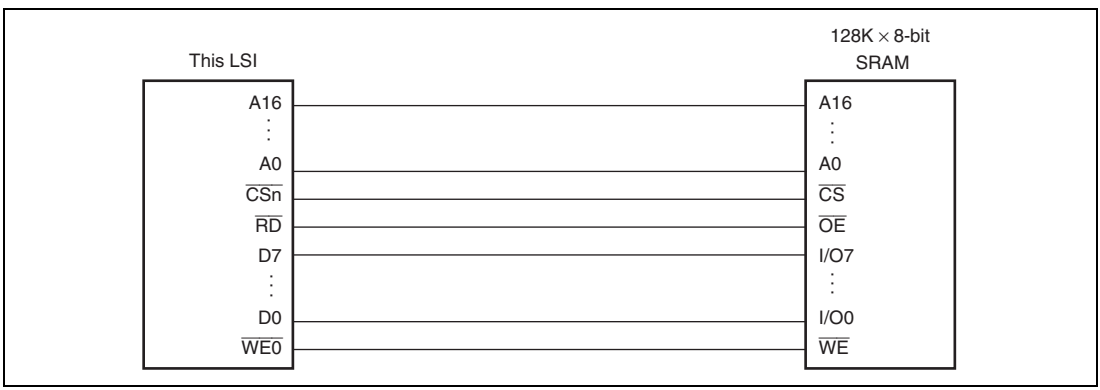


Figure 9.7 Example of 8-Bit Data-Width SRAM Connection

chanhvinh.com

9.5.3 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, 3, and 6 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 9.8.

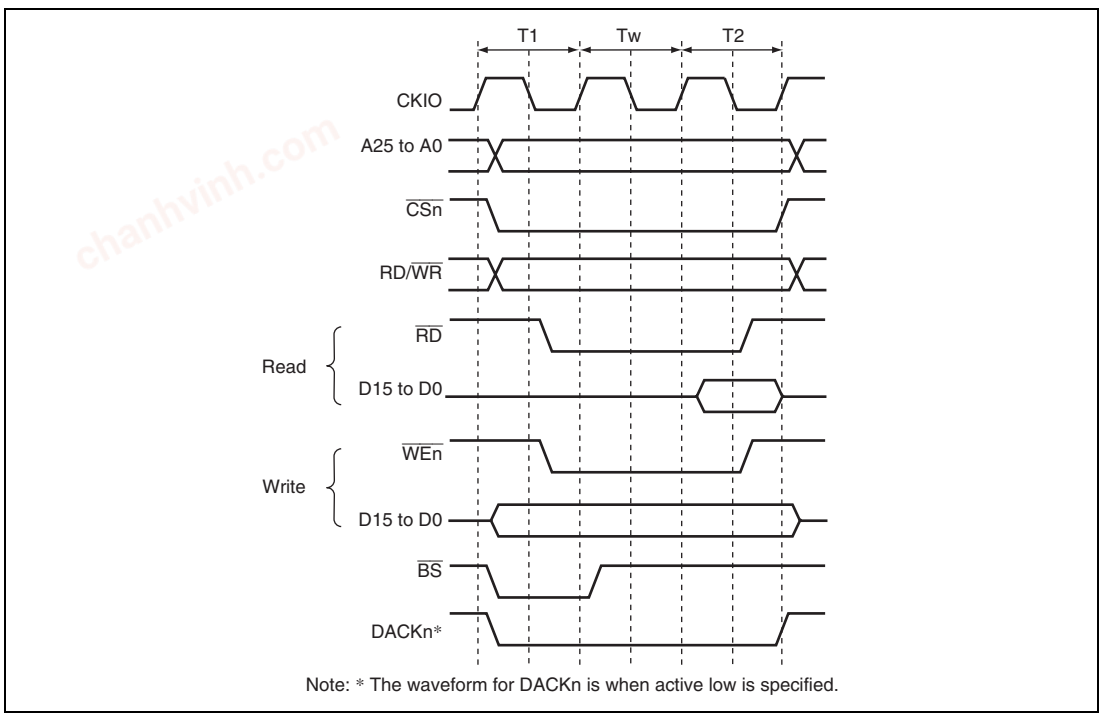


Figure 9.8 Wait Timing for Normal Space Access (Software Wait Only)

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 9.9. 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.

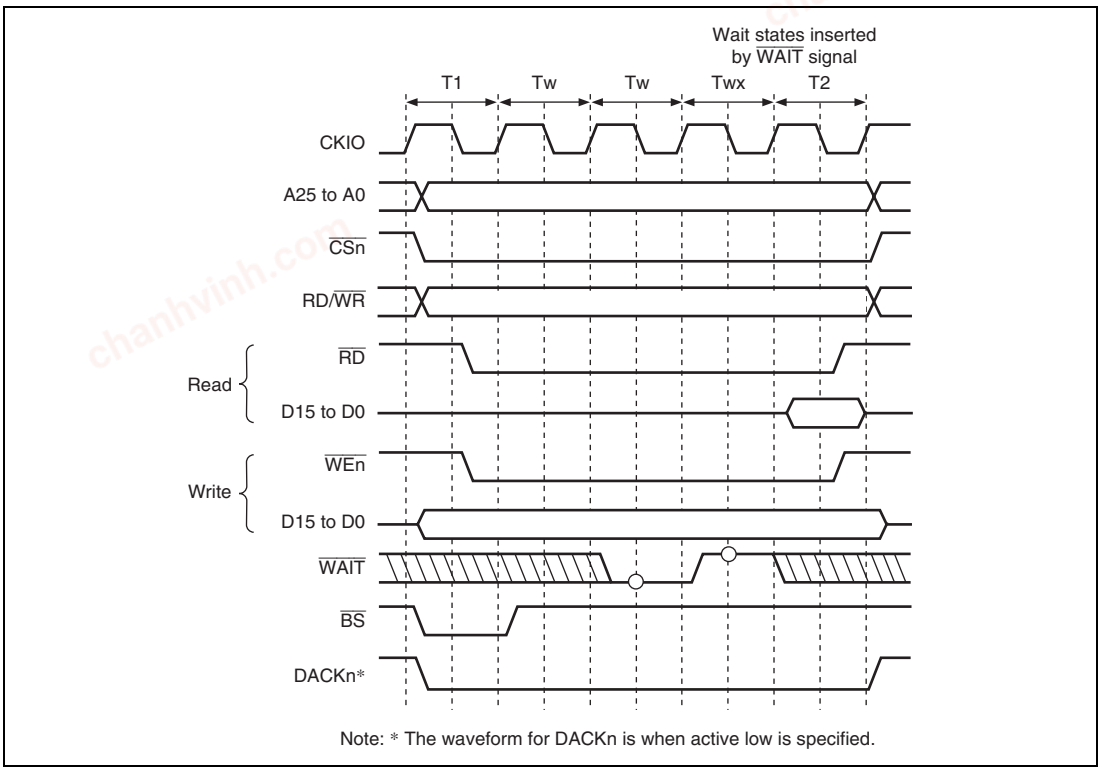


Figure 9.9 Wait Cycle Timing for Normal Space Access (Wait Cycle Insertion Using $\overline{\text{WAIT}}$ Signal)

chanhvinh.com

9.5.4 $\overline{\text{CSn}}$ 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 9.10 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.

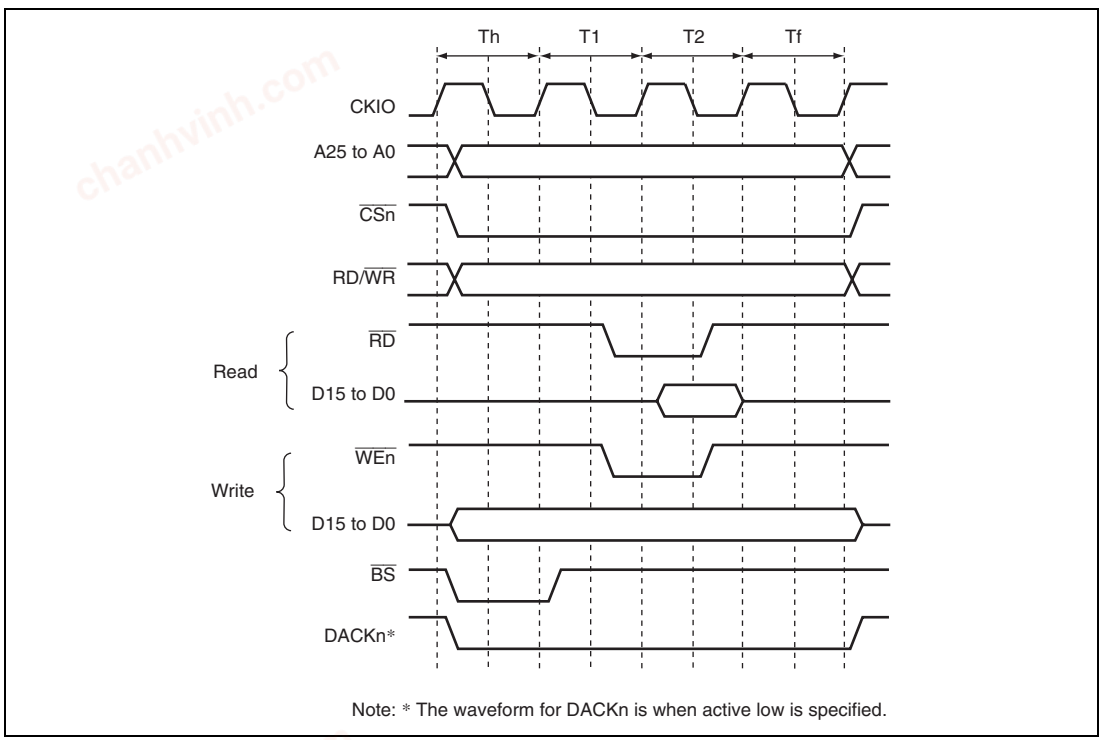


Figure 9.10 $\overline{\text{CSn}}$ Assert Period Expansion

chanhvinh.com

9.5.5 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.

Timing charts are shown in figures 9.11 to 9.13.

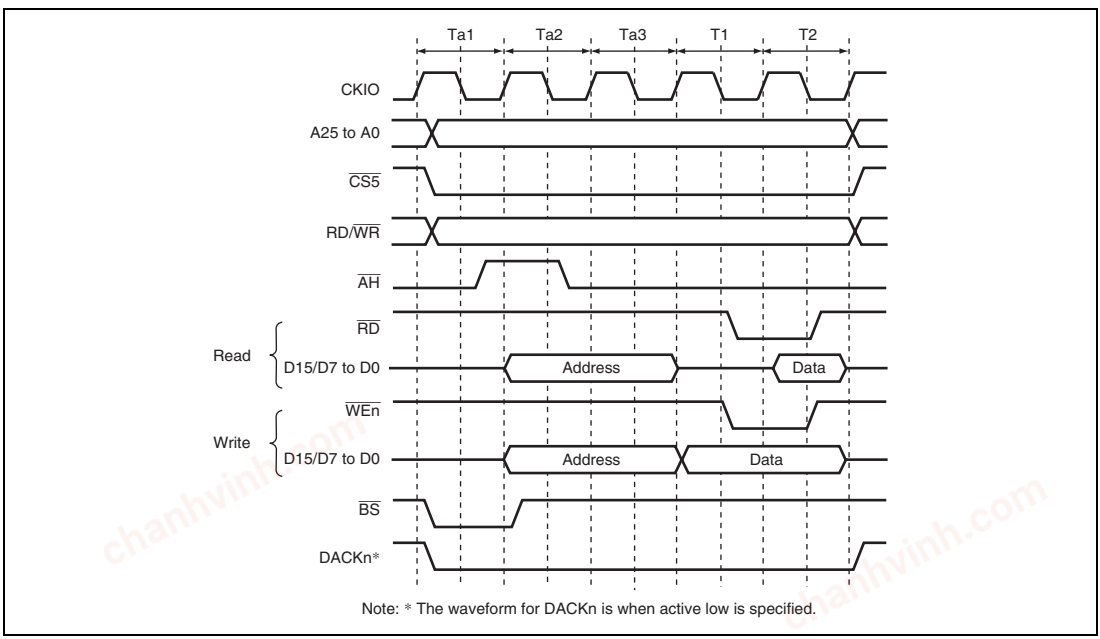
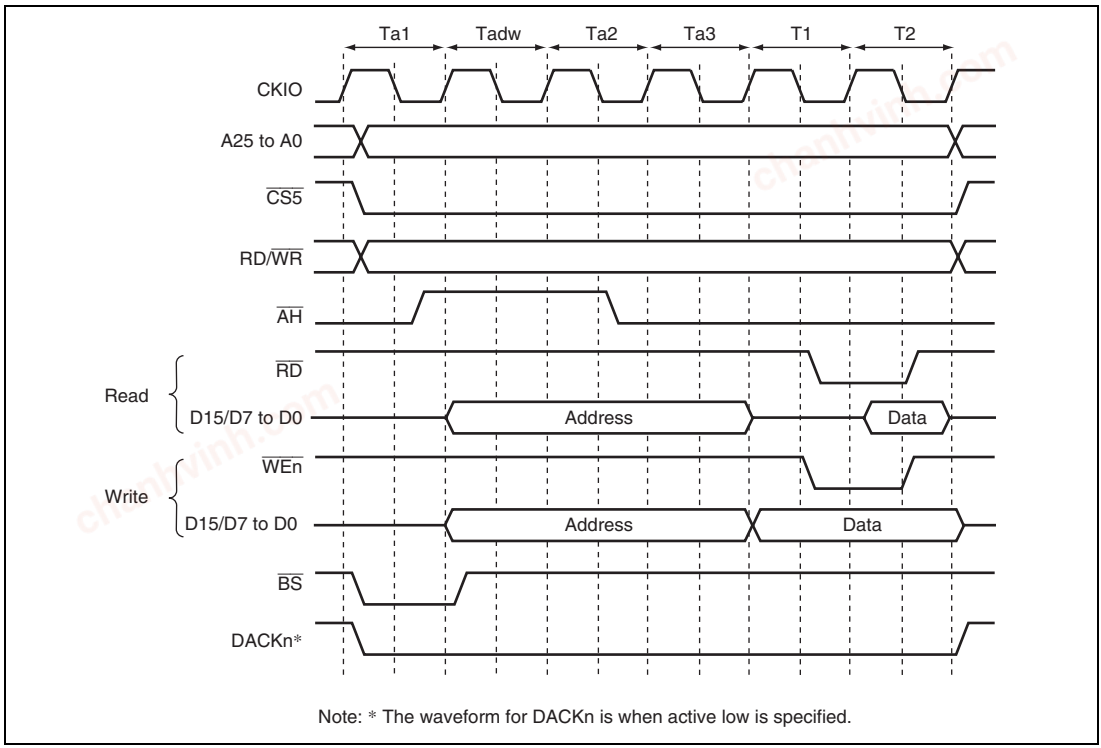


Figure 9.11 Access Timing for MPX Space (Address Cycle No Wait, Data Cycle No Wait)

chanhvinh.com



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

Figure 9.12 Access Timing for MPX Space (Address Cycle Wait 1, Data Cycle No Wait)

chanhvinh.com

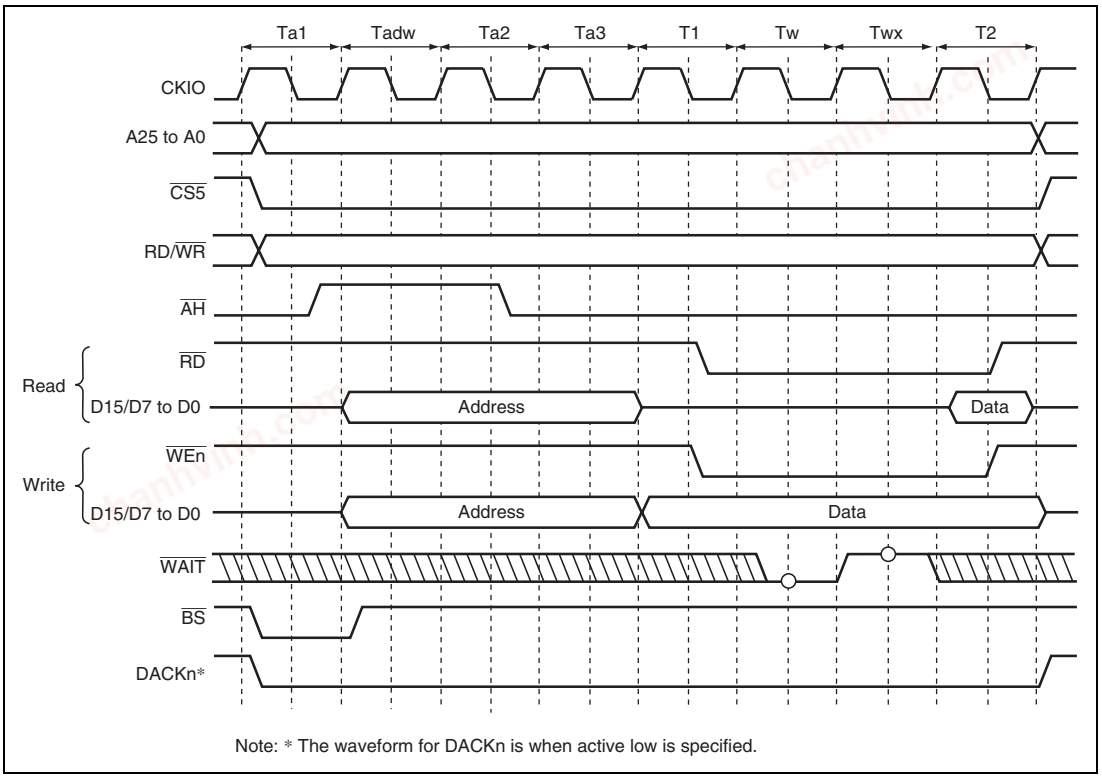


Figure 9.13 Access Timing for MPX Space
(Address Cycle Access Wait 1, Data Cycle Wait 1, External Wait 1)

chanhvinh.com

9.5.6 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/$\overline{WR}$, DQMU, DQML, 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.

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 DQMU and DQML. Reading or writing is performed for a byte whose corresponding DQMx is low. For details on the relationship between DQMx and the byte to be accessed, see section 9.5.1, Endian/Access Size and Data Alignment.

chanhvinh.com

Figures 9.14 shows an example of the connection of the SDRAM with the LSI.

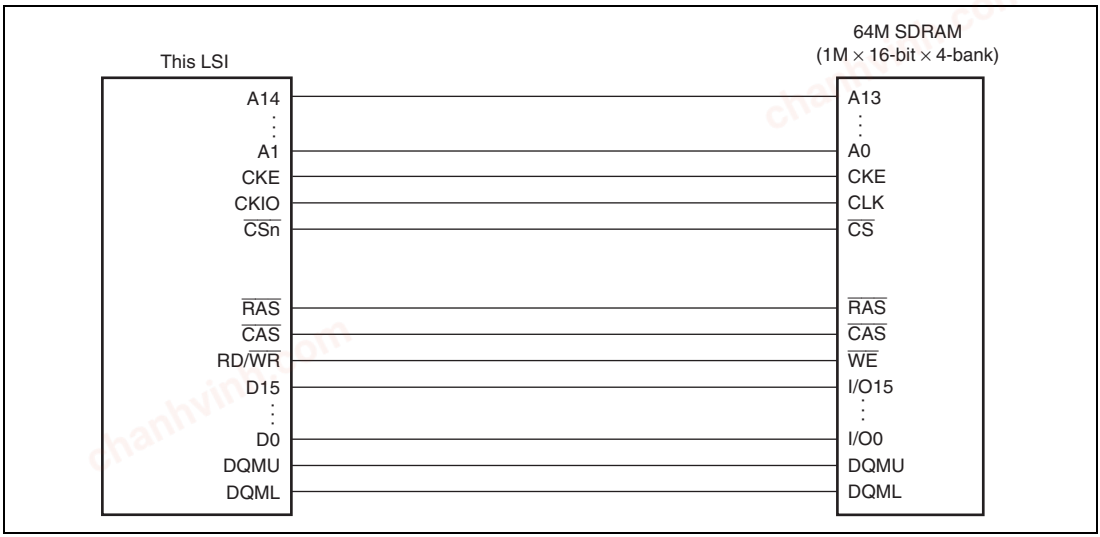


Figure 9.14 Example of 16-Bit Data Width SDRAM Connection

(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. Tables 9.9 to 9.11 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 word 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.

chanhvinh.com

Table 9.9 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 (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				

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

chanhvinh.com

Table 9.9 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]**		
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 Mword × 16 bits × 4 banks, column 8 bits product): 1				

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

chanhvinh.com

Table 9.10 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]**		
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				

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

chanhvinh.com

Table 9.10 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]**		
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				

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

chanhvinh.com

Table 9.11 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (3)-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				

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

chanhvinh.com

Table 9.11 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (3)-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				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
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-byte transfer in cache miss.
- 16-byte transfer in the direct memory access controller

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 9.12 shows the relationship between the access size and the number of bursts.

Table 9.12 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

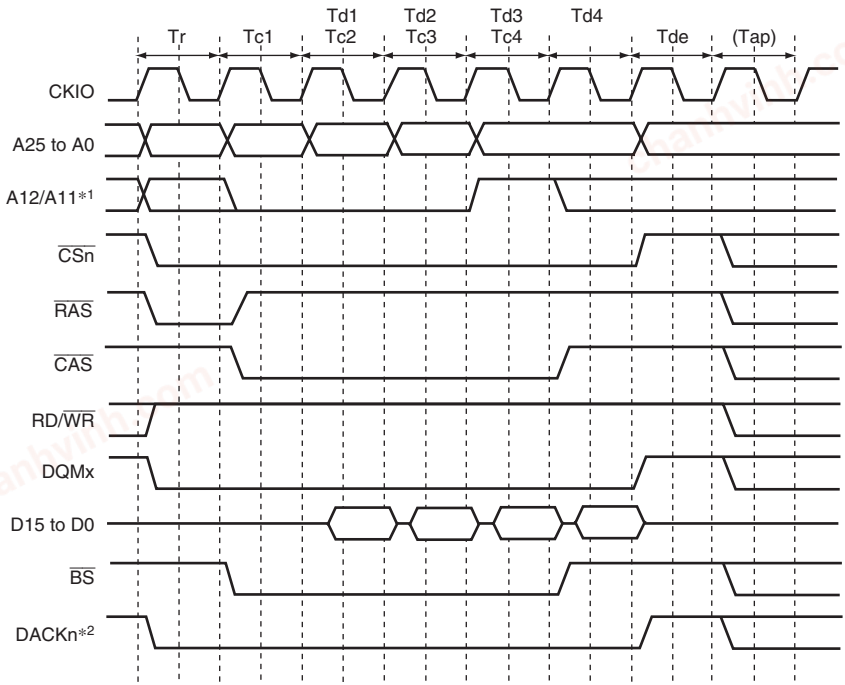
chanhvinh.com

Figures 9.15 and 9.16 show a timing chart 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 9.16 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 cycles 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.

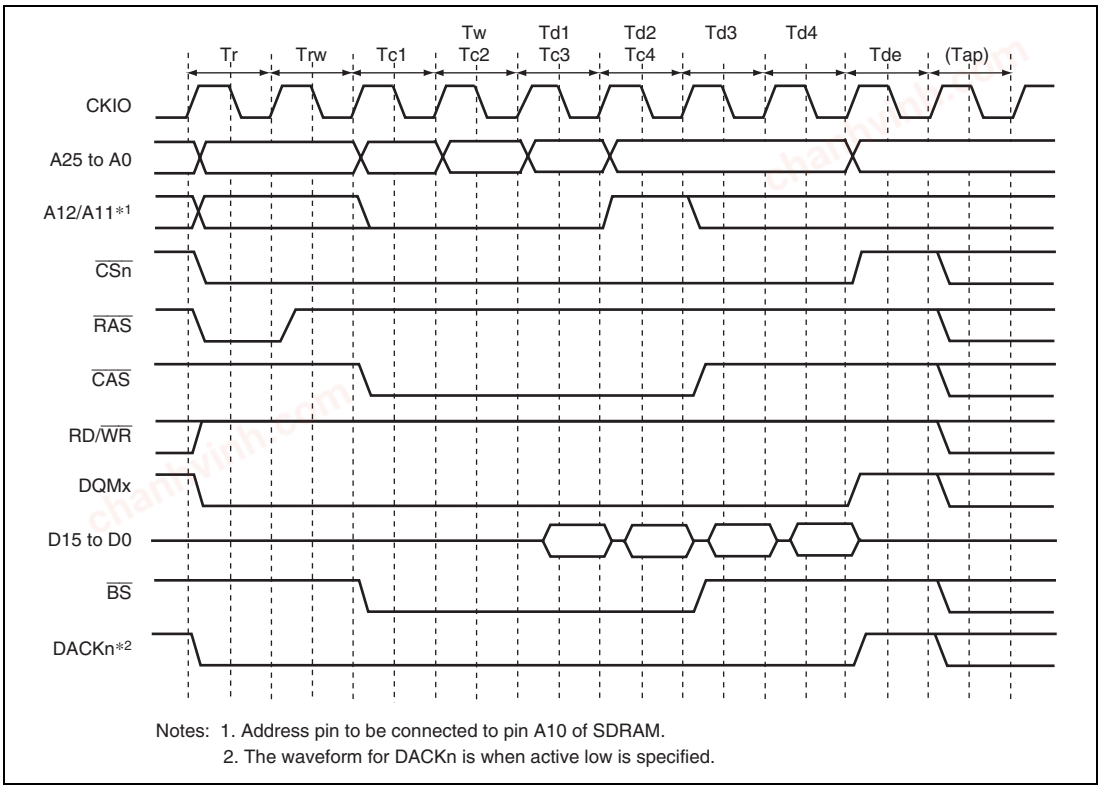
chanhvinh.com



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

Figure 9.15 Burst Read Basic Timing (CAS Latency 1, Auto Pre-Charge)

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 9.16 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 data exists in a cache-disabled space and 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 9.17 shows the single read basic timing.

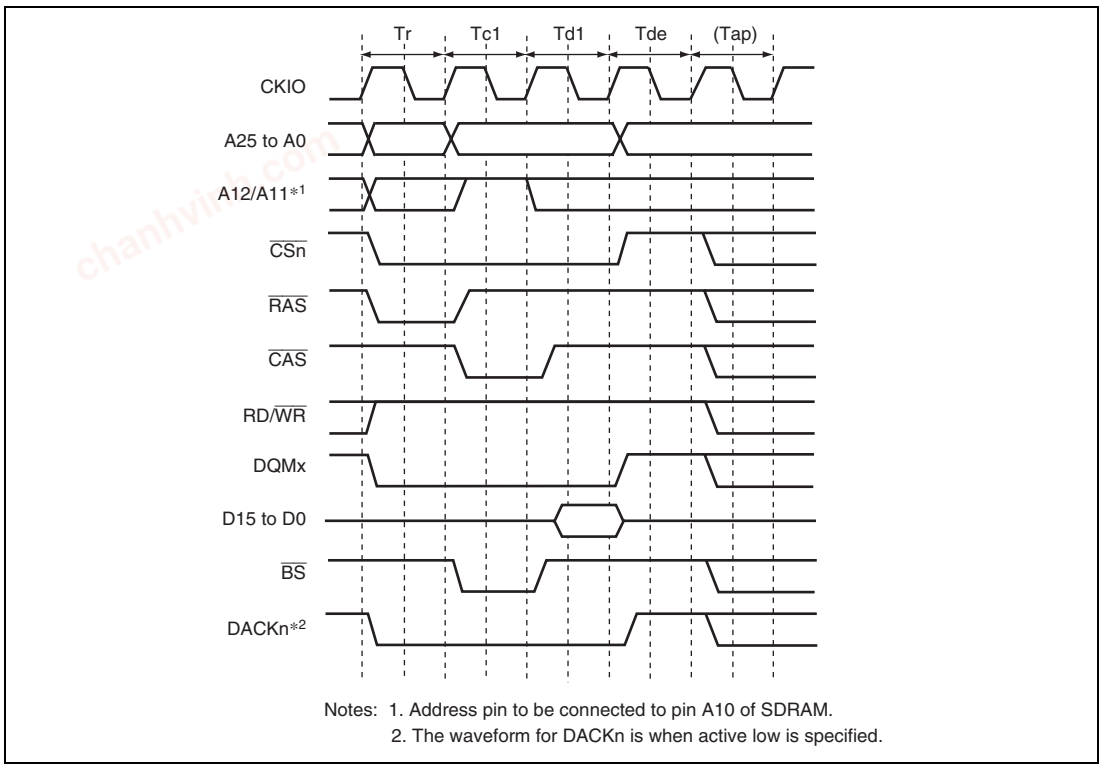


Figure 9.17 Basic Timing for Single Read (CAS Latency 1, Auto Pre-Charge)

chanhvinh.com

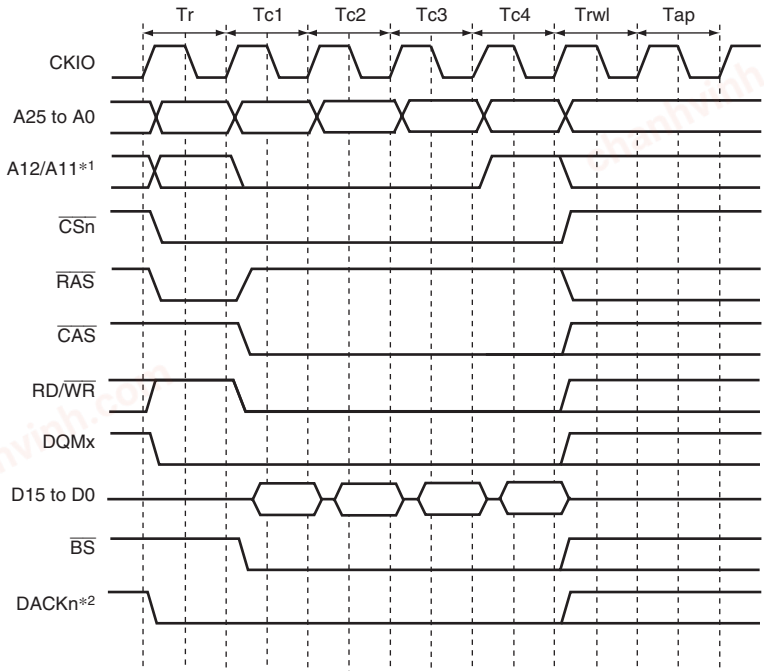
(5) Burst Write

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

- Access size in writing is larger than data bus width.
- Write-back of the cache
- 16-byte transfer in the direct memory access controller

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 9.12. Figure 9.18 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.

chanhvinh.com



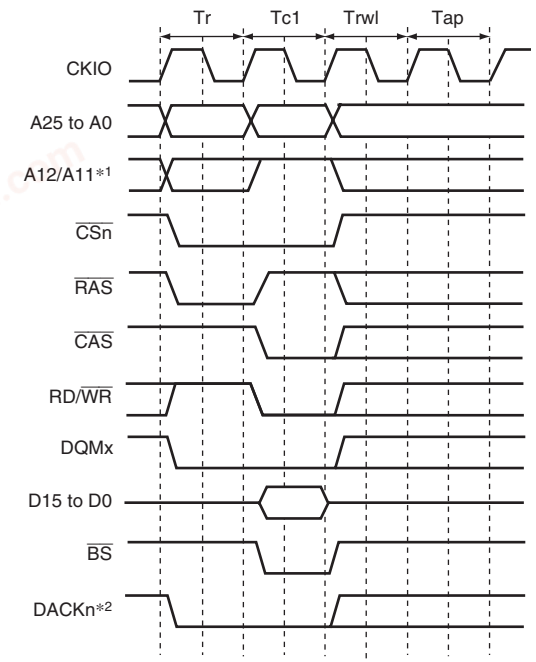
Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 9.18 Basic Timing for Burst Write (Auto Pre-Charge)

chanhvinh.com

(6) Single Write

A write access ends in one cycle when data is written in a cache-disabled space and 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 9.19 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 9.19 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 9.20, a burst read cycle for the same row address in figure 9.21, and a burst read cycle for different row addresses in figure 9.22. Similarly, a single write cycle without auto-precharge is shown in figure 9.23, a single write cycle for the same row address in figure 9.24, and a single write cycle for different row addresses in figure 9.25.

In figure 9.21, 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 DQMx 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 DQMx signal is asserted after the Tc cycle.

chanhvinh.com

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 9.20 or 9.23, followed by repetition of the cycle in figure 9.21 or 9.24. 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 9.22 or 9.25 is executed instead of that in figure 9.21 or 9.24. In bank active mode, too, all banks become inactive after a refresh cycle or after the bus is released as the result of bus arbitration.

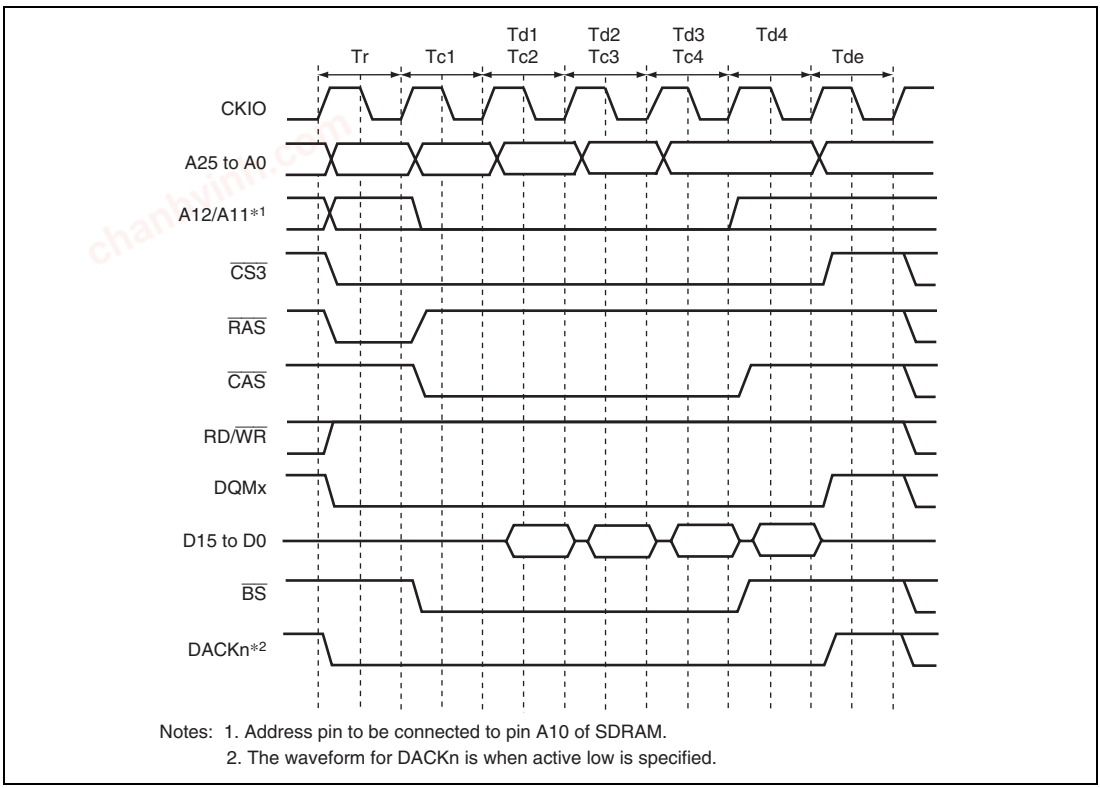
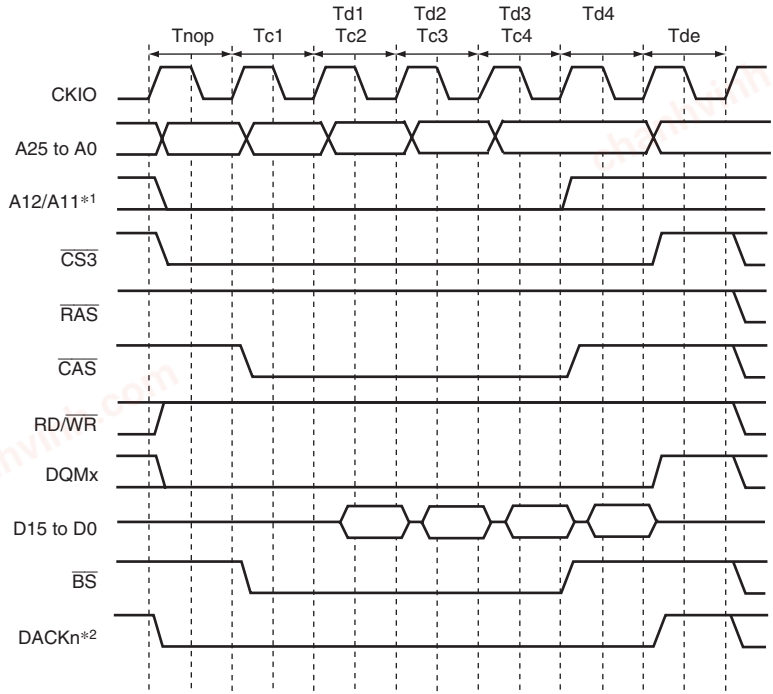


Figure 9.20 Burst Read Timing (Bank Active, Different Bank, CAS Latency 1)

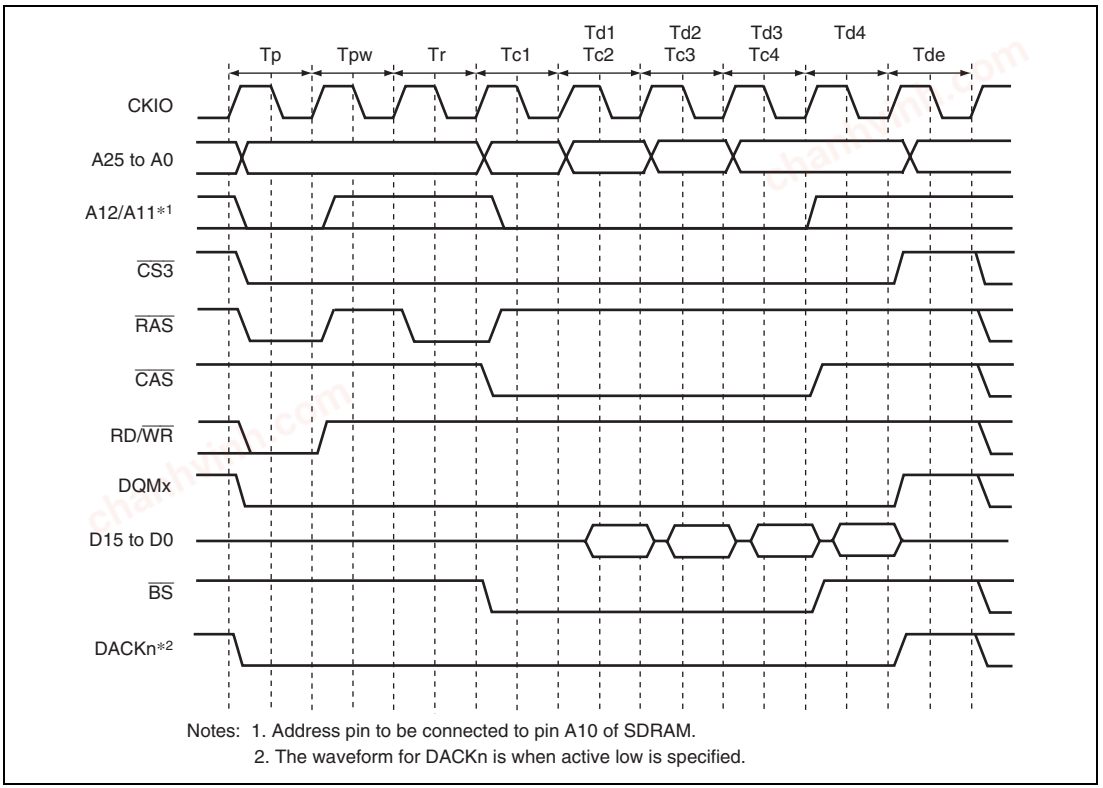
chanhvinh.com



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

Figure 9.21 Burst Read Timing
(Bank Active, Same Row Addresses in the Same Bank, CAS Latency 1)

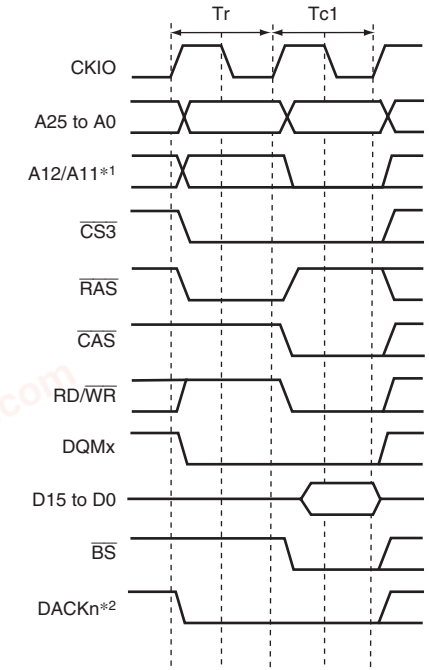
chanhvinh.com



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

Figure 9.22 Burst Read Timing
(Bank Active, Different Row Addresses in the Same Bank, CAS Latency 1)

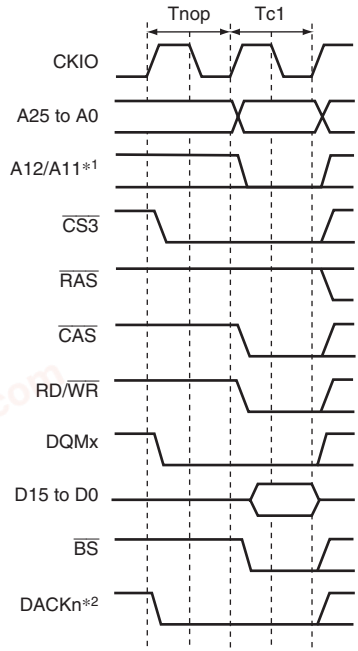
chanhvinh.com



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

Figure 9.23 Single Write Timing (Bank Active, Different Bank)

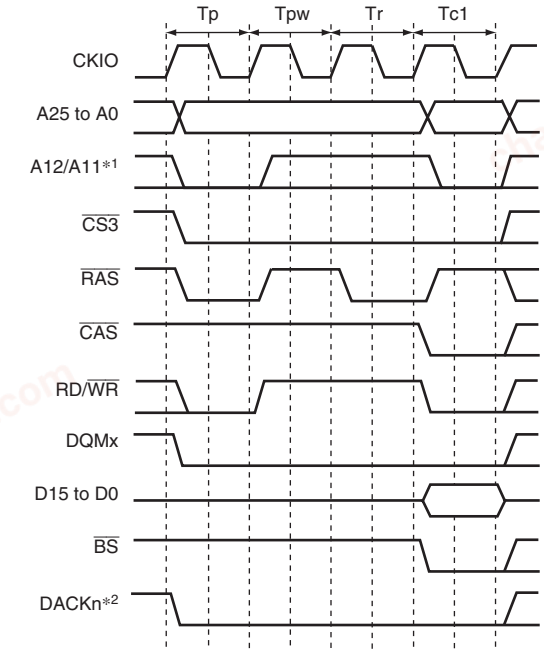
chanhvinh.com



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

Figure 9.24 Single Write Timing (Bank Active, Same Row Addresses in the Same 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 9.25 Single Write Timing (Bank Active, Different Row Addresses in the Same Bank)

(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.

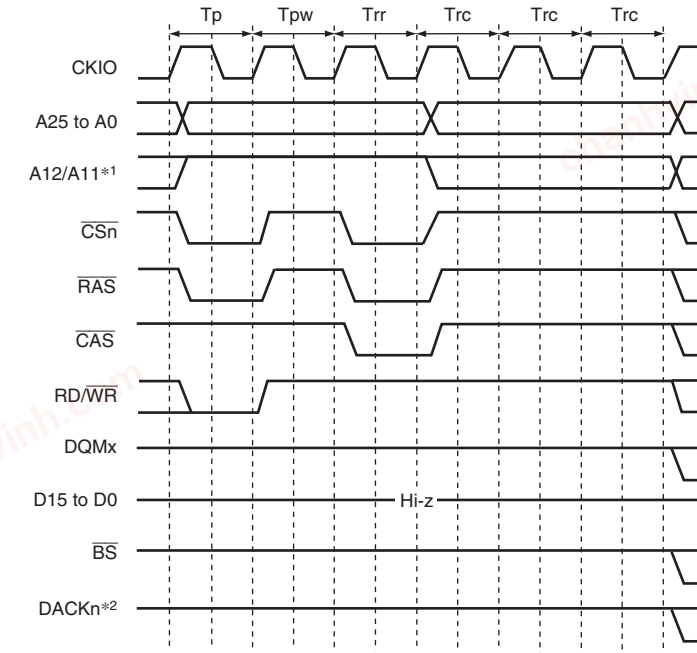
chanhvinh.com

(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 9.26 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.

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 9.26 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.

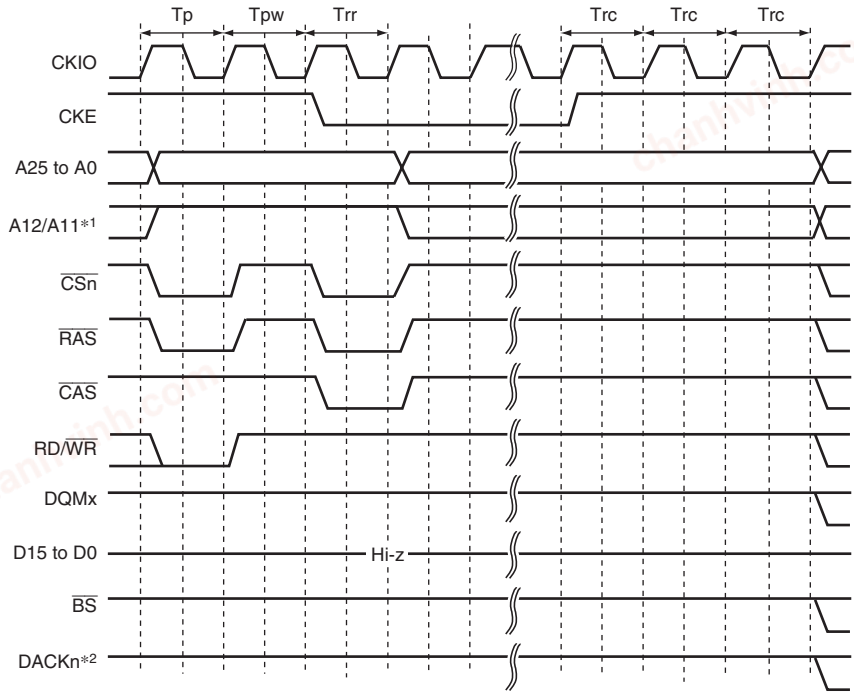
Self-refresh timing is shown in figure 9.27. Settings must be made so that self-refresh clearing and data retention are performed correctly, and auto-refreshing is performed at the correct intervals. When self-refreshing is activated from the state in which auto-refreshing is set, auto-refreshing is 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.

When the multiplication rate for the PLL circuit is changed, the CKIO output will become unstable or will be fixed low. For details on the CKIO output, see section 5, Clock Pulse Generator. The contents of SDRAM can be retained by placing the SDRAM in the self-refresh state before changing the multiplication rate.

The self-refresh state is not cleared by a manual reset. In case of a power-on reset, the bus state controller's registers are initialized, and therefore the self-refresh state is cleared.

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 9.27 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 refresh request occurs while the bus is released by the bus arbitration function, the refresh will not be executed until the bus mastership is acquired.

If the external bus does not return the bus for a period longer than the specified refresh interval, refresh cannot be executed and the SDRAM contents may be lost.

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 or the bus mastership occupation must be prevented from occurring.

If a bus mastership is requested during self-refresh, the bus will not be released until the refresh is completed.

(10) Low-Frequency Mode

When the SLOW bit in SDCR is set to 1, output of commands, addresses, and write data, and fetch of read data are performed at a timing suitable for operating SDRAM at a low frequency.

Figure 9.28 shows the access timing in low-frequency mode. In this mode, commands, addresses, and write data are output in synchronization with the falling edge of CKIO, which is half a cycle delayed than the normal timing. Read data is fetched at the rising edge of CKIO, which is half a cycle faster than the normal timing. This timing allows the hold time of commands, addresses, write data, and read data to be extended.

If SDRAM is operated at a high frequency with the SLOW bit set to 1, the setup time of commands, addresses, write data, and read data are not guaranteed. Take the operating frequency and timing design into consideration when making the SLOW bit setting.

chanhvinh.com

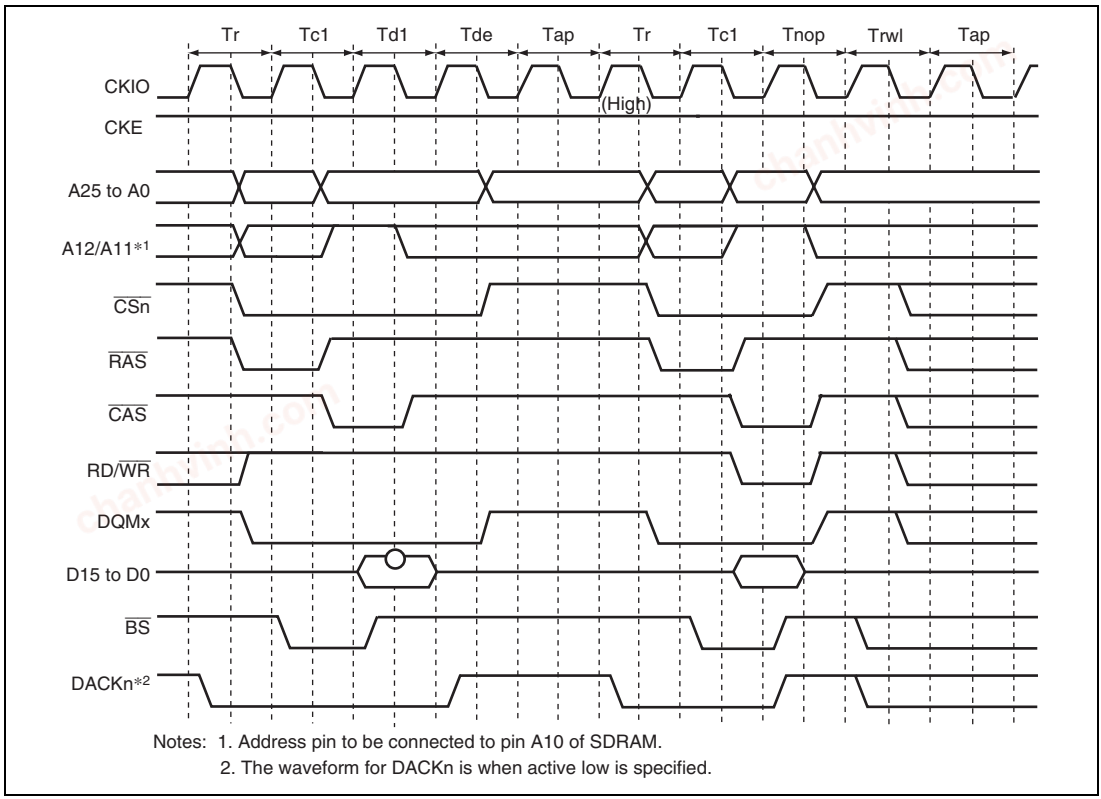


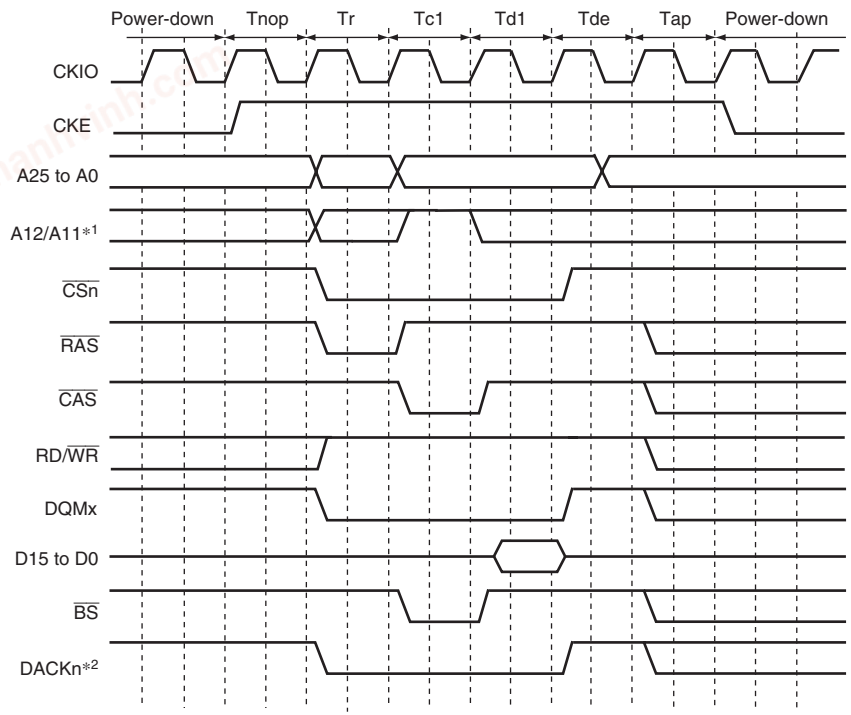
Figure 9.28 Low-Frequency Mode Access Timing

chanhvinh.com

(11) 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, please 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 9.29 shows the access timing in power-down mode.



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

Figure 9.29 Power-Down Mode Access Timing

chanhvinh.com

(12) 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 $\overline{CS}$n, $\overline{RAS}$, $\overline{CAS}$, 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 word write to address H'FFFC4000 + X for area 2 SDRAM, and to address H'FFFC5000 + 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 a byte-size access to the addresses shown in table 9.13. In this time 0 is output at the external address pins of A12 or later.

Table 9.13 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'FFFC4440	H'0000440
	3	H'FFFC4460	H'0000460

Burst read/burst write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'FFFC4040	H'0000040
	3	H'FFFC4060	H'0000060

chanhvinh.com

- 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'FFFC5440	H'0000440
	3	H'FFFC5460	H'0000460

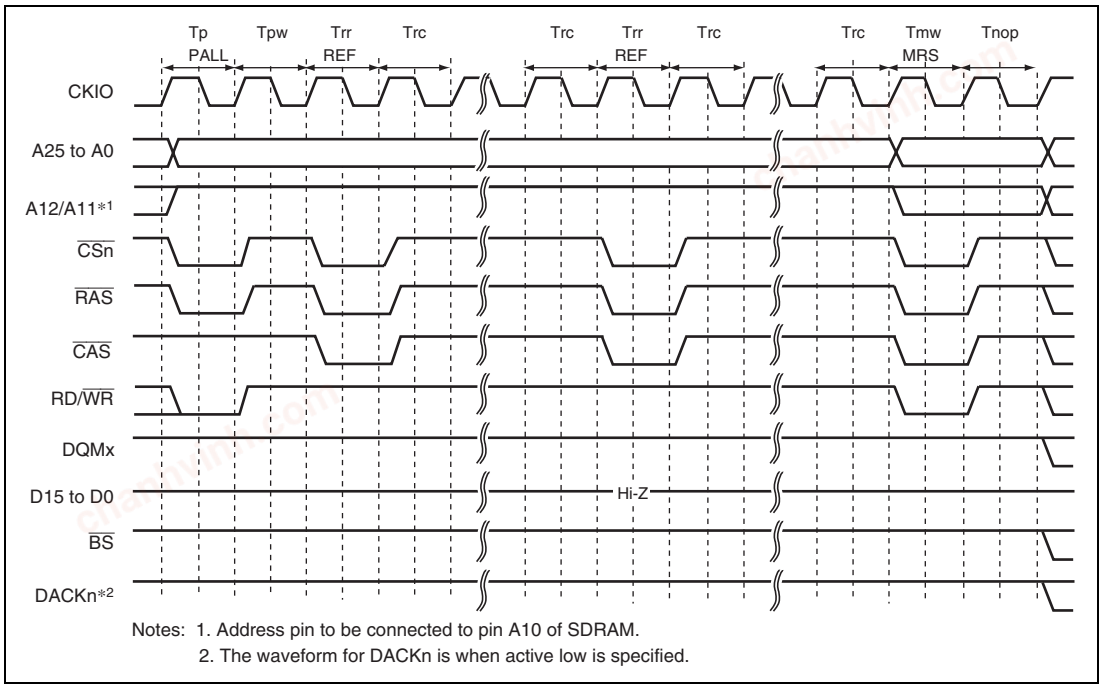
  Burst read/burst write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'FFFC5040	H'0000040
	3	H'FFFC5060	H'0000060

Mode register setting timing is shown in figure 9.30. 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.

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 9.30 SDRAM Mode Write Timing (Based on JEDEC)

chanhvinh.com

(13) 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, please refer to the Data Sheet for the low-power SDRAM to be used.

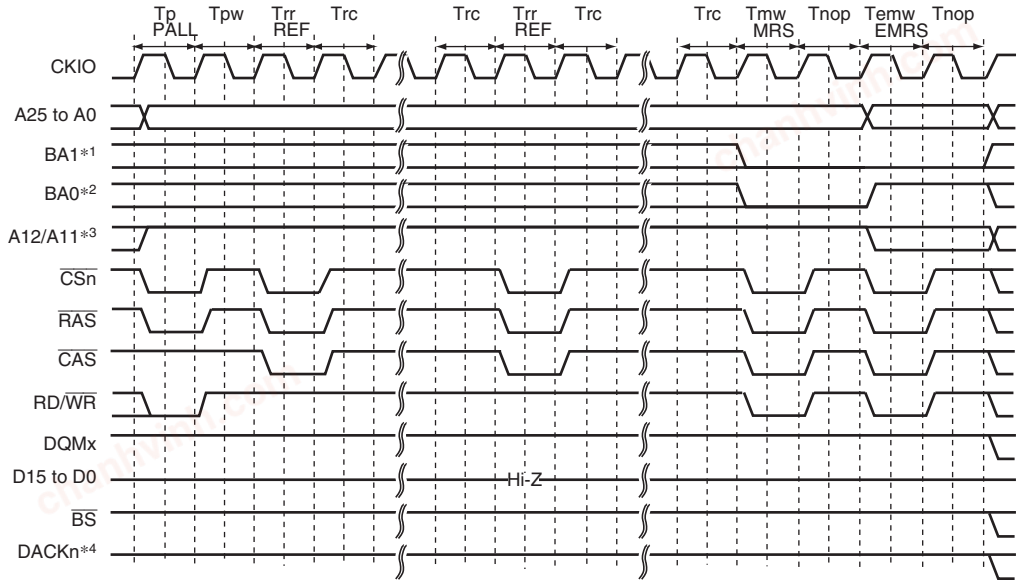
The low-power SDRAM supports the extension mode register in addition to the mode registers as the normal SDRAM. This LSI supports issuing of the extension mode register write command (EMRS).

The EMRS command is issued according to the conditions specified in table below. For example, if data H'0YYYYYYY is written to address H'FFFC5XX0 in longword, 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'FFFC5XX0 in longword, the commands are issued to the CS3 space in the following sequence: PALL -> MRS -> EMRS.

Table 9.14 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'FFFC4XX0	H'********	16 bits	H'0000XX0	—
CS3 MRS	H'FFFC5XX0	H'********	16 bits	H'0000XX0	—
CS2 MRS + EMRS (with refresh)	H'FFFC4XX0	H'0YYYYYYY	32 bits	H'0000XX0	H'YYYYYYY
CS3 MRS + EMRS (with refresh)	H'FFFC5XX0	H'0YYYYYYY	32 bits	H'0000XX0	H'YYYYYYY
CS2 MRS + EMRS (without refresh)	H'FFFC4XX0	H'1YYYYYYY	32 bits	H'0000XX0	H'YYYYYYY
CS3 MRS + EMRS (without refresh)	H'FFFC5XX0	H'1YYYYYYY	32 bits	H'0000XX0	H'YYYYYYY

chanhvinh.com



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 9.31 EMRS Command Issue Timing

chanhvinh.com

- 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.

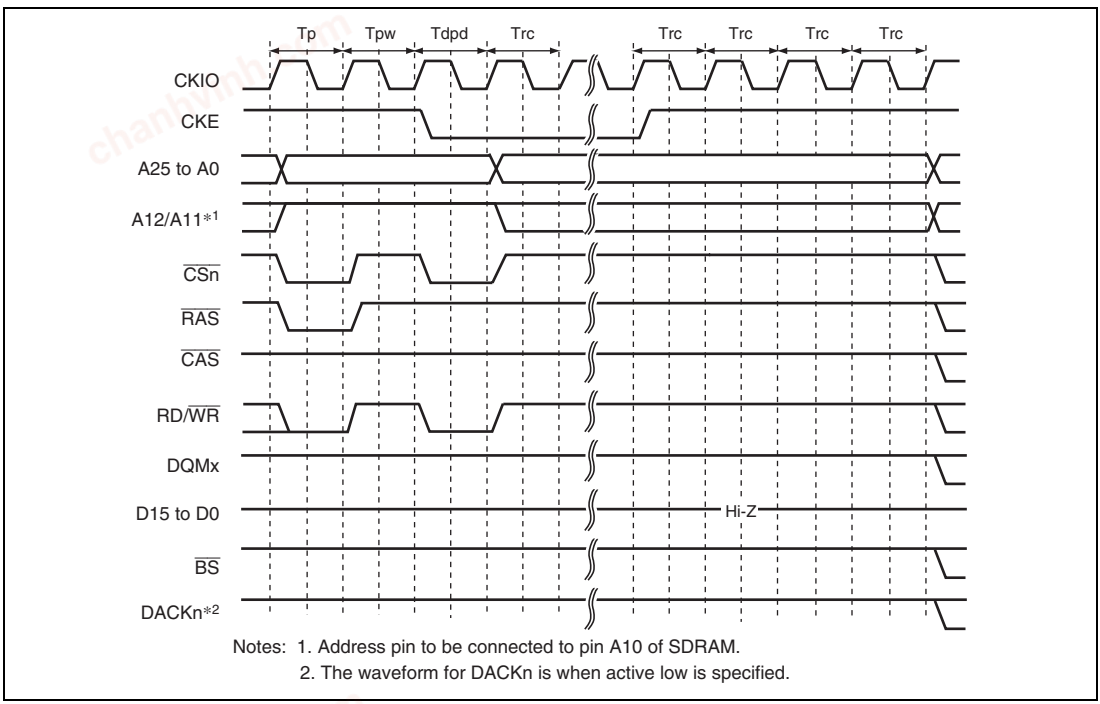


Figure 9.32 Deep Power-Down Mode Transition Timing

chanhvinh.com

9.5.7 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 9.15 lists a relationship between bus width, access size, and the number of bursts. Figure 9.33 shows a timing chart.

Table 9.15 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

chanhvinh.com

Bus Width	Access Size	CSnWCR. BST[1:0] Bits	Number of Bursts	Access Count
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*	4	2
			2, 4, 2	3

Note: * When the bus width is 16 bits, the access size is 16 bits, 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.

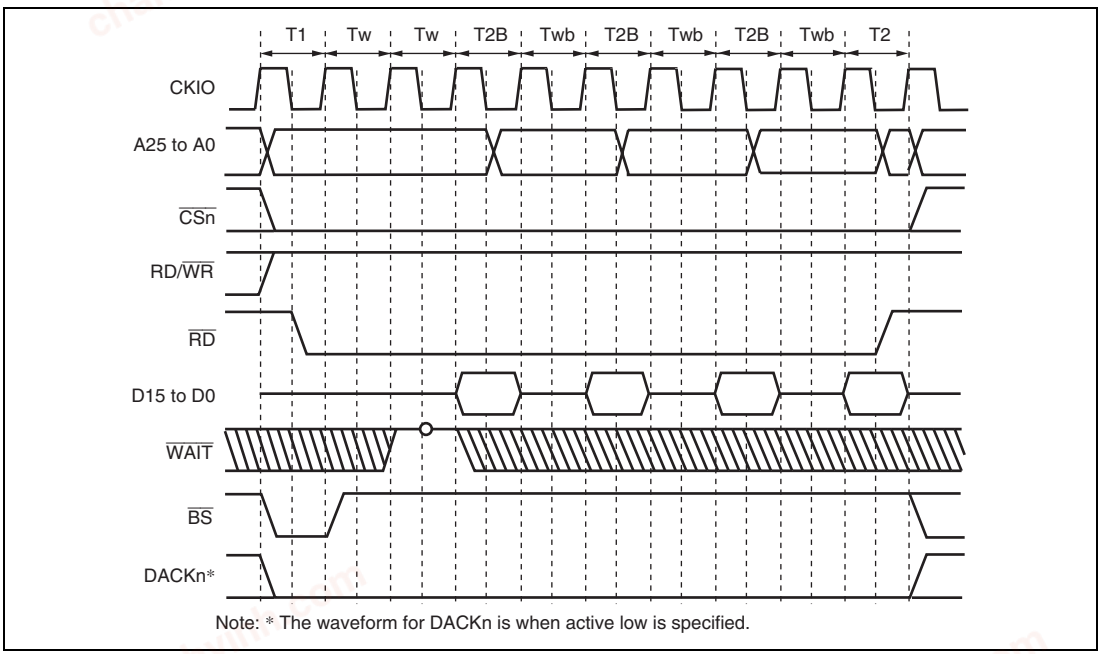


Figure 9.33 Burst ROM Access Timing (Clocked Asynchronous)
(Bus Width = 16Bits, 16-Byte Transfer (Number of Burst 4-4), Wait Cycles Inserted in First Access = 2, Wait Cycles Inserted in Second and Subsequent Access Cycles = 1)

chanhvinh.com

9.5.8 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 UB and 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 9.34. In write access, data is written to the memory according to the timing of the byte-selection pin ($\overline{\text{WEn}}$). For details, please 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 9.35 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 9.36 shows the access timing when a software wait is specified.

chanhvinh.com

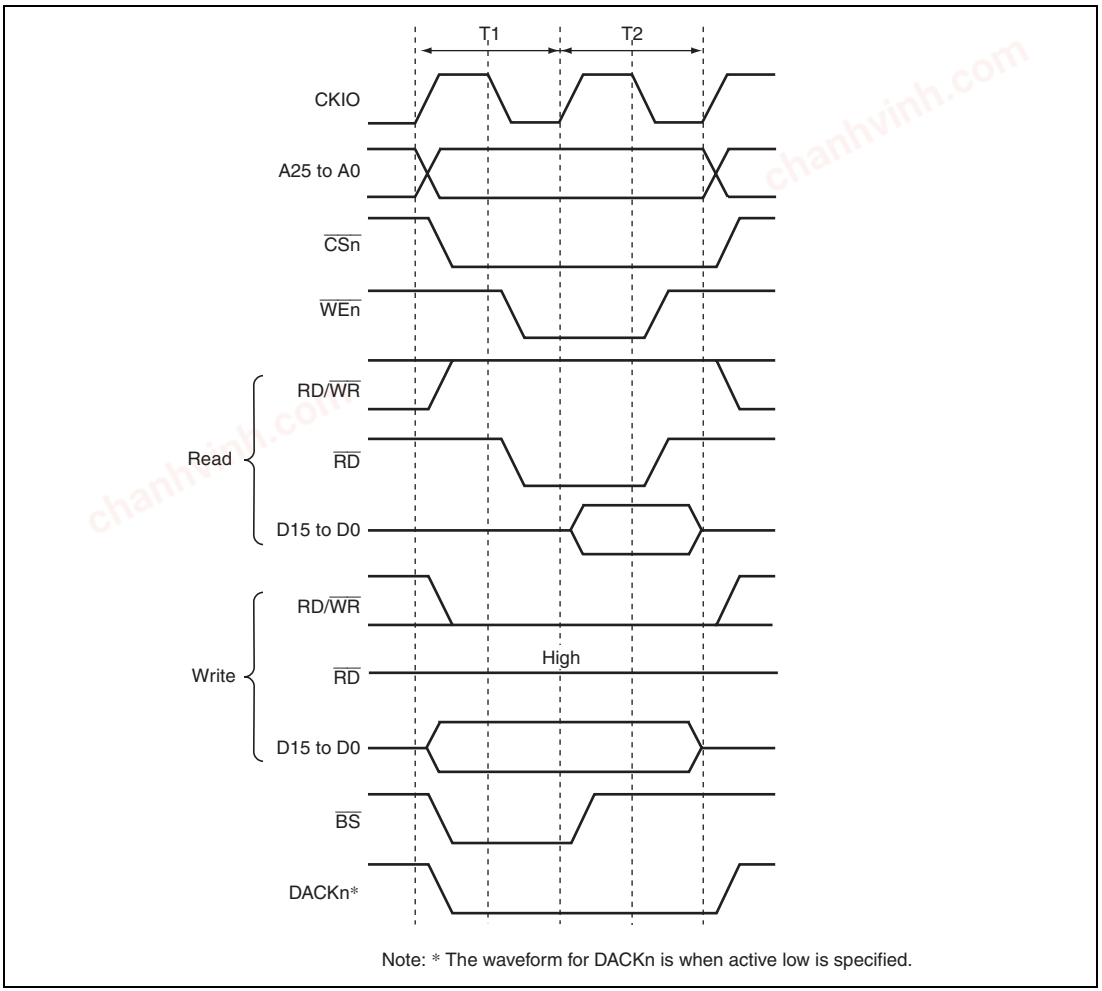


Figure 9.34 Basic Access Timing for SRAM with Byte Selection (BAS = 0)

chanhvinh.com

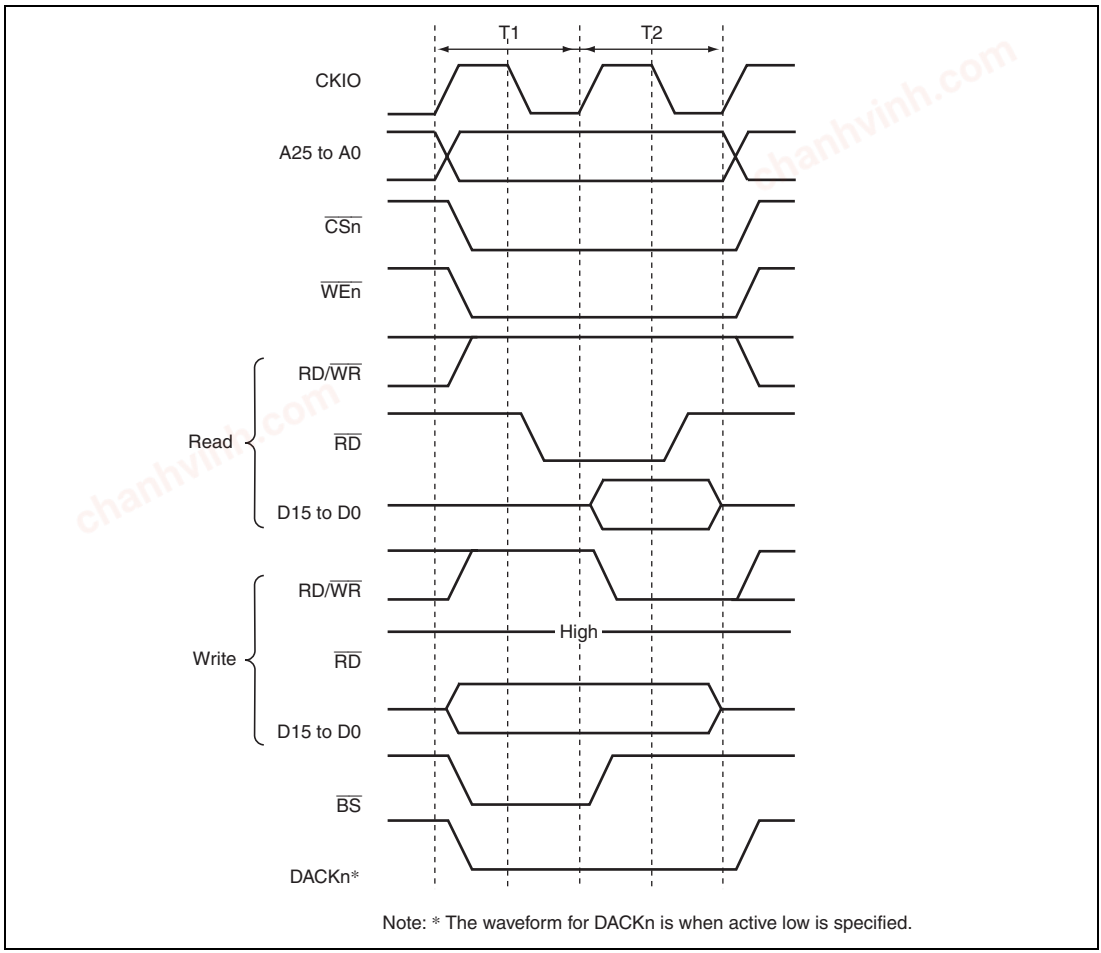


Figure 9.35 Basic Access Timing for SRAM with Byte Selection (BAS = 1)

chanhvinh.com

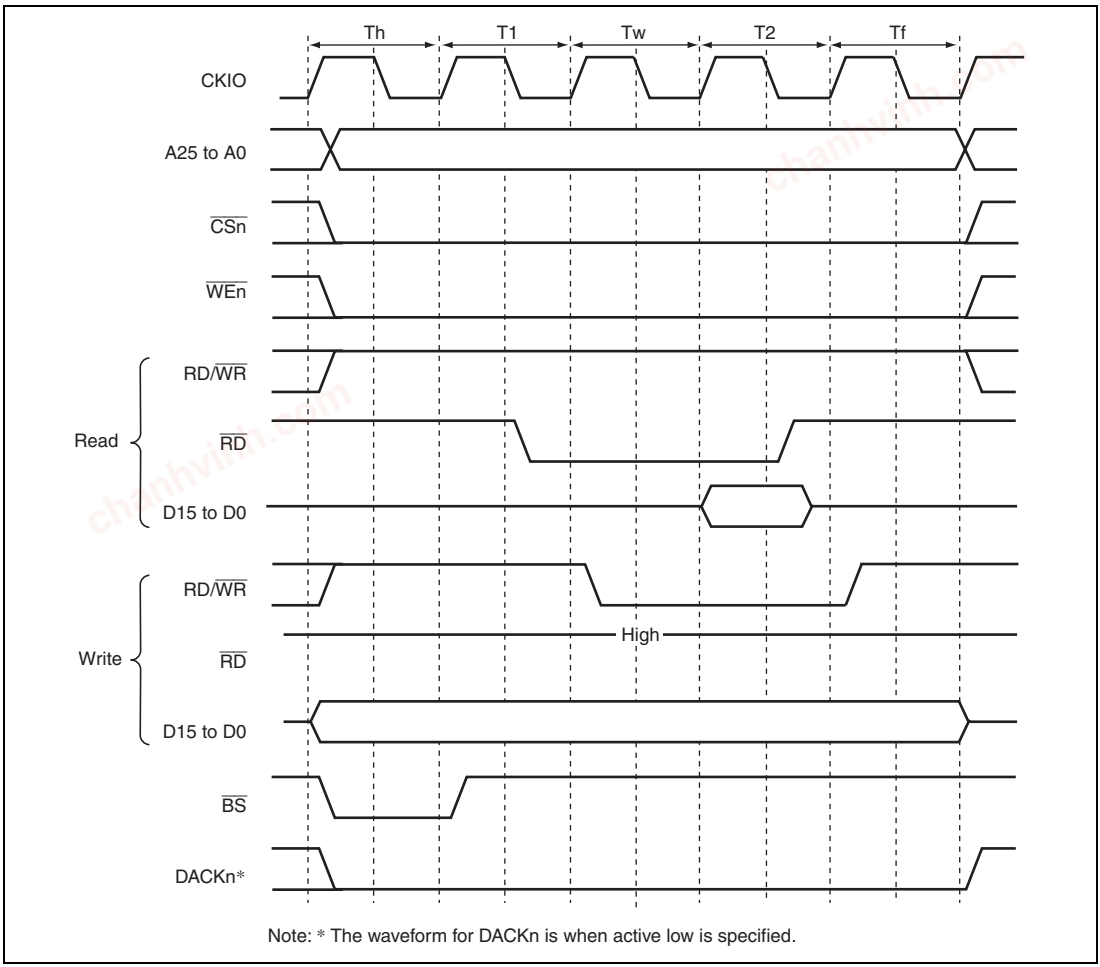


Figure 9.36 Wait Timing for SRAM with Byte Selection (BAS = 1) (SW[1:0] = 01, WR[3:0] = 0001, HW[1:0] = 01)

chanhvinh.com

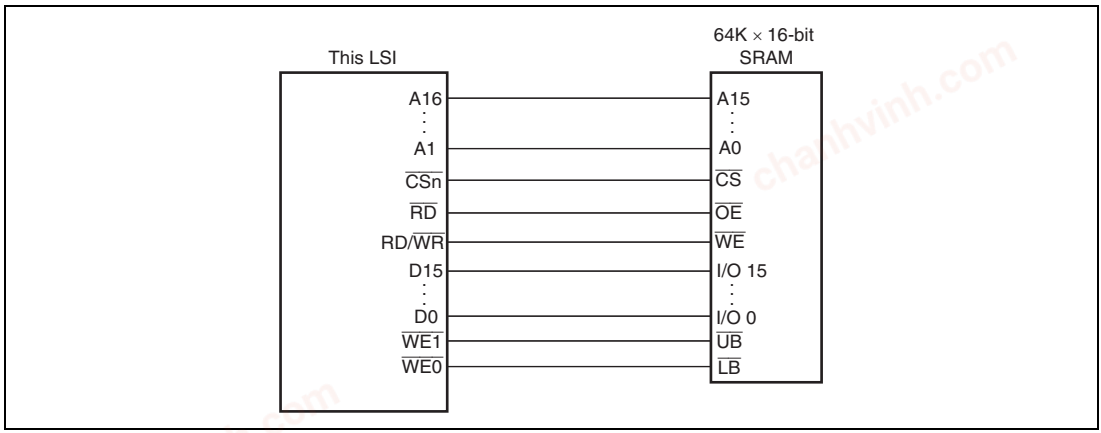


Figure 9.37 Example of Connection with 16-Bit Data-Width SRAM with Byte Selection

9.5.9 PCMCIA Interface

With this LSI, areas 5 and 6 can be used for the IC memory card and I/O card interface defined in the JEIDA specifications version 4.2 (PCMCIA2.1 Rev. 2.1) by specifying bits TYPE[2:0] in CSnBCR (n = 5 and 6) to B'101. In addition, the bits SA[1:0] in CSnWCR (n = 5 and 6) assign the upper or lower 32 Mbytes of each area to IC memory card or I/O card interface. For example, if the bits SA1 and SA0 in CS5WCR are set to 1 and cleared to 0, respectively, the upper 32 Mbytes of area 5 are used for IC memory card interface and the lower 32 Mbytes are used for I/O card interface.

When the PCMCIA interface is used, the bus size must be specified as 8 bits or 16 bits using the bits BSZ[1:0] in CS5BCR or CS6BCR.

Figure 9.38 shows an example of connection between this LSI and a PCMCIA card. To enable hot swapping (insertion and removal of the PCMCIA card with the system power turned on), tri-state buffers must be connected between the bus interface of this LSI and the PCMCIA card.

In the JEIDA and PCMCIA standards, operation in big endian mode is not clearly defined. Consequently, the provided PCMCIA interface in big endian mode is available only for this LSI.

chanhvinh.com

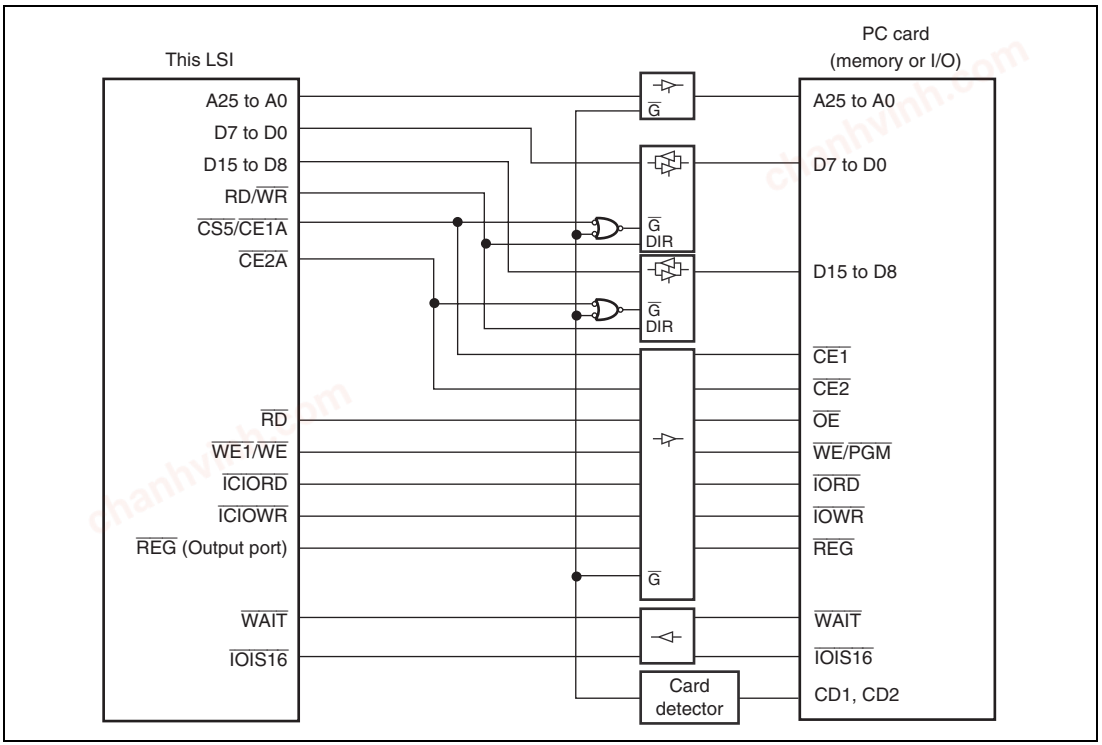


Figure 9.38 Example of PCMCIA Interface Connection

chanhvinh.com

(1) Basic Timing for Memory Card Interface

Figure 9.39 shows the basic timing of the PCMCIA IC memory card interface. When areas 5 and 6 are specified as the PCMCIA interface, the bus is accessed with the IC memory card interface according to the SA[1:0] bit settings in CS5WCR and CS6WCR. If the external bus frequency (CKIO) increases, the setup times and hold times for the address pins (A25 to A0), card enable signals ($\overline{CE1A}$, $\overline{CE2A}$, $\overline{CE1B}$, $\overline{CE2B}$), and write data (D15 to D0) to the $\overline{RD}$ and $\overline{WE}$ signals become insufficient. To prevent this error, this LSI enables the setup times and hold times for areas 5 and 6 to be specified independently, using CS5WCR and CS6WCR. In the PCMCIA interface, as in the normal space interface, a software wait or hardware wait using the $\overline{WAIT}$ pin can be inserted. Figure 9.40 shows the PCMCIA memory bus wait timing.

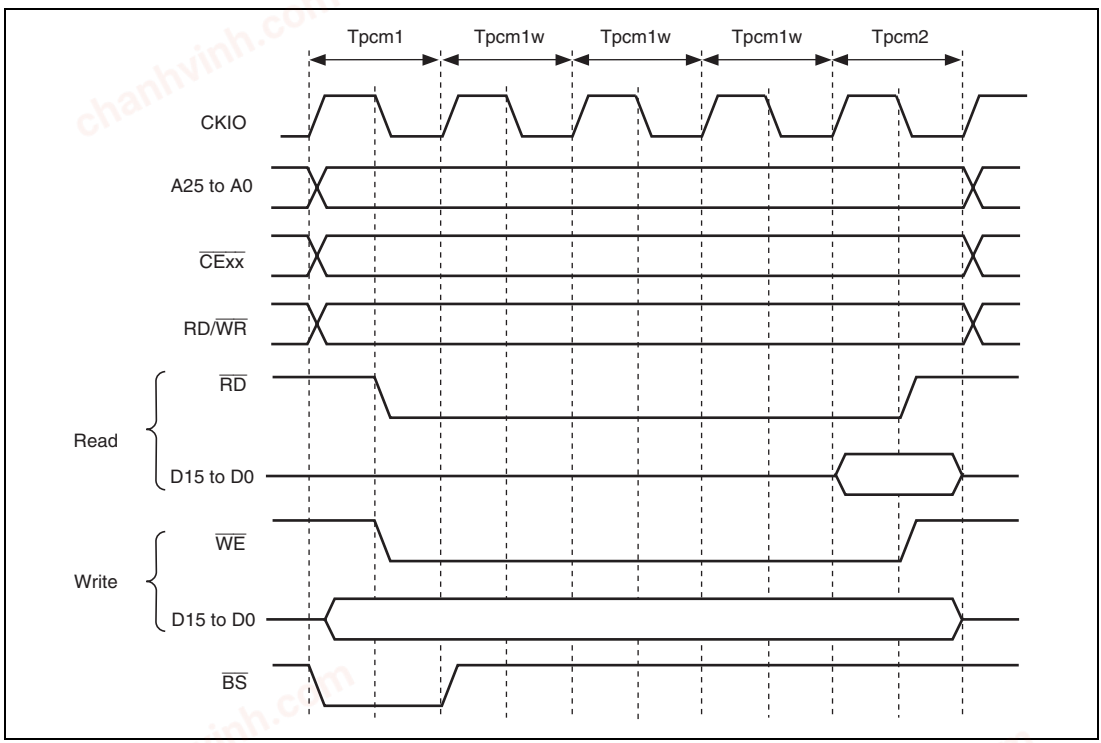


Figure 9.39 Basic Access Timing for PCMCIA Memory Card Interface

chanhvinh.com

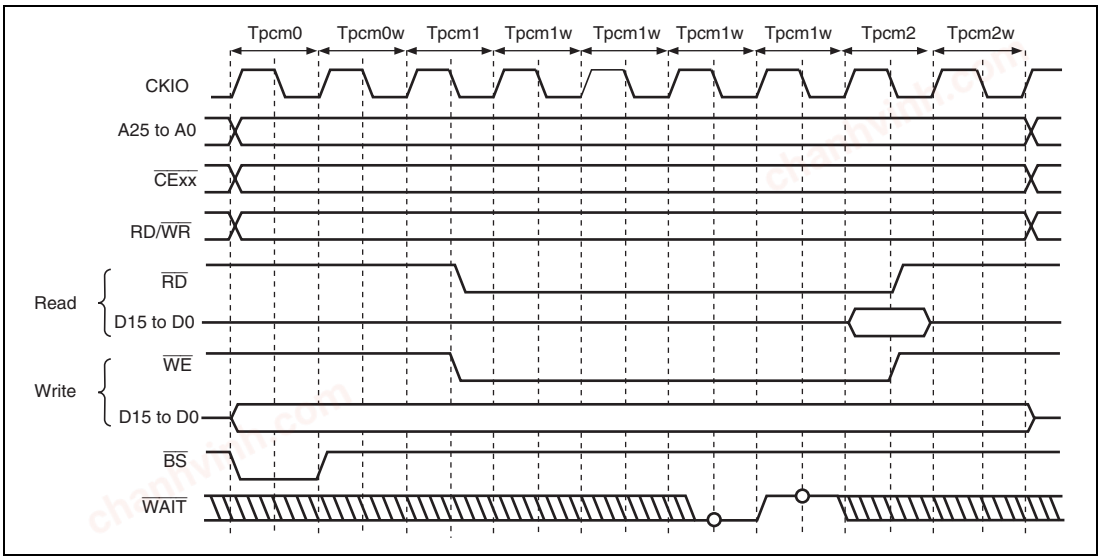


Figure 9.40 Wait Timing for PCMCIA Memory Card Interface (TED[3:0] = B'0010, PCW[3:0] = B'0000, TEH[3:0] = B'0001, Hardware Wait = 1)

A port is used to generate the $\overline{\text{REG}}$ signal that switches between the common memory and attribute memory. As shown in the example in figure 9.41, when the total memory space necessary for the common memory and attribute memory is 32 Mbytes or less, pin A24 can be used as the $\overline{\text{REG}}$ signal to allocate a 16-Mbyte common memory space and a 16-Mbyte attribute memory space.

chanhvinh.com

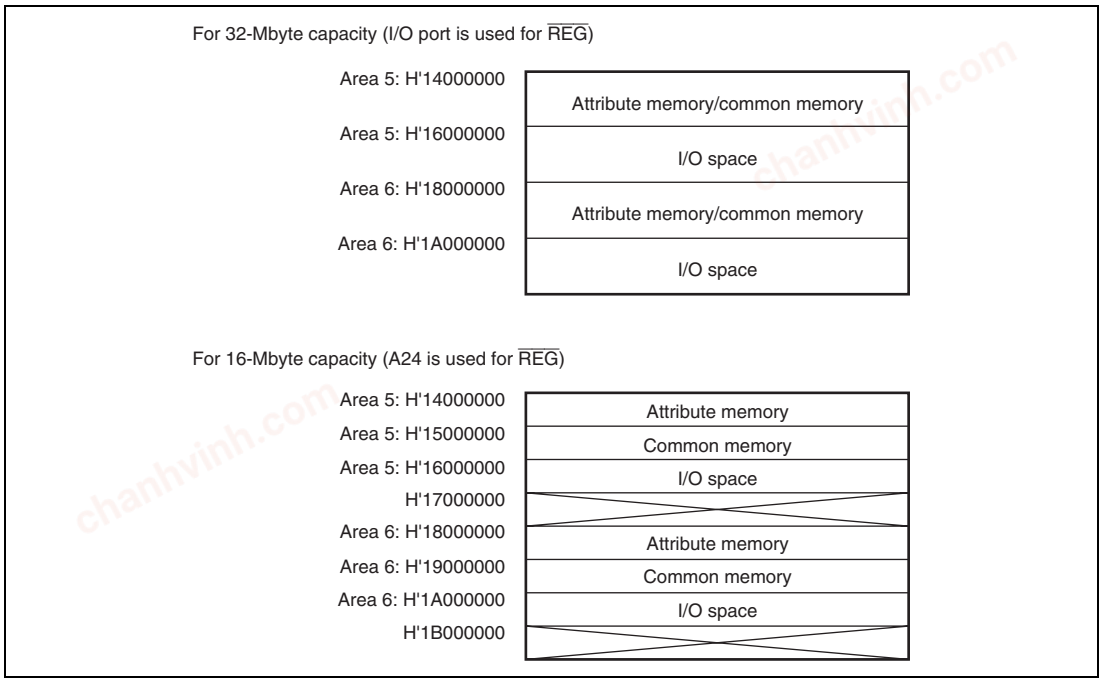


Figure 9.41 Example of PCMCIA Space Allocation (CS5WCR.SA[1:0] = B'10, CS6WCR.SA[1:0] = B'10)

chanhvinh.com

(2) Basic Timing for I/O Card Interface

Figures 9.42 and 9.43 show the basic timing for the PCMCIA I/O card interface.

When accessing an I/O card through the PCMCIA interface, be sure to access the space as cache-disabled.

Switching between I/O card and IC memory card interfaces in the respective address spaces is accomplished by the SA[1:0] bit settings in CS5WCR and CS6WCR.

The $\overline{\text{IOIS16}}$ pin can be used for dynamic adjustment of the width of the I/O bus in access to an I/O card via the PCMCIA interface when little endian mode has been selected. When the bus width of area 5 or 6 is set to 16 bits and the $\overline{\text{IOIS16}}$ signal is driven high during a cycle of word-unit access to the I/O card bus, the bus width will be recognized as 8 bits and only 8 bits of data will be accessed during the current cycle of the I/O card bus. Operation will automatically continue with access to the remaining 8 bits of data.

The $\overline{\text{IOIS16}}$ signal is sampled on falling edges of the CKIO in Tpci0 as well as all Tpci0w cycles for which the TED3 to TED0 bits are set to 1.5 cycles or more, and the $\overline{\text{CE2A}}$ and $\overline{\text{CE2B}}$ signals are updated after 1.5 cycles of the CKIO signal from the sampling point of Tpci0. Ensure that the $\overline{\text{IOIS16}}$ signal is defined at all sampling points and does not change along the way.

Set the TED3 to TED0 bits to satisfy the requirement of the PC card in use with regard to setup timing from $\overline{\text{ICIORD}}$ or $\overline{\text{ICIOWR}}$ to $\overline{\text{CE1}}$ or $\overline{\text{CE2}}$.

The basic waveforms for dynamic bus-size adjustment are shown in figure 9.43.

Since the $\overline{\text{IOIS16}}$ signal is not supported in big endian mode, the $\overline{\text{IOIS16}}$ signal should be fixed to the low level when big endian mode has been selected.

chanhvinh.com

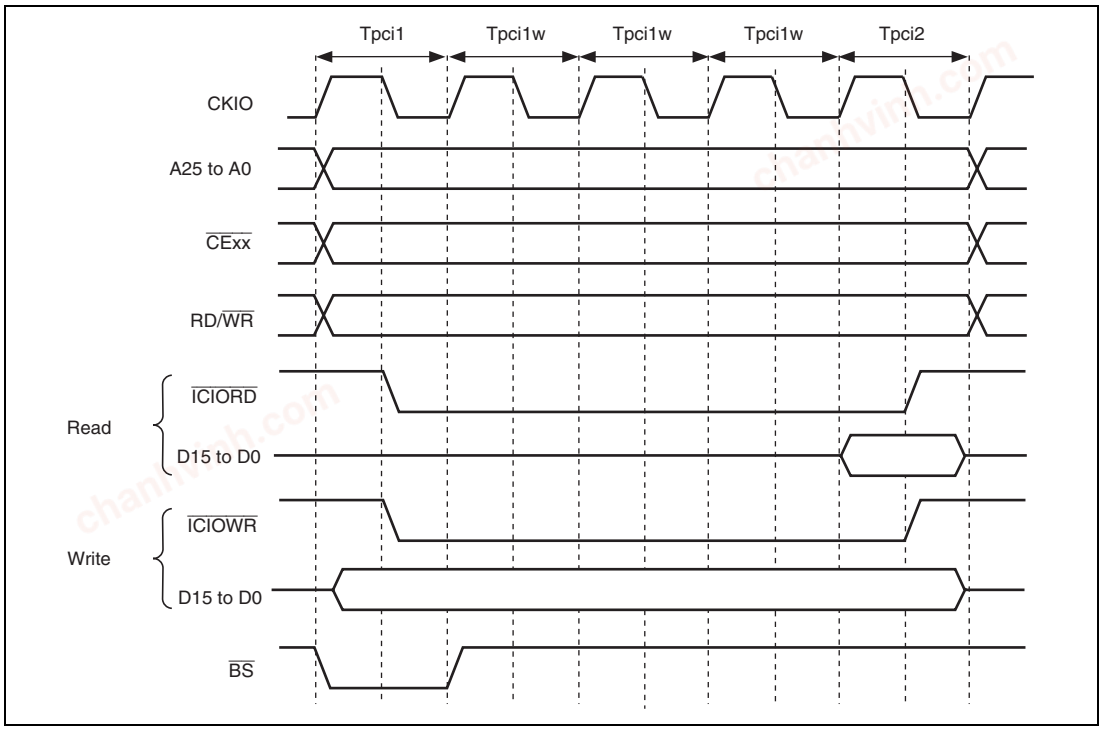


Figure 9.42 Basic Access Timing for PCMCIA I/O Card Interface

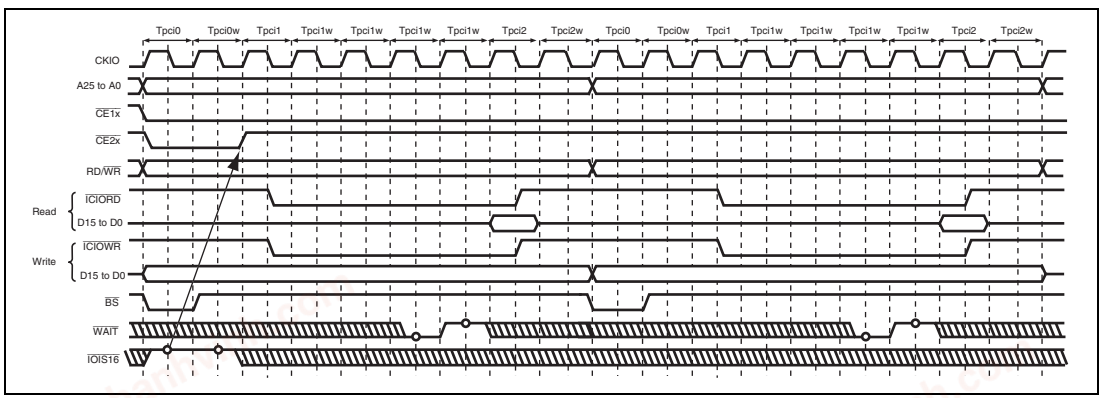


Figure 9.43 Dynamic Bus-Size Adjustment Timing for PCMCIA I/O Card Interface (TED[3:0] = B'0010, PCW[3:0] = B'0000, TEH[3:0] = B'0001, Hardware Wait = 1)

chanhvinh.com

9.5.10 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.

Since the bus width is 16 bits, the burst length must be specified as 8. 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 longword 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 16-byte read by cache fill in the cache-enabled spaces or 16-byte read by the DMA. The burst ROM interface performs write access in the same way as normal space access.

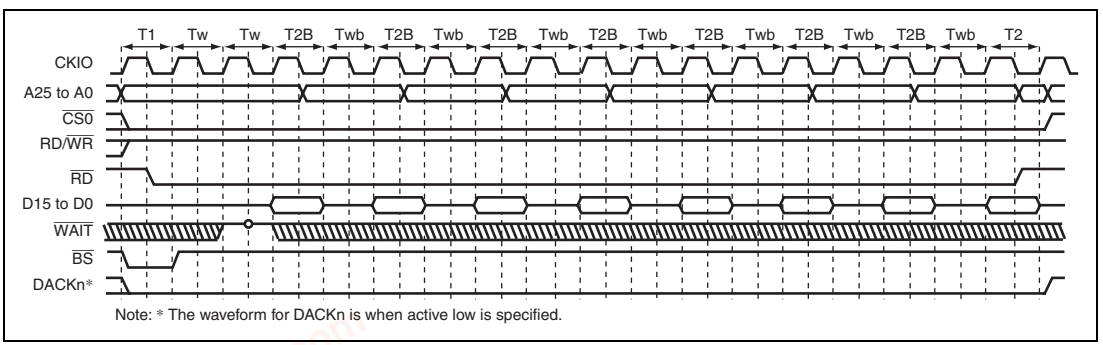


Figure 9.44 Burst ROM Access Timing (Clocked Synchronous) (Burst Length = 8, Wait Cycles Inserted in First Access = 2, Wait Cycles Inserted in Second and Subsequent Access Cycles = 1)

chanhvinh.com

9.5.11 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, and bits DMAIW2 to DMAIW0 and DMAIWA in CMNCR. 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
6. Data output from an external device caused by DMA single address transfer is followed by data output from another device that includes this LSI (DMAIWA = 0)
7. Data output from an external device caused by DMA single address transfer is followed by any type of access (DMAIWA = 1)

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. Here, $\overline{\text{CSn}}$ and $\overline{\text{CSm}}$ also include $\overline{\text{CE2A}}$ and $\overline{\text{CE2B}}$ for PCMCIA.

There are eight conditions that determine the number of idle cycles on the external bus as shown in table 9.16. The effects of these conditions are shown in figure 9.45.

chanhvinh.com

Table 9.16 Conditions for Determining Number of Idle Cycles

No.	Condition	Description	Range	Note
[1]	DMAIW[2:0] in CMNCR	These bits specify the number of idle cycles for DMA single address transfer. This condition is effective only for single address transfer and generates idle cycles after the access is completed.	0 to 12	When 0 is specified for the number of idle cycles, the DACK signal may be asserted continuously. This causes a discrepancy between the number of cycles detected by the device with DACK and the direct memory access controller transfer count, resulting in a malfunction.
[2]	IW***[2:0] in CSnBCR	These bits specify the number of idle cycles for access other than single address transfer. 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.
[3]	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.

chanhvinh.com

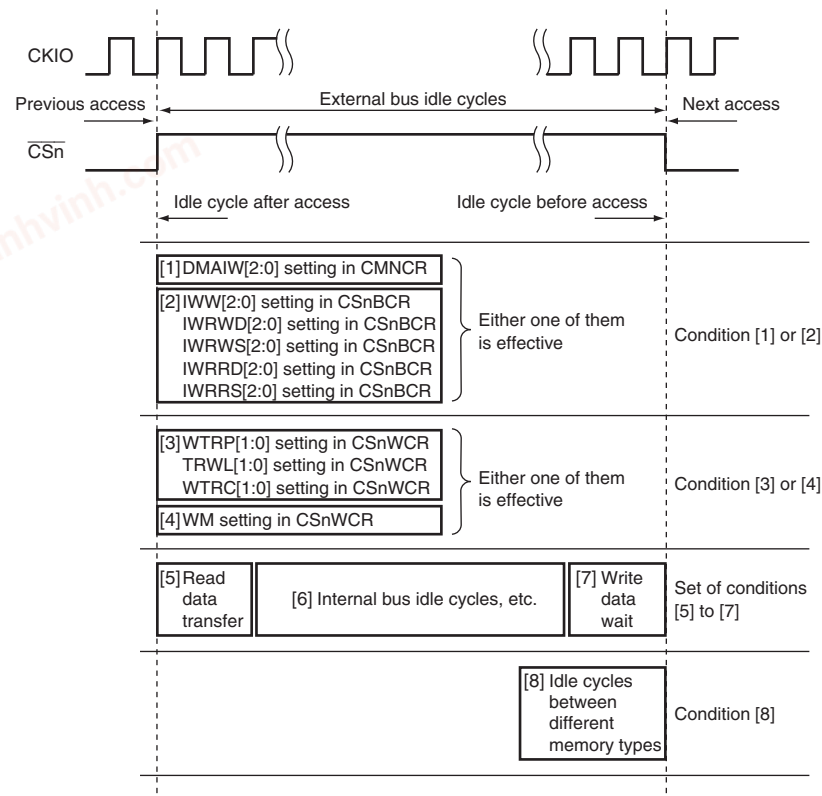
No.	Condition	Description	Range	Note
[4]	WM in CSnWCR	This bit enables or disables external $\overline{\text{WAIT}}$ pin input for the memory types other than SDRAM. When this bit is cleared to 0 (external $\overline{\text{WAIT}}$ enabled), one idle cycle is inserted to check the external $\overline{\text{WAIT}}$ pin input after the access is completed. When this bit is set to 1 (disabled), no idle cycle is generated.	0 or 1	
[5]	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 or PCMCIA interface.
[6]	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 Iϕ:Bϕ clock ratio. Tables 9.17 and 9.18 show the relationship between the clock ratio and the minimum number of internal bus idle cycles.

chanhvinh.com

No.	Condition	Description	Range	Note
[7]	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 are frequently 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.
[8]	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 2.5	The number of idle cycles depends on the target memory types. See table 9.19.

chanhvinh.com

In the above conditions, a total of four conditions, that is, condition [1] or [2] (either one is effective), condition [3] or [4] (either one is effective), a set of conditions [5] to [7] (these are generated successively, and therefore the sum of them should be taken as one set of idle cycles), and condition [8] 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] or [2].



Note: A total of four conditions (condition [1] or [2], condition [3] or [4], a set of conditions [5] to [7], and condition [8]) generate idle cycle at the same time. Accordingly, the maximum number of cycles among these four conditions become the number of idle cycles.

Figure 9.45 Idle Cycle Conditions

chanhvinh.com

Table 9.17 Minimum Number of Idle Cycles on Internal Bus (CPU Operation)

	Clock Ratio (Iϕ:Bϕ)					
CPU Operation	**8:1**	**6:1**	**4:1**	**3:1**	**2:1**	**1:1**
Write → write	1	1	2	2	2	3
Write → read	0	0	0	0	0	1
Read → write	1	1	2	2	2	3
Read → read	0	0	0	0	0	1

Table 9.18 Minimum Number of Idle Cycles on Internal Bus (Direct Memory Access Controller Operation)

Direct Memory Access Controller Operation	**Transfer Mode**	
	Dual Address	**Single Address**
Write → write	0	2
Write → read	0 or 2	0
Read → write	0	0
Read → read	0	2

Notes: 1. The write → write and read → read columns in dual address transfer indicate the cycles in the divided access cycles.
2. For the write → read cycles in dual address transfer, 0 means different channels are activated successively and 2 means when the same channel is activated successively.
3. The write → read and read → write columns in single address transfer indicate the case when different channels are activated successively. The "write" means transfer from a device with DACK to external memory and the "read" means transfer from external memory to a device with DACK.

chanhvinh.com

Table 9.19 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**	**SDRAM (Low-Frequency Mode)**	**PCMCIA**	**Burst ROM (Synchronous)**
SRAM	0	0	1	0	1	1	1.5	0	0
Burst ROM (asynchronous)	0	0	1	0	1	1	1.5	0	0
MPX-I/O	1	1	0	1	1	1	1.5	1	1
Byte SRAM (BAS = 0)	0	0	1	0	1	1	1.5	0	0
Byte SRAM (BAS = 1)	1	1	2	1	0	0	1.5	1	1
SDRAM	1	1	2	1	0	0	—	1	1
SDRAM (low-frequency mode)	1.5	1.5	2.5	1.5	0.5	—	1	1.5	1.5
PCMCIA	0	0	1	0	1	1	1.5	0	0
Burst ROM (synchronous)	0	0	1	0	1	1	1.5	0	0

Figure 9.46 shows sample estimation of idle cycles between access cycles. In the actual operation, the idle cycles may become shorter than the estimated value due to the write buffer effect or may become longer due to internal bus idle cycles caused by stalling in the pipeline due to CPU instruction execution or CPU register conflicts. Please consider these errors when estimating the idle cycles.

chanhvinh.com

Sample Estimation of Idle Cycles between Access Cycles

This example estimates the idle cycles for data transfer from the CS1 space to CS2 space by CPU access. Transfer is repeated in the following order: CS1 read → CS1 read → CS2 write → CS2 write → CS1 read → ...

- Conditions

  The bits for setting the idle cycles between access cycles in CS1BCR and CS2BCR are all set to 0.
  In CS1WCR and CS2WCR, the WM bit is set to 1 (external $\overline{\text{WAIT}}$ pin disabled) and the HW[1:0] bits are set to 00 (CS negation is not extended).
  Iϕ:Bϕ is set to 4:1, and no other processing is done during transfer.
  For both the CS1 and CS2 spaces, normal SRAM devices are connected, the bus width is 32 bits, and access size is also 32 bits.

The idle cycles generated under each condition are estimated for each pair of access cycles. In the following table, R indicates a read cycle and W indicates a write cycle.

Condition	R → R	R → W	W → W	W → R	Note
[1] or [2]	0	0	0	0	CSnBCR is set to 0.
[3] or [4]	0	0	0	0	The WM bit is set to 1.
[5]	1	1	0	0	Generated after a read cycle.
[6]	0	2	2	0	See the Iϕ:Bϕ = 4:1 columns in table 9.17.
[7]	0	1	0	0	No idle cycle is generated for the second time due to the write buffer effect.
[5] + [6] + [7]	1	4	2	0	
[8]	0	0	0	0	Value for SRAM → SRAM access
Estimated idle cycles	1	4	2	0	Maximum value among conditions [1] or [2], [3] or [4], [5] + [6] + [7], and [8]
Actual idle cycles	1	4	2	1	The estimated value does not match the actual value in the W → R cycles because the internal idle cycles due to condition [6] is estimated as 0 but actually an internal idle cycle is generated due to execution of a loop condition check instruction.

Figure 9.46 Comparison between Estimated Idle Cycles and Actual Value

chanhvinh.com

9.5.12 Bus Arbitration

The bus arbitration of this LSI has the bus mastership in the normal state and releases the bus mastership after receiving a bus request from another device.

Bus mastership is transferred at the boundary of bus cycles. Namely, bus mastership is released immediately after receiving a bus request when a bus cycle is not being performed. The release of bus mastership is delayed until the bus cycle is complete when a bus cycle is in progress. Even when from outside the LSI it looks like a bus cycle is not being performed, a bus cycle may be performing internally, started by inserting wait cycles between access cycles. Therefore, it cannot be immediately determined whether or not bus mastership has been released by looking at the $\overline{\text{CS}}$n signal or other bus control signals. The states that do not allow bus mastership release are shown below.

1. 16-byte transfer because of a cache miss
2. During write-back operation for the cache
3. Between the read and write cycles of a TAS instruction
4. Multiple bus cycles generated when the data bus width is smaller than the access size (for example, between bus cycles when longword access is made to a memory with a data bus width of 8 bits)
5. 16-byte transfer by the direct memory access controller
6. Setting the BLOCK bit in CMNCR to 1
7. During access to the external flash memory by the NAND flash memory controller

Moreover, by using DPRTY bit in CMNCR, whether the bus mastership request is received or not can be selected during burst transfer by the direct memory access controller.

The LSI has the bus mastership until a bus request is received from another device. Upon acknowledging the assertion (low level) of the external bus request signal $\overline{\text{BREQ}}$, the LSI releases the bus at the completion of the current bus cycle and asserts the $\overline{\text{BACK}}$ signal. After the LSI acknowledges the negation (high level) of the $\overline{\text{BREQ}}$ signal that indicates the external device has released the bus, it negates the $\overline{\text{BACK}}$ signal and resumes the bus usage.

With the SDRAM interface, all bank pre-charge commands (PALLs) are issued when active banks exist and the bus is released after completion of a PALL command.

chanhvinh.com

The bus sequence is as follows. The address bus and data bus are placed in a high-impedance state synchronized with the rising edge of CKIO. The bus mastership enable signal is asserted 0.5 cycles after the above timing, synchronized with the falling edge of CKIO. The bus control signals ($\overline{BS}$, $\overline{CSn}$, $\overline{RAS}$, $\overline{CAS}$, CKE, DQMx, $\overline{WEn}$, $\overline{RD}$, and RD/$\overline{WR}$) are placed in the high-impedance state at subsequent rising edges of CKIO. These bus control signals are driven high one or more cycles before they are placed in the high-impedance state. Bus request signals are sampled at the falling edge of CKIO. Note that CKE, $\overline{RAS}$, and $\overline{CAS}$ can be continued to be driven at the previous value even in the bus-released state by setting the HIZCNT bit in CMNCR.

The sequence for reclaiming the bus mastership from an external device is described below. 1.5 cycles after the negation of $\overline{BREQ}$ is detected at the falling edge of CKIO, the bus control signals are driven high. The bus acknowledge signal is negated at the next falling edge of the clock. The fastest timing at which actual bus cycles can be resumed after bus control signal assertion is at the rising edge of the CKIO where address and data signals are driven. Figure 9.47 shows the bus arbitration timing.

When it is necessary to refresh SDRAM while releasing the bus mastership, the bus mastership should be returned. If the bus mastership is not returned for a specified refreshing period or longer, the contents of SDRAM cannot be guaranteed because a refreshing cannot be executed.

While releasing the bus mastership, the SLEEP instruction (to enter sleep mode, deep standby mode, or software standby mode), as well as a manual reset, cannot be executed until the LSI obtains the bus mastership.

The $\overline{BREQ}$ input signal is ignored in software standby mode or deep standby mode and the $\overline{BACK}$ output signal is placed in the high impedance state. If the bus mastership request is required in this state, the bus mastership must be released by pulling down the $\overline{BACK}$ pin to enter software standby mode or deep standby mode.

The bus mastership release ($\overline{BREQ}$ signal for high level negation) after the bus mastership request ($\overline{BREQ}$ signal for low level assertion) must be performed after the bus usage permission ($\overline{BACK}$ signal for low level assertion). If the $\overline{BREQ}$ signal is negated before the $\overline{BACK}$ signal is asserted, only one cycle of the $\overline{BACK}$ signal is asserted depending on the timing of the $\overline{BREQ}$ signal to be negated and this may cause a bus contention between the external device and the LSI.

chanhvinh.com

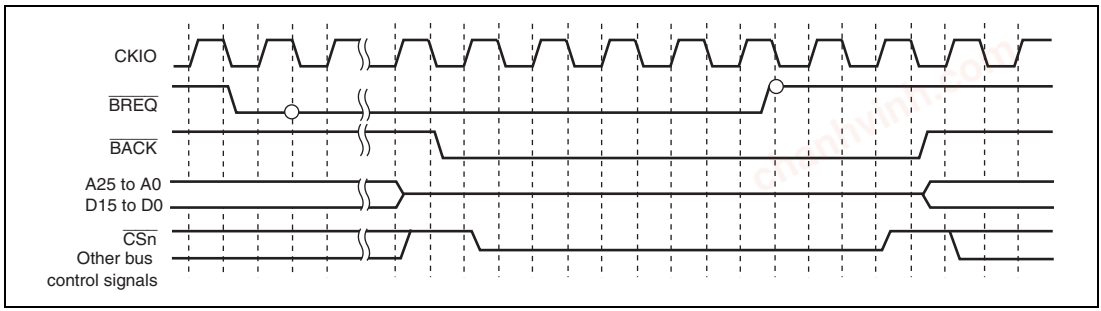


Figure 9.47 Bus Arbitration Timing

9.5.13 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, sleep, and manual reset, control registers of the bus state controller are not initialized. At manual reset, only the current bus cycle being executed is completed. Since the RTCNT continues counting up during manual reset signal assertion, a refresh request occurs to initiate the refresh cycle.

(2) Access from the Side of the LSI Internal Bus Master

There are three types of LSI internal buses: a CPU bus, internal bus, and peripheral bus. The CPU and cache memory are connected to the CPU bus. The bus state controller and internal bus masters other than the CPU are connected to the internal bus. Low-speed peripheral modules are connected to the peripheral bus. Internal memories other than the cache memory are connected bidirectionally to the CPU bus and internal bus. Access from the CPU bus to the internal bus is enabled but access from the internal bus to the CPU bus is disabled. This gives rise to the following problems.

On-chip bus masters such as the direct memory access controller other than the CPU can access internal memory other than the cache memory but cannot access the cache memory. If an on-chip bus master other than the CPU writes data to an external memory other than the cache, the contents of the external memory may differ from that of the cache memory. To prevent this problem, if the external memory whose contents is cached is written by an on-chip bus master other than the CPU, the corresponding cache memory should be purged by software.

In a cache-enabled space, if the CPU initiates read access, the cache is searched. If the cache stores data, the CPU latches the data and completes the read access. If the cache does not store data, the

chanhvinh.com

CPU performs four contiguous longword read cycles to perform cache fill operations via the internal bus. If a cache miss occurs in byte or word operand access or at a branch to an odd word boundary (4n + 2), the CPU performs four contiguous longword access cycles to perform a cache fill operation on the external interface. For a cache-disabled space, the CPU performs access according to the actual access addresses. For an instruction fetch to an even word boundary (4n), the CPU performs longword access. For an instruction fetch to an odd word boundary (4n + 2), the CPU performs word access.

For a read cycle of an on-chip peripheral module, the cycle is initiated through the internal bus and peripheral bus. The read data is sent to the CPU via the peripheral bus, internal bus, and CPU bus.

In a write cycle for the cache-enabled space, the write cycle operation differs according to the cache write methods.

In write-back mode, the cache is first searched. If data is detected at the address corresponding to the cache, the data is then re-written to the cache. In the actual memory, data will not be re-written until data in the corresponding address is re-written. If data is not detected at the address corresponding to the cache, the cache is modified. In this case, data to be modified is first saved to the internal buffer, 16-byte data including the data corresponding to the address is then read, and data in the corresponding access of the cache is finally modified. Following these operations, a write-back cycle for the saved 16-byte data is executed.

In write-through mode, the cache is first searched. If data is detected at the address corresponding to the cache, the data is re-written to the cache simultaneously with the actual write via the internal bus. If data is not detected at the address corresponding to the cache, the cache is not modified but an actual write is performed via the internal bus.

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.

chanhvinh.com

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 dual address 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 (Pϕ) cycles are required. Care must be taken in system design.

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 SLEEP instruction must be performed after setting the STBY bit in the STBCR register to 1. However a dummy read of the STBCR register is required before executing the SLEEP instruction. If a dummy read is omitted, the CPU executes the SLEEP instruction before the STBY bit is set to 1, thus the system enters sleep mode not software standby mode. A dummy read of the STBCR 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.

(4) External Flash Memory Access by NAND Flash Memory Controller

In this product, a part of the external data bus is used also as data bus for the NAND flash memory controller. The use of the data bus is controlled by the NAND flash memory controller. Memory access by the NAND flash memory controller is started after the preceding access to the external device by this module is completed. If an access to the external device by this module occurs during the access by the NAND flash memory controller, it must wait until the completion of the access by the NAND flash memory controller.

chanhvinh.com

When a memory access request by the NAND flash memory controller and an external bus release request conflict with each other, the request accepted first has higher priority. When the two requests occur at the same time, the access by the NAND flash memory controller has higher priority.

Auto-refresh operation and self-refresh operation are executed even during a memory access by the NAND flash memory controller.

chanhvinh.com

Section 10 Direct Memory Access Controller

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.

10.1 Features

- Number of channels: 16 channels (channels 0 to 15) selectable
  Two channels (channels 0 and 1*) can receive external requests.
- 4-Gbyte physical address space
- Data transfer unit is selectable: Byte, word (two bytes), longword (four bytes), and 16 bytes (longword × 4)
- Maximum transfer count: 16,777,216 transfers (24 bits)
- Address mode: Dual address mode and single address mode are supported.
- Transfer requests
  — External request
  — On-chip peripheral module request
  — Auto request
  The following modules can issue on-chip peripheral module requests.
  — Serial communication interface with FIFO: 16 sources
  — I^2C bus interface 3: six sources
  — A/D converter: one source
  — Multi-function timer pulse unit 2: five sources
  — Compare match timer: two sources
  — USB 2.0 host/function module: two sources
  — NAND flash memory controller: two sources
  — Controller area network: two sources
  — Serial sound interface: five sources
  — Sampling rate converter: four sources
  — Renesas SPDIF interface: two sources
  — CD-ROM decoder: one source
  — SD host interface: two sources
  — Renesas serial peripheral interface: four sources
  — Clock synchronous serial I/O with FIFO: two sources
  — Motor control PWM timer: two sources

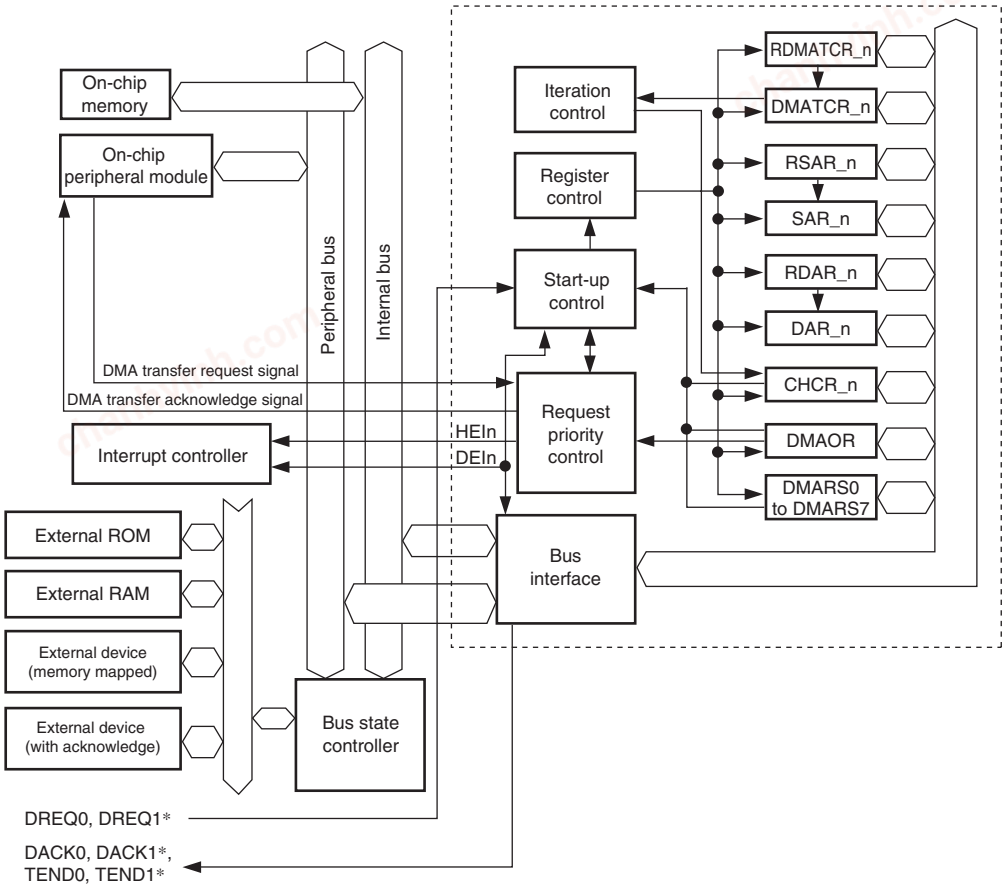
chanhvinh.com

  - Decompression unit: two sources
- Selectable bus modes
  - Cycle steal mode (normal mode or intermittent mode)
  - Burst mode
- Selectable channel priority levels: The channel priority levels are selectable between two fixed modes.
- Interrupt request: An interrupt request can be sent to the CPU on completion of half- or full-data transfer. Through the HE and HIE bits in CHCR, an interrupt is specified to be issued to the CPU when half of the initially specified DMA transfer is completed.
- External request detection: There are following four types of DREQ input detection.
  - Low level detection
  - High level detection
  - Rising edge detection
  - Falling edge detection
- Transfer request acknowledge and transfer end signals: Active levels for DACK and TEND can be set independently.
- Support of reload functions in DMA transfer information registers: DMA transfer using the same information as the current transfer can be repeated automatically without specifying the information again. Modifying the reload registers during DMA transfer enables next DMA transfer to be done using different transfer information. The reload function can be enabled or disabled independently in each channel or reload register.

Note: * Channel 1 can receive external requests only in the SH7264 Group.

chanhvinh.com

Figure 10.1 shows the block diagram of this module.



[Legend]

RDMATCR:	DMA reload transfer count register	CHCR:	DMA channel control register
DMATCR:	DMA transfer count register	DMAOR:	DMA operation register
RSAR:	DMA reload source address register	DMARS0 to DMARS7:	DMA extension resource selectors 0 to 7
SAR:	DMA source address register	HEIn:	DMA transfer half-end interrupt request to the CPU
RDAR:	DMA reload destination address register	DEIn:	DMA transfer end interrupt request to the CPU
DAR:	DMA destination address register	n = 0 to 15	

Note: * Pins in channel 1 can be used only in the SH7264 Group.

Figure 10.1 Block Diagram

chanhvinh.com

10.2 Input/Output Pins

Table 10.1 lists the pin configuration of this module. This module has pins for two channels (channels 0 and 1*) for external bus use.

Table 10.1 Pin Configuration

Channel	Name	Abbreviation	I/O	Function
0	DMA transfer request	DREQ0	I	DMA transfer request input from an external device to channel 0
	DMA transfer request acknowledge	DACK0	O	DMA transfer request acknowledge output from channel 0 to an external device
	DMA transfer end	TEND0	O	DMA transfer end output for channel 0
1	DMA transfer request*	DREQ1	I	DMA transfer request input from an external device to channel 1
	DMA transfer request acknowledge*	DACK1	O	DMA transfer request acknowledge output from channel 1 to an external device
	DMA transfer end*	TEND1	O	DMA transfer end output for channel 1

Note: * Pins in channel 1 can be used only in the SH7264 Group.

chanhvinh.com

10.3 Register Descriptions

This module has the registers listed in table 10.2. There are four control registers and three reload registers for each channel, and one common control register is 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 SAR_0 for SAR in channel 0.

Table 10.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
0	DMA source address register_0	SAR_0	R/W	H'00000000	H'FFFE1000	16, 32
	DMA destination address register_0	DAR_0	R/W	H'00000000	H'FFFE1004	16, 32
	DMA transfer count register_0	DMATCR_0	R/W	H'00000000	H'FFFE1008	16, 32
	DMA channel control register_0	CHCR_0	R/W*[1]	H'00000000	H'FFFE100C	8, 16, 32
	DMA reload source address register_0	RSAR_0	R/W	H'00000000	H'FFFE1100	16, 32
	DMA reload destination address register_0	RDAR_0	R/W	H'00000000	H'FFFE1104	16, 32
	DMA reload transfer count register_0	RDMATCR_0	R/W	H'00000000	H'FFFE1108	16, 32
1	DMA source address register_1	SAR_1	R/W	H'00000000	H'FFFE1010	16, 32
	DMA destination address register_1	DAR_1	R/W	H'00000000	H'FFFE1014	16, 32
	DMA transfer count register_1	DMATCR_1	R/W	H'00000000	H'FFFE1018	16, 32
	DMA channel control register_1	CHCR_1	R/W*[1]	H'00000000	H'FFFE101C	8, 16, 32
	DMA reload source address register_1	RSAR_1	R/W	H'00000000	H'FFFE1110	16, 32
	DMA reload destination address register_1	RDAR_1	R/W	H'00000000	H'FFFE1114	16, 32
	DMA reload transfer count register_1	RDMATCR_1	R/W	H'00000000	H'FFFE1118	16, 32

chanhvinh.com

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
2	DMA source address register_2	SAR_2	R/W	H'00000000	H'FFFE1020	16, 32
	DMA destination address register_2	DAR_2	R/W	H'00000000	H'FFFE1024	16, 32
	DMA transfer count register_2	DMATCR_2	R/W	H'00000000	H'FFFE1028	16, 32
	DMA channel control register_2	CHCR_2	R/W*[1]	H'00000000	H'FFFE102C	8, 16, 32
	DMA reload source address register_2	RSAR_2	R/W	H'00000000	H'FFFE1120	16, 32
	DMA reload destination address register_2	RDAR_2	R/W	H'00000000	H'FFFE1124	16, 32
	DMA reload transfer count register_2	RDMATCR_2	R/W	H'00000000	H'FFFE1128	16, 32
3	DMA source address register_3	SAR_3	R/W	H'00000000	H'FFFE1030	16, 32
	DMA destination address register_3	DAR_3	R/W	H'00000000	H'FFFE1034	16, 32
	DMA transfer count register_3	DMATCR_3	R/W	H'00000000	H'FFFE1038	16, 32
	DMA channel control register_3	CHCR_3	R/W*[1]	H'00000000	H'FFFE103C	8, 16, 32
	DMA reload source address register_3	RSAR_3	R/W	H'00000000	H'FFFE1130	16, 32
	DMA reload destination address register_3	RDAR_3	R/W	H'00000000	H'FFFE1134	16, 32
	DMA reload transfer count register_3	RDMATCR_3	R/W	H'00000000	H'FFFE1138	16, 32

chanhvinh.com

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
4	DMA source address register_4	SAR_4	R/W	H'00000000	H'FFFE1040	16, 32
	DMA destination address register_4	DAR_4	R/W	H'00000000	H'FFFE1044	16, 32
	DMA transfer count register_4	DMATCR_4	R/W	H'00000000	H'FFFE1048	16, 32
	DMA channel control register_4	CHCR_4	R/W*[1]	H'00000000	H'FFFE104C	8, 16, 32
	DMA reload source address register_4	RSAR_4	R/W	H'00000000	H'FFFE1140	16, 32
	DMA reload destination address register_4	RDAR_4	R/W	H'00000000	H'FFFE1144	16, 32
	DMA reload transfer count register_4	RDMATCR_4	R/W	H'00000000	H'FFFE1148	16, 32
5	DMA source address register_5	SAR_5	R/W	H'00000000	H'FFFE1050	16, 32
	DMA destination address register_5	DAR_5	R/W	H'00000000	H'FFFE1054	16, 32
	DMA transfer count register_5	DMATCR_5	R/W	H'00000000	H'FFFE1058	16, 32
	DMA channel control register_5	CHCR_5	R/W*[1]	H'00000000	H'FFFE105C	8, 16, 32
	DMA reload source address register_5	RSAR_5	R/W	H'00000000	H'FFFE1150	16, 32
	DMA reload destination address register_5	RDAR_5	R/W	H'00000000	H'FFFE1154	16, 32
	DMA reload transfer count register_5	RDMATCR_5	R/W	H'00000000	H'FFFE1158	16, 32

chanhvinh.com

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
6	DMA source address register_6	SAR_6	R/W	H'00000000	H'FFFE1060	16, 32
	DMA destination address register_6	DAR_6	R/W	H'00000000	H'FFFE1064	16, 32
	DMA transfer count register_6	DMATCR_6	R/W	H'00000000	H'FFFE1068	16, 32
	DMA channel control register_6	CHCR_6	R/W*[1]	H'00000000	H'FFFE106C	8, 16, 32
	DMA reload source address register_6	RSAR_6	R/W	H'00000000	H'FFFE1160	16, 32
	DMA reload destination address register_6	RDAR_6	R/W	H'00000000	H'FFFE1164	16, 32
	DMA reload transfer count register_6	RDMATCR_6	R/W	H'00000000	H'FFFE1168	16, 32
7	DMA source address register_7	SAR_7	R/W	H'00000000	H'FFFE1070	16, 32
	DMA destination address register_7	DAR_7	R/W	H'00000000	H'FFFE1074	16, 32
	DMA transfer count register_7	DMATCR_7	R/W	H'00000000	H'FFFE1078	16, 32
	DMA channel control register_7	CHCR_7	R/W*[1]	H'00000000	H'FFFE107C	8, 16, 32
	DMA reload source address register_7	RSAR_7	R/W	H'00000000	H'FFFE1170	16, 32
	DMA reload destination address register_7	RDAR_7	R/W	H'00000000	H'FFFE1174	16, 32
	DMA reload transfer count register_7	RDMATCR_7	R/W	H'00000000	H'FFFE1178	16, 32

chanhvinh.com

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
8	DMA source address register_8	SAR_8	R/W	H'00000000	H'FFFE1080	16, 32
	DMA destination address register_8	DAR_8	R/W	H'00000000	H'FFFE1084	16, 32
	DMA transfer count register_8	DMATCR_8	R/W	H'00000000	H'FFFE1088	16, 32
	DMA channel control register_8	CHCR_8	R/W*[1]	H'00000000	H'FFFE108C	8, 16, 32
	DMA reload source address register_8	RSAR_8	R/W	H'00000000	H'FFFE1180	16, 32
	DMA reload destination address register_8	RDAR_8	R/W	H'00000000	H'FFFE1184	16, 32
	DMA reload transfer count register_8	RDMATCR_8	R/W	H'00000000	H'FFFE1188	16, 32
9	DMA source address register_9	SAR_9	R/W	H'00000000	H'FFFE1090	16, 32
	DMA destination address register_9	DAR_9	R/W	H'00000000	H'FFFE1094	16, 32
	DMA transfer count register_9	DMATCR_9	R/W	H'00000000	H'FFFE1098	16, 32
	DMA channel control register_9	CHCR_9	R/W*[1]	H'00000000	H'FFFE109C	8, 16, 32
	DMA reload source address register_9	RSAR_9	R/W	H'00000000	H'FFFE1190	16, 32
	DMA reload destination address register_9	RDAR_9	R/W	H'00000000	H'FFFE1194	16, 32
	DMA reload transfer count register_9	RDMATCR_9	R/W	H'00000000	H'FFFE1198	16, 32

chanhvinh.com

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
10	DMA source address register_10	SAR_10	R/W	H'00000000	H'FFFE10A0	16, 32
	DMA destination address register_10	DAR_10	R/W	H'00000000	H'FFFE10A4	16, 32
	DMA transfer count register_10	DMATCR_10	R/W	H'00000000	H'FFFE10A8	16, 32
	DMA channel control register_10	CHCR_10	R/W*[1]	H'00000000	H'FFFE10AC	8, 16, 32
	DMA reload source address register_10	RSAR_10	R/W	H'00000000	H'FFFE11A0	16, 32
	DMA reload destination address register_10	RDAR_10	R/W	H'00000000	H'FFFE11A4	16, 32
	DMA reload transfer count register_10	RDMATCR_10	R/W	H'00000000	H'FFFE11A8	16, 32
11	DMA source address register_11	SAR_11	R/W	H'00000000	H'FFFE10B0	16, 32
	DMA destination address register_11	DAR_11	R/W	H'00000000	H'FFFE10B4	16, 32
	DMA transfer count register_11	DMATCR_11	R/W	H'00000000	H'FFFE10B8	16, 32
	DMA channel control register_11	CHCR_11	R/W*[1]	H'00000000	H'FFFE10BC	8, 16, 32
	DMA reload source address register_11	RSAR_11	R/W	H'00000000	H'FFFE11B0	16, 32
	DMA reload destination address register_11	RDAR_11	R/W	H'00000000	H'FFFE11B4	16, 32
	DMA reload transfer count register_11	RDMATCR_11	R/W	H'00000000	H'FFFE11B8	16, 32

chanhvinh.com

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
12	DMA source address register_12	SAR_12	R/W	H'00000000	H'FFFE10C0	16, 32
	DMA destination address register_12	DAR_12	R/W	H'00000000	H'FFFE10C4	16, 32
	DMA transfer count register_12	DMATCR_12	R/W	H'00000000	H'FFFE10C8	16, 32
	DMA channel control register_12	CHCR_12	R/W*[1]	H'00000000	H'FFFE10CC	8, 16, 32
	DMA reload source address register_12	RSAR_12	R/W	H'00000000	H'FFFE11C0	16, 32
	DMA reload destination address register_12	RDAR_12	R/W	H'00000000	H'FFFE11C4	16, 32
	DMA reload transfer count register_12	RDMATCR_12	R/W	H'00000000	H'FFFE11C8	16, 32
13	DMA source address register_13	SAR_13	R/W	H'00000000	H'FFFE10D0	16, 32
	DMA destination address register_13	DAR_13	R/W	H'00000000	H'FFFE10D4	16, 32
	DMA transfer count register_13	DMATCR_13	R/W	H'00000000	H'FFFE10D8	16, 32
	DMA channel control register_13	CHCR_13	R/W*[1]	H'00000000	H'FFFE10DC	8, 16, 32
	DMA reload source address register_13	RSAR_13	R/W	H'00000000	H'FFFE11D0	16, 32
	DMA reload destination address register_13	RDAR_13	R/W	H'00000000	H'FFFE11D4	16, 32
	DMA reload transfer count register_13	RDMATCR_13	R/W	H'00000000	H'FFFE11D8	16, 32

chanhvinh.com

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
14	DMA source address register_14	SAR_14	R/W	H'00000000	H'FFFE10E0	16, 32
	DMA destination address register_14	DAR_14	R/W	H'00000000	H'FFFE10E4	16, 32
	DMA transfer count register_14	DMATCR_14	R/W	H'00000000	H'FFFE10E8	16, 32
	DMA channel control register_14	CHCR_14	R/W*[1]	H'00000000	H'FFFE10EC	8, 16, 32
	DMA reload source address register_14	RSAR_14	R/W	H'00000000	H'FFFE11E0	16, 32
	DMA reload destination address register_14	RDAR_14	R/W	H'00000000	H'FFFE11E4	16, 32
	DMA reload transfer count register_14	RDMATCR_14	R/W	H'00000000	H'FFFE11E8	16, 32
15	DMA source address register_15	SAR_15	R/W	H'00000000	H'FFFE10F0	16, 32
	DMA destination address register_15	DAR_15	R/W	H'00000000	H'FFFE10F4	16, 32
	DMA transfer count register_15	DMATCR_15	R/W	H'00000000	H'FFFE10F8	16, 32
	DMA channel control register_15	CHCR_15	R/W*[1]	H'00000000	H'FFFE10FC	8, 16, 32
	DMA reload source address register_15	RSAR_15	R/W	H'00000000	H'FFFE11F0	16, 32
	DMA reload destination address register_15	RDAR_15	R/W	H'00000000	H'FFFE11F4	16, 32
	DMA reload transfer count register_15	RDMATCR_15	R/W	H'00000000	H'FFFE11F8	16, 32

chanhvinh.com

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Common	DMA operation register	DMAOR	R/W*[2]	H'0000	H'FFFE1200	8, 16
0 and 1	DMA extension resource selector 0	DMARS0	R/W	H'0000	H'FFFE1300	16
2 and 3	DMA extension resource selector 1	DMARS1	R/W	H'0000	H'FFFE1304	16
4 and 5	DMA extension resource selector 2	DMARS2	R/W	H'0000	H'FFFE1308	16
6 and 7	DMA extension resource selector 3	DMARS3	R/W	H'0000	H'FFFE130C	16
8 and 9	DMA extension resource selector 4	DMARS4	R/W	H'0000	H'FFFE1310	16
10 and 11	DMA extension resource selector 5	DMARS5	R/W	H'0000	H'FFFE1314	16
12 and 13	DMA extension resource selector 6	DMARS6	R/W	H'0000	H'FFFE1318	16
14 and 15	DMA extension resource selector 7	DMARS7	R/W	H'0000	H'FFFE131C	16

Notes: 1. For the HE and TE bits in CHCR_n, only 0 can be written to clear the flags after 1 is read.
2. For the AE and NMIF bits in DMAOR, only 0 can be written to clear the flags after 1 is read.

chanhvinh.com

10.3.1 DMA Source Address Registers (SAR)

The DMA source address registers (SAR) are 32-bit readable/writable registers that specify the source address of a DMA transfer. During a DMA transfer, these registers indicate the next source address. When the data of an external device with DACK is transferred in single address mode, SAR is ignored.

To transfer data in word (2-byte), longword (4-byte), or 16-byte unit, specify the address with 2-byte, 4-byte, or16-byte address boundary respectively.

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/W	R/W	R/W	R/W	R/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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

10.3.2 DMA Destination Address Registers (DAR)

The DMA destination address registers (DAR) are 32-bit readable/writable registers that specify the destination address of a DMA transfer. During a DMA transfer, these registers indicate the next destination address. When the data of an external device with DACK is transferred in single address mode, DAR is ignored.

To transfer data in word (2-byte), longword (4-byte), or 16-byte unit, specify the address with 2-byte, 4-byte, or 16-byte address boundary respectively.

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/W	R/W	R/W	R/W	R/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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

chanhvinh.com

10.3.3 DMA Transfer Count Registers (DMATCR)

The DMA transfer count registers (DMATCR) are 32-bit readable/writable registers that specify the number of DMA transfers. The transfer count is 1 when the setting is H'00000001, 16,777,215 when H'00FFFFFF is set, and 16,777,216 (the maximum) when H'00000000 is set. During a DMA transfer, these registers indicate the remaining transfer count.

The upper eight bits of DMATCR are always read as 0, and the write value should always be 0. To transfer data in 16 bytes, one 16-byte transfer (128 bits) counts one.

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
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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

10.3.4 DMA Channel Control Registers (CHCR)

The DMA channel control registers (CHCR) are 32-bit readable/writable registers that control the DMA transfer mode.

The DO, AM, AL, DL, DS, and TL bits which specify the DREQ, DACK, and TEND external pin functions can be read and written to in channels 0 and 1*[1], but they are reserved in channels 2 to 15*[2].

Notes: 1. Only channel 0 can be used in the SH7262 Group.
2. Channels 1 to 15 can be used in the SH7262 Group.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TC	-	RLD SAR	RLD DAR	-	DAF	SAF	-	DO	TL	-	TE MASK	HE	HIE	AM	AL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R/W	R/W	R	R/W	R/W	R	R/W	R/W	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
	DM[1:0]		SM[1:0]		RS[3:0]				DL	DS	TB	TS[1:0]		IE	TE	DE
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: * Only 0 can be written to clear the flag after 1 is read.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
31	TC	0	R/W	Transfer Count Mode Specifies whether to transmit data once or for the count specified in DMATCR by one transfer request. This function is valid only in on-chip peripheral module request mode. Note that when this bit is set to 0, the TB bit must not be set to 1 (burst mode). Do not set TC to 1 when a module other than multi-function timer pulse unit 2, compare-match timer, controller area network, CD-ROM decoder, or A/D converter is set as the transfer request source. 0: Transmits data once by one transfer request 1: Transmits data for the count specified in DMATCR by one transfer request
30	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
29	RLDSAR	0	R/W	SAR Reload Function ON/OFF Enables (ON) or disables (OFF) the function to reload SAR and DMATCR. 0: Disables (OFF) the function to reload SAR and DMATCR 1: Enables (ON) the function to reload SAR and DMATCR
28	RLDDAR	0	R/W	DAR Reload Function ON/OFF Enables (ON) or disables (OFF) the function to reload DAR and DMATCR. 0: Disables (OFF) the function to reload DAR and DMATCR 1: Enables (ON) the function to reload DAR and DMATCR
27	—	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
26	DAF	0	R/W	Fixed Destination Address 16-Byte Transfer Enabled when the transfer size (set in TS[1:0]) is 16 bytes and the destination address mode (set in DM[1:0]) is fixed address. 0: 16 bytes of data are transferred to the address set in the DAR register. The write destination address is the address set in the DAR register + H'0, + H'4, + H'8, or + H'C. 1: Four bytes of data are transferred four times to the address set in the DAR register. The write destination address is fixed at the address set in the DAR register. This function is exclusively for use with the CD-ROM decoder, USB 2.0 host/function module, and sampling rate converter.
25	SAF	0	R/W	Fixed Source Address 16-Byte Transfer Enabled when the transfer size (set in TS[1:0]) is 16 bytes and the source address mode (set in SM[1:0]) is fixed address. 0: 16 bytes of data are transferred from the address specified in SAR. The read address is the address set in the SAR register + H'0, + H'4, + H'8, or + H'C. 1: Four bytes of data are transferred four times from the address specified in SAR. The read address is fixed at the address set in the SAR register. This function is exclusively for use with the CD-ROM decoder, USB 2.0 host/function module, and sampling rate converter.
24	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
23	DO	0	R/W	DMA Overrun Selects whether DREQ is detected by overrun 0 or by overrun 1. This bit is valid only in level detection by CHCR_0 and CHCR_1*[1]. This bit is reserved in CHCR_2 to CHCR_15*[2]; it is always read as 0 and the write value should always be 0. 0: Detects DREQ by overrun 0 1: Detects DREQ by overrun 1

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
22	TL	0	R/W	Transfer End Level Specifies the TEND signal output is high active or low active. This bit is valid only in CHCR_0 and CHCR_1*[1]. This bit is reserved in CHCR_2 to CHCR_15*[2]; it is always read as 0 and the write value should always be 0. 0: Low-active output from TEND 1: High-active output from TEND
21	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
20	TEMASK	0	R/W	TE Set Mask Specifies that DMA transfer does not stop even if the TE bit is set to 1. If this bit is set to 1 along with the bit for SAR/DAR reload function, DMA transfer can be performed until the transfer request is cancelled. In auto request mode or when a rising/falling edge of the DREQ signal is detected in external request mode, the setting of this bit is ignored and DMA transfer stops if the TE bit is set to 1. Note that this function is enabled only when either the RLDSAR bit or the RLDDAR bit is set to 1. 0: DMA transfer stops if the TE bit is set 1: DMA transfer does not stop even if the TE bit is set

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
19	HE	0	R/(W) *3	Half-End Flag This bit is set to 1 when the transfer count reaches half of the DMATCR value that was specified before transfer starts. If DMA transfer ends because of an NMI interrupt, a DMA address error, or clearing of the DE bit or the DME bit in DMAOR before the transfer count reaches half of the initial DMATCR value, the HE bit is not set to 1. If DMA transfer ends due to an NMI interrupt, a DMA address error, or clearing of the DE bit or the DME bit in DMAOR after the HE bit is set to 1, the bit remains set to 1. To clear the HE bit, write 0 to it after HE = 1 is read.*4 0: DMATCR > (DMATCR set before transfer starts)/2 during DMA transfer or after DMA transfer is terminated [Clearing condition] • Writing 0 after reading HE = 1.*4 1: DMATCR ≤ (DMATCR set before transfer starts)/2
18	HIE	0	R/W	Half-End Interrupt Enable Specifies whether to issue an interrupt request to the CPU when the transfer count reaches half of the DMATCR value that was specified before transfer starts. When the HIE bit is set to 1, this module requests an interrupt to the CPU when the HE bit becomes 1. 0: Disables an interrupt to be issued when DMATCR = (DMATCR set before transfer starts)/2 1: Enables an interrupt to be issued when DMATCR = (DMATCR set before transfer starts)/2

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
17	AM	0	R/W	Acknowledge Mode Specifies whether DACK and TEND are output in data read cycle or in data write cycle in dual address mode. In single address mode, DACK and TEND are always output regardless of the specification by this bit. This bit is valid only in CHCR_0 and CHCR_1*[1]. This bit is reserved in CHCR_2 to CHCR_15*[2]; it is always read as 0 and the write value should always be 0. 0: DACK and TEND output in read cycle (dual address mode) 1: DACK and TEND output in write cycle (dual address mode)
16	AL	0	R/W	Acknowledge Level Specifies the DACK (acknowledge) signal output is high active or low active. This bit is valid only in CHCR_0 and CHCR_1*[1]. This bit is reserved in CHCR_2 to CHCR_15*[2]; it is always read as 0 and the write value should always be 0. 0: Low-active output from DACK 1: High-active output from DACK

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
15, 14	DM[1:0]	00	R/W	Destination Address Mode These bits select whether the DMA destination address is incremented, decremented, or left fixed. (In single address mode, DM1 and DM0 bits are ignored when data is transferred to an external device with DACK.) 00: Fixed destination address 01: Destination address is incremented (+1 in byte-unit transfer, +2 in word-unit transfer, +4 in longword-unit transfer, +16 in 16-byte-unit transfer) 10: Destination address is decremented (–1 in byte-unit transfer, –2 in word-unit transfer, –4 in longword-unit transfer, setting prohibited in 16-byte-unit transfer) 11: Setting prohibited
13, 12	SM[1:0]	00	R/W	Source Address Mode These bits select whether the DMA source address is incremented, decremented, or left fixed. (In single address mode, SM1 and SM0 bits are ignored when data is transferred from an external device with DACK.) 00: Fixed source address 01: Source address is incremented (+1 in byte-unit transfer, +2 in word-unit transfer, +4 in longword-unit transfer, +16 in 16-byte-unit transfer) 10: Source address is decremented (–1 in byte-unit transfer, –2 in word-unit transfer, –4 in longword-unit transfer, setting prohibited in 16-byte-unit transfer) 11: Setting prohibited

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
11 to 8	RS[3:0]	0000	R/W	Resource Select These bits specify which transfer requests will be sent to this module. The changing of transfer request source should be done in the state when DMA enable bit (DE) is set to 0. 0000: External request, dual address mode 0001: Setting prohibited 0010: External request/single address mode External address space → External device with DACK 0011: External request/single address mode External device with DACK → External address space 0100: Auto request 0101: Setting prohibited 0110: Setting prohibited 0111: Setting prohibited 1000: DMA extension resource selector 1001: Controller area network, channel 0 1010: Controller area network, channel 1 1011: Setting prohibited 1100: Setting prohibited 1101: Setting prohibited 1110: Setting prohibited 1111: Setting prohibited Note: External request specification is valid only in CHCR_0 and CHCR_1. External request should not be specified for channels CHCR_2 to CHCR_15.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
7	DL	0	R/W	DREQ Level
6	DS	0	R/W	DREQ Edge Select These bits specify the sampling method of the DREQ pin input and the sampling level. These bits are valid only in CHCR_0 and CHCR_1*[1]. These bits are reserved in CHCR_2 to CHCR_15*[2]; they are always read as 0 and the write value should always be 0. If the transfer request source is specified as an on-chip peripheral module or if an auto-request is specified, the specification by these bits is ignored. 00: DREQ detected in low level 01: DREQ detected at falling edge 10: DREQ detected in high level 11: DREQ detected at rising edge
5	TB	0	R/W	Transfer Bus Mode Specifies the bus mode at DMA transfer. Note that the burst mode must not be selected when TC = 0. 0: Cycle steal mode 1: Burst mode
4, 3	TS[1:0]	00	R/W	Transfer Size These bits specify the size of data to be transferred. Select the size of data to be transferred when the source or destination is an on-chip peripheral module register of which transfer size is specified. 00: Byte unit 01: Word unit (two bytes) 10: Longword unit (four bytes) 11: 16-byte (four longword) unit

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
2	IE	0	R/W	Interrupt Enable Specifies whether or not an interrupt request is generated to the CPU at the end of the DMA transfer. Setting this bit to 1 generates an interrupt request (DEI) to the CPU when TE bit is set to 1. 0: Disables an interrupt request 1: Enables an interrupt request
1	TE	0	R/(W)*[3]	Transfer End Flag This bit is set to 1 when DMATCR becomes 0 and DMA transfer ends.*[4] The TE bit is not set to 1 in the following cases. • DMA transfer ends due to an NMI interrupt or DMA address error before DMATCR becomes 0. • DMA transfer is ended by clearing the DE bit and DME bit in DMA operation register (DMAOR). To clear the TE bit, write 0 after reading TE = 1. Even if the DE bit is set to 1 while the TEMASK bit is 0 and this bit is 1, transfer is not enabled. 0: During the DMA transfer or DMA transfer has been terminated [Clearing condition] • Writing 0 after reading TE = 1*[4] 1: DMA transfer ends by the specified count (DMATCR = 0)

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
0	DE	0	R/W	DMA Enable Enables or disables the DMA transfer. In auto request mode, DMA transfer starts by setting the DE bit and DME bit in DMAOR to 1. In this case, all of the bits TE, NMIF in DMAOR, and AE must be 0. In an external request or peripheral module request, DMA transfer starts if DMA transfer request is generated by the devices or peripheral modules after setting the bits DE and DME to 1. If the DREQ signal is detected by low/high level in external request mode, or in peripheral module request mode, the NMIF bit and the AE bit must be 0 if the TEMASK bit is 1. If the TEMASK bit is 0, the TE bit must also be 0. If the DREQ signal is detected by a rising/falling edge in external request mode, all of the bits TE, NMIF, and AE must be 0 as in the case of auto request mode. Clearing the DE bit to 0 can terminate the DMA transfer. 0: DMA transfer disabled 1: DMA transfer enabled

Notes: 1. Only CHCR_0 can be used in the SH7262 Group.
2. Channels 1 to 15 can be used in the SH7262 Group.
3. Only 0 can be written to clear the flag after 1 is read.
4. If the flag is read at the same timing it is set to 1, the read data will be 0, but the internal state may be the same as reading 1. Therefore, if 0 is written to the flag, the flag will be cleared to 0 because the internal state is the same as when writing 0 after reading 1. For details, refer to section 10.5.2, Notes on Using Flag Bits.

chanhvinh.com

10.3.5 DMA Reload Source Address Registers (RSAR)

The DMA reload source address registers (RSAR) are 32-bit readable/writable registers.

When the SAR reload function is enabled, the RSAR value is written to the source address register (SAR) at the end of the current DMA transfer. In this case, a new value for the next DMA transfer can be preset in RSAR during the current DMA transfer. When the SAR reload function is disabled, RSAR is ignored.

To transfer data in word (2-byte), longword (4-byte), or 16-byte unit, specify the address with 2-byte, 4-byte, or16-byte address boundary respectively.

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/W	R/W	R/W	R/W	R/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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

10.3.6 DMA Reload Destination Address Registers (RDAR)

The DMA reload destination address registers (RDAR) are 32-bit readable/writable registers.

When the DAR reload function is enabled, the RDAR value is written to the destination address register (DAR) at the end of the current DMA transfer. In this case, a new value for the next DMA transfer can be preset in RDAR during the current DMA transfer. When the DAR reload function is disabled, RDAR is ignored.

To transfer data in word (2-byte), longword (4-byte), or 16-byte unit, specify the address with 2-byte, 4-byte, or16-byte address boundary respectively.

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/W	R/W	R/W	R/W	R/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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

chanhvinh.com

10.3.7 DMA Reload Transfer Count Registers (RDMATCR)

The DMA reload transfer count registers (RDMATCR) are 32-bit readable/writable registers.

When the SAR/DAR reload function is enabled, the RDMATCR value is written to the transfer count register (DMATCR) at the end of the current DMA transfer. In this case, a new value for the next DMA transfer can be preset in RDMATCR during the current DMA transfer. When the SAR/DAR reload function is disabled, RDMATCR is ignored.

The upper eight bits of RDMATCR are always read as 0, and the write value should always be 0.

As in DMATCR, the transfer count is 1 when the setting is H'00000001, 16,777,215 when H'00FFFFFF is set, and 16,777,216 (the maximum) when H'00000000 is set. To transfer data in 16 bytes, one 16-byte transfer (128 bits) counts one.

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
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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

chanhvinh.com

10.3.8 DMA Operation Register (DMAOR)

The DMA operation register (DMAOR) is a 16-bit readable/writable register that specifies the priority level of channels at the DMA transfer. This register also shows the DMA transfer status.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	CMS[1:0]		-	-	PR[1:0]		-	-	-	-	-	AE	NMIF	DME
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	R	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, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	CMS[1:0]	00	R/W	Cycle Steal Mode Select These bits select either normal mode or intermittent mode in cycle steal mode. It is necessary that the bus modes of all channels be set to cycle steal mode to make the intermittent mode valid. 00: Normal mode 01: Setting prohibited 10: Intermittent mode 16 Executes one DMA transfer for every 16 cycles of Bϕ clock. 11: Intermittent mode 64 Executes one DMA transfer for every 64 cycles of Bϕ clock.
11, 10	—	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
9, 8	PR[1:0]	00	R/W	Priority Mode These bits select the priority level between channels when there are transfer requests for multiple channels simultaneously. 00: Fixed mode 1: CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7 > CH8 > CH9 > CH10 > CH11 > CH12 > CH13 > CH14 > CH15 01: Fixed mode 2: CH0 > CH8 > CH1 > CH9 > CH2 > CH10 > CH3 > CH11 > CH4 > CH12 > CH5 > CH13 > CH6 > CH14 > CH7 > CH15 10: Setting prohibited 11: Setting prohibited
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	AE	0	R/(W)*[1]	Address Error Flag Indicates whether an address error has occurred by this module. When this bit is set, even if the DE bit in CHCR and the DME bit in DMAOR are set to 1, DMA transfer is not enabled. This bit can only be cleared by writing 0 after reading 1.*[2] 0: No address error occurred by this module 1: Address error occurred by this module [Clearing condition] • Writing 0 after reading AE = 1*[2]

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
1	NMIF	0	R/(W)*1	NMI Flag Indicates that an NMI interrupt occurred. When this bit is set, even if the DE bit in CHCR and the DME bit in DMAOR are set to 1, DMA transfer is not enabled. This bit can only be cleared by writing 0 after reading 1.*2 When the NMI is input, the DMA transfer in progress can be done in one transfer unit. Even if the NMI interrupt is input while this module is not in operation, the NMIF bit is set to 1. 0: No NMI interrupt 1: NMI interrupt occurred [Clearing condition] Writing 0 after reading NMIF = 1*2
0	DME	0	R/W	DMA Master Enable Enables or disables DMA transfer on all channels. If the DME bit and DE bit in CHCR are set to 1, DMA transfer is enabled. However, transfer is enabled only when the TE bit in CHCR of the transfer corresponding channel, the NMIF bit in DMAOR, and the AE bit are all cleared to 0. Clearing the DME bit to 0 can terminate the DMA transfer on all channels. 0: DMA transfer is disabled on all channels 1: DMA transfer is enabled on all channels

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

2. If the flag is read at the same timing it is set to 1, the read data will be 0, but the internal state may be the same as reading 1. Therefore, if 0 is written to the flag, the flag will be cleared to 0 because the internal state is the same as when writing 0 after reading 1. For details, refer to section 10.5.2, Notes on Using Flag Bits.

chanhvinh.com

If the priority mode bits are modified after a DMA transfer, the channel priority is initialized. If fixed mode 2 is specified, the channel priority is specified as CH0 > CH8 > CH1 > CH9 > CH2 > CH10 > CH3 > CH11 > CH4 > CH12 > CH5 > CH13 > CH6 > DH14 > CH7 > CH15. If fixed mode 1 is specified, the channel priority is specified as CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7 > CH8 > CH9 > CH10 > CH11 > CH12 > CH13 > CH14 > CH15.

The internal operation of this module for an address error is as follows:

- No address error: Read (source to interior of this module) → Write (interior of this module to destination)
- Address error in source address: Nop → Nop
- Address error in destination address: Read → Nop

chanhvinh.com

10.3.9 DMA Extension Resource Selectors 0 to 7 (DMARS0 to DMARS7)

The DMA extension resource selectors (DMARS) are 16-bit readable/writable registers that specify the source of the DMA transfer request from peripheral modules in each channel. DMARS0 to DMARS7 are for channels 0 and 1, 2 and 3, 4 and 5, 6 and 7, 8 and 9, 10 and 11, 12 and 13, and 14 and 15, respectively. Table 10.3 shows the specifiable combinations.

DMARS can specify the following transfer request sources (The following modules can issue on-chip peripheral module requests):

- Serial communication interface with FIFO: 16 sources
- I^2C bus interface 3: six sources
- A/D converter: one source
- Multi-function timer pulse unit 2: five sources
- Compare match timer: two sources
- USB 2.0 host/function module: two sources
- NAND flash memory controller: two sources
- Controller area network: two sources
- Serial sound interface: five sources
- Sampling rate converter: four sources
- Renesas SPDIF interface: two sources
- CD-ROM decoder: one source
- SD host interface: two sources
- Renesas serial peripheral interface: four sources
- Clock synchronous serial I/O with FIFO: two sources
- Motor control PWM timer: two sources
- Decompression unit: two sources

Two transfer request sources for the controller area network do not need to be specified by these registers, for they can be specified using the RS3 to RS0 bits in the DMA channel control register (CHCR).

chanhvinh.com

- DMARS0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH1 MID[5:0]						CH1 RID[1:0]		CH0 MID[5:0]						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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- DMARS1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH3 MID[5:0]						CH3 RID[1:0]		CH2 MID[5:0]						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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- DMARS2

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH5 MID[5:0]						CH5 RID[1:0]		CH4 MID[5:0]						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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- DMARS3

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH7 MID[5:0]						CH7 RID[1:0]		CH6 MID[5:0]						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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- DMARS4

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH9 MID[5:0]						CH9 RID[1:0]		CH8 MID[5:0]						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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- DMARS5

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH11 MID[5:0]						CH11 RID[1:0]		CH10 MID[5:0]						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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

chanhvinh.com

- DMARS6

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH13 MID[5:0]						CH13 RID[1:0]		CH12 MID[5:0]						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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- DMARS7

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH15 MID[5:0]						CH15 RID[1:0]		CH14 MID[5:0]						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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Transfer requests from the various modules specify MID and RID as shown in table 10.3.

Table 10.3 DMARS Settings

Peripheral Module	Setting Value for One Channel ({MID, RID})	MID	RID	Function
USB 2.0 host/function module	H'03	B'000000	B'11	Channel 0 FIFO
	H'07	B'000001	B'11	Channel 1 FIFO
Renesas SPDIF interface	H'09	B'000010	B'01	Transmit
	H'0A	B'000010	B'10	Receive
SD host interface	H'11	B'000100	B'01	SD_BUF write
	H'12		B'10	SD_BUF read
Clock synchronous serial I/O with FIFO	H'19	B'000110	B'01	Transmit
	H'1A		B'10	Receive
Serial sound interface Channel 0	H'21	B'001000	B'01	Transmit
	H'22		B'10	Receive
Serial sound interface Channel 1	H'27	B'001001	B'11	—
Serial sound interface Channel 2	H'2B	B'001010	B'11	—

chanhvinh.com

Peripheral Module	Setting Value for One Channel ({MID, RID})	MID	RID	Function
Serial sound interface Channel 3	H'2F	B'001011	B'11	—
Motor control PWM timer Channel 1	H'33	B'001100	B'11	—
Motor control PWM timer Channel 2	H'37	B'001101	B'11	—
Sampling rate converter Channel 0	H'41	B'010000	B'01	Input data FIFO empty
	H'42		B'10	Output data FIFO full
Sampling rate converter Channel 1	H'45	B'010001	B'01	Input data FIFO empty
	H'46		B'10	Output data FIFO full
Renesas serial peripheral interface Channel 0	H'51	B'010100	B'01	Transmit
	H'52		B'10	Receive
Renesas serial peripheral interface Channel 1	H'55	B'010101	B'01	Transmit
	H'56		B'10	Receive
I^2C bus interface 3 Channel 0	H'61	B'011000	B'01	Transmit
	H'62		B'10	Receive
I^2C bus interface 3 Channel 1	H'65	B'011001	B'01	Transmit
	H'66		B'10	Receive
I^2C bus interface 3 Channel 2	H'69	B'011010	B'01	Transmit
	H'6A		B'10	Receive
CD-ROM decoder	H'73	B'011100	B'11	—
Serial communication interface with FIFO Channel 0	H'81	B'100000	B'01	Transmit
	H'82		B'10	Receive
Serial communication interface with FIFO Channel 1	H'85	B'100001	B'01	Transmit
	H'86		B'10	Receive

chanhvinh.com

Peripheral Module	Setting Value for One Channel ({MID, RID})	MID	RID	Function
Serial communication interface with FIFO Channel 2	H'89	B'100010	B'01	Transmit
	H'8A		B'10	Receive
Serial communication interface with FIFO Channel 3	H'8D	B'100011	B'01	Transmit
	H'8E		B'10	Receive
Serial communication interface with FIFO Channel 4	H'91	B'100100	B'01	Transmit
	H'92		B'10	Receive
Serial communication interface with FIFO Channel 5	H'95	B'100101	B'01	Transmit
	H'96		B'10	Receive
Serial communication interface with FIFO Channel 6	H'99	B'100110	B'01	Transmit
	H'9A		B'10	Receive
Serial communication interface with FIFO Channel 7	H'9D	B'100111	B'01	Transmit
	H'9E		B'10	Receive
A/D converter	H'B3	B'101100	B'11	—
NAND flash memory controller	H'BB	B'101110	B'11	Transmit/ receive data
	H'BF	B'101111	B'11	Transmit/ receive control code
Decompression unit	H'D1	B'110100	B'01	Input data FIFO empty
	H'D2		B'10	Output data FIFO full
Multi-function timer pulse unit 2 Channel 0	H'E3	B'111000	B'11	—
Multi-function timer pulse unit 2 Channel 1	H'E7	B'111001	B'11	—
Multi-function timer pulse unit 2 Channel 2	H'EB	B'111010	B'11	—

chanhvinh.com

Peripheral Module	Setting Value for One Channel ({MID, RID})	MID	RID	Function
Multi-function timer pulse unit 2 Channel 3	H'EF	B'111011	B'11	—
Multi-function timer pulse unit 2 Channel 4	H'F3	B'111100	B'11	—
Compare match timer Channel 0	H'FB	B'111110	B'11	—
Compare match timer Channel 1	H'FF	B'111111	B'11	—

When MID or RID other than the values listed in table 10.3 is set, the operation of this LSI is not guaranteed. The transfer request from DMARS is valid only when the resource select bits (RS3 to RS0) in CHCR0 to CHCR15 have been set to B'1000. Otherwise, even if DMARS has been set, the transfer request source is not accepted.

chanhvinh.com

10.4 Operation

When there is a DMA transfer request, this module 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. In bus mode, the burst mode or the cycle steal mode can be selected.

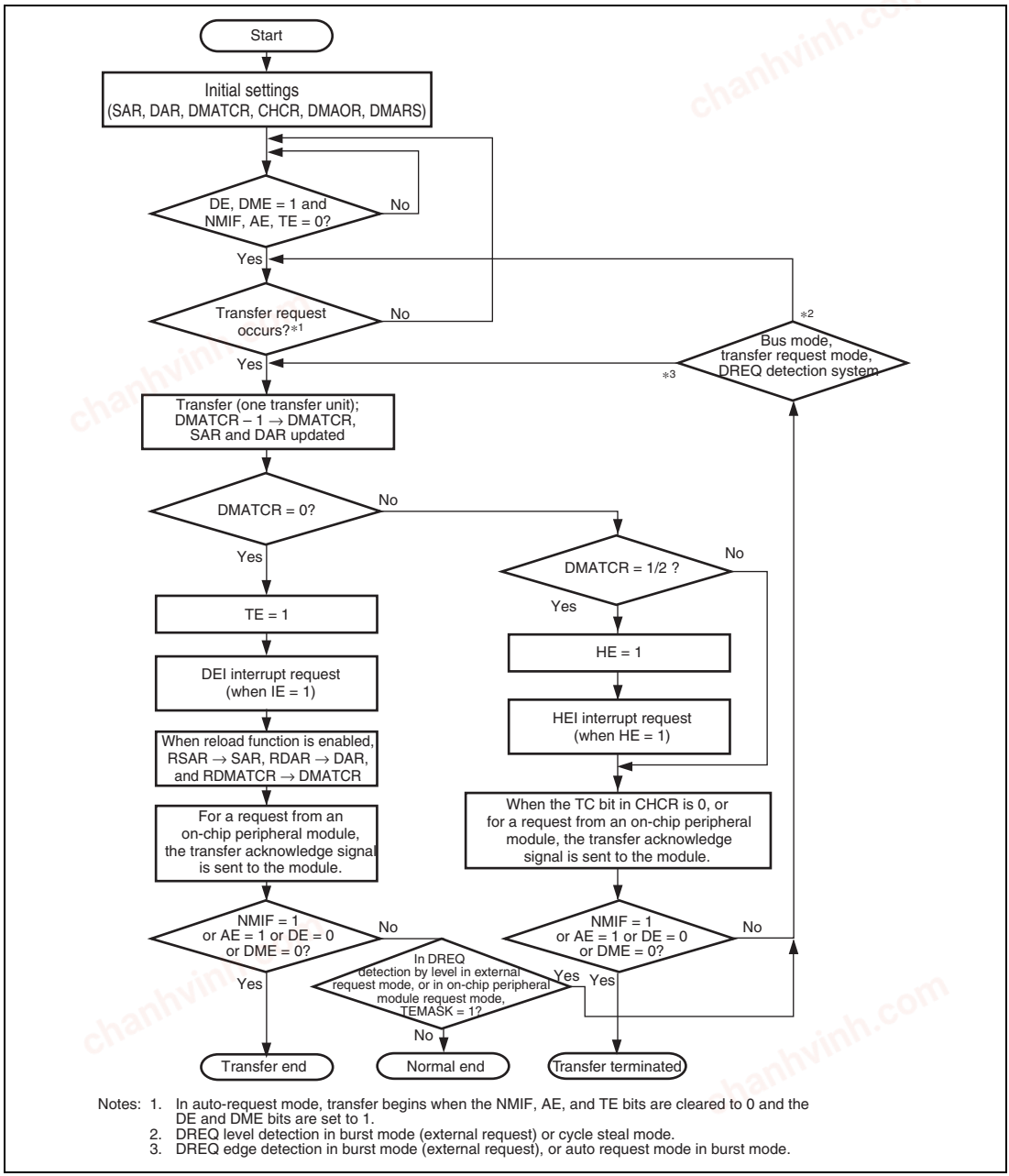
10.4.1 Transfer Flow

After the DMA source address registers (SAR), DMA destination address registers (DAR), DMA transfer count registers (DMATCR), DMA channel control registers (CHCR), DMA operation register (DMAOR), three reload registers (RSAR, RDAR, RDMATCR) and DMA extension resource selector (DMARS) are set for the target transfer conditions, this module transfers data according to the following procedure:

1. Checks to see if transfer is enabled (DE = 1, DME = 1, TEMASK = 0 or 1 (TE = 0 when TEMASK = 0), AE = 0, NMIF = 0).
2. When a transfer request comes and transfer is enabled, this module transfers one transfer unit of data (depending on the settings of the TS1 and TS0 bits). For an auto request, the transfer begins automatically when the DE bit and DME bit are set to 1. The DMATCR value will be decremented by 1 for each transfer. The actual transfer flows vary by address mode and bus mode.
3. When half of the specified transfer count is exceeded (when DMATCR reaches half of the initial value), an HEI interrupt is sent to the CPU if the HIE bit in CHCR is set to 1.
4. When transfer has been completed for the specified count (when DMATCR reaches 0) while the TEMASK bit is 0, the transfer ends normally. If the IE bit in CHCR is set to 1 at this time, a DEI interrupt is sent to the CPU. When DMATCR reaches 0 while the TEMASK bit is 1, the TE bit is set to 1 and then the values set in RSAR, RDAR and RDMATCR are reloaded in SAR, DAR and DMATCR, respectively to continue transfer operation until the DMA transfer request is cancelled.
5. When an address error in this module or an NMI interrupt is generated, the transfer is terminated. Transfers are also terminated when the DE bit in CHCR or the DME bit in DMAOR is cleared to 0.

chanhvinh.com

Figure 10.2 is a flowchart of this procedure.



Notes: 1. In auto-request mode, transfer begins when the NMIF, AE, and TE bits are cleared to 0 and the DE and DME bits are set to 1.
2. DREQ level detection in burst mode (external request) or cycle steal mode.
3. DREQ edge detection in burst mode (external request), or auto request mode in burst mode.

Figure 10.2 DMA Transfer Flowchart

chanhvinh.com

10.4.2 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. The request mode is selected by the RS[3:0] bits in CHCR_0 to CHCR_15 and DMARS0 to DMARS7.

(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, the auto-request mode allows this module to automatically generate a transfer request signal internally. When the DE bits in CHCR_0 to CHCR_15 and the DME bit in DMAOR are set to 1, the transfer begins so long as the TE bits in CHCR_0 to CHCR_15, and the AE and NMIF bits in DMAOR are 0.

(2) External Request Mode

In this mode a transfer is performed at the request signals (DREQ0 and DREQ1)*[1] of an external device. Choose one of the modes shown in table 10.4 according to the application system. When the DMA transfer is enabled (DE = 1, DME = 1, TEMASK = 0 or 1 (TE = 0 when TEMASK = 0), AE = 0, NMIF = 0 for level detection; DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0 for edge detection), DMA transfer is performed upon a request at the DREQ input.

Table 10.4 Selecting External Request Modes with the RS Bits

RS[3]	RS[2]	RS[1]	RS[0]	Address Mode	Transfer Source	Transfer Destination
0	0	0	0	Dual address mode	Any	Any
0	0	1	0	Single address mode	External memory, memory-mapped external device	External device with DACK
			1		External device with DACK	External memory, memory-mapped external device

chanhvinh.com

Choose to detect DREQ by either the edge or level of the signal input with the DL and DS bits in CHCR_0 and CHCR_1*[2] as shown in table 10.5. The source of the transfer request does not have to be the data transfer source or destination. When DREQ is detected by a rising/falling edge and DMA transfer is performed in burst mode, the transfer continues until DMATCR reaches 0 by one DMA transfer request. In cycle steal mode, one DMA transfer is performed by one request.

Notes: 1. Only DREQ0 can be used in the SH7262 Group.
2. Only CHCR_0 can be used in the SH7262 Group.

Table 10.5 Selecting External Request Detection with DL and DS Bits

CHCR		
DL Bit	**DS Bit**	**Detection of External Request**
0	0	Low-level detection
	1	Falling-edge detection
1	0	High-level detection
	1	Rising-edge detection

When DREQ is accepted, the DREQ pin enters the request accept disabled state (non-sensitive period). After issuing acknowledge DACK signal for the accepted DREQ, the DREQ pin again enters the request accept enabled state.

When DREQ is used by level detection, there are following two cases by the timing to detect the next DREQ after outputting DACK.

- Overrun 0: Transfer is terminated after the same number of transfer has been performed as requests.
- Overrun 1: Transfer is terminated after transfers have been performed for (the number of requests plus 1) times.

The DO bit in CHCR selects this overrun 0 or overrun 1.

Table 10.6 Selecting External Request Detection with DO Bit

CHCR	
DO Bit	**External Request**
0	Overrun 0
1	Overrun 1

chanhvinh.com

(3) On-Chip Peripheral Module Request

In this mode, the transfer is performed in response to the DMA transfer request signal from an on-chip peripheral module.

Table 10.7 lists the DMA transfer request signals sent from on-chip peripheral modules to this module.

If DMA transfer is enabled (DE = 1, DME = 1, TEMASK = 0 or 1 (TE = 0 when TEMASK = 0), AE = 0, and NMIF = 0) in on-chip peripheral module request mode, DMA transfer is started by a transfer request signal.

In on-chip peripheral module request mode, there are cases where transfer source or destination is fixed. For details, see table 10.7.

Table 10.7 Selecting On-Chip Peripheral Module Request Modes with RS3 to RS0 Bits

CHCR	DMARS		DMA Transfer Request Source	DMA Transfer Request Signal	Transfer Source	Transfer Destination	Bus Mode
RS[3:0]	MID	RID					
1001	Any	Any	Controller area network Channel 0	RM0 (reception end)	MB0	Any	Cycle steal
1010	Any	Any	Controller area network Channel 1	RM0 (reception end)	MB0	Any	
1000	000000	11	USB 2.0 host/function module	USB_DMA0 (receive FIFO in channel 0 full)	D0FIFO	Any	
				USB_DMA0 (transmit FIFO in channel 0 empty)	Any	D0FIFO	
	000001	11		USB_DMA1 (receive FIFO in channel 1 full)	D1FIFO	Any	
				USB_DMA1 (transmit FIFO in channel 1 empty)	Any	D1FIFO	

chanhvinh.com

CHCR	DMARS		DMA Transfer Request Source	DMA Transfer Request Signal	Transfer Source	Transfer Destination	Bus Mode
RS[3:0]	MID	RID					
1000	000010	01	Renesas SPDIF interface	SPDIFTXI (DMA transfer from transmission module)	Any	TDAD	Cycle steal
		10		SPDIFRXI (DMA transfer to reception module)	RDAD	Any	
	000100	01	SD host interface	SD_BUF write	Any	Data register	
		10		SD_BUF read	Data register	Any	
	000110	01	Clock synchronous serial I/O with FIFO	TXI transmit data transfer)	Any	SITDR	
		10		RXI (receive data transfer)	SIRDR	Any	
	001000	01	Serial sound interface Channel 0	SSITXI0 (transmit data empty)	Any	SSIFTDR_0	
		10		SSIRXI0 (receive data full)	SSIFRDR_0	Any	
	001001	11	Serial sound interface Channel 1	SSIRTI1 (transmit data empty)	Any	SSIFTDR_1	
				SSIRTI1 (receive data full)	SSIFRDR_1	Any	
	001010	11	Serial sound interface Channel 2	SSIRTI2 (transmit data empty)	Any	SSIFTDR_2	
				SSIRTI2 (receive data full)	SSIFRDR_2	Any	
	001011	11	Serial sound interface Channel 3	SSIRTI3 (transmit data empty)	Any	SSIFTDR_3	
				SSIRTI3 (receive data full)	SSIFRDR_3	Any	
	001100	11	Motor control PWM timer Channel 1	CMI1 (PWM compare match)	Any	PWBFR1	
	001101	11	Motor control PWM timer Channel 2	CMI2 (PWM compare match)	Any	PWBFR2	
	010000	01	Sampling rate converter Channel 0	IDEI0 (input data empty)	Any	SRCIDR_0	
		10		ODFI0 (output data full)	SRCODR_0	Any	

chanhvinh.com

CHCR	DMARS		DMA Transfer Request Source	DMA Transfer Request Signal	Transfer Source	Transfer Destination	Bus Mode
RS[3:0]	MID	RID					
1000	010001	01	Sampling rate converter Channel 1	IDEI1 (input data empty)	Any	SRCIDR_1	Cycle steal
		10		ODFI1 (output data full)	SRCODR_1	Any	
	010100	01	Renesas serial peripheral interface Channel 0	SPTI0 (transmit buffer empty)	Any	SPDR_0	
		10		SPRI0 (receive buffer full)	SPDR_0	Any	
	010101	01	Renesas serial peripheral interface Channel 1	SPTI1 (transmit buffer empty)	Any	SPDR_1	
		10		SPRI1 (receive buffer full)	SPDR_1	Any	
	011000	01	I^2C bus interface 3 Channel 0	TXI0 (transmit data empty)	Any	ICDRT_0	
		10		RXI0 (receive data full)	ICDRR_0	Any	
	011001	01	I^2C bus interface 3 Channel 1	TXI1 (transmit data empty)	Any	ICDRT_1	
		10		RXI1 (receive data full)	ICDRR_1	Any	
	011010	01	I^2C bus interface 3 Channel 2	TXI2 (transmit data empty)	Any	ICDRT_2	
		10		RXI2 (receive data full)	ICDRR_2	Any	
	011100	11	CD-ROM decoder	IREADY (decode end)	STRMDOUT	Any	Cycle steal or burst
	100000	01	Serial communication interface with FIFO Channel 0	TXI0 (transmit FIFO data empty)	Any	SCFTDR_0	Cycle steal
		10		RXI0 (receive FIFO data full)	SCFRDR_0	Any	
	100001	01	Serial communication interface with FIFO Channel 1	TXI1 (transmit FIFO data empty)	Any	SCFTDR_1	
		10		RXI1 (receive FIFO data full)	SCFRDR_1	Any	
	100010	01	Serial communication interface with FIFO Channel 2	TXI2 (transmit FIFO data empty)	Any	SCFTDR_2	
		10		RXI2 (receive FIFO data full)	SCFRDR_2	Any	

chanhvinh.com

CHCR	DMARS		DMA Transfer Request Source	DMA Transfer Request Signal	Transfer Source	Transfer Destination	Bus Mode
RS[3:0]	MID	RID					
1000	100011	01	Serial communication interface with FIFO Channel 3	TXI3 (transmit FIFO data empty)	Any	SCFTDR_3	Cycle steal
		10		RXI3 (receive FIFO data full)	SCFRDR_3	Any	
	100100	01	Serial communication interface with FIFO Channel 4	TXI4 (transmit FIFO data empty)	Any	SCFTDR_4	
		10		RXI4 (receive FIFO data full)	SCFRDR_4	Any	
	100101	01	Serial communication interface with FIFO Channel 5	TXI5 (transmit FIFO data empty)	Any	SCFTDR_5	
		10		RXI5 (receive FIFO data full)	SCFRDR_5	Any	
	100110	01	Serial communication interface with FIFO Channel 6	TXI6 (transmit FIFO data empty)	Any	SCFTDR_6	
		10		RXI6 (receive FIFO data full)	SCFRDR_6	Any	
	100111	01	Serial communication interface with FIFO Channel 7	TXI7 (transmit FIFO data empty)	Any	SCFTDR_7	
		10		RXI7 (receive FIFO data full)	SCFRDR_7	Any	
	101100	11	A/D converter	ADI (A/D conversion end)	ADDR	Any	
	101110	11	NAND flash memory controller	Data part Transmission FIFO data empty	Any	FLDTFIFO	
				Data part Reception FIFO data full	FLDTFIFO	Any	
	101111	11		Control code part Transmission FIFO data empty	Any	FLECFIFO	
				Control code part Reception FIFO data full	FLECFIFO	Any	
	110100	01	Decompression unit	IFEI (input FIFO empty)	Any	Input buffer register	
		10		OFFI (output FIFO full)	Output buffer register	Any	

chanhvinh.com

CHCR	DMARS		DMA Transfer Request Source	DMA Transfer Request Signal	Transfer Source	Transfer Destination	Bus Mode
RS[3:0]	MID	RID					
1000	111000	11	Multi-function timer pulse unit 2 Channel 0	TGI0A (input capture or compare match)	Any	Any	Cycle steal or burst
	111001	11	Multi-function timer pulse unit 2 Channel 1	TGI1A (input capture or compare match)	Any	Any	
	111010	11	Multi-function timer pulse unit 2 Channel 2	TGI2A (input capture or compare match)	Any	Any	
	111011	11	Multi-function timer pulse unit 2 Channel 3	TGI3A (input capture or compare match)	Any	Any	
	111100	11	Multi-function timer pulse unit 2 Channel 4	TGI4A (input capture or compare match)	Any	Any	
	111110	11	Compare match timer Channel 0	CMI0 (compare match)	Any	Any	
	111111	11	Compare match timer Channel 1	CMI1 (compare match)	Any	Any	

chanhvinh.com

10.4.3 Channel Priority

When this module receives simultaneous transfer requests on two or more channels, it selects a channel according to a predetermined priority order. Two modes (fixed mode 1 and fixed mode 2) are selected.

In these mode, the priority levels among the channels are as follows:

Fixed mode 1: CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7 > CH8 > CH9 > CH10 > CH11> CH12> CH13 > CH14 > CH15

Fixed mode 2: CH0 > CH8 > CH1 > CH9 > CH2 > CH10 > CH3 > CH11> CH4 > CH12 > CH5 > CH13 > CH6> CH14 > CH7 > CH15

These are selected by the PR1 and PR0 bits in the DMA operation register (DMAOR).

10.4.4 DMA Transfer Types

DMA transfer has two types; single address mode transfer and dual address mode transfer. They depend on the number of bus cycles of access to the transfer source and destination. A data transfer timing depends on the bus mode, which is the cycle steal mode or burst mode. This module supports the transfers shown in table 10.8.

Table 10.8 Supported DMA Transfers

	Transfer Destination				
Transfer Source	**External Device with DACK**	**External Memory**	**Memory-Mapped External Device**	**On-Chip Peripheral Module**	**On-Chip Memory**
External device with DACK	Not available	Dual, single	Dual, single	Not available	Not available
External memory	Dual, single	Dual	Dual	Dual	Dual
Memory-mapped external device	Dual, single	Dual	Dual	Dual	Dual
On-chip peripheral module	Not available	Dual	Dual	Dual	Dual
On-chip memory	Not available	Dual	Dual	Dual	Dual

Notes: 1. Dual: Dual address mode
2. Single: Single address mode
3. 16-byte transfer is available only for on-chip peripheral modules that support longword access.

chanhvinh.com

(1) Address Modes

(a) Dual Address Mode

In dual address mode, both the transfer source and destination are accessed (selected) by an address. The transfer source and destination can be located externally or internally.

DMA transfer requires two bus cycles because data is read from the transfer source in a data read cycle and written to the transfer destination in a data write cycle. At this time, transfer data is temporarily stored in this module. In the transfer between external memories as shown in figure 10.3, data is read to this module from one external memory in a data read cycle, and then that data is written to the other external memory in a data write cycle.

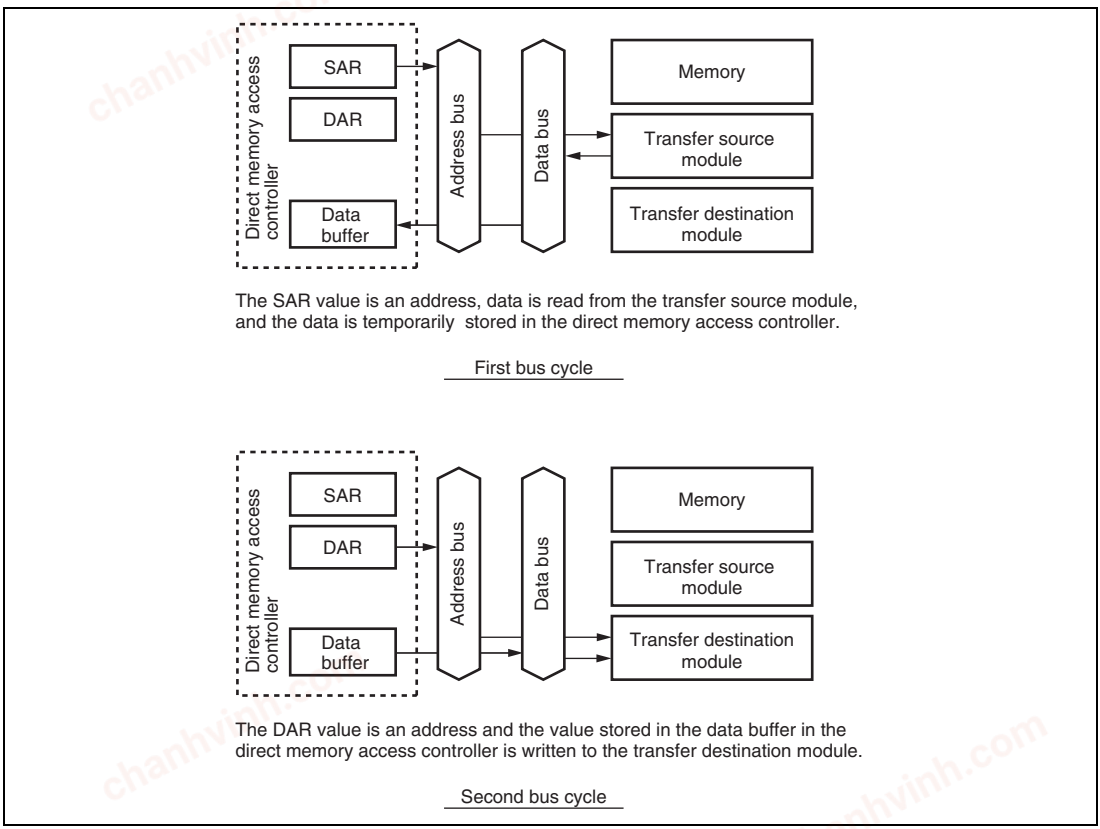


Figure 10.3 Data Flow of Dual Address Mode

chanhvinh.com

Auto request, external request, and on-chip peripheral module request are available for the transfer request. DACK can be output in read cycle or write cycle in dual address mode. The AM bit in the channel control register (CHCR) can specify whether the DACK is output in read cycle or write cycle.

Figure 10.4 shows an example of DMA transfer timing in dual address mode.

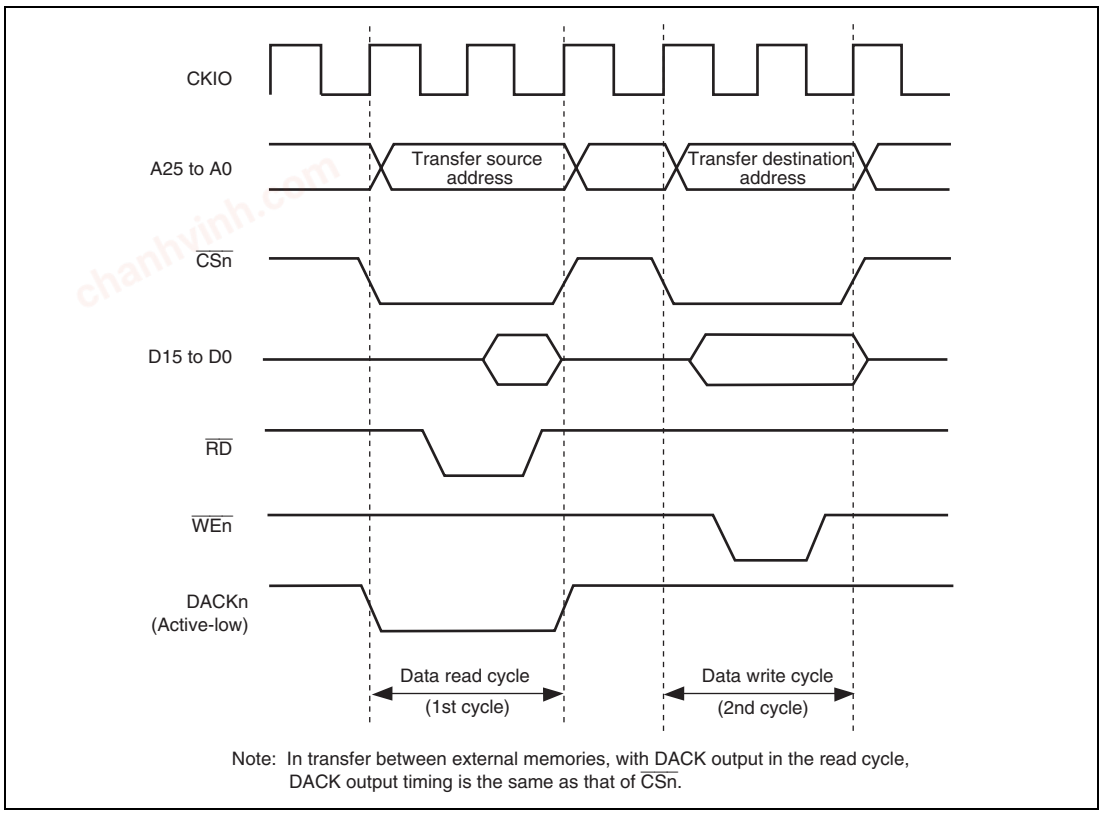


Figure 10.4 Example of DMA Transfer Timing in Dual Mode (Transfer Source: Normal Memory, Transfer Destination: Normal Memory)

chanhvinh.com

(b) Single Address Mode

In single address mode, both the transfer source and destination are external devices, either of them is accessed (selected) by the DACK signal, and the other device is accessed by an address. In this mode, this module performs one DMA transfer in one bus cycle, accessing one of the external devices by outputting the DACK transfer request acknowledge signal to it, and at the same time outputting an address to the other device involved in the transfer. For example, in the case of transfer between external memory and an external device with DACK shown in figure 10.5, when the external device outputs data to the data bus, that data is written to the external memory in the same bus cycle.

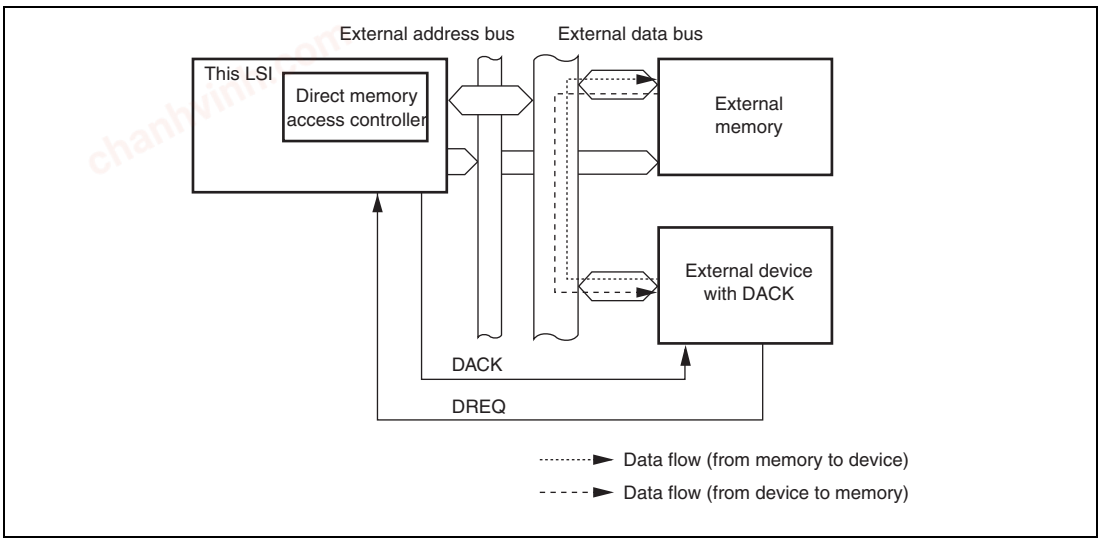


Figure 10.5 Data Flow in Single Address Mode

Two kinds of transfer are possible in single address mode: (1) transfer between an external device with DACK and a memory-mapped external device, and (2) transfer between an external device with DACK and external memory. In both cases, only the external request signal (DREQ) is used for transfer requests.

chanhvinh.com

Figure 10.6 shows an example of DMA transfer timing in single address mode.

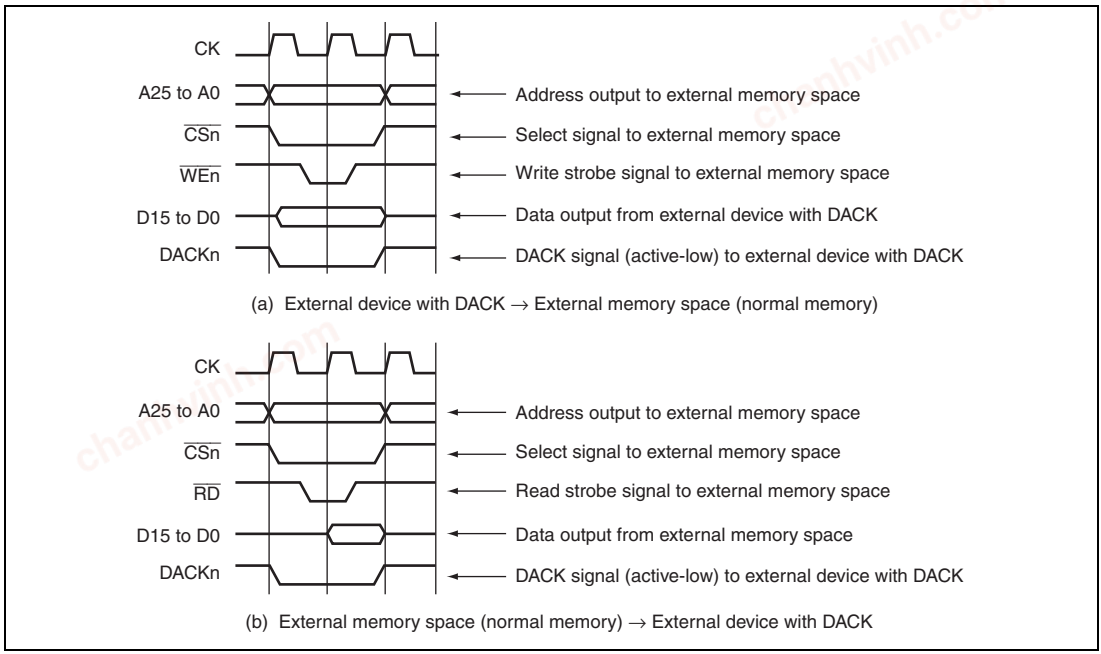


Figure 10.6 Example of DMA Transfer Timing in Single Address Mode

(2) Bus Modes

There are two bus modes; cycle steal and burst. Select the mode by the TB bits in the channel control registers (CHCR).

(a) Cycle Steal Mode

- Normal mode

  In normal mode of cycle steal, the bus mastership is given to another bus master after a one-transfer-unit (byte, word, longword, or 16-byte unit) DMA transfer. When another transfer request occurs, the bus mastership is obtained from another bus master and a transfer is performed for one transfer unit. When that transfer ends, the bus mastership is passed to another bus master. This is repeated until the transfer end conditions are satisfied.

  The cycle-steal normal mode can be used for any transfer section; transfer request source, transfer source, and transfer destination.

chanhvinh.com

Figure 10.7 shows an example of DMA transfer timing in cycle-steal normal mode. Transfer conditions shown in the figure are;

— Dual address mode

— DREQ low level detection

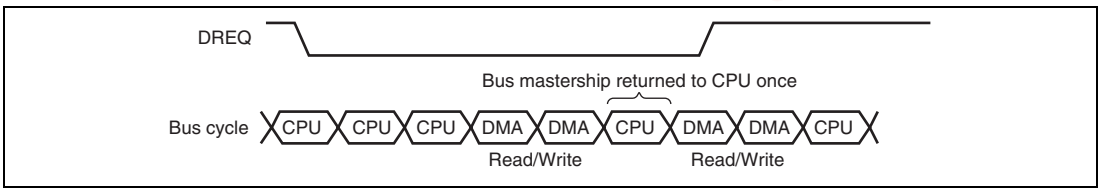


Figure 10.7 DMA Transfer Example in Cycle-Steal Normal Mode (Dual Address, DREQ Low Level Detection)

- Intermittent Mode 16 and Intermittent Mode 64

  In intermittent mode of cycle steal, this module returns the bus mastership to other bus master whenever a unit of transfer (byte, word, longword, or 16 bytes) is completed. If the next transfer request occurs after that, this module obtains the bus mastership from other bus master after waiting for 16 or 64 cycles of Bϕ clock. This module then transfers data of one unit and returns the bus mastership to other bus master. These operations are repeated until the transfer end condition is satisfied. It is thus possible to make lower the ratio of bus occupation by DMA transfer than the normal mode of cycle steal.

  When this module obtains again the bus mastership, DMA transfer may be postponed in case of entry updating due to cache miss.

  The cycle-steal intermittent mode can be used for any transfer section; transfer request source, transfer source, and transfer destination. The bus modes, however, must be cycle steal mode in all channels.

  Figure 10.8 shows an example of DMA transfer timing in cycle-steal intermittent mode. Transfer conditions shown in the figure are;

  — Dual address mode

  — DREQ low level detection

chanhvinh.com

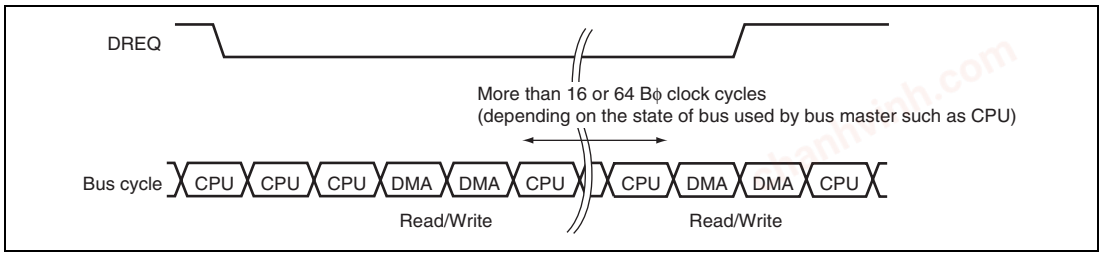


Figure 10.8 Example of DMA Transfer in Cycle-Steal Intermittent Mode (Dual Address, DREQ Low Level Detection)

(b) Burst Mode

In burst mode, once this module obtains the bus mastership, it does not release the bus mastership and continues to perform transfer until the transfer end condition is satisfied. In external request mode with low-level detection of the DREQ pin, however, when the DREQ pin is driven high, the bus mastership is passed to another bus master after the DMA transfer request that has already been accepted ends, even if the transfer end conditions have not been satisfied.

Figure 10.9 shows DMA transfer timing in burst mode.

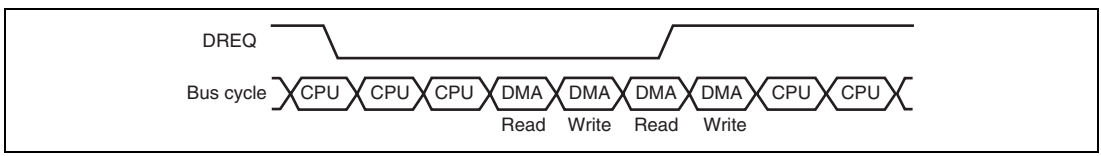


Figure 10.9 DMA Transfer Example in Burst Mode (Dual Address, DREQ Low Level Detection)

chanhvinh.com

(3) Relationship between Request Modes and Bus Modes by DMA Transfer Category

Table 10.9 shows the relationship between request modes and bus modes by DMA transfer category.

Table 10.9 Relationship of Request Modes and Bus Modes by DMA Transfer Category

Address Mode	Transfer Category	Request Mode	Bus Mode	Transfer Size (Bits)	Usable Channels
Dual	External device with DACK and external memory	External	B/C	8/16/32/128	0, 1*6
	External device with DACK and memory-mapped external device	External	B/C	8/16/32/128	0, 1*6
	External memory and external memory	All*4	B/C	8/16/32/128	0 to 15*3
	External memory and memory-mapped external device	All*4	B/C	8/16/32/128	0 to 15*3
	Memory-mapped external device and memory-mapped external device	All*4	B/C	8/16/32/128	0 to 15*3
	External memory and on-chip peripheral module	All*1	B/C*5	8/16/32/128*2	0 to 15*3
	Memory-mapped external device and on-chip peripheral module	All*1	B/C*5	8/16/32/128*2	0 to 15*3
	On-chip peripheral module and on-chip peripheral module	All*1	B/C*5	8/16/32/128*2	0 to 15*3
	On-chip memory and on-chip memory	All*4	B/C	8/16/32/128	0 to 15*3
	On-chip memory and memory-mapped external device	All*4	B/C	8/16/32/128	0 to 15*3
	On-chip memory and on-chip peripheral module	All*1	B/C*5	8/16/32/128*2	0 to 15*3
	On-chip memory and external memory	All*4	B/C	8/16/32/128	0 to 15*3
Single	External device with DACK and external memory	External	B/C	8/16/32/128	0, 1*6
	External device with DACK and memory-mapped external device	External	B/C	8/16/32/128	0, 1*6

[Legend]

B: Burst

C: Cycle steal

Notes: 1. External requests, auto requests, and on-chip peripheral module requests are all available. However, in the case of internal module request, along with the exception of the multi-function timer pulse unit 2 and the compare match timer as the transfer request source, the requesting module must be designated as the transfer source or the transfer destination.

chanhvinh.com

2. Access size permitted for the on-chip peripheral module register functioning as the transfer source or transfer destination.
3. If the transfer request is an external request, channels 0 and 1 are only available (Only channel 0 in the SH7262 Group).
4. External requests, auto requests, and on-chip peripheral module requests are all available. In the case of on-chip peripheral module requests, however, the compare match timer and the multi-function timer pulse unit 2 are only available.
5. In the case of on-chip peripheral module request, only cycle steal except for the CD-ROM decoder, the multi-function timer pulse unit 2, and the compare match timer as the transfer request source.
6. Channel 0 is only available in the SH7262 Group.

(4) Bus Mode and Channel Priority

In priority fixed mode (CH0 > CH1), when channel 1 is transferring data in burst mode and a request arrives for transfer on channel 0, which has higher-priority, the data transfer on channel 0 will begin immediately. In this case, if the transfer on channel 0 is also in burst mode, the transfer on channel 1 will only resume on completion of the transfer on channel 0.

When channel 0 is in cycle steal mode, one transfer-unit of data on this channel, which has the higher priority, is transferred. Data is then transferred continuously to channel 1 without releasing the bus. The bus mastership will then switch between the two in this order: channel 0, channel 1, channel 0, channel 1, etc. That is, the CPU cycle after the data transfer in cycle steal mode is replaced with a burst-mode transfer cycle (priority execution of burst-mode cycle). An example of this is shown in figure 10.10.

When multiple channels are in burst mode, data transfer on the channel that has the highest priority is given precedence. When DMA transfer is being performed on multiple channels, the bus mastership is not released to another bus-master device until all of the competing burst-mode transfers have been completed.

chanhvinh.com

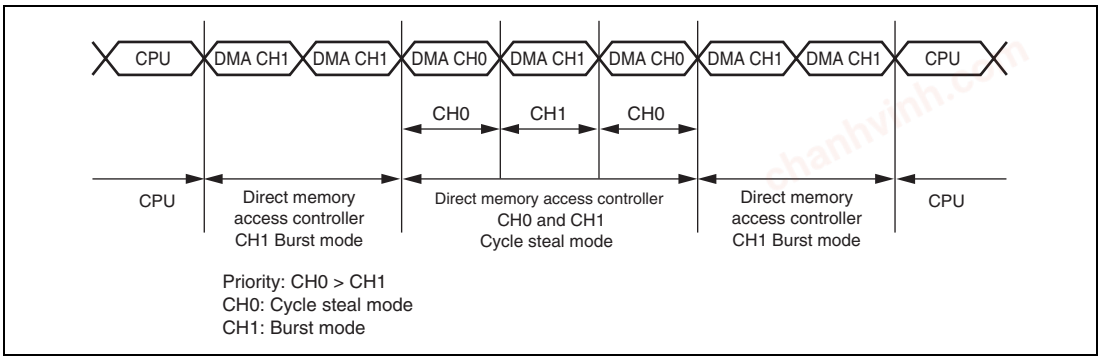


Figure 10.10 Bus State when Multiple Channels are Operating

10.4.5 Number of Bus Cycles and DREQ Pin Sampling Timing

(1) Number of Bus Cycles

When this module is the bus master, the number of bus cycles is controlled by the bus state controller in the same way as when the CPU is the bus master. For details, see section 9, Bus State Controller.

(2) DREQ Pin Sampling Timing

Figures 10.11 to 10.14 show the DREQ input sampling timings in each bus mode.

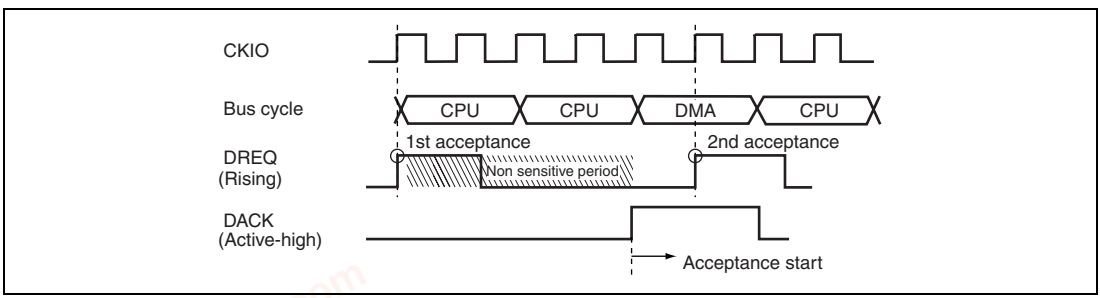


Figure 10.11 Example of DREQ Input Detection in Cycle Steal Mode Edge Detection

chanhvinh.com

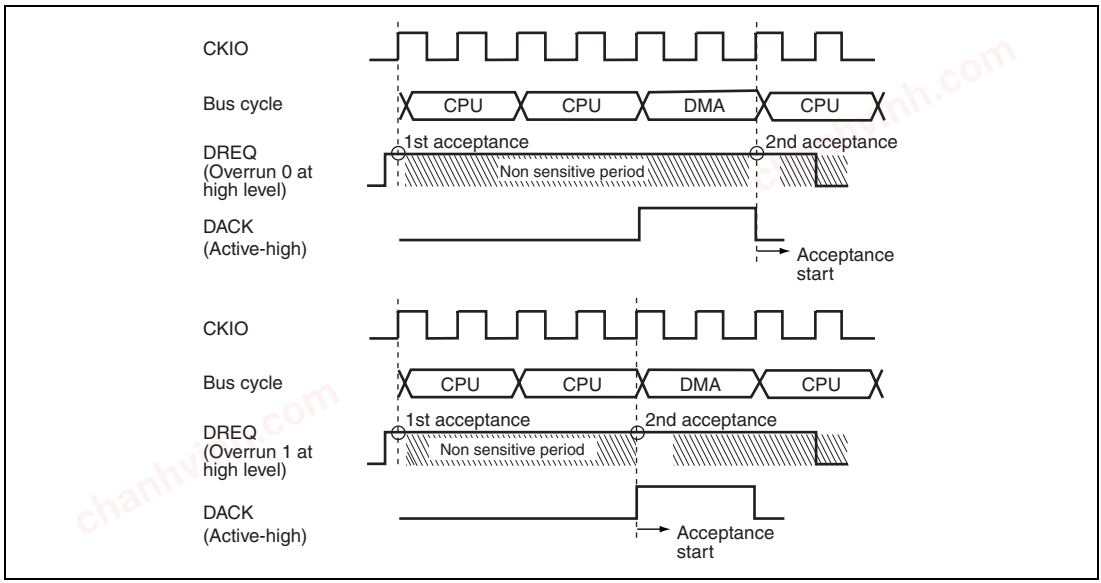


Figure 10.12 Example of DREQ Input Detection in Cycle Steal Mode Level Detection

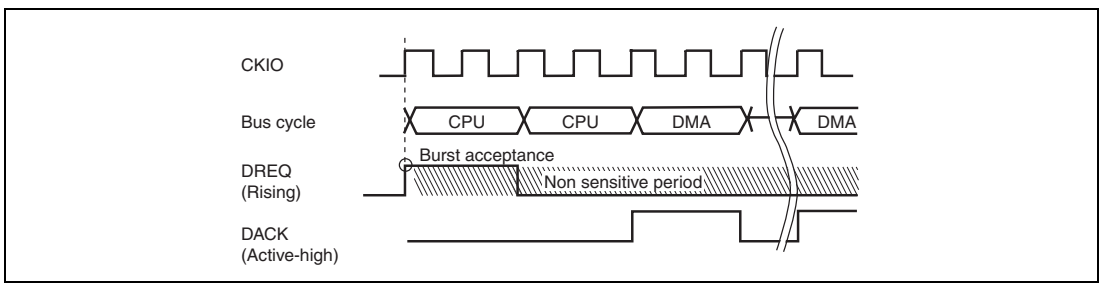


Figure 10.13 Example of DREQ Input Detection in Burst Mode Edge Detection

chanhvinh.com

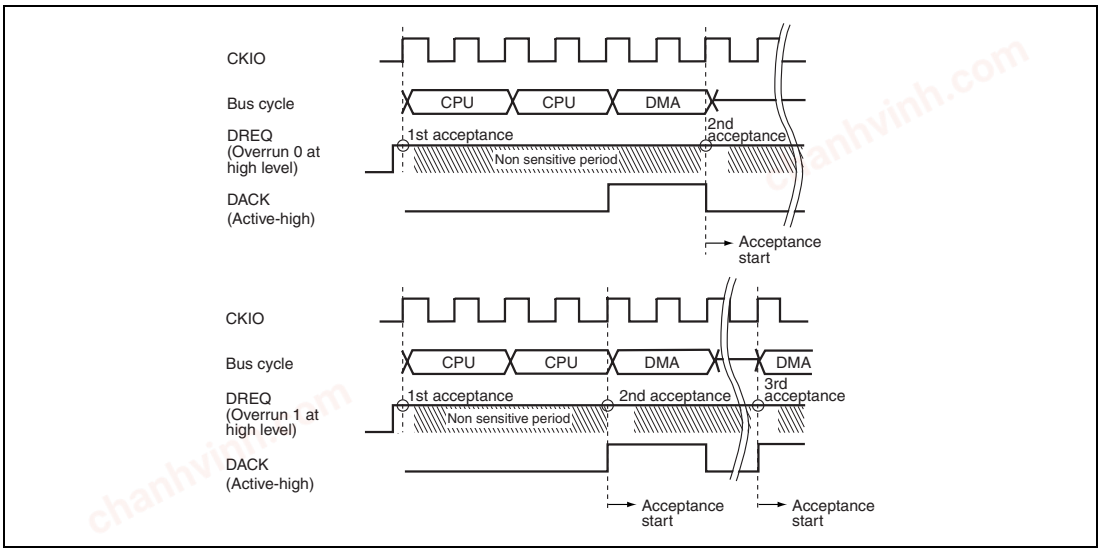


Figure 10.14 Example of DREQ Input Detection in Burst Mode Level Detection

Figure 10.15 shows the TEND output timing.

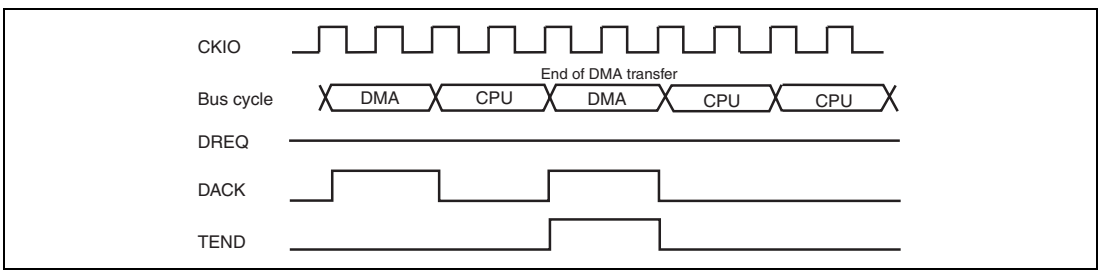
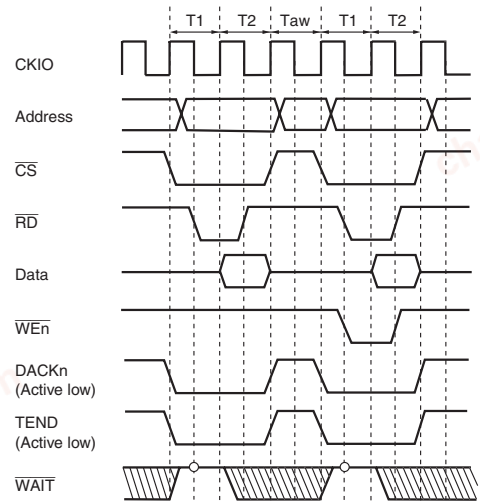


Figure 10.15 Example of DMA Transfer End Signal Timing (Cycle Steal Mode Level Detection)

The unit of the DMA transfer is divided into multiple bus cycles when 16-byte transfer is performed for an 8-bit or16-bit external device or when word transfer is performed for an 8-bit external device. When a setting is made so that the DMA transfer size is divided into multiple bus cycles and the $\overline{CS}$ signal is negated between bus cycles, note that DACK and TEND are divided like the $\overline{CS}$ signal for data alignment as shown in figure 10.16. Figures 10.11 to 10.15 show the cases where DACK and TEND are not divided in the DMA transfer.

chanhvinh.com



Note: TEND is asserted for the last unit of DMA transfer. If a transfer unit is divided into multiple bus cycles and the CS is negated between the bus cycles, TEND is also divided.

Figure 10.16 Bus State Controller Normal Memory Access (No Wait, Idle Cycle 1, Longword Access to 16-Bit Device)

chanhvinh.com

10.5 Usage Notes

10.5.1 Timing of DACK and TEND Outputs

When the external memory is the MPX-I/O, the DACK output is asserted with the timing of the data cycle. For details, see section 9.5.5, MPX-I/O Interface in section 9, Bus State Controller.

When the memory is other than the MPX-I/O, the DACK output is asserted with the same timing as the corresponding CS signal.

The TEND output does not depend on the type of memory and is always asserted with the same timing as the corresponding CS signal.

10.5.2 Notes on Using Flag Bits

The notes on using the following flag bits are described here.

- DMA channel control register (CHCR)
  HE (Half-End) and TE (Transfer End Flag) bits
- DMA operation register (DMAOR)
  AE (Address Error Flag) and NMIF (NMI Flag) bits

If a flag is read at the same timing it is set to 1, the read data will be 0, but the internal state may be the same as reading 1. Therefore, if 0 is written to the flag, the flag will be cleared to 0 because the internal state is the same as when writing 0 after reading 1.

In the case of using a flag, to prevent from unintentionally clearing the flag bit to 0, perform read/write as follows:

(a) In the case of intended bit clear, write 0 to the flag bit after reading it as 1.
(b) In other cases, write 1 to the flag bit.

If a flag is not used, just writing 0 to the flag bit does not generate errors (in the case of intended bit clear, write 0 to the flag bit after reading it as 1).

chanhvinh.com

Section 11 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.

11.1 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
- 28 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 11.1 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		√	—	—	√	√

chanhvinh.com

Item	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4
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
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

chanhvinh.com

Item	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4
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 11.1 shows a block diagram.

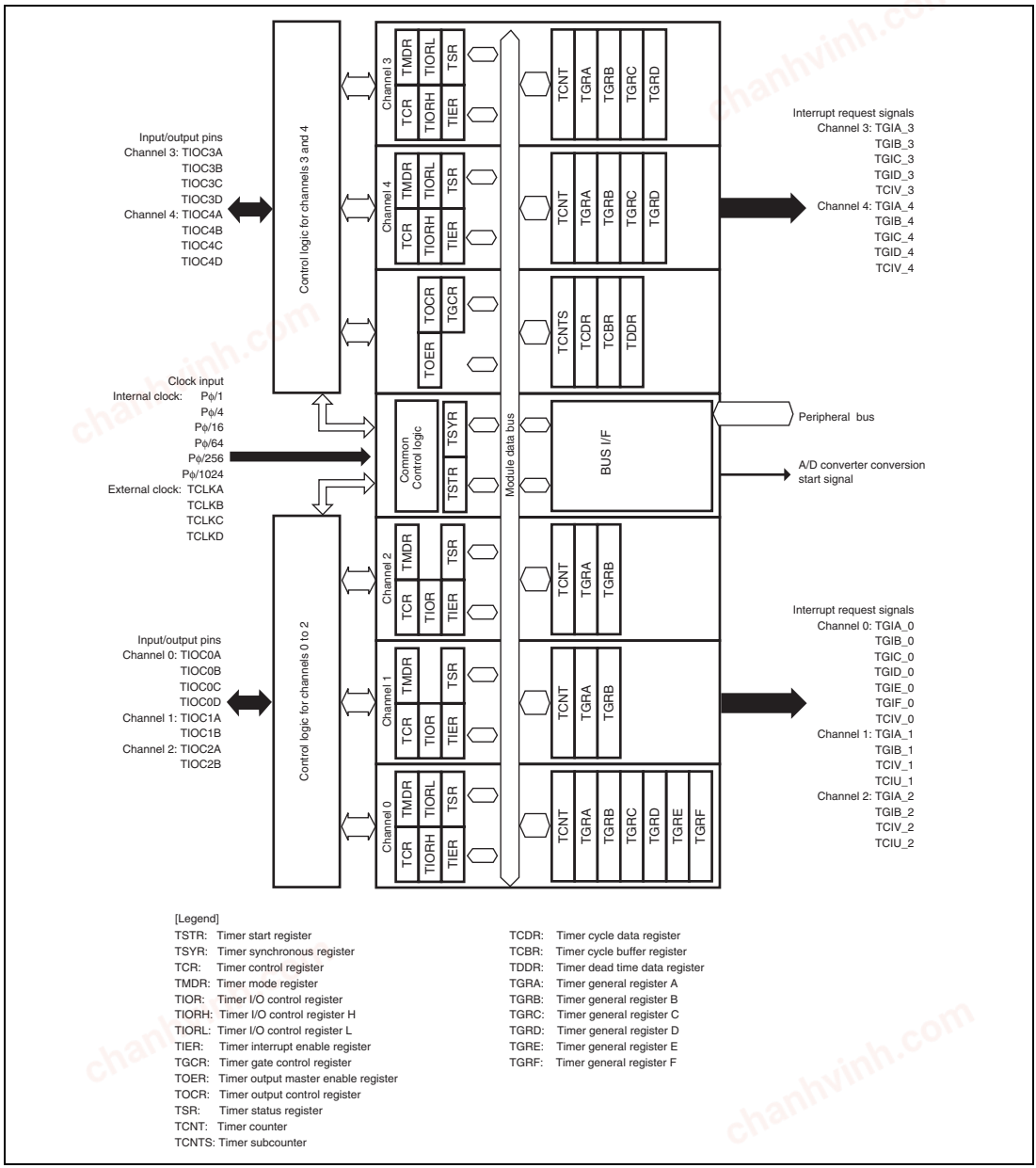


Figure 11.1 Block Diagram

chanhvinh.com

11.2 Input/Output Pins

Table 11.2 shows the pin configuration.

Table 11.2 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 11.54 in section 11.4.8, Complementary PWM Mode.

chanhvinh.com

11.3 Register Descriptions

Table 11.3 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 11.3 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'FFFE4300	8
	Timer mode register_0	TMDR_0	R/W	H'00	H'FFFE4301	8
	Timer I/O control register H_0	TIORH_0	R/W	H'00	H'FFFE4302	8
	Timer I/O control register L_0	TIORL_0	R/W	H'00	H'FFFE4303	8
	Timer interrupt enable register_0	TIER_0	R/W	H'00	H'FFFE4304	8
	Timer status register_0	TSR_0	R/W	H'C0	H'FFFE4305	8
	Timer counter_0	TCNT_0	R/W	H'0000	H'FFFE4306	16
	Timer general register A_0	TGRA_0	R/W	H'FFFF	H'FFFE4308	16
	Timer general register B_0	TGRB_0	R/W	H'FFFF	H'FFFE430A	16
	Timer general register C_0	TGRC_0	R/W	H'FFFF	H'FFFE430C	16
	Timer general register D_0	TGRD_0	R/W	H'FFFF	H'FFFE430E	16
	Timer general register E_0	TGRE_0	R/W	H'FFFF	H'FFFE4320	16
	Timer general register F_0	TGRF_0	R/W	H'FFFF	H'FFFE4322	16
	Timer interrupt enable register 2_0	TIER2_0	R/W	H'00	H'FFFE4324	8
	Timer status register 2_0	TSR2_0	R/W	H'C0	H'FFFE4325	8
	Timer buffer operation transfer mode register_0	TBTM_0	R/W	H'00	H'FFFE4326	8
1	Timer control register_1	TCR_1	R/W	H'00	H'FFFE4380	8
	Timer mode register_1	TMDR_1	R/W	H'00	H'FFFE4381	8
	Timer I/O control register_1	TIOR_1	R/W	H'00	H'FFFE4382	8
	Timer interrupt enable register_1	TIER_1	R/W	H'00	H'FFFE4384	8
	Timer status register_1	TSR_1	R/W	H'C0	H'FFFE4385	8

chanhvinh.com

Channel	Register Name	Abbreviation	R/W	Initial value	Address	Access Size
1	Timer counter_1	TCNT_1	R/W	H'0000	H'FFFE4386	16
	Timer general register A_1	TGRA_1	R/W	H'FFFF	H'FFFE4388	16
	Timer general register B_1	TGRB_1	R/W	H'FFFF	H'FFFE438A	16
	Timer input capture control register	TICCR	R/W	H'00	H'FFFE4390	8
2	Timer control register_2	TCR_2	R/W	H'00	H'FFFE4000	8
	Timer mode register_2	TMDR_2	R/W	H'00	H'FFFE4001	8
	Timer I/O control register_2	TIOR_2	R/W	H'00	H'FFFE4002	8
	Timer interrupt enable register_2	TIER_2	R/W	H'00	H'FFFE4004	8
	Timer status register_2	TSR_2	R/W	H'C0	H'FFFE4005	8
	Timer counter_2	TCNT_2	R/W	H'0000	H'FFFE4006	16
	Timer general register A_2	TGRA_2	R/W	H'FFFF	H'FFFE4008	16
	Timer general register B_2	TGRB_2	R/W	H'FFFF	H'FFFE400A	16
3	Timer control register_3	TCR_3	R/W	H'00	H'FFFE4200	8
	Timer mode register_3	TMDR_3	R/W	H'00	H'FFFE4202	8
	Timer I/O control register H_3	TIORH_3	R/W	H'00	H'FFFE4204	8
	Timer I/O control register L_3	TIORL_3	R/W	H'00	H'FFFE4205	8
	Timer interrupt enable register_3	TIER_3	R/W	H'00	H'FFFE4208	8
	Timer status register_3	TSR_3	R/W	H'C0	H'FFFE422C	8
	Timer counter_3	TCNT_3	R/W	H'0000	H'FFFE4210	16
	Timer general register A_3	TGRA_3	R/W	H'FFFF	H'FFFE4218	16
	Timer general register B_3	TGRB_3	R/W	H'FFFF	H'FFFE421A	16
	Timer general register C_3	TGRC_3	R/W	H'FFFF	H'FFFE4224	16
	Timer general register D_3	TGRD_3	R/W	H'FFFF	H'FFFE4226	16
	Timer buffer operation transfer mode register_3	TBTM_3	R/W	H'00	H'FFFE4238	8
4	Timer control register_4	TCR_4	R/W	H'00	H'FFFE4201	8
	Timer mode register_4	TMDR_4	R/W	H'00	H'FFFE4203	8
	Timer I/O control register H_4	TIORH_4	R/W	H'00	H'FFFE4206	8
	Timer I/O control register L_4	TIORL_4	R/W	H'00	H'FFFE4207	8

chanhvinh.com

Channel	Register Name	Abbreviation	R/W	Initial value	Address	Access Size
4	Timer interrupt enable register_4	TIER_4	R/W	H'00	H'FFFE4209	8
	Timer status register_4	TSR_4	R/W	H'C0	H'FFFE422D	8
	Timer counter_4	TCNT_4	R/W	H'0000	H'FFFE4212	16
	Timer general register A_4	TGRA_4	R/W	H'FFFF	H'FFFE421C	16
	Timer general register B_4	TGRB_4	R/W	H'FFFF	H'FFFE421E	16
	Timer general register C_4	TGRC_4	R/W	H'FFFF	H'FFFE4228	16
	Timer general register D_4	TGRD_4	R/W	H'FFFF	H'FFFE422A	16
	Timer buffer operation transfer mode register_4	TBTM_4	R/W	H'00	H'FFFE4239	8
	Timer A/D converter start request control register	TADCR	R/W	H'0000	H'FFFE4240	16
	Timer A/D converter start request cycle set register A_4	TADCORA_4	R/W	H'FFFF	H'FFFE4244	16
	Timer A/D converter start request cycle set register B_4	TADCORB_4	R/W	H'FFFF	H'FFFE4246	16
	Timer A/D converter start request cycle set buffer register A_4	TADCOBRA_4	R/W	H'FFFF	H'FFFE4248	16
	Timer A/D converter start request cycle set buffer register B_4	TADCOBRB_4	R/W	H'FFFF	H'FFFE424A	16
Common	Timer start register	TSTR	R/W	H'00	H'FFFE4280	8
	Timer synchronous register	TSYR	R/W	H'00	H'FFFE4281	8
	Timer read/write enable register	TRWER	R/W	H'01	H'FFFE4284	8

chanhvinh.com

Channel	Register Name	Abbreviation	R/W	Initial value	Address	Access Size
Common to 3 and 4	Timer output master enable register	TOER	R/W	H'C0	H'FFFE420A	8
	Timer output control register 1	TOCR1	R/W	H'00	H'FFFE420E	8
	Timer output control register 2	TOCR2	R/W	H'00	H'FFFE420F	8
	Timer gate control register	TGCR	R/W	H80	H'FFFE420D	8
	Timer cycle data register	TCDR	R/W	H'FFFF	H'FFFE4214	16
	Timer dead time data register	TDDR	R/W	H'FFFF	H'FFFE4216	16
	Timer subcounter	TCNTS	R	H'0000	H'FFFE4220	16
	Timer cycle buffer register	TCBR	R/W	H'FFFF	H'FFFE4222	16
	Timer interrupt skipping set register	TITCR	R/W	H'00	H'FFFE4230	8
	Timer interrupt skipping counter	TITCNT	R	H'00	H'FFFE4231	8
	Timer buffer transfer set register	TBTER	R/W	H'00	H'FFFE4232	8
	Timer dead time enable register	TDER	R/W	H'01	H'FFFE4234	8
	Timer waveform control register	TWCR	R/W	H'00	H'FFFE4260	8
	Timer output level buffer register	TOLBR	R/W	H'00	H'FFFE4236	8

chanhvinh.com

11.3.1 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:	7	6	5	4	3	2	1	0
	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]		
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 5	CCLR[2:0]	000	R/W	Counter Clear 0 to 2 These bits select the TCNT counter clearing source. See tables 11.4 and 11.5 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 tables 11.6 to 11.9 for details.

[Legend]
x: Don't care

chanhvinh.com

Table 11.4 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]

Notes: 1. Synchronous operation is set by setting the SYNC bit in TSYR to 1.
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.

Table 11.5 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]

Notes: 1. Synchronous operation is selected by setting the SYNC bit in TSYR to 1.
2. Bit 7 is reserved in channels 1 and 2. It is always read as 0 and cannot be modified.

chanhvinh.com

Table 11.6 TPSC0 to TPSC2 (Channel 0)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
0	0	0	0	Internal clock: counts on Pϕ/1
			1	Internal clock: counts on Pϕ/4
		1	0	Internal clock: counts on Pϕ/16
			1	Internal clock: counts on Pϕ/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 11.7 TPSC0 to TPSC2 (Channel 1)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
1	0	0	0	Internal clock: counts on Pϕ/1
			1	Internal clock: counts on Pϕ/4
		1	0	Internal clock: counts on Pϕ/16
			1	Internal clock: counts on Pϕ/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 Pϕ/256
			1	Counts on TCNT_2 overflow/underflow

Note: This setting is ignored when channel 1 is in phase counting mode.

chanhvinh.com

Table 11.8 TPSC0 to TPSC2 (Channel 2)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
2	0	0	0	Internal clock: counts on Pϕ/1
			1	Internal clock: counts on Pϕ/4
		1	0	Internal clock: counts on Pϕ/16
			1	Internal clock: counts on Pϕ/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 Pϕ/1024

Note: This setting is ignored when channel 2 is in phase counting mode.

Table 11.9 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 Pϕ/1
			1	Internal clock: counts on Pϕ/4
		1	0	Internal clock: counts on Pϕ/16
			1	Internal clock: counts on Pϕ/64
	1	0	0	Internal clock: counts on Pϕ/256
			1	Internal clock: counts on Pϕ/1024
		1	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input

chanhvinh.com

11.3.2 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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 11.10 for details.

chanhvinh.com

Table 11.10 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 2 (transmit at crest and trough)*[3]

[Legend]

X: Don't care

Notes: 1. PWM mode 2 cannot be set for channels 3 and 4.

2. Phase counting mode cannot be set for channels 0, 3, and 4.

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

11.3.3 Timer I/O Control Register (TIOR)

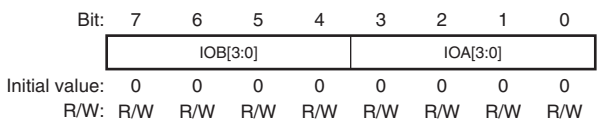
The TIOR registers are 8-bit readable/writable registers that control the TGR registers. This module has a total of eight TIOR registers, two each for channels 0, 3, and 4, one each for channels 1 and 2.

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.

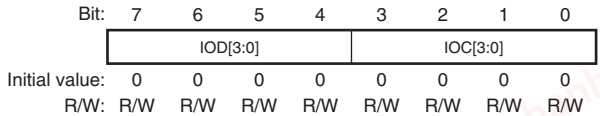
- TIORH_0, TIOR_1, TIOR_2, TIORH_3, TIORH_4



Bit	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 11.11 TIOR_1: Table 11.13 TIOR_2: Table 11.14 TIORH_3: Table 11.15 TIORH_4: Table 11.17
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 11.19 TIOR_1: Table 11.21 TIOR_2: Table 11.22 TIORH_3: Table 11.23 TIORH_4: Table 11.25

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 11.12 TIORL_3: Table 11.16 TIORL_4: Table 11.18
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 11.20 TIORL_3: Table 11.24 TIORL_4: Table 11.26

chanhvinh.com

Table 11.11 TIORH_0 (Channel 0)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**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 11.12 TIORL_0 (Channel 0)

				Description	
Bit 7 IOD3	**Bit 6 IOD2**	**Bit 5 IOD1**	**Bit 4 IOD0**	**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

Notes: 1. After power-on reset, 0 is output until TIOR is set.

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 11.13 TIOR_1 (Channel 1)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**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 11.14 TIOR_2 (Channel 2)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**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 11.15 TIORH_3 (Channel 3)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**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 11.16 TIORL_3 (Channel 3)

				Description	
Bit 7 IOD3	**Bit 6 IOD2**	**Bit 5 IOD1**	**Bit 4 IOD0**	**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

Notes: 1. After power-on reset, 0 is output until TIOR is set.

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 11.17 TIORH_4 (Channel 4)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**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 11.18 TIORL_4 (Channel 4)

				Description	
Bit 7 IOD3	**Bit 6 IOD2**	**Bit 5 IOD1**	**Bit 4 IOD0**	**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

Notes: 1. After power-on reset, 0 is output until TIOR is set.

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 11.19 TIORH_0 (Channel 0)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**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 11.20 TIORL_0 (Channel 0)

				Description	
Bit 3 IOC3	**Bit 2 IOC2**	**Bit 1 IOC1**	**Bit 0 IOC0**	**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

Notes: 1. After power-on reset, 0 is output until TIOR is set.

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 11.21 TIOR_1 (Channel 1)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**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 11.22 TIOR_2 (Channel 2)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**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 11.23 TIORH_3 (Channel 3)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**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 11.24 TIORL_3 (Channel 3)

				Description	
Bit 3 IOC3	**Bit 2 IOC2**	**Bit 1 IOC1**	**Bit 0 IOC0**	**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

Notes: 1. After power-on reset, 0 is output until TIOR is set.

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 11.25 TIORH_4 (Channel 4)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**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 11.26 TIORL_4 (Channel 4)

				Description	
Bit 3 IOC3	**Bit 2 IOC2**	**Bit 1 IOC1**	**Bit 0 IOC0**	**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

Notes: 1. After power-on reset, 0 is output until TIOR is set.

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

11.3.4 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 Converter Start Request Enable Enables or disables generation of A/D converter 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 Converter Start Request Enable 2 Enables or disables generation of A/D converter 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 converter start request generation by TCNT_4 underflow (trough) disabled 1: A/D converter 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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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
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

chanhvinh.com

- 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

11.3.5 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)

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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

Notes: 1. Writing 0 to this bit after reading it as 1 clears the flag.
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

Notes: 1. Writing 0 to this bit after reading it as 1 clears the flag.
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

11.3.6 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. 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. 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. 0: When compare match A occurs in each channel 1: When TCNT is cleared in each channel

chanhvinh.com

11.3.7 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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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

11.3.8 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

Note: * Do not set to 1 when complementary PWM mode is not selected.

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 11.27.
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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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
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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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

Notes: 1. TADCR must not be accessed in eight bits; it should always be accessed in 16 bits.
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).
3. If link with interrupt skipping is enabled while interrupt skipping is disabled, A/D converter start requests will not be issued.
* Do not set to 1 when complementary PWM mode is not selected.

Table 11.27 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]

Notes: 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.
2. These settings are prohibited when complementary PWM mode is not selected.

chanhvinh.com

11.3.9 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.

11.3.10 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.

chanhvinh.com

11.3.11 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.

11.3.12 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

11.3.13 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

11.3.14 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
6	SYNC3	0	R/W	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
5 to 3	—	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
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

11.3.15 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 TCNT4.

chanhvinh.com

11.3.16 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.

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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 11.3.17, Timer Output Control Register 1 (TOCR1), and section 11.3.18, 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

11.3.17 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.

chanhvinh.com

Bit	Bit Name	Initial value	R/W	Description
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 This bit selects the reverse phase output level in reset-synchronized PWM mode/complementary PWM mode. See table 11.28.
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 11.29.

Notes: 1. Setting the TOCL bit to 1 prevents accidental modification when the CPU goes out of control.
2. Clearing the TOCS0 bit to 0 makes this bit setting valid.
3. After power-on reset, 1 can be written only once. After 1 has been written, 0 cannot be written.

Table 11.28 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 reverse phase waveform initial output value changes to active level after elapse of the dead time after count start.

chanhvinh.com

Table 11.29 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 11.2 shows an example of complementary PWM mode output (1 phase) when OLSN = 1, OLSP = 1.

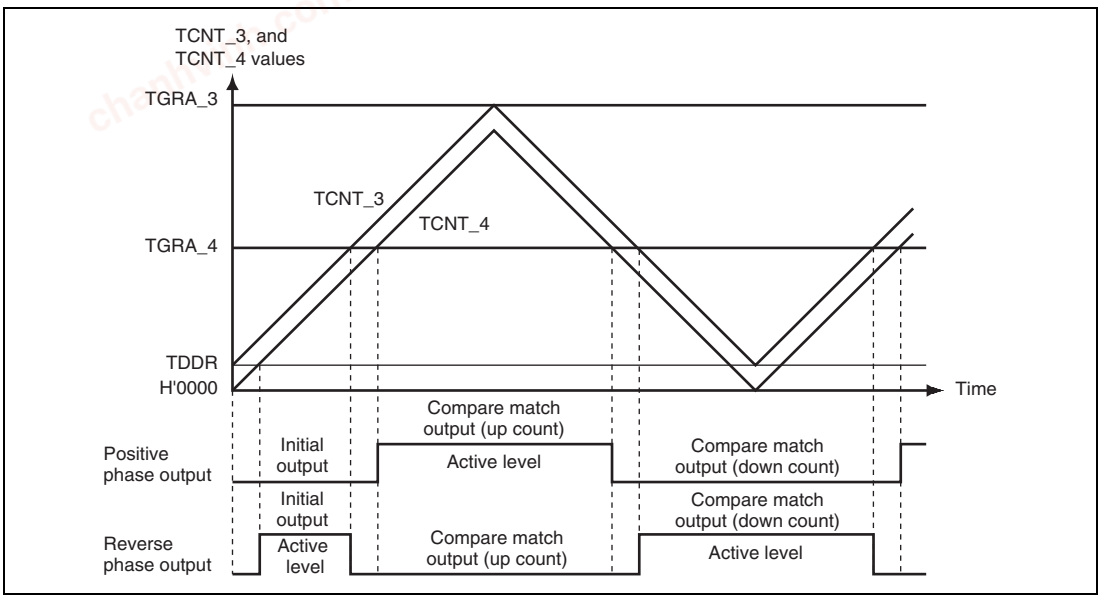


Figure 11.2 Complementary PWM Mode Output Level Example

chanhvinh.com

11.3.18 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 11.30.
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 11.31.
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 11.32.
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 11.33.
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 11.34.
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 11.35.

chanhvinh.com

Bit	Bit Name	Initial value	R/W	Description
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 11.36.

Note: * Setting the TOCS bit in TOCR1 to 1 makes this bit setting valid.

Table 11.30 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 11.31 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 reverse phase waveform initial output value changes to the active level after elapse of the dead time after count start.

chanhvinh.com

Table 11.32 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 11.33 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 reverse phase waveform initial output value changes to the active level after elapse of the dead time after count start.

Table 11.34 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

Table 11.35 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 reverse phase waveform initial output value changes to the active level after elapse of the dead time after count start.

chanhvinh.com

Table 11.36 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

11.3.19 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.

chanhvinh.com

Figure 11.3 shows an example of the PWM output level setting procedure in buffer operation.

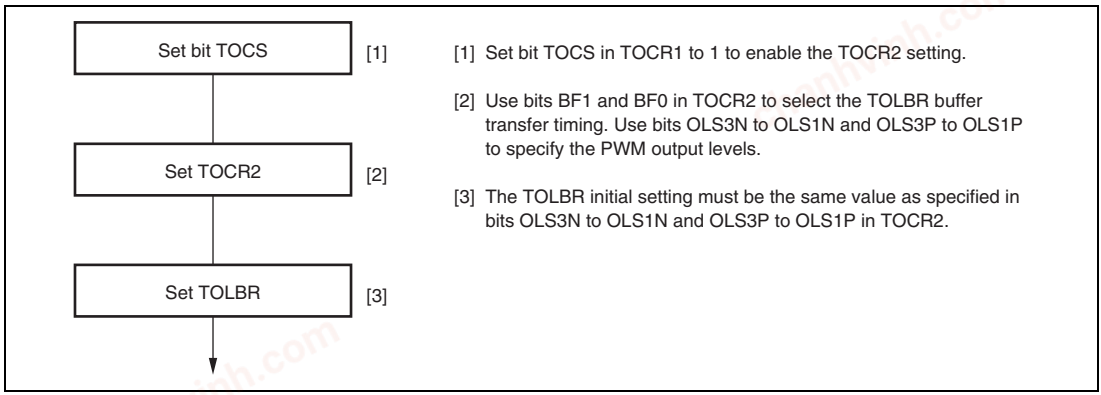


Figure 11.3 PWM Output Level Setting Procedure in Buffer Operation

11.3.20 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

chanhvinh.com

Bit	Bit Name	Initial value	R/W	Description
5	N	0	R/W	Reverse 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 pin (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/reverse 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 11.37.
1	VF	0	R/W	
0	UF	0	R/W	

chanhvinh.com

Table 11.37 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

11.3.21 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.

chanhvinh.com

11.3.22 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.

11.3.23 Timer Cycle Data Register (TCDR)

TCDR is a 16-bit register used only in complementary PWM mode. Set half the PWM carrier sync value 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.

chanhvinh.com

11.3.24 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.

11.3.25 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:	7	6	5	4	3	2	1	0
	T3AEN	3ACOR[2:0]			T4VEN	4VCOR[2:0]		
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	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 11.38.
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

chanhvinh.com

Bit	Bit Name	Initial value	R/W	Description
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 11.39.

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 (TICNT).

Table 11.38 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 11.39 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

11.3.26 Timer Interrupt Skipping Counter (TITCNT)

TITCNT is an 8-bit readable/writable 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 4VCOR2 value in TITCR • When the T4VEN bit in TITCR is cleared to 0 • When the 4VCOR2 to 4VCOR2 bits in TITCR are cleared to 0

Note: To clear the TITCNT, clear the bits T3AEN and T4VEN in TITCR to 0.

chanhvinh.com

11.3.27 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 11.40.

Note: * Applicable buffer registers:
TGRC_3, TGRD_3, TGRC_4, TGRD_4, and TCBR

chanhvinh.com

Table 11.40 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

Notes: 1. Data is transferred according to the MD3 to MD0 bit setting in TMDR. For details, refer to section 11.4.8, Complementary PWM Mode.

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

11.3.28 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

11.3.29 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)

Note: * Do not set to 1 when complementary PWM mode is not selected.

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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 11.40. 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 is not selected.

11.3.30 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. A 16-bit data bus to the bus master enables 16-bit read/writes. 8-bit read/write is not possible. Always access in 16-bit units.

All registers other than the above registers are 8-bit registers. These are connected to the CPU by a 16-bit data bus, so 16-bit read/writes and 8-bit read/writes are both possible.

chanhvinh.com

11.4 Operation

11.4.1 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 11.4 shows an example of the count operation setting procedure.

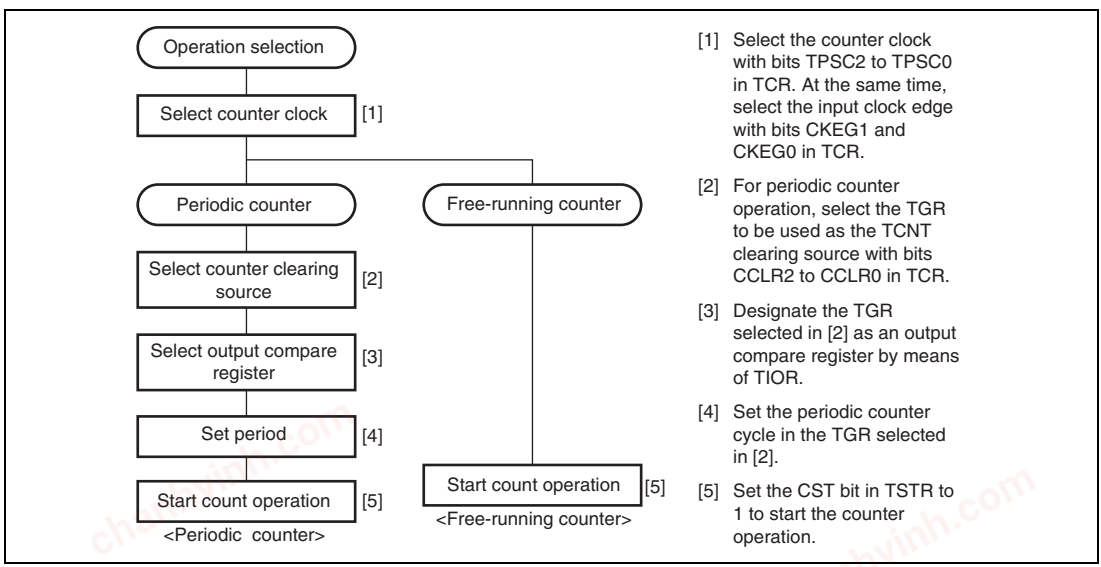


Figure 11.4 Example of Counter Operation Setting Procedure

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 11.5 illustrates free-running counter operation.

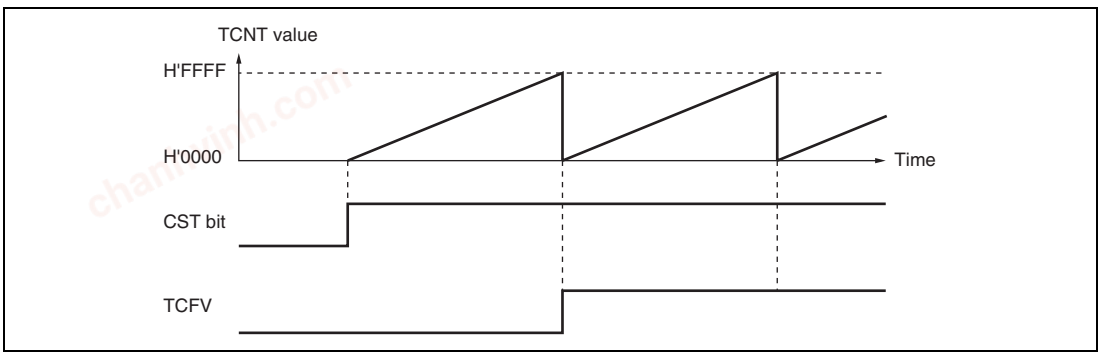


Figure 11.5 Free-Running Counter Operation

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.

chanhvinh.com

Figure 11.6 illustrates periodic counter operation.

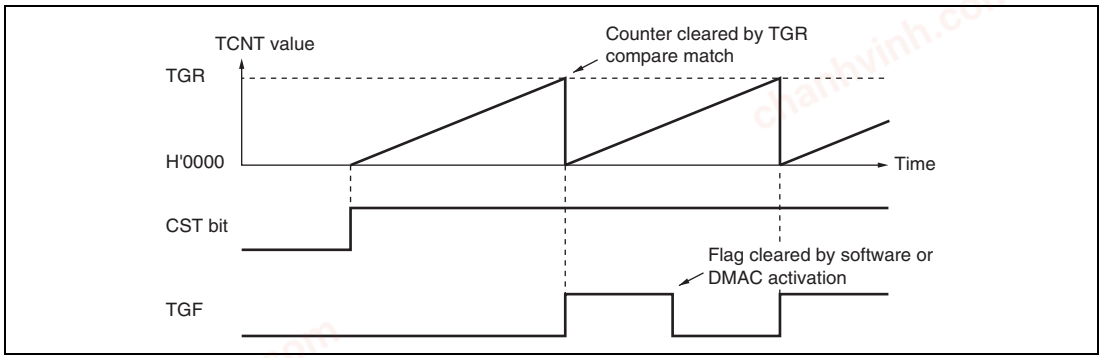


Figure 11.6 Periodic Counter Operation

(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 11.7 shows an example of the setting procedure for waveform output by compare match

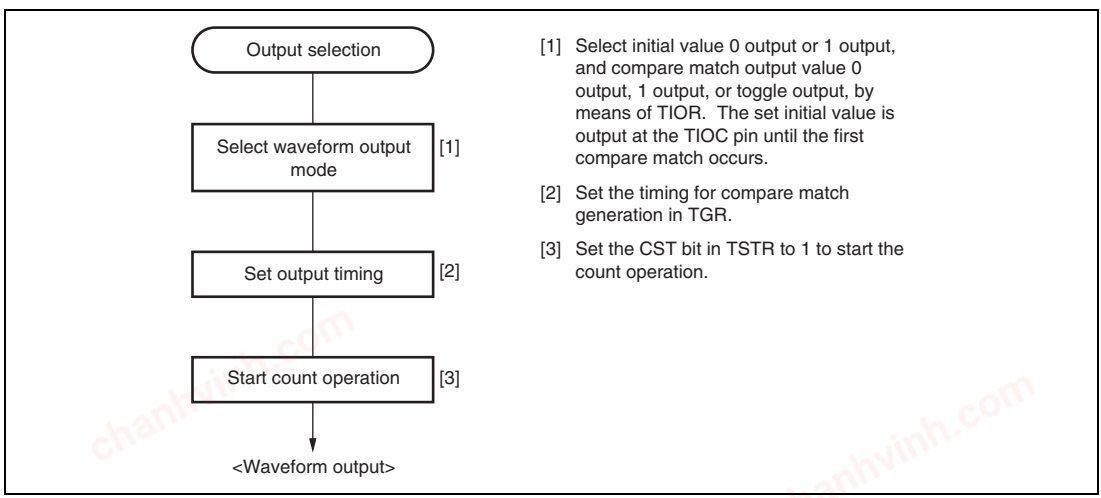


Figure 11.7 Example of Setting Procedure for Waveform Output by Compare Match

chanhvinh.com

(b) Examples of Waveform Output Operation:

Figure 11.8 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.

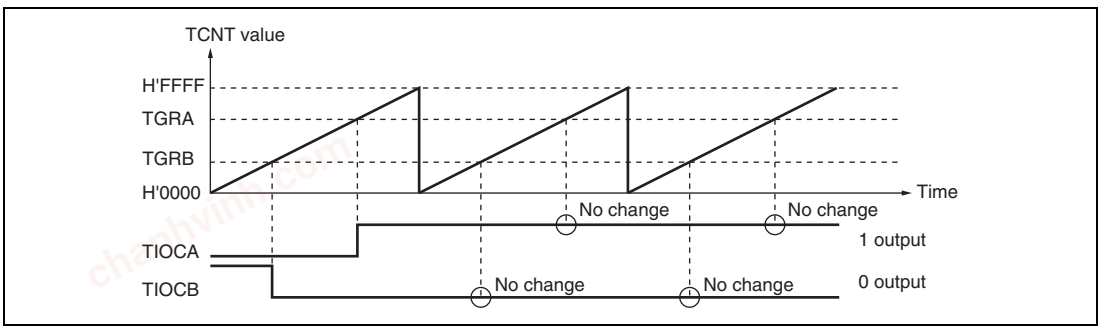


Figure 11.8 Example of 0 Output/1 Output Operation

Figure 11.9 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.

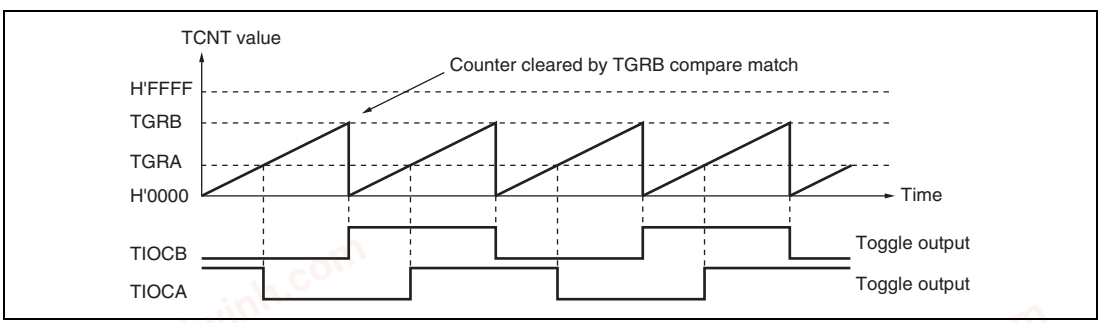


Figure 11.9 Example of Toggle Output Operation

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 11.10 shows an example of the input capture operation setting procedure.

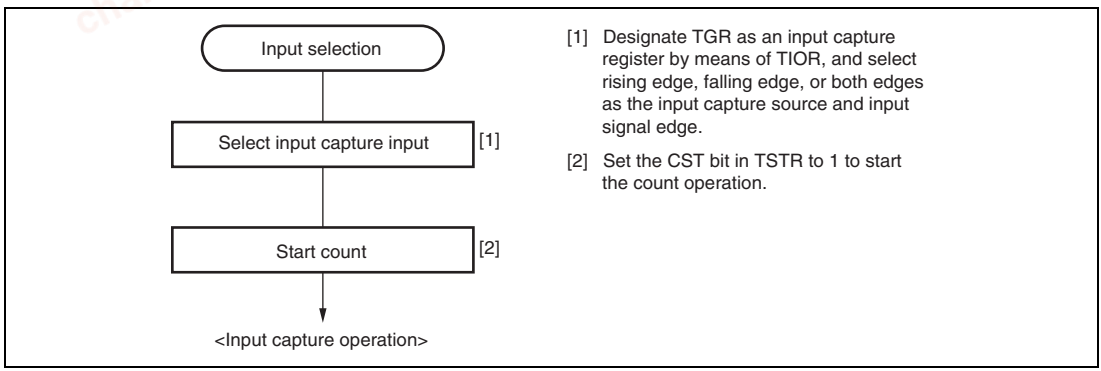


Figure 11.10 Example of Input Capture Operation Setting Procedure

chanhvinh.com

(b) Example of Input Capture Operation

Figure 11.11 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.

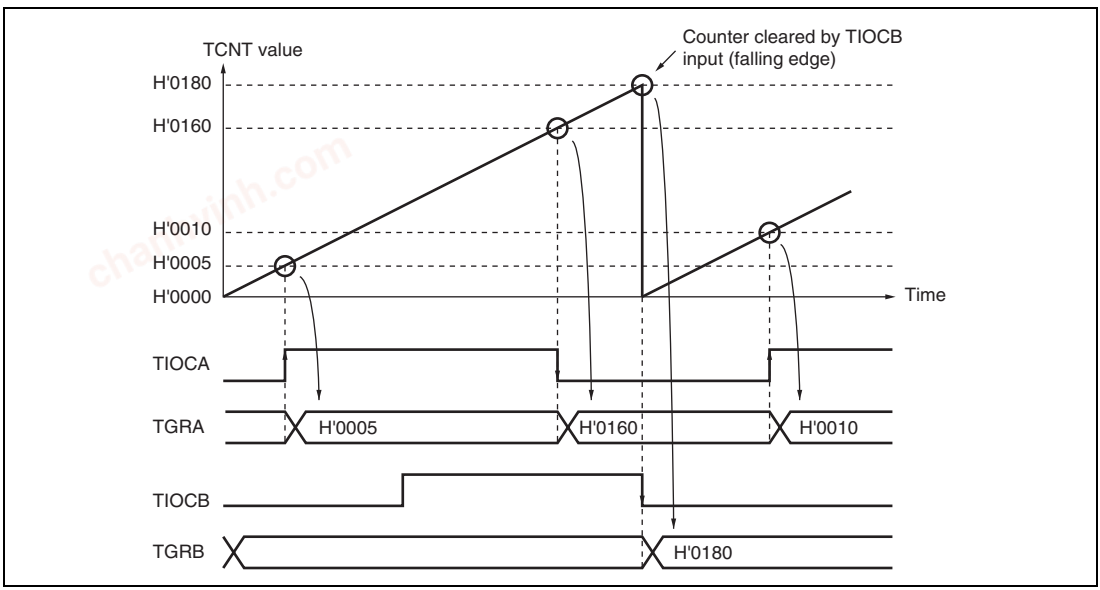


Figure 11.11 Example of Input Capture Operation

chanhvinh.com

11.4.2 Synchronous Operation

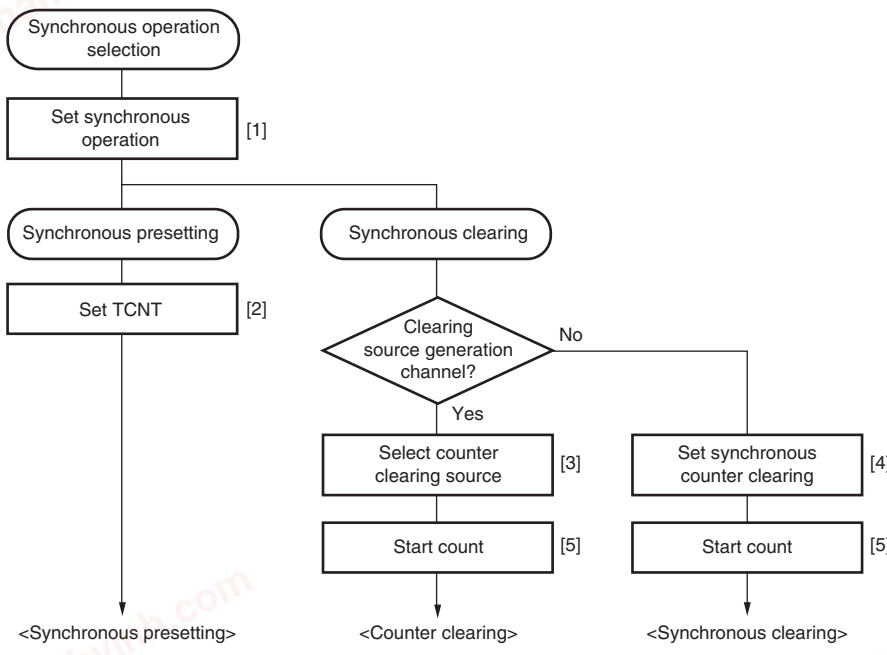
In synchronous operation, the values in a number of TCNT counters can be rewritten simultaneously (synchronous presetting). Also, a number of TCNT counters can be cleared simultaneously by making the appropriate setting in TCR (synchronous clearing).

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. Channel 5 cannot be used for synchronous operation.

(1) Example of Synchronous Operation Setting Procedure

Figure 11.12 shows an example of the synchronous operation setting procedure.



[1] Set to 1 the SYNC bits in TSYR corresponding to the channels to be designated for synchronous operation.
[2] When the TCNT counter of any of the channels designated for synchronous operation is written to, the same value is simultaneously written to the other TCNT counters.
[3] Use bits CCLR2 to CCLR0 in TCR to specify TCNT clearing by input capture/output compare, etc.
[4] Use bits CCLR2 to CCLR0 in TCR to designate synchronous clearing for the counter clearing source.
[5] Set to 1 the CST bits in TSTR for the relevant channels, to start the count operation.

Figure 11.12 Example of Synchronous Operation Setting Procedure

chanhvinh.com

(2) Example of Synchronous Operation

Figure 11.13 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 11.4.5, PWM Modes.

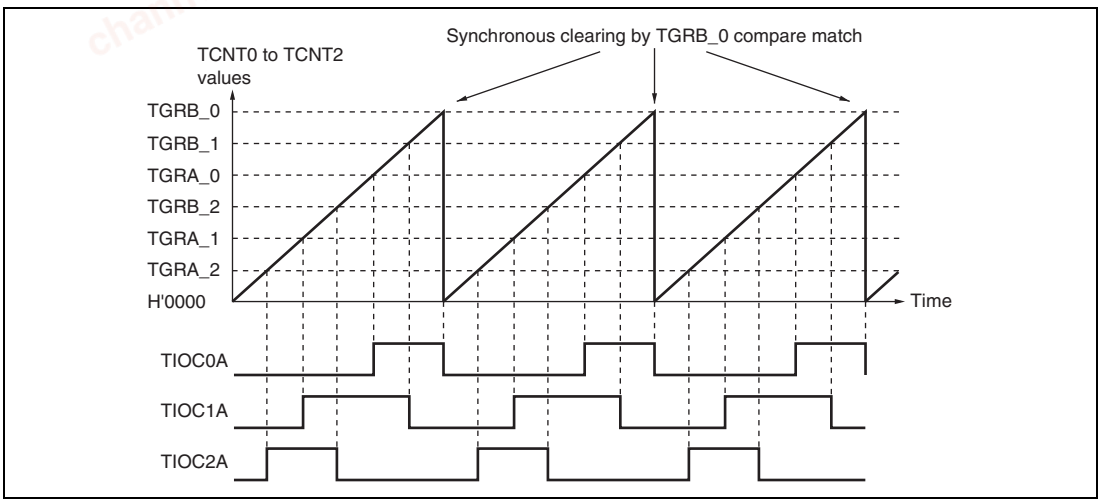


Figure 11.13 Example of Synchronous Operation

chanhvinh.com

11.4.3 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 11.41 shows the register combinations used in buffer operation.

Table 11.41 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 11.14.

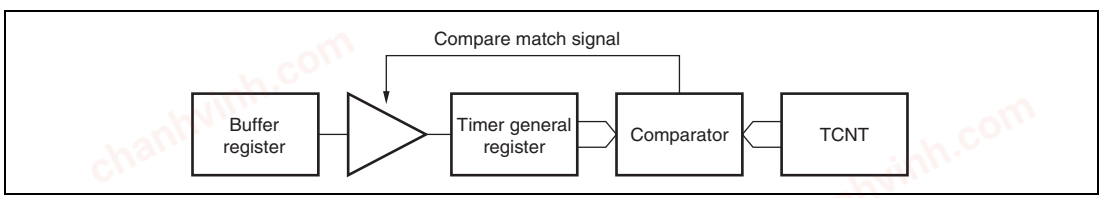


Figure 11.14 Compare Match Buffer Operation

chanhvinh.com

- 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 11.15.

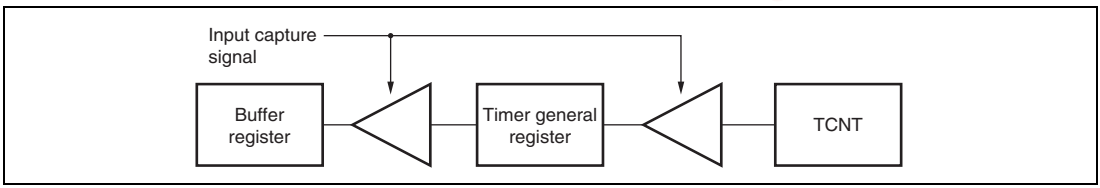


Figure 11.15 Input Capture Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 11.16 shows an example of the buffer operation setting procedure.

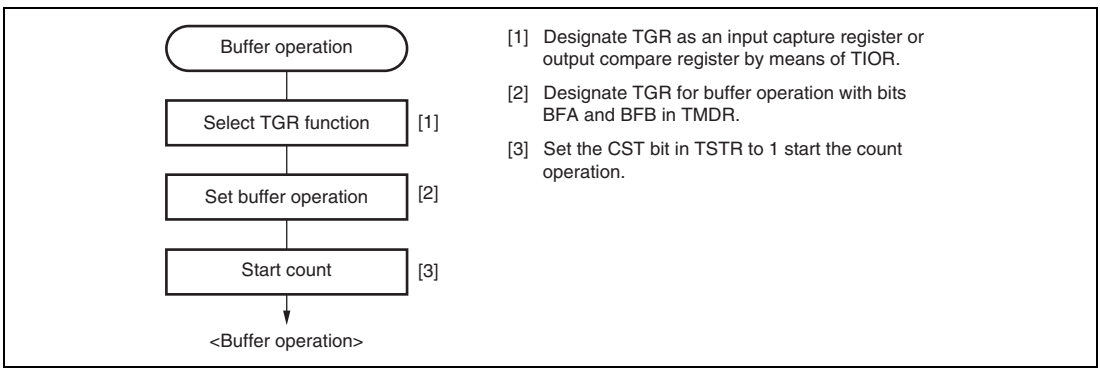


Figure 11.16 Example of Buffer Operation Setting Procedure

chanhvinh.com

(2) Examples of Buffer Operation

(a) When TGR is an output compare register

Figure 11.17 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 11.4.5, PWM Modes.

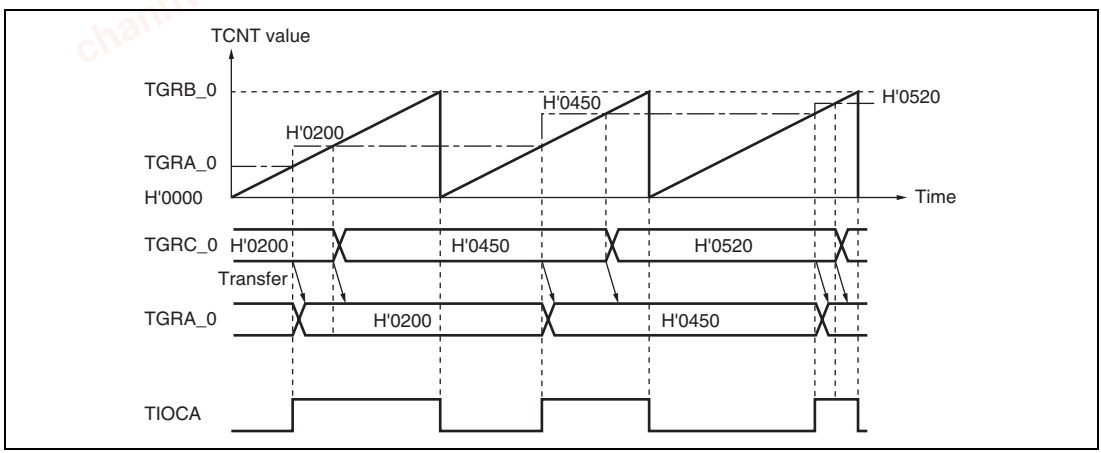


Figure 11.17 Example of Buffer Operation (1)

(b) When TGR is an input capture register

Figure 11.18 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.

chanhvinh.com

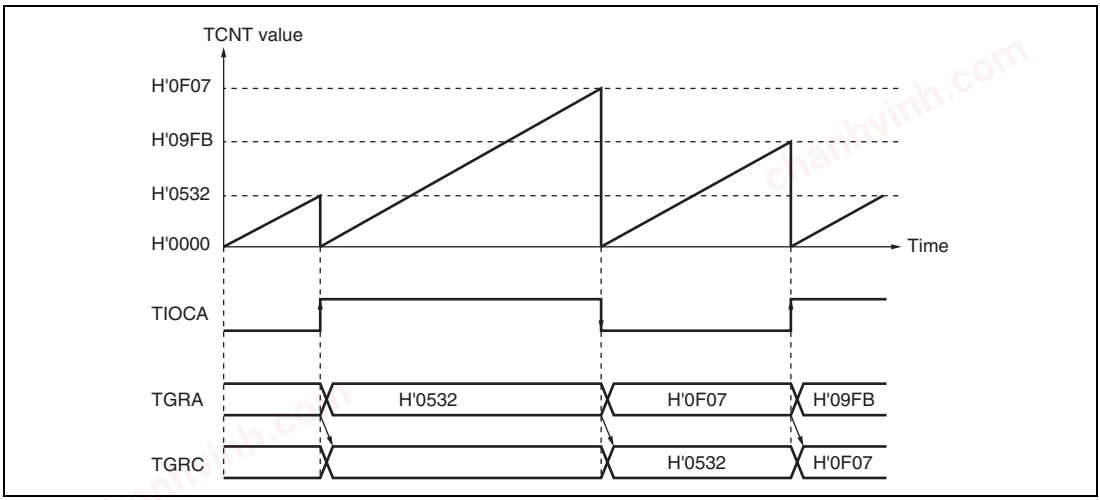


Figure 11.18 Example of Buffer Operation (2)

(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 11.19 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.

chanhvinh.com

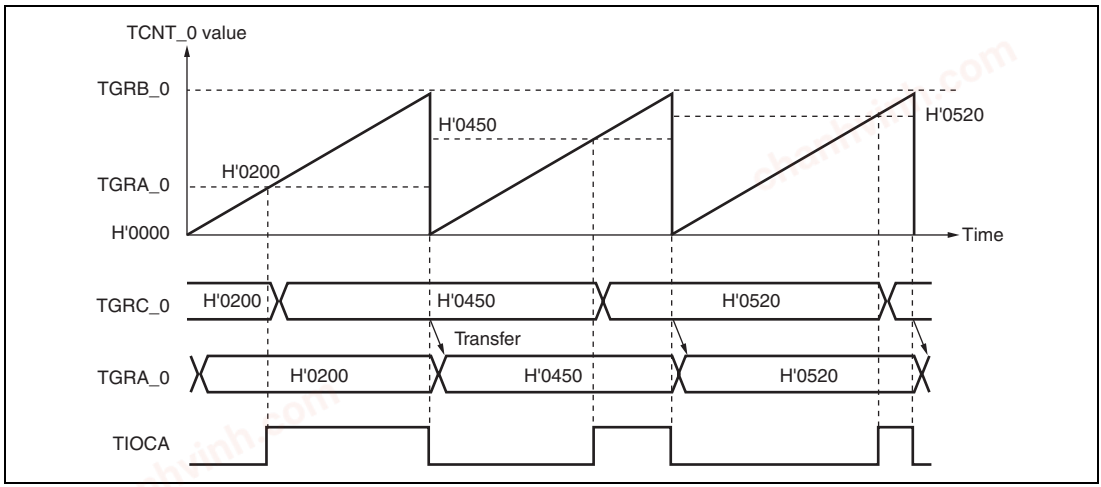


Figure 11.19 Example of Buffer Operation When TCNT_0 Clearing is Selected for TGRC_0 to TGRA_0 Transfer Timing

11.4.4 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 11.42 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 11.42 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). For input capture in cascade connection, refer to section 11.7.22, Simultaneous Capture of TCNT_1 and TCNT_2 in Cascade Connection.

chanhvinh.com

Table 11.43 show the TICCR setting and input capture input pins.

Table 11.43 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

(1) Example of Cascaded Operation Setting Procedure

Figure 11.20 shows an example of the setting procedure for cascaded operation.

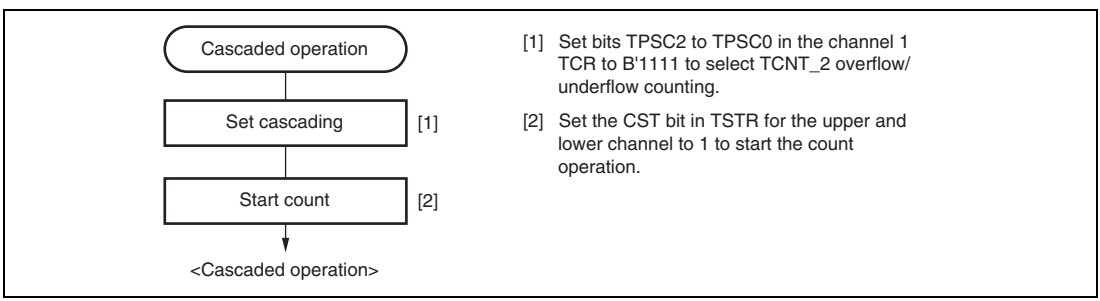


Figure 11.20 Cascaded Operation Setting Procedure

(2) Cascaded Operation Example (a)

Figure 11.21 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.

chanhvinh.com

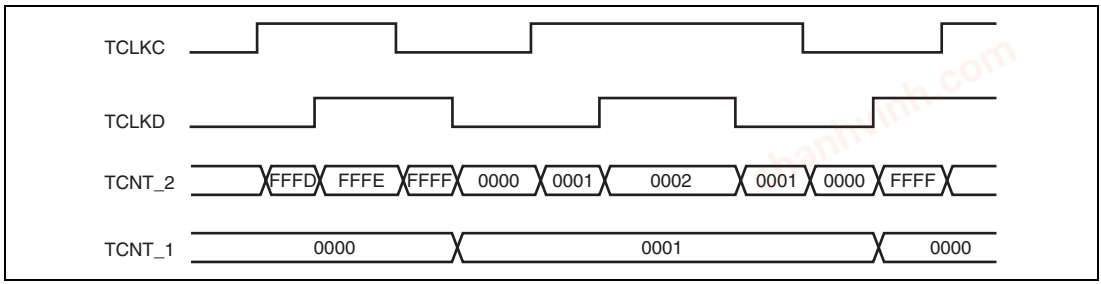


Figure 11.21 Cascaded Operation Example (a)

(3) Cascaded Operation Example (b)

Figure 11.22 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.

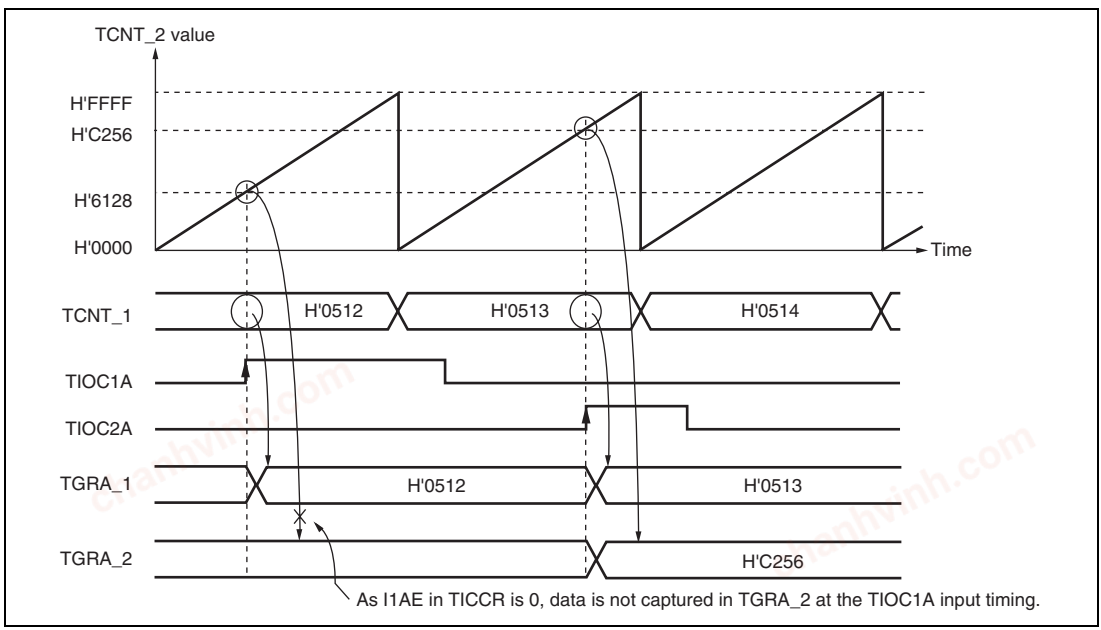


Figure 11.22 Cascaded Operation Example (b)

chanhvinh.com

(4) Cascaded Operation Example (c)

Figure 11.23 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.

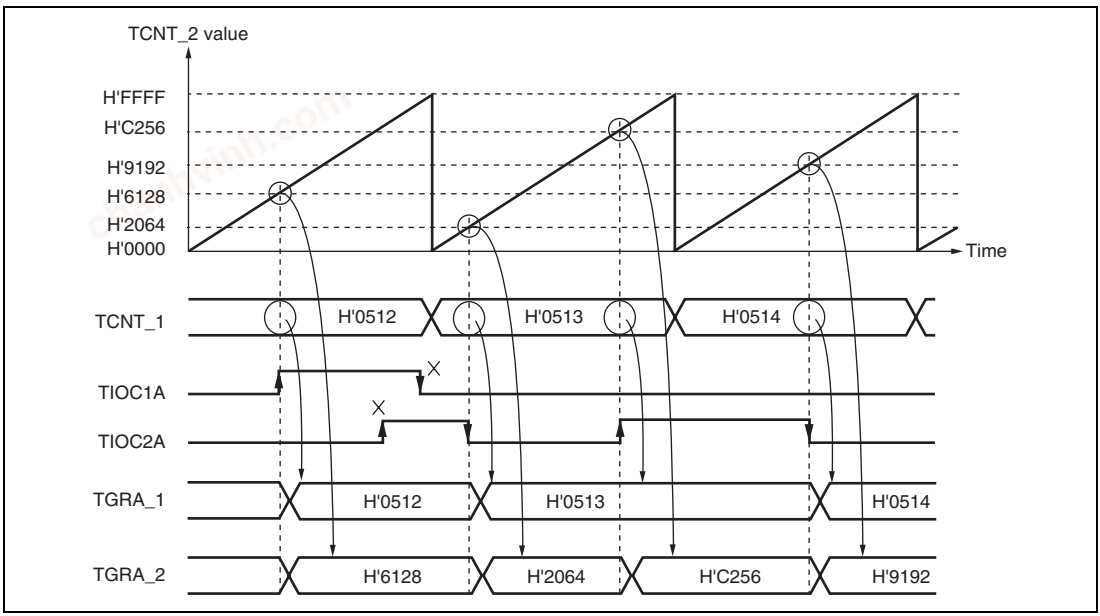


Figure 11.23 Cascaded Operation Example (c)

chanhvinh.com

(5) Cascaded Operation Example (d)

Figure 11.24 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.

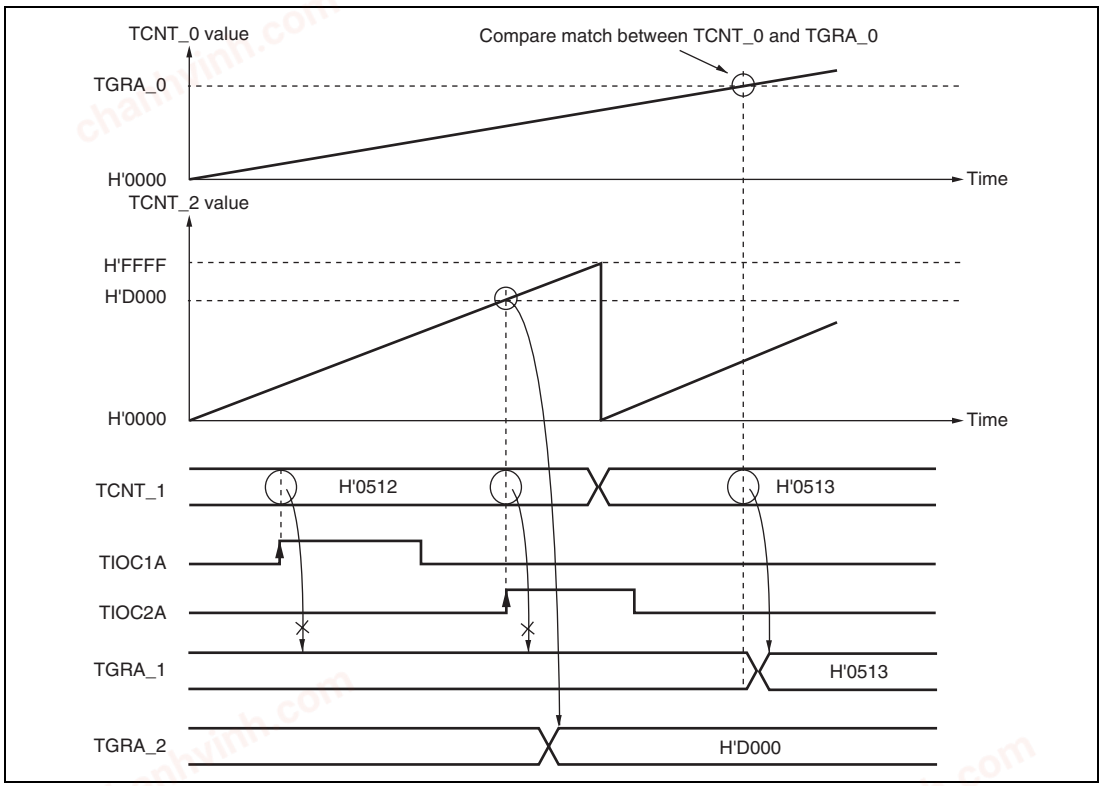


Figure 11.24 Cascaded Operation Example (d)

chanhvinh.com

11.4.5 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 synchronization 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 11.44.

chanhvinh.com

Table 11.44 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 11.25 shows an example of the PWM mode setting procedure.

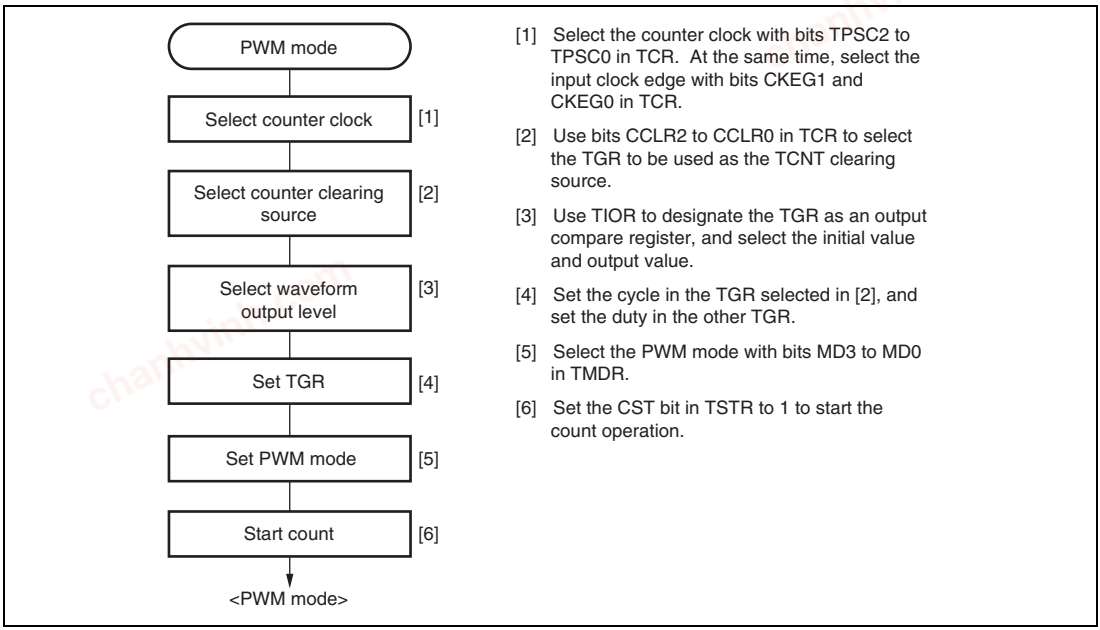


Figure 11.25 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 11.26 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.

chanhvinh.com

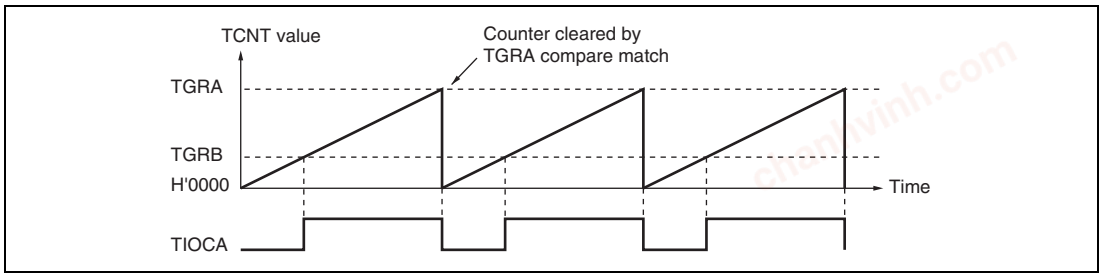


Figure 11.26 Example of PWM Mode Operation (1)

Figure 11.27 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.

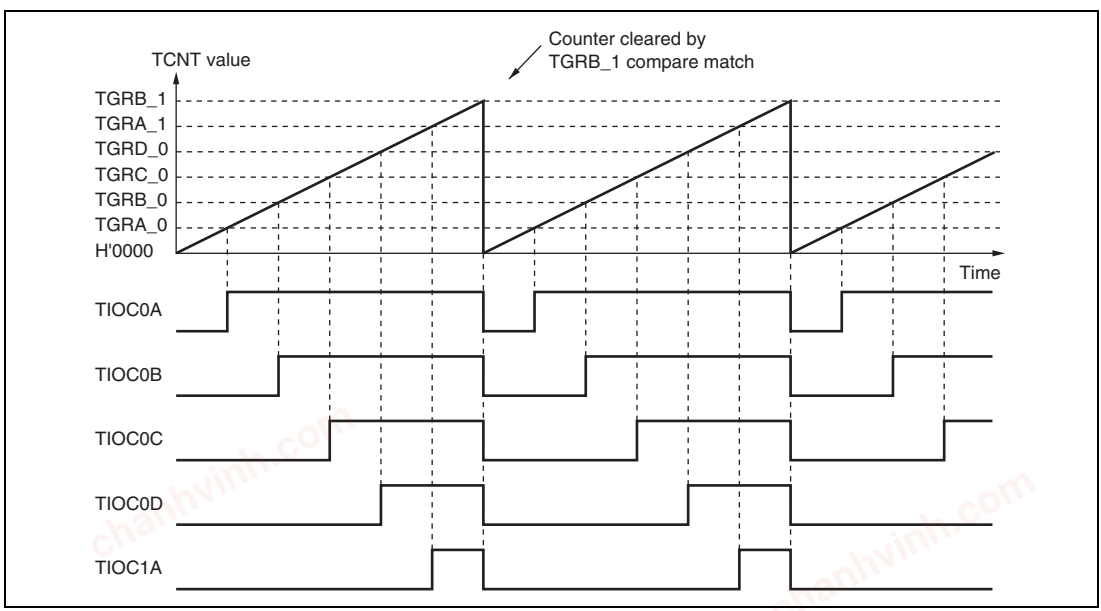


Figure 11.27 Example of PWM Mode Operation (2)

chanhvinh.com

Figure 11.28 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode.

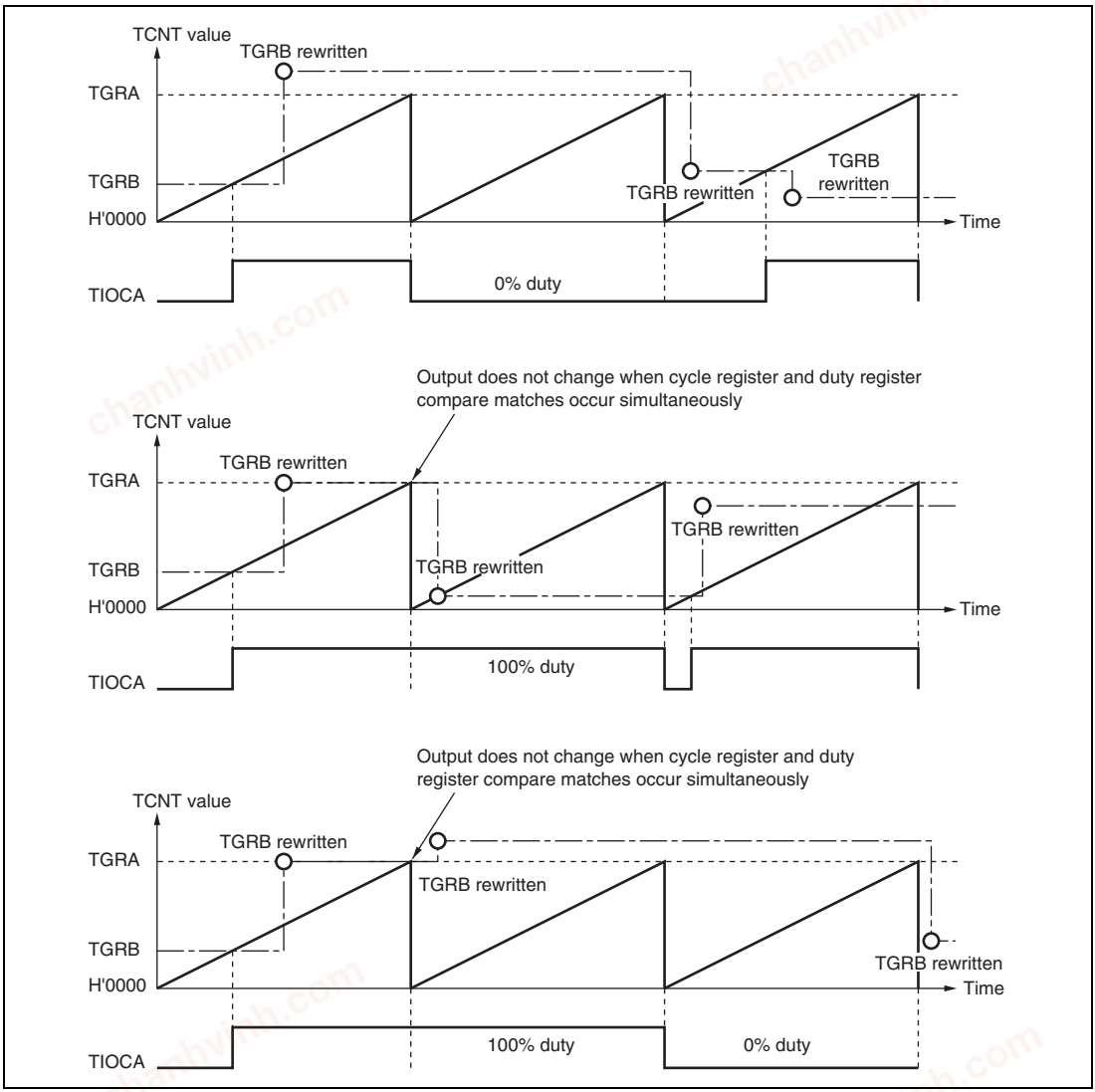


Figure 11.28 Example of PWM Mode Operation (3)

chanhvinh.com

11.4.6 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 11.45 shows the correspondence between external clock pins and channels.

Table 11.45 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 11.29 shows an example of the phase counting mode setting procedure.

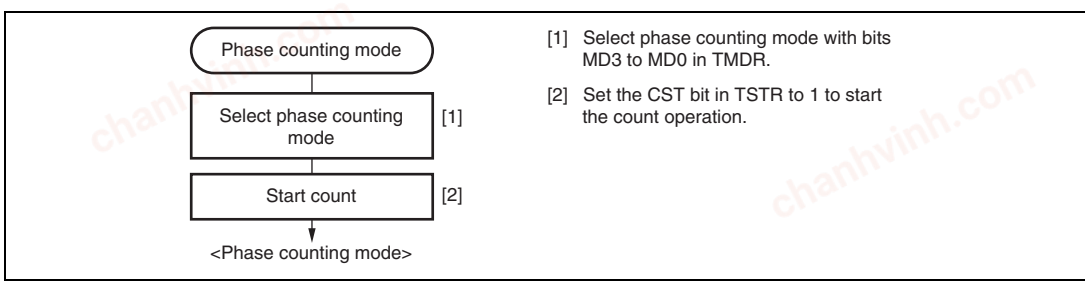


Figure 11.29 Example of Phase Counting Mode Setting Procedure

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 11.30 shows an example of phase counting mode 1 operation, and table 11.46 summarizes the TCNT up/down-count conditions.

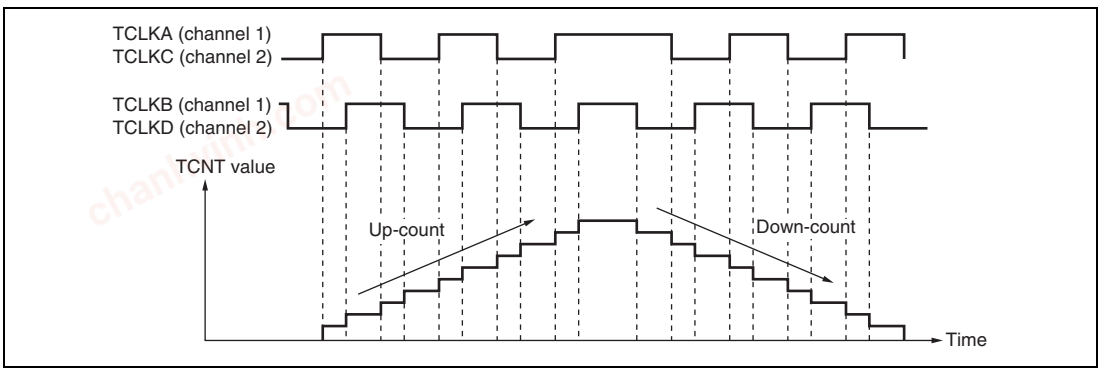


Figure 11.30 Example of Phase Counting Mode 1 Operation

Table 11.46 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 11.31 shows an example of phase counting mode 2 operation, and table 11.47 summarizes the TCNT up/down-count conditions.

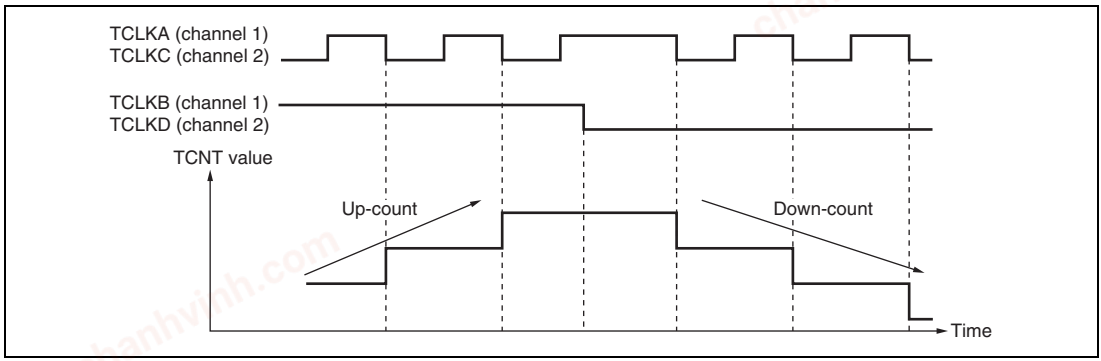


Figure 11.31 Example of Phase Counting Mode 2 Operation

Table 11.47 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 11.32 shows an example of phase counting mode 3 operation, and table 11.48 summarizes the TCNT up/down-count conditions.

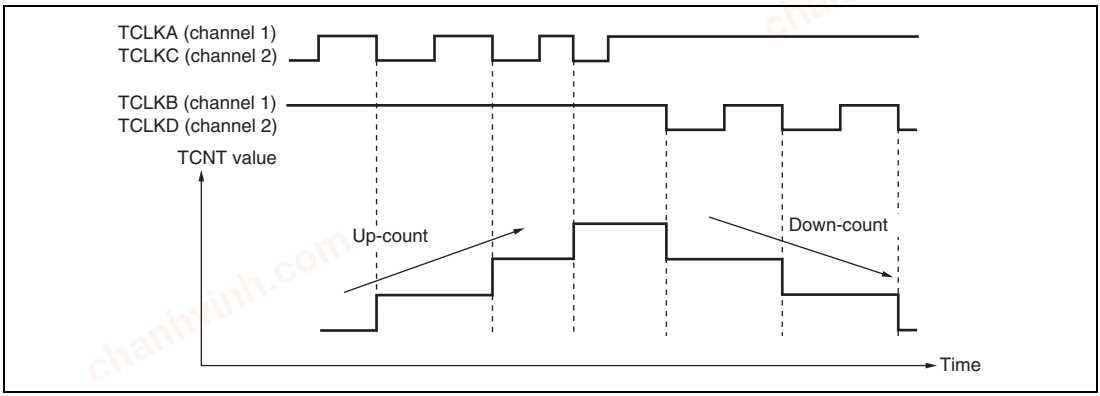


Figure 11.32 Example of Phase Counting Mode 3 Operation

Table 11.48 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 11.33 shows an example of phase counting mode 4 operation, and table 11.49 summarizes the TCNT up/down-count conditions.

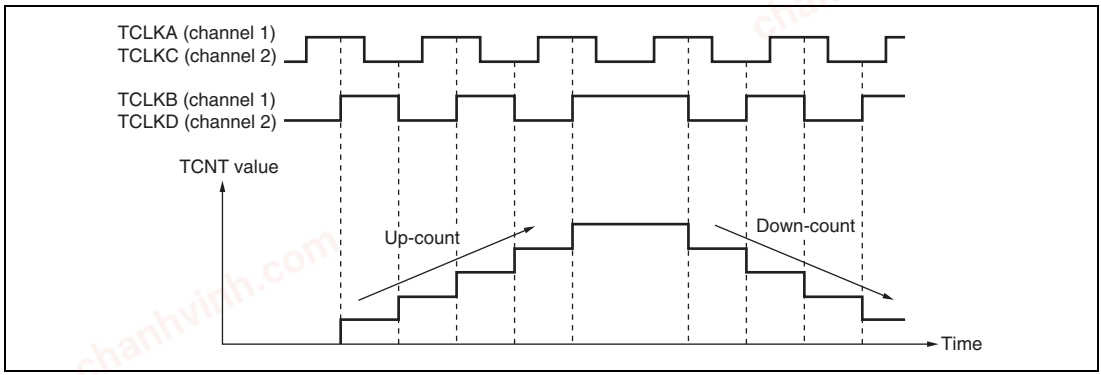


Figure 11.33 Example of Phase Counting Mode 4 Operation

Table 11.49 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 11.34 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.

chanhvinh.com

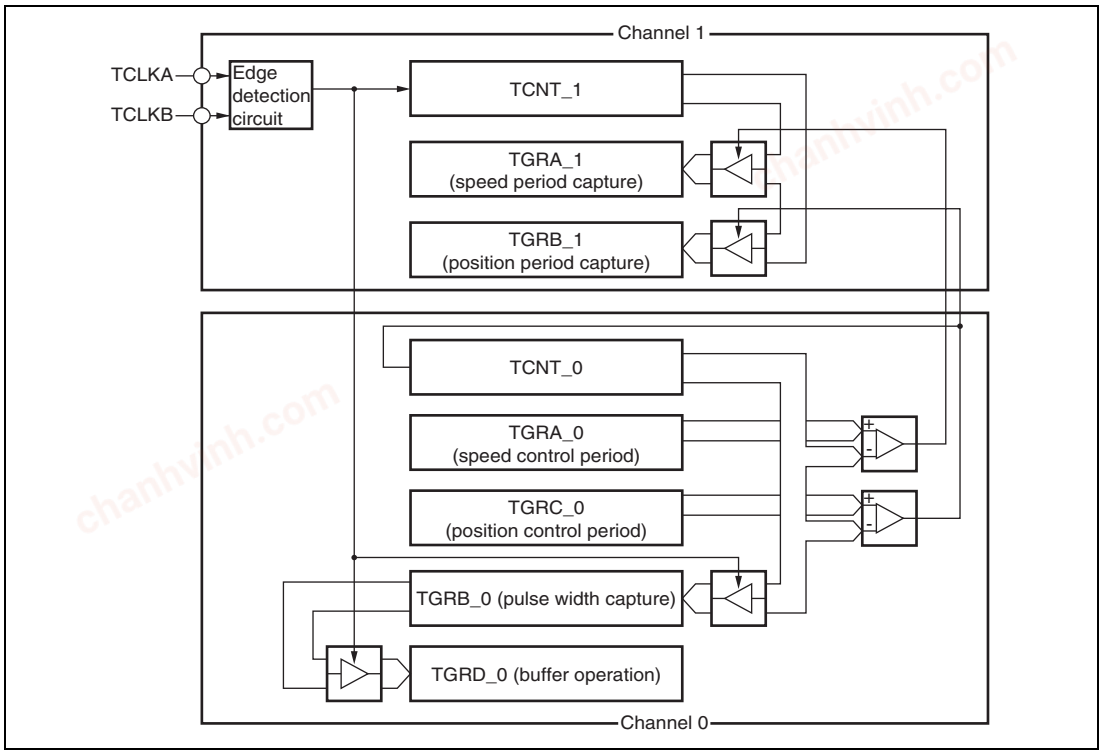


Figure 11.34 Phase Counting Mode Application Example

chanhvinh.com

11.4.7 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 TCNT3 functions as an upcounter.

Table 11.50 shows the PWM output pins used. Table 11.51 shows the settings of the registers.

Table 11.50 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)

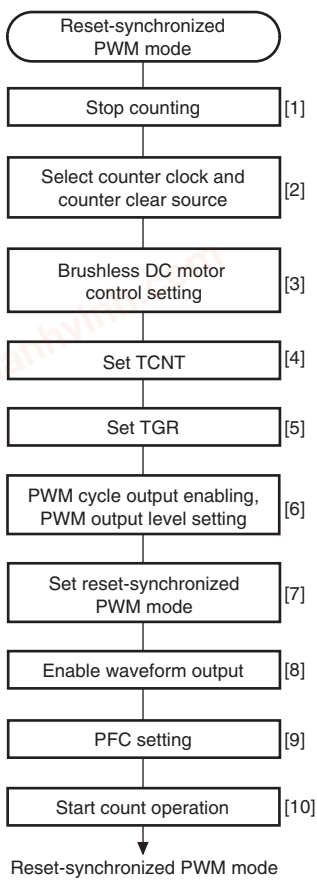
Table 11.51 Register Settings for Reset-Synchronized PWM Mode

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 11.35 shows an example of procedure for selecting the reset synchronized PWM mode.



[1] Clear the CST3 and CST4 bits in the TSTR to 0 to halt the counting of TCNT. The reset-synchronized PWM mode must be set up while TCNT_3 and TCNT_4 are halted.

[2] Set bits TPSC2-TPSC0 and CKEG1 and CKEG0 in the TCR_3 to select the counter clock and clock edge for channel 3. Set bits CCLR2-CCLR0 in the TCR_3 to select TGRA compare-match as a counter clear source.

[3] When performing brushless DC motor control, set bit BDC in the timer gate control register (TGCR) and set the feedback signal input source and output chopping or gate signal direct output.

[4] Reset TCNT_3 and TCNT_4 to H'0000.

[5] TGRA_3 is the period register. Set the waveform period value in TGRA_3. Set the transition timing of the PWM output waveforms in TGRB_3, TGRA_4, and TGRB_4. Set times within the compare-match range of TCNT_3.
$X \leq TGRA_3$ (X: set value).

[6] Select enabling/disabling of toggle output synchronized with the PMW cycle using bit PSYE in the timer output control register (TOCR), and set the PWM output level with bits OLSP and OLSN. When specifying the PWM output level by using TOLBR as a buffer for TOCR_2, see figure 11.3.

[7] Set bits MD3-MD0 in TMDR_3 to B'1000 to select the reset-synchronized PWM mode. Do not set to TMDR_4.

[8] Set the enabling/disabling of the PWM waveform output pin in TOER.

[9] Set the port control register and the port I/O register.

[10] Set the CST3 bit in the TSTR to 1 to start the count operation.

Note: The output waveform starts to toggle operation at the point of TCNT_3 = TGRA_3 = X by setting X = TGRA, i.e., cycle = duty.

Figure 11.35 Procedure for Selecting Reset-Synchronized PWM Mode

chanhvinh.com

(2) Reset-Synchronized PWM Mode Operation

Figure 11.36 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.

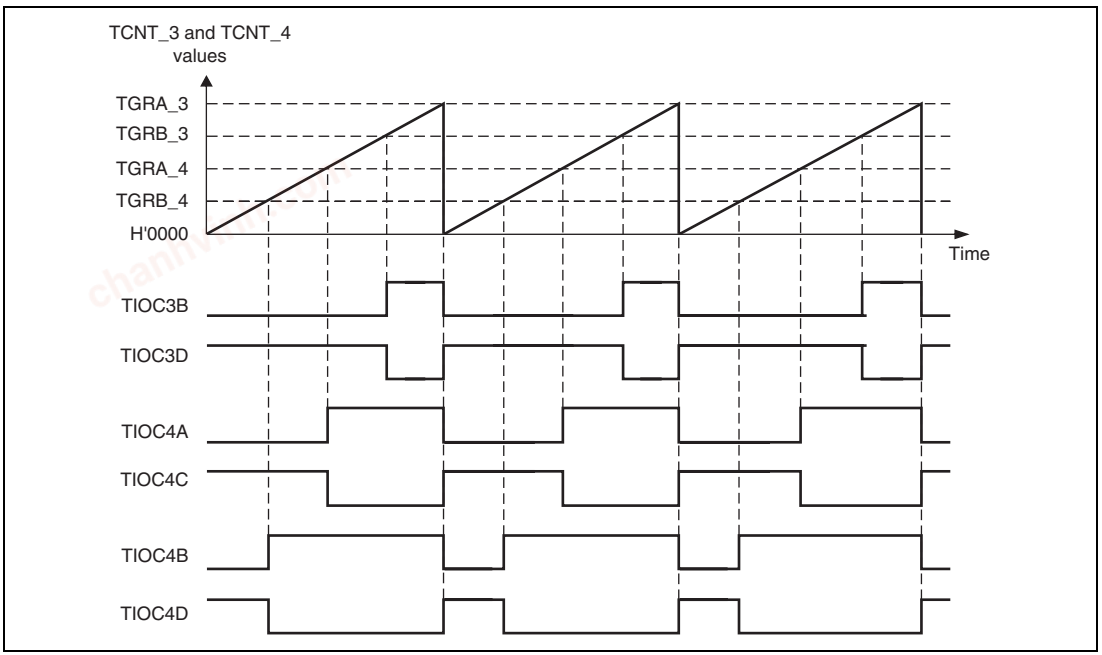


Figure 11.36 Reset-Synchronized PWM Mode Operation Example (When TOCR's OLSN = 1 and OLSP = 1)

chanhvinh.com

11.4.8 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 11.52 shows the PWM output pins used. Table 11.53 shows the settings of the registers used.

A function to directly cut off the PWM output by using an external signal is supported as a port function.

Table 11.52 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 11.53 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

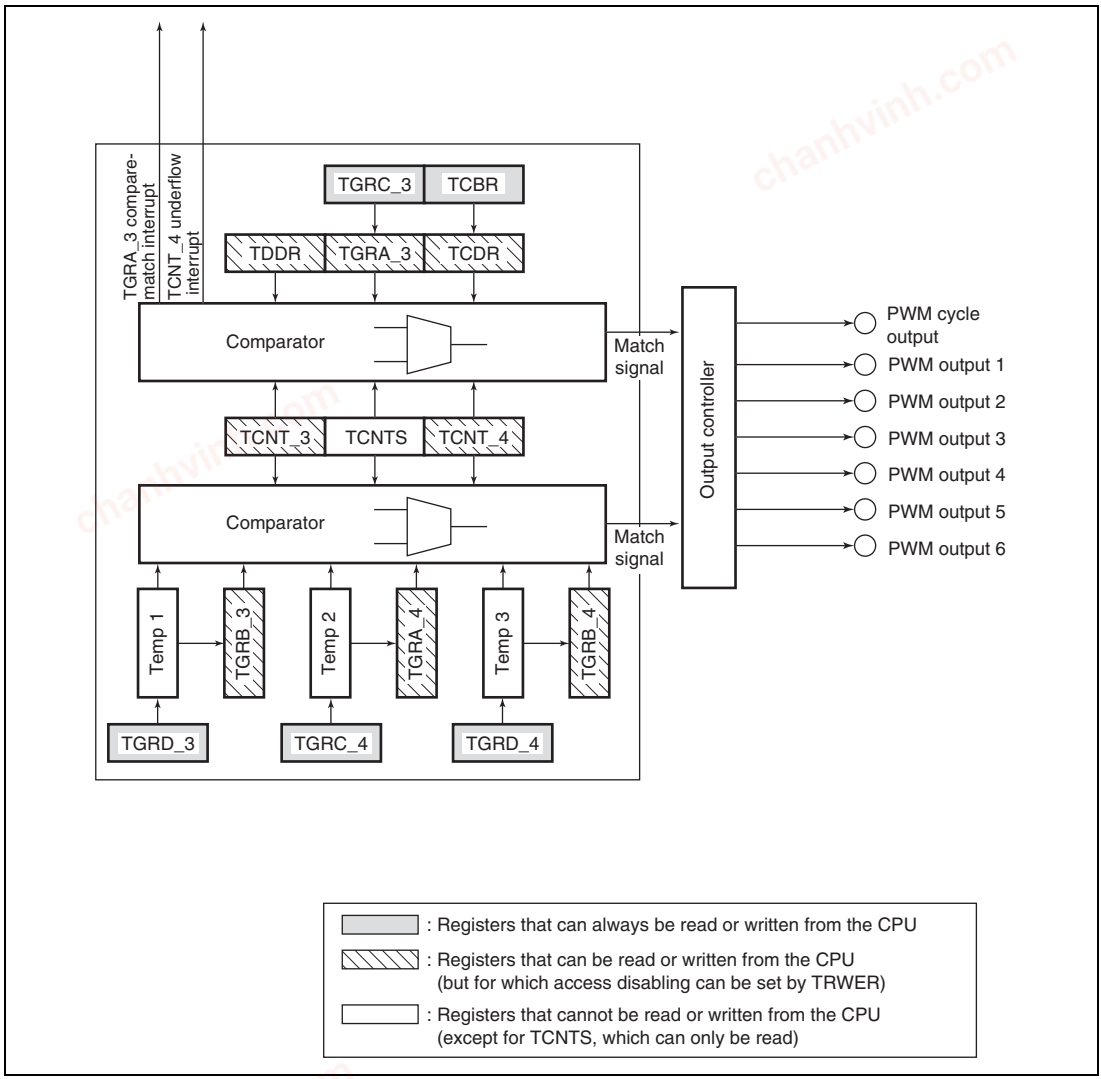


Figure 11.37 Block Diagram of Channels 3 and 4 in Complementary PWM Mode

chanhvinh.com

(1) Example of Complementary PWM Mode Setting Procedure

An example of the complementary PWM mode setting procedure is shown in figure 11.38.

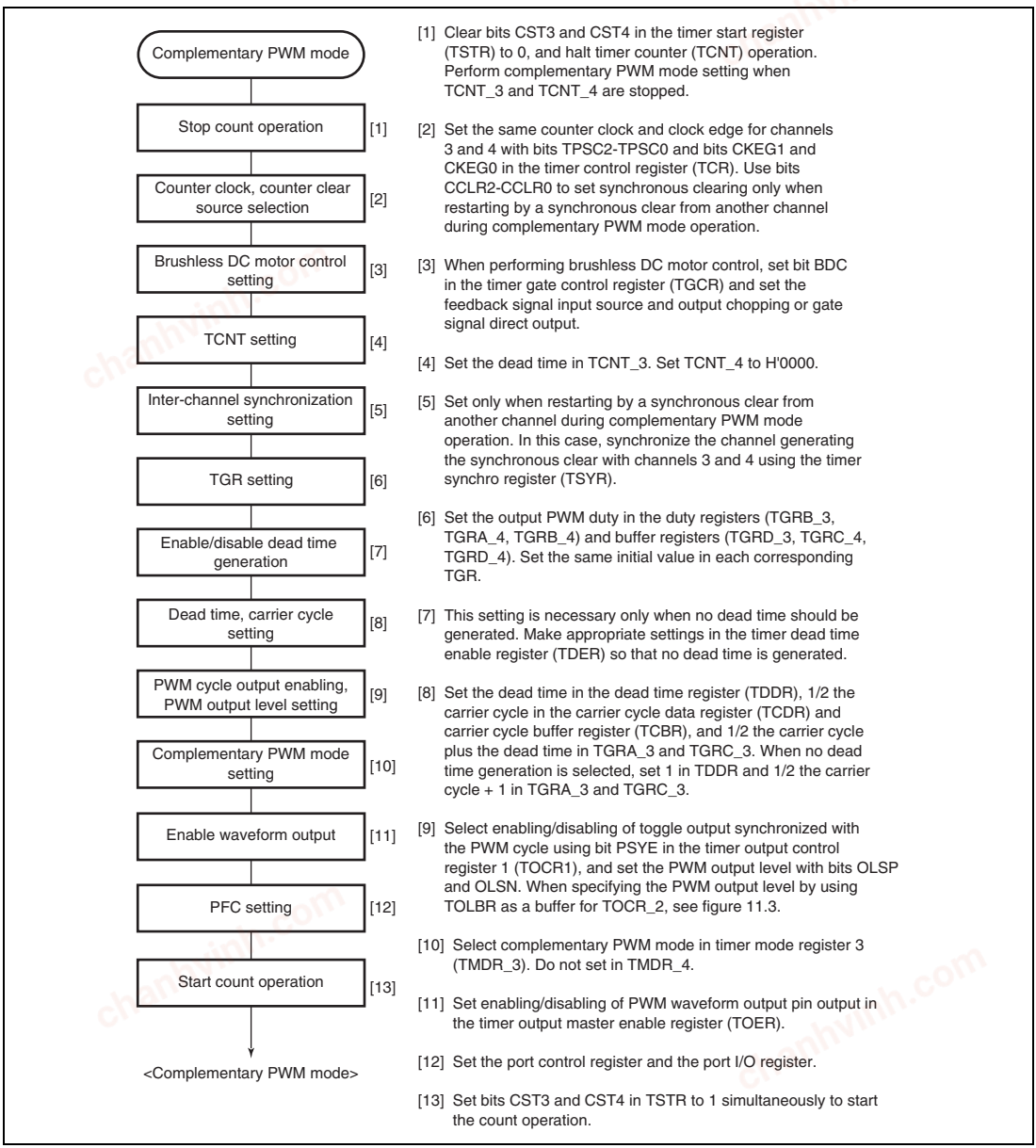


Figure 11.38 Example of Complementary PWM Mode Setting Procedure

chanhvinh.com

(2) Outline of Complementary PWM Mode Operation

In complementary PWM mode, 6-phase PWM output is possible. Figure 11.39 illustrates counter operation in complementary PWM mode, and figure 11.40 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 TCNT3 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, TCNT4 counts up in synchronization with TCNT_3, and switches to down-counting when it matches TCDR. On reaching H'0000, TCNT4 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.

chanhvinh.com

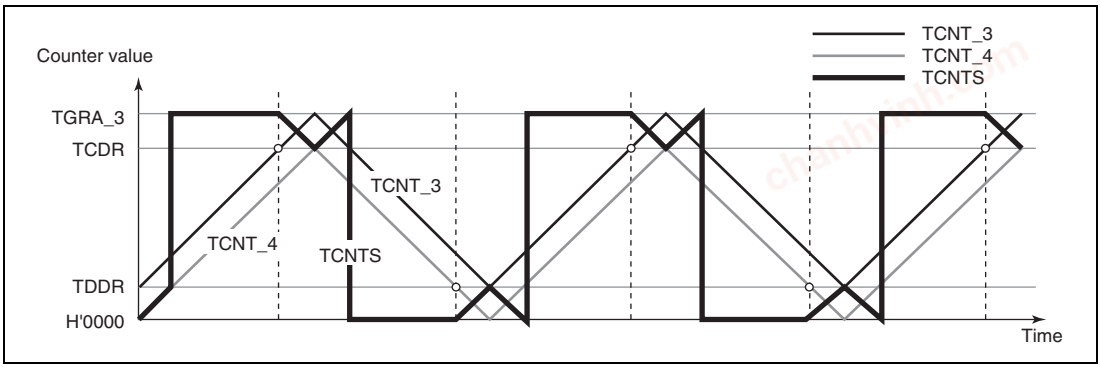


Figure 11.39 Complementary PWM Mode Counter Operation

(b) Register Operation

In complementary PWM mode, nine registers are used, comprising compare registers, buffer registers, and temporary registers. Figure 11.40 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 11.40 shows an example in which the mode is selected in which the change is made in the trough.

chanhvinh.com

In the tb interval (tb1 in figure 11.40) 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

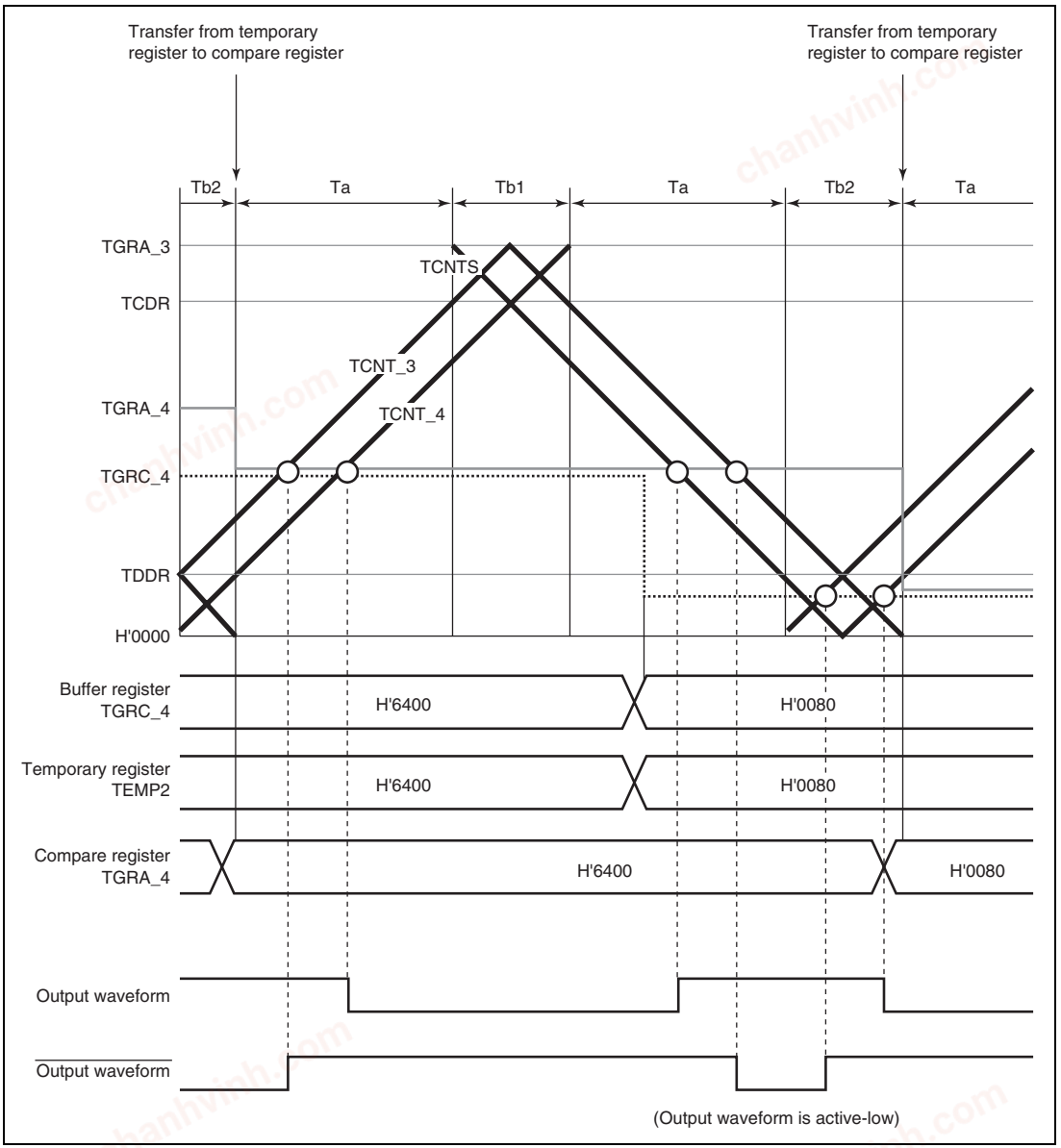


Figure 11.40 Example of Complementary PWM Mode Operation

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 11.54 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 11.41 shows an example of operation without dead time.

chanhvinh.com

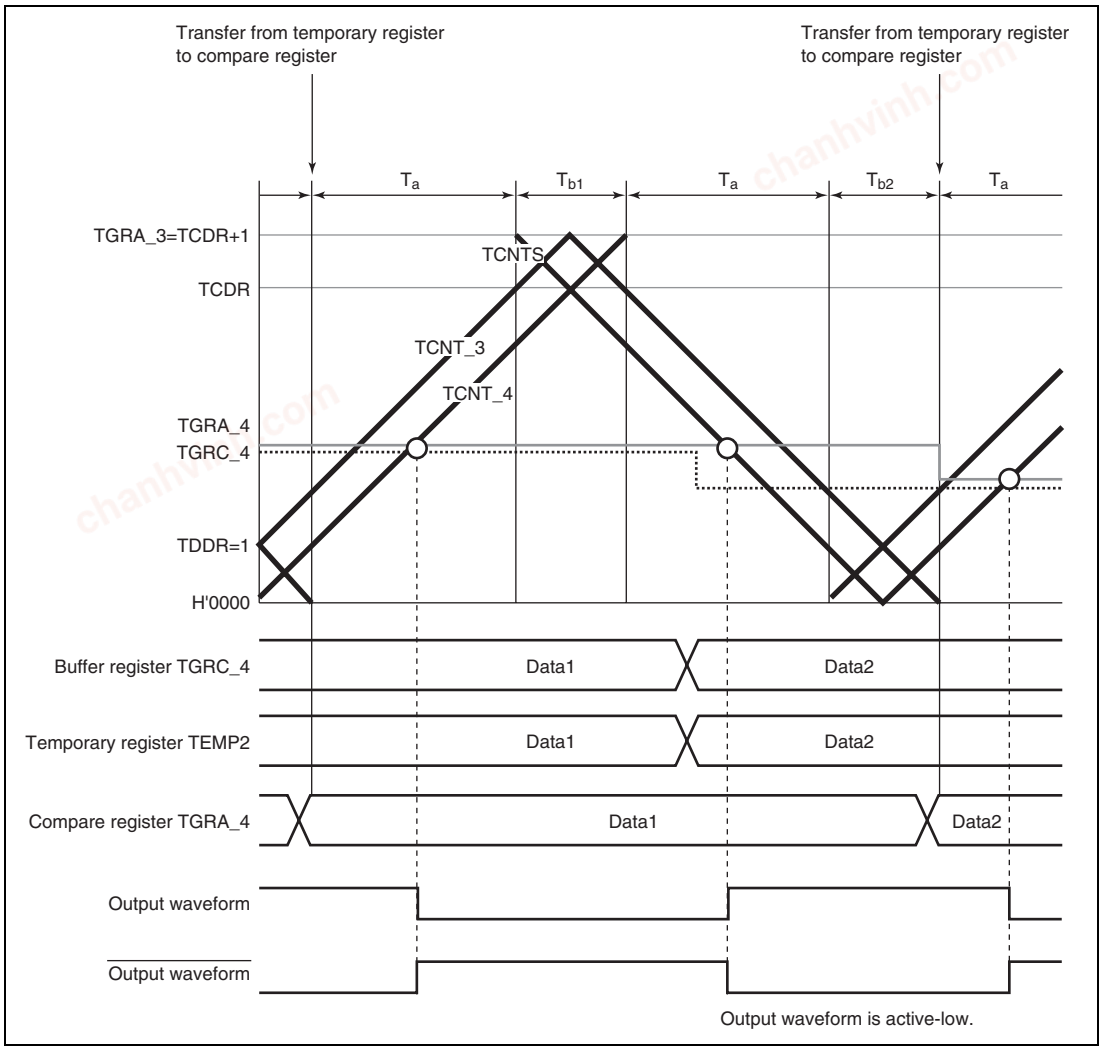


Figure 11.41 Example of Operation without Dead Time

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 set value = TCDR set value + TDDR set value
Without dead time: TGRA_3 set value = TCDR set value + 1

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 11.42 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.

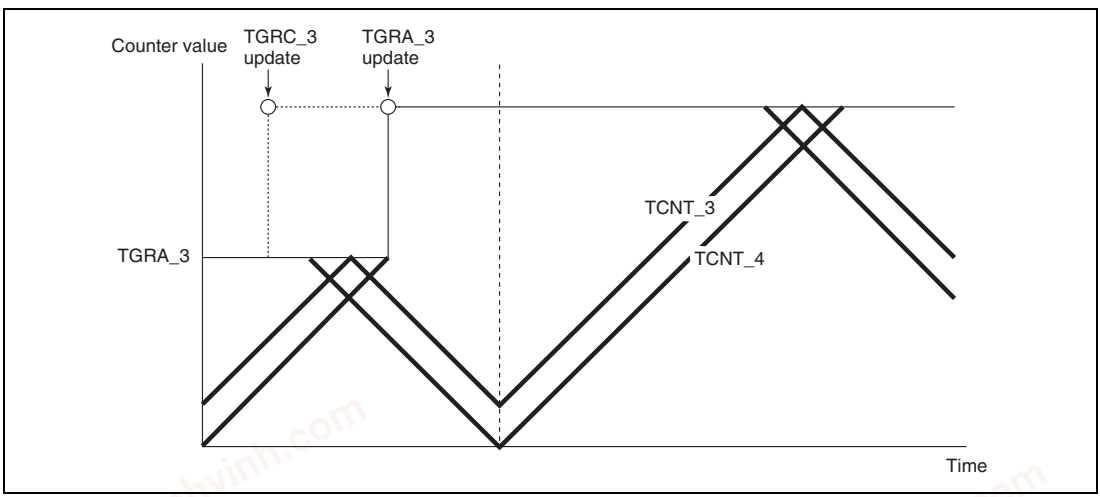


Figure 11.42 Example of PWM Cycle Updating

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 11.43 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

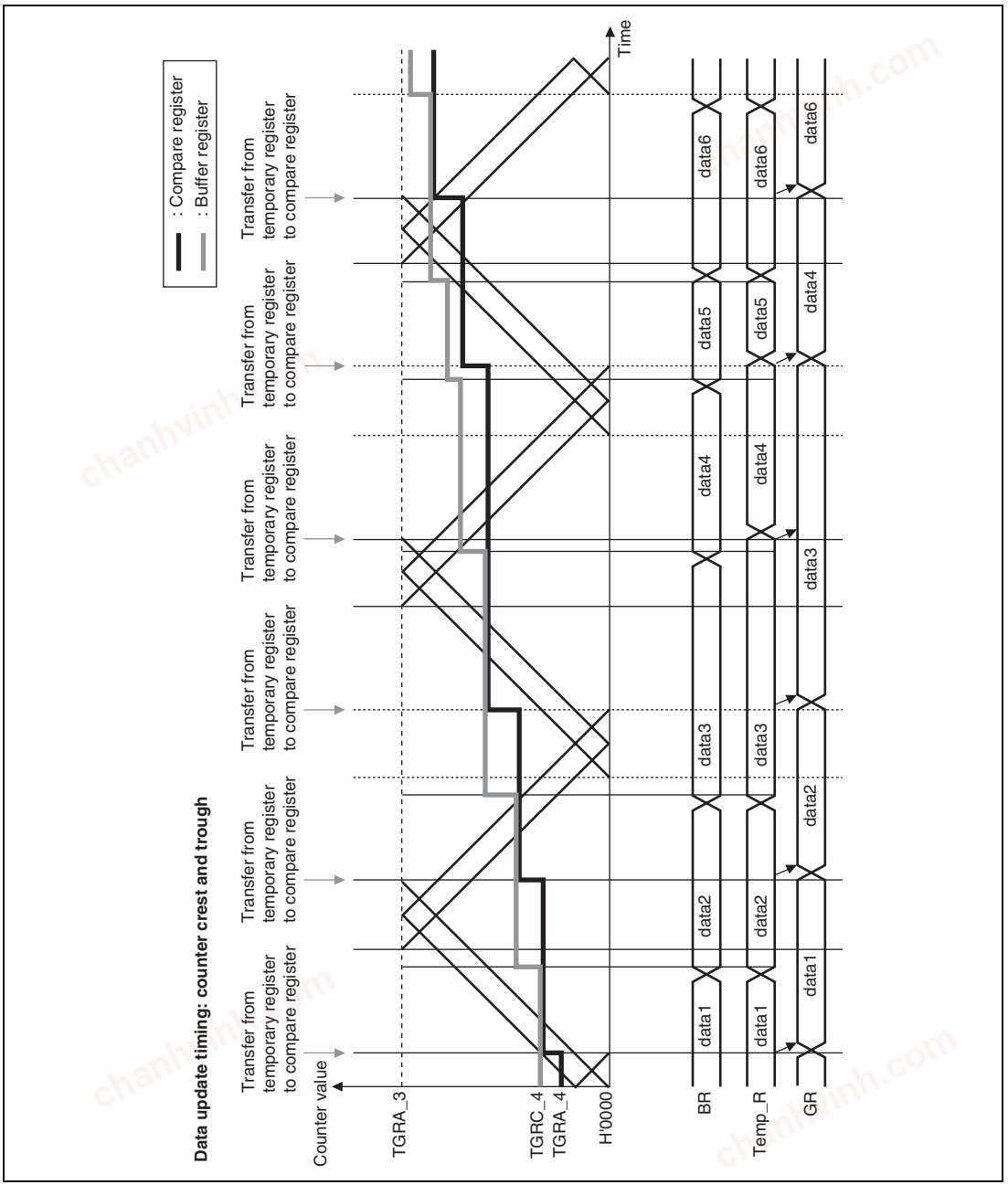


Figure 11.43 Example of Data Update in Complementary PWM Mode

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 11.44 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 11.45.

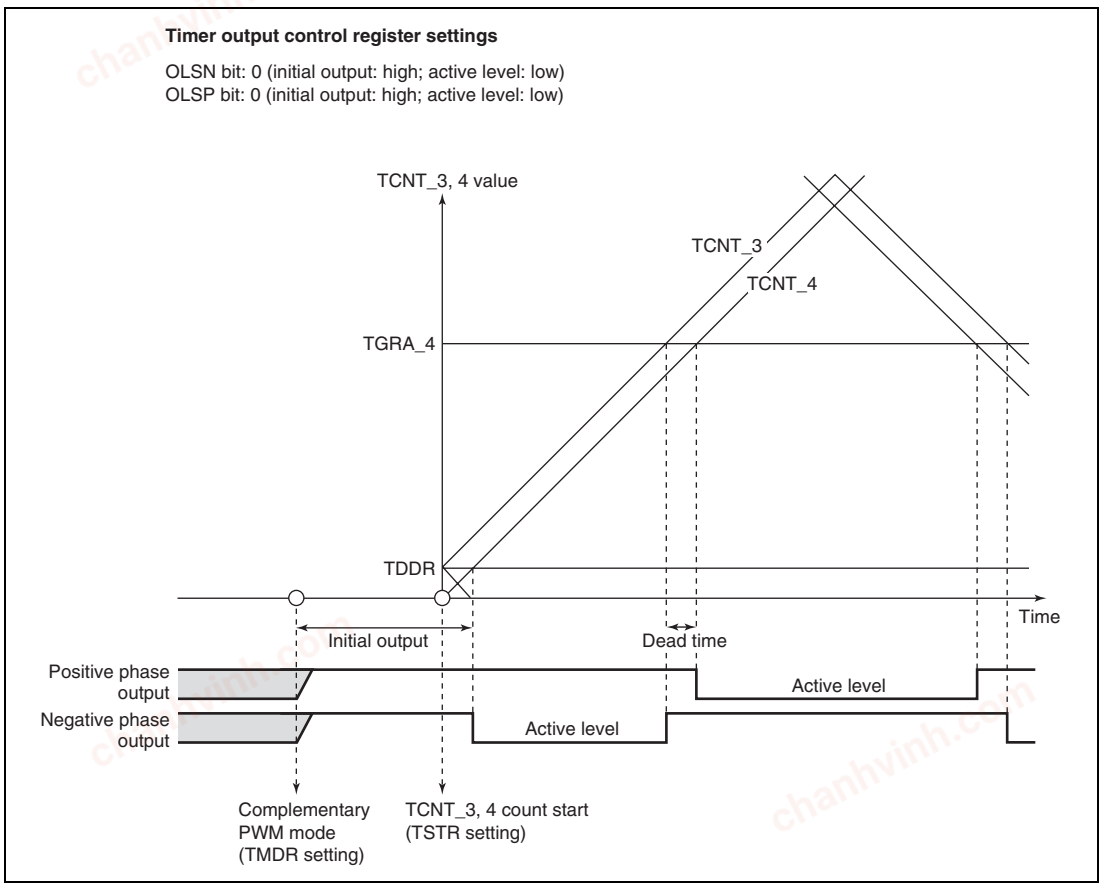


Figure 11.44 Example of Initial Output in Complementary PWM Mode (1)

chanhvinh.com

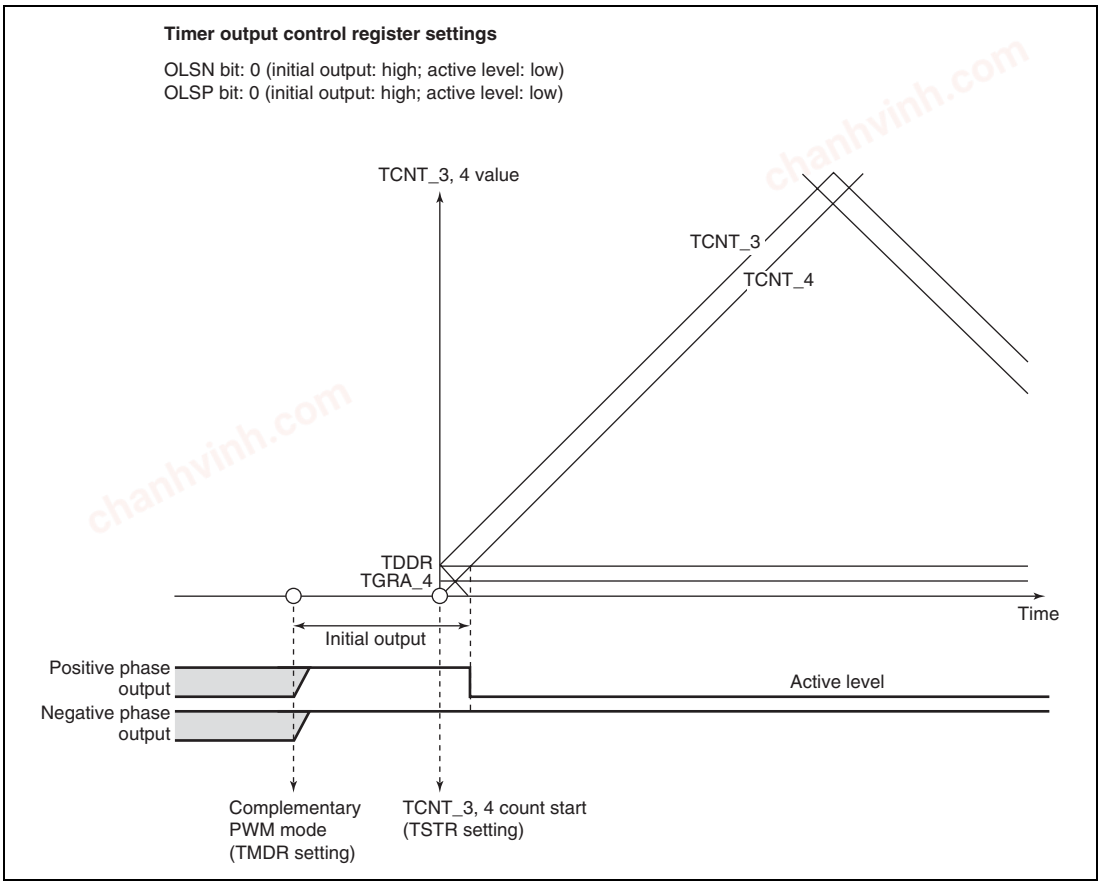


Figure 11.45 Example of Initial Output in Complementary PWM Mode (2)

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 data register. While TCNTS is counting, data 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. Figures 11.46 to 11.48 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 11.46.

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 11.47, compare-match **b** is ignored, and the negative phase is turned off 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 11.48, 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

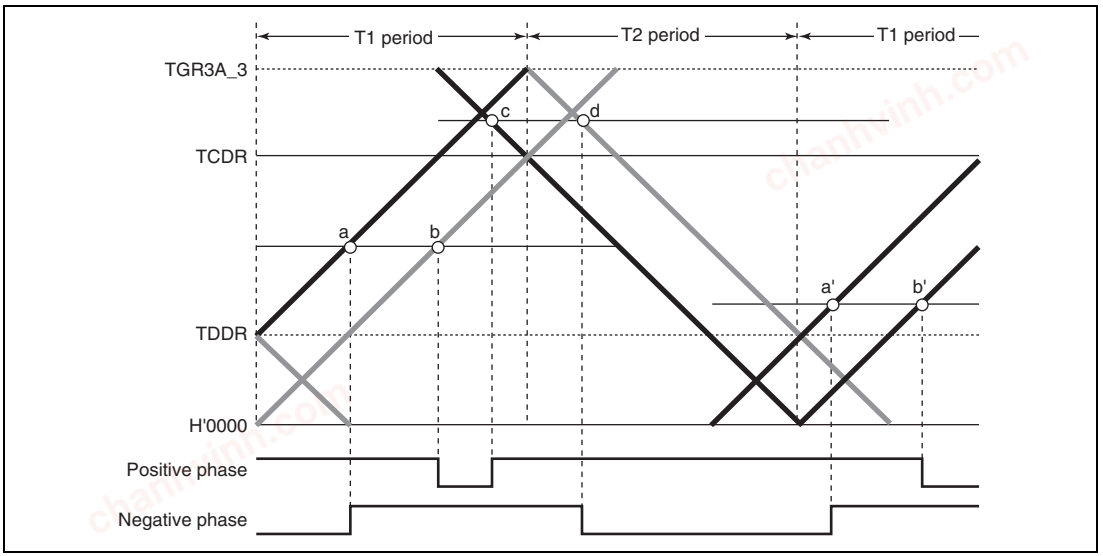


Figure 11.46 Example of Complementary PWM Mode Waveform Output (1)

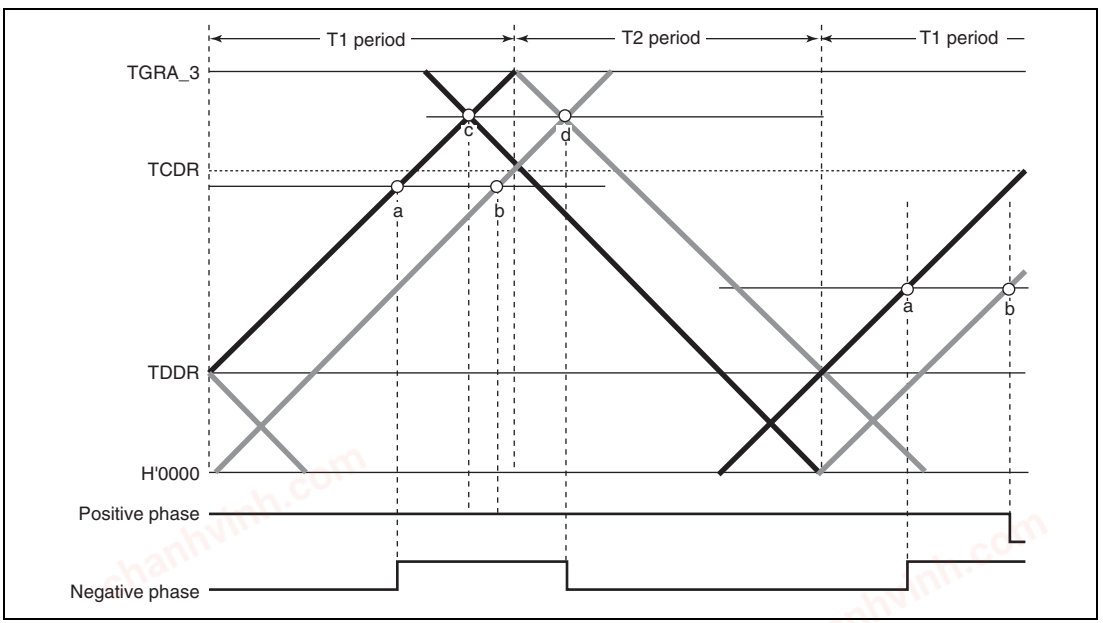


Figure 11.47 Example of Complementary PWM Mode Waveform Output (2)

chanhvinh.com

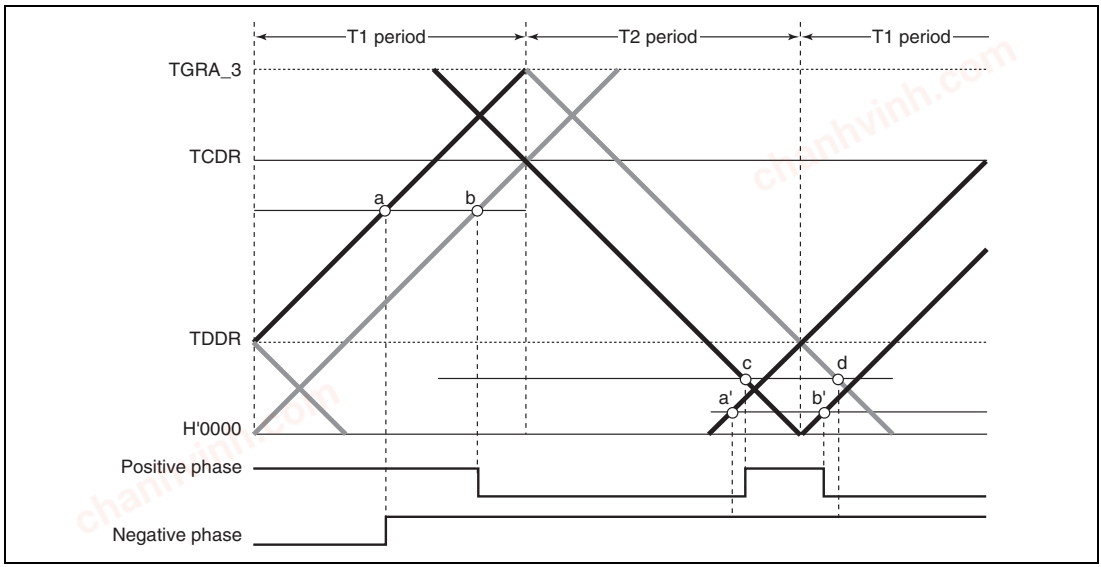


Figure 11.48 Example of Complementary PWM Mode Waveform Output (3)

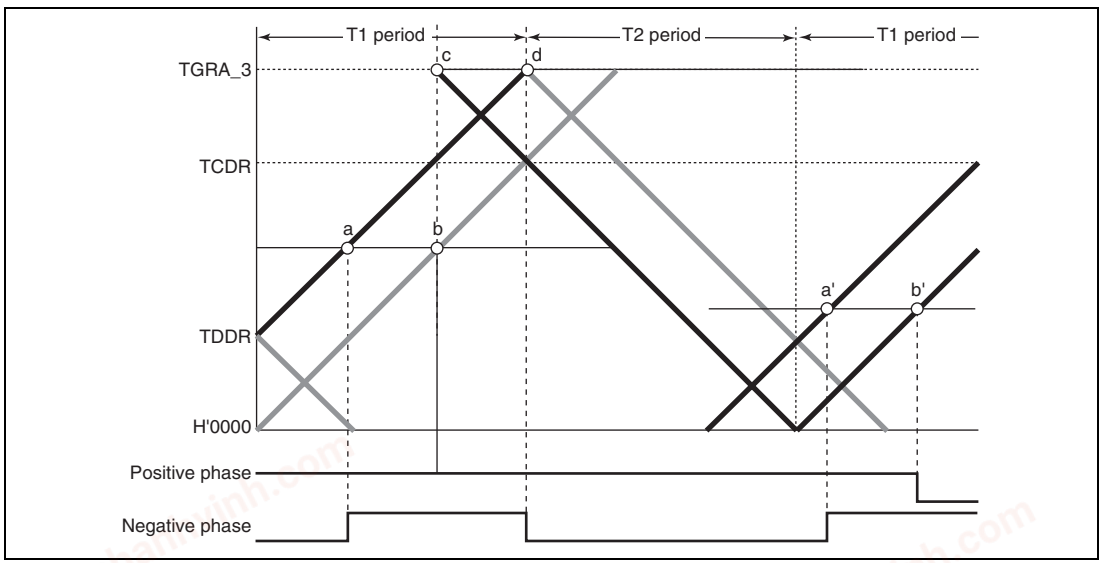


Figure 11.49 Example of Complementary PWM Mode 0% and 100% Waveform Output (1)

chanhvinh.com

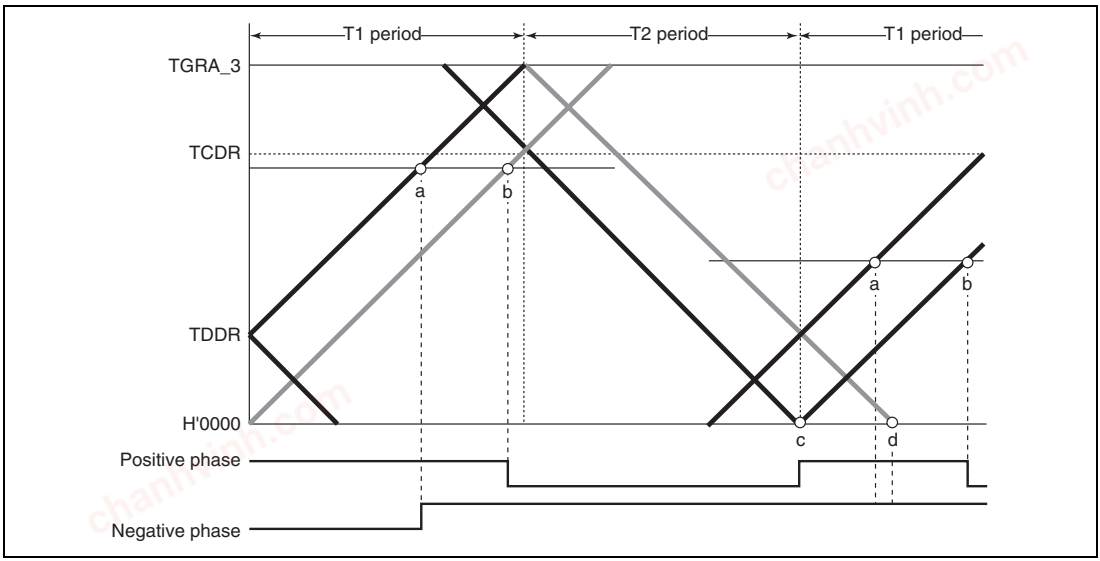


Figure 11.50 Example of Complementary PWM Mode 0% and 100% Waveform Output (2)

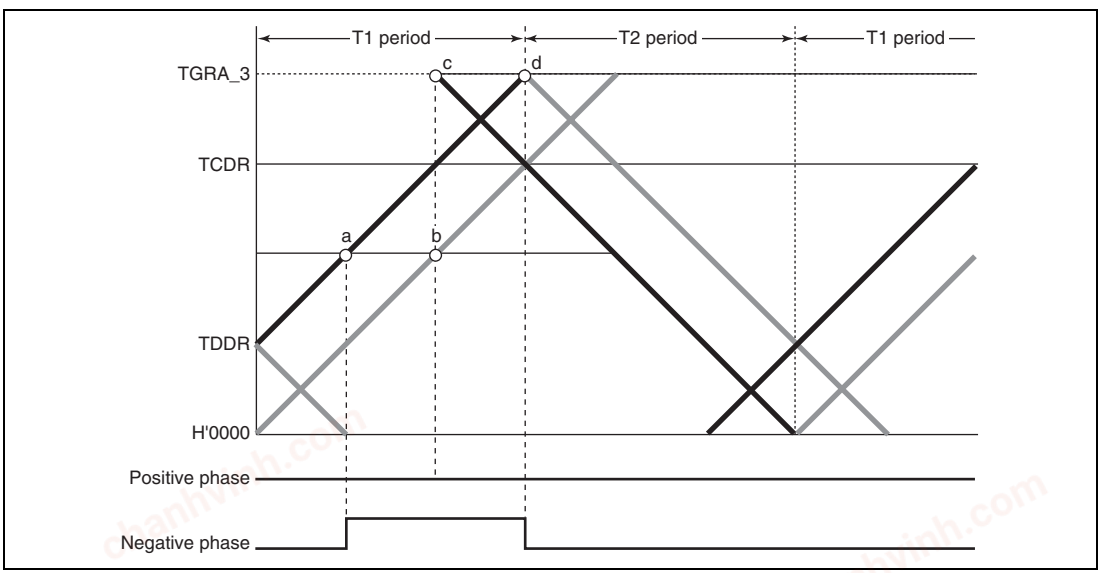


Figure 11.51 Example of Complementary PWM Mode 0% and 100% Waveform Output (3)

chanhvinh.com

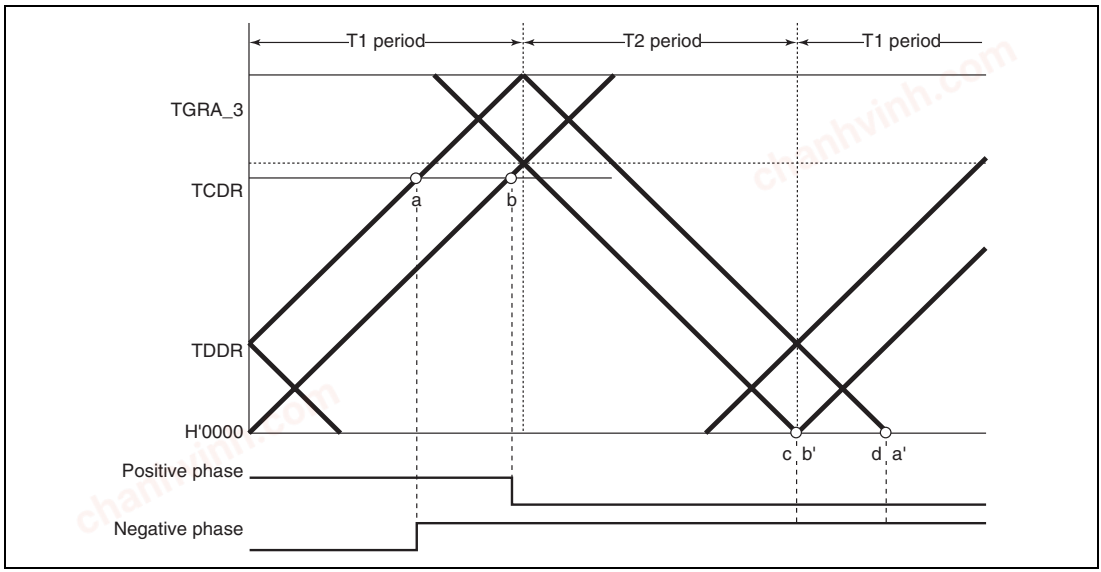


Figure 11.52 Example of Complementary PWM Mode 0% and 100% Waveform Output (4)

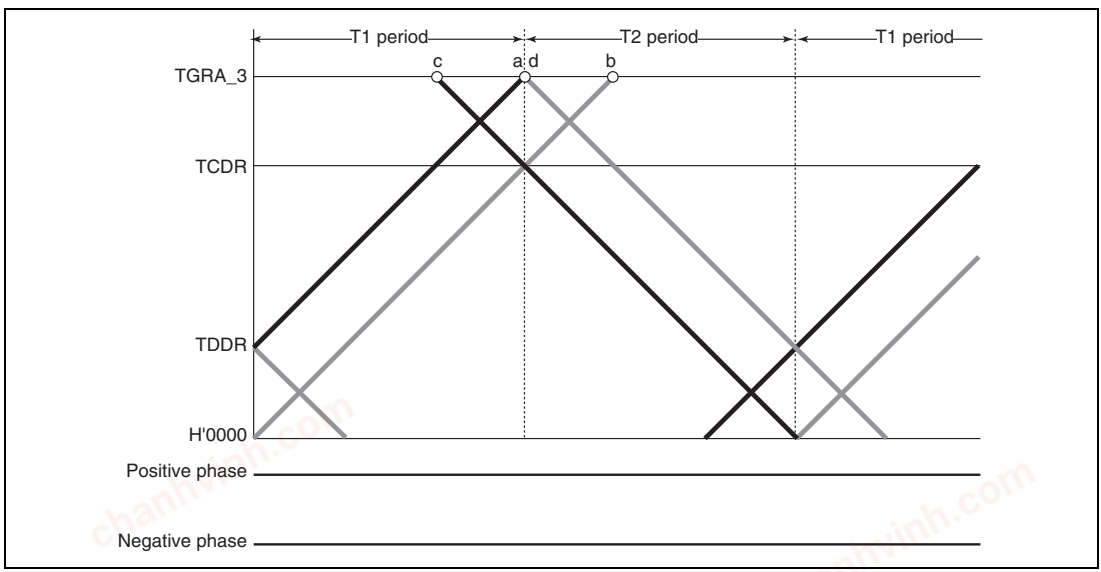


Figure 11.53 Example of Complementary PWM Mode 0% and 100% Waveform Output (5)

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. Figures 11.49 to 11.53 show output examples.

100% duty output is performed when the data 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 data 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 11.54.

This output is toggled by a compare-match between TCNT_3 and TGRA_3 and a compare-match between TCNT4 and H'0000.

The output pin for this toggle output is the TIOC3A pin. The initial output is 1.

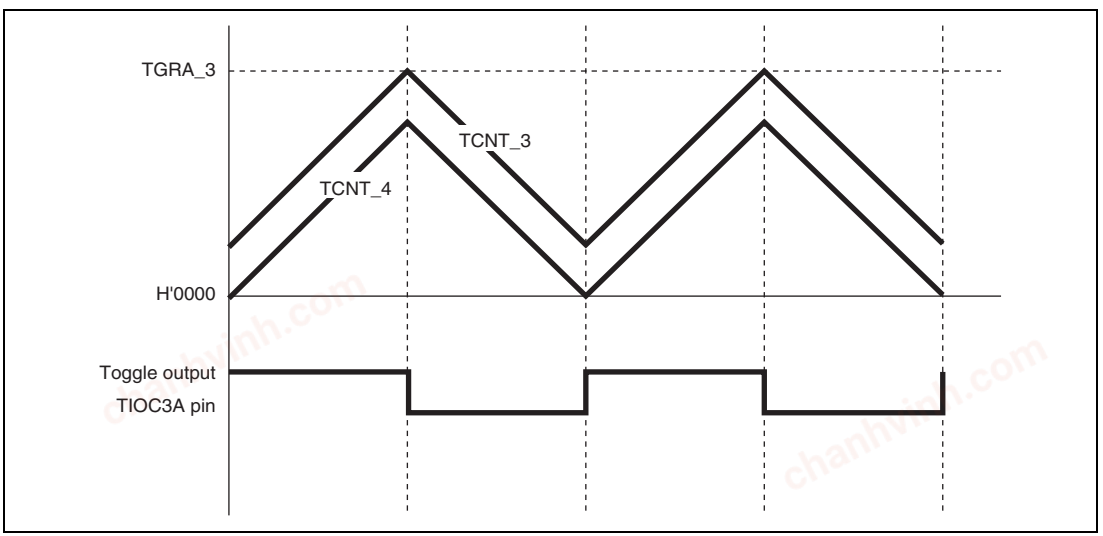


Figure 11.54 Example of Toggle Output Waveform Synchronized with PWM Output

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 11.55 illustrates the operation.

Use of this function enables counter clearing and restarting to be performed by means of an external signal.

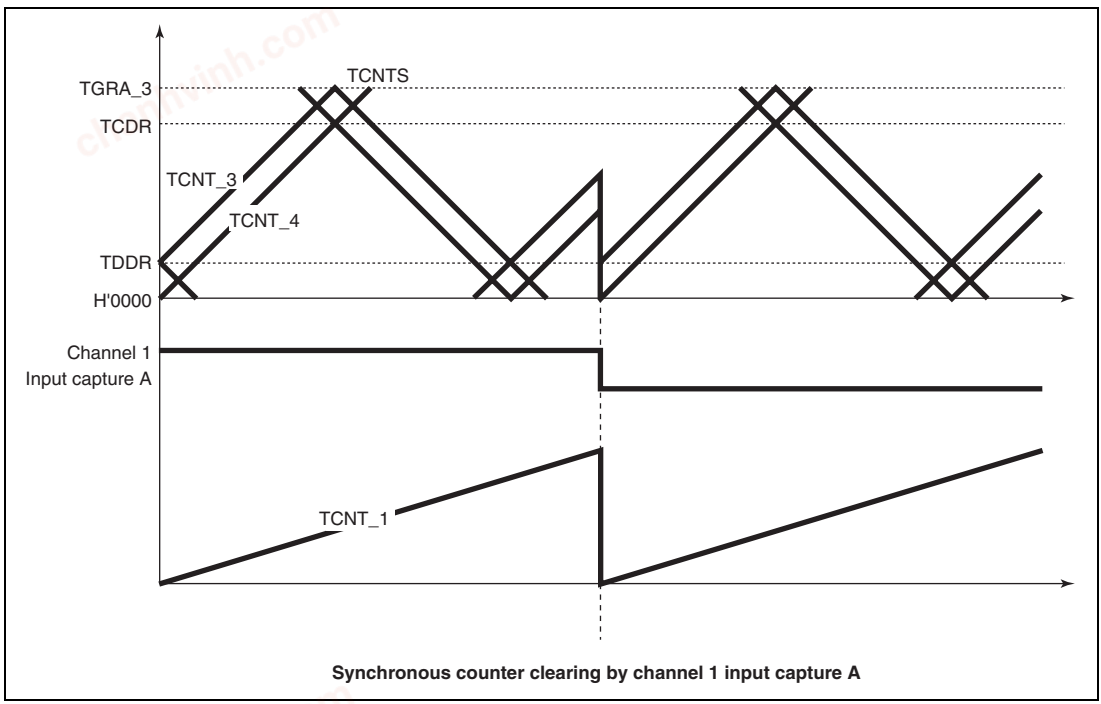


Figure 11.55 Counter Clearing Synchronized with Another Channel

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 11.56. 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 11.56) 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 11.7.23, Notes on Output Waveform Control During Synchronous Counter Clearing in Complementary PWM Mode.

chanhvinh.com

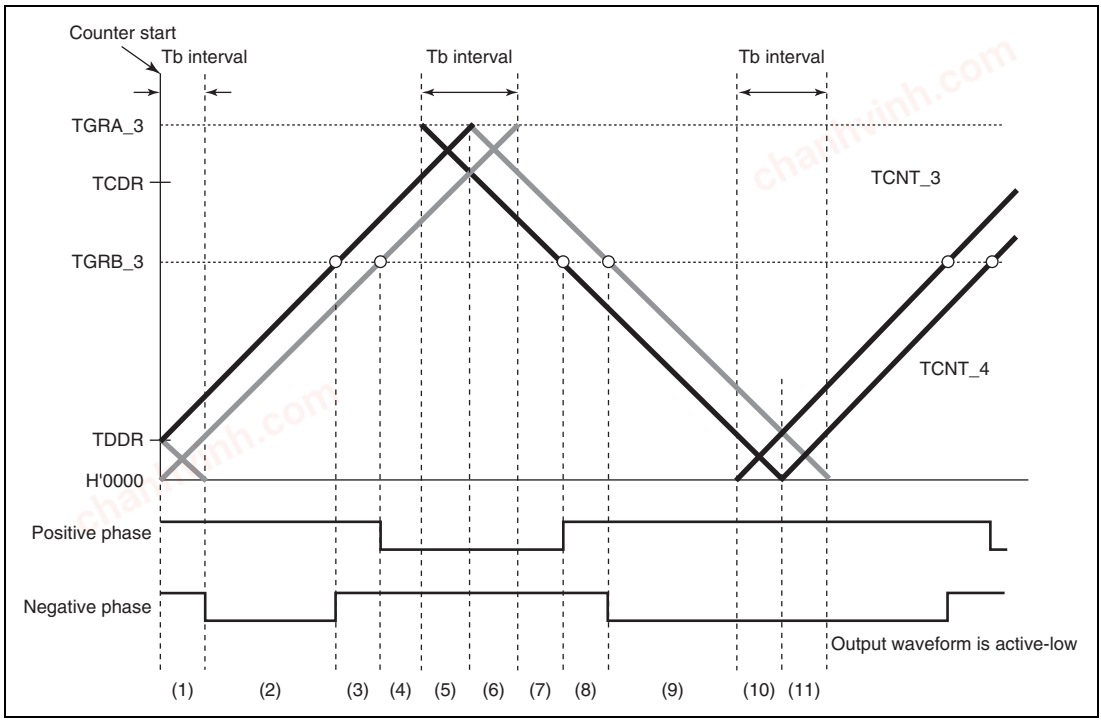


Figure 11.56 Timing for Synchronous Counter Clearing

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 11.57.

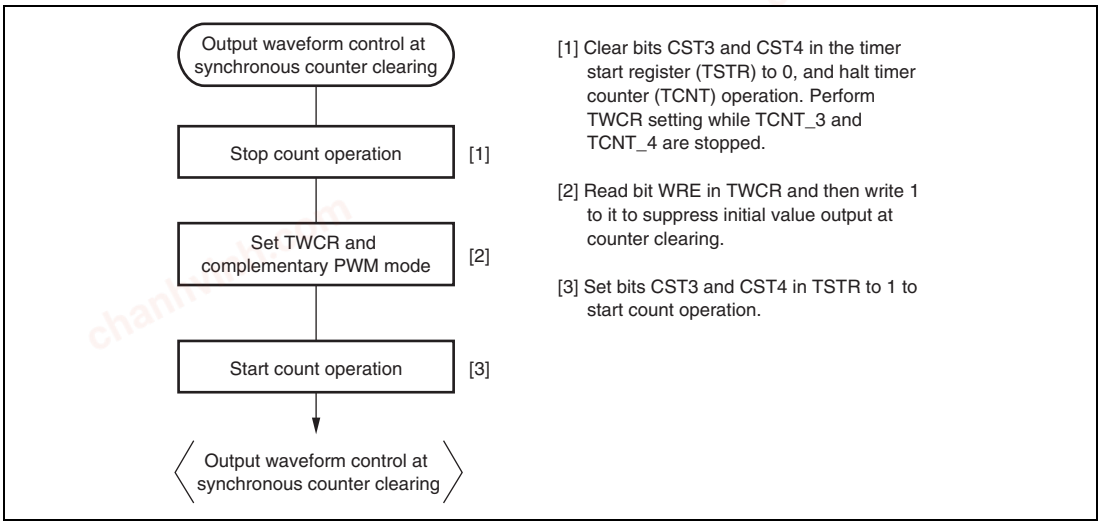


Figure 11.57 Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

chanhvinh.com

- Examples of Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

  Figures 11.58 to 11.61 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 figures 11.58 to 11.61, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in figure 11.56, respectively.

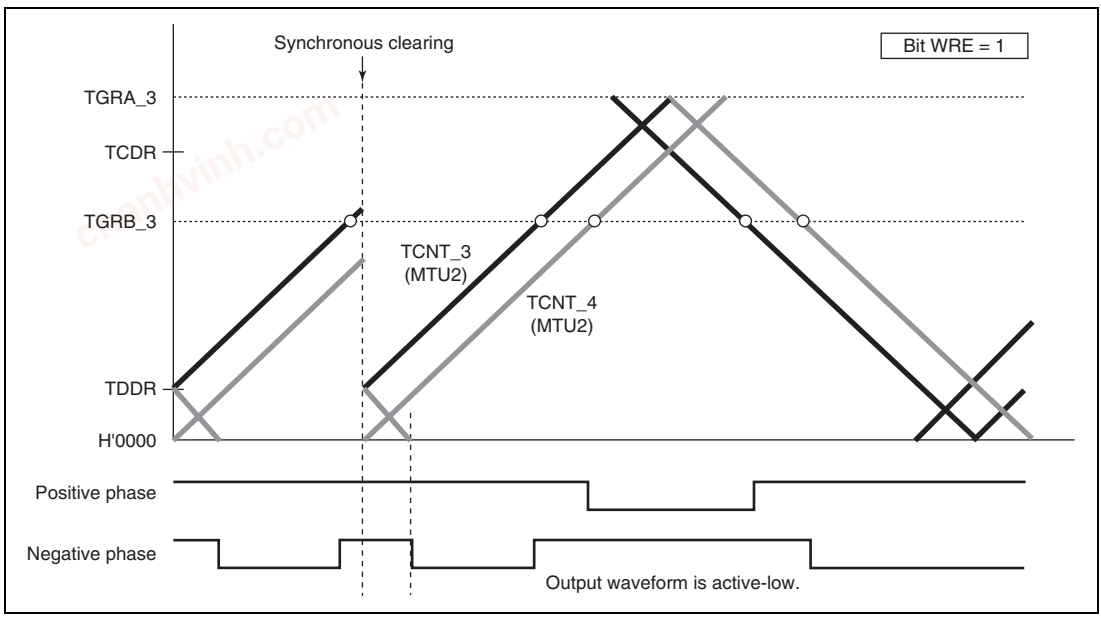


Figure 11.58 Example of Synchronous Clearing in Dead Time during Up-Counting (Timing (3) in Figure 11.56; Bit WRE of TWCR is 1)

chanhvinh.com

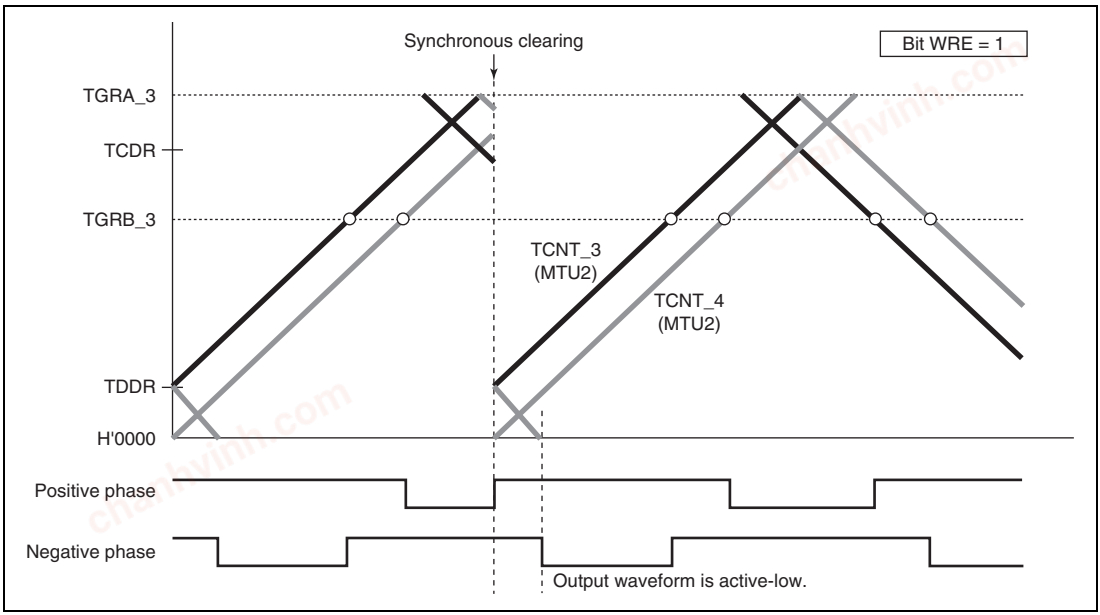


Figure 11.59 Example of Synchronous Clearing in Interval Tb at Crest (Timing (6) in Figure 11.56; Bit WRE of TWCR is 1)

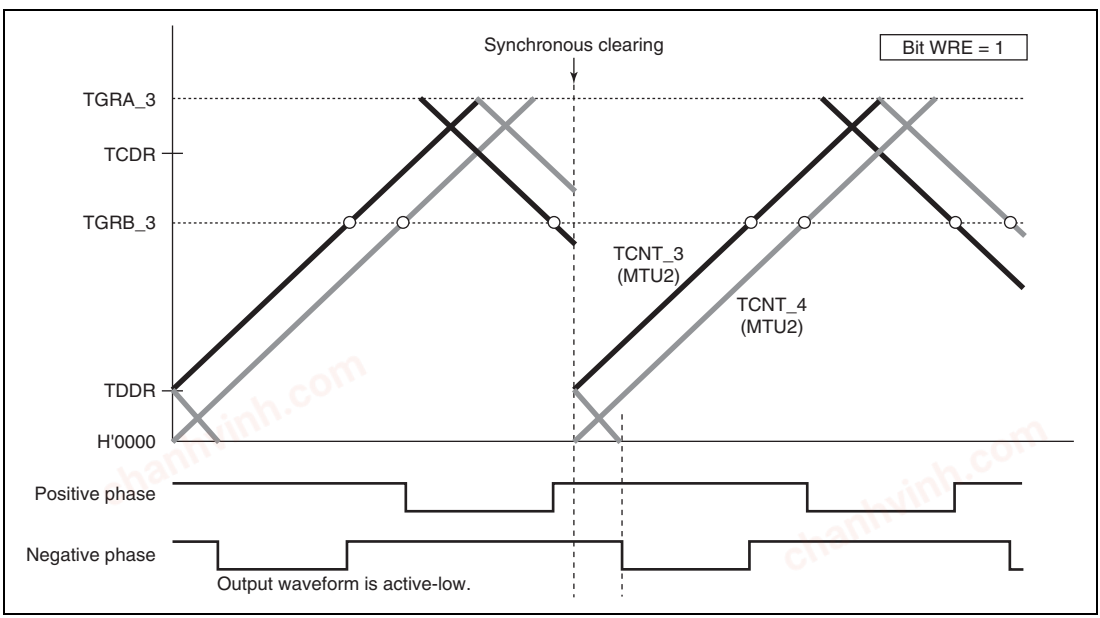


Figure 11.60 Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 11.56; Bit WRE of TWCR is 1)

chanhvinh.com

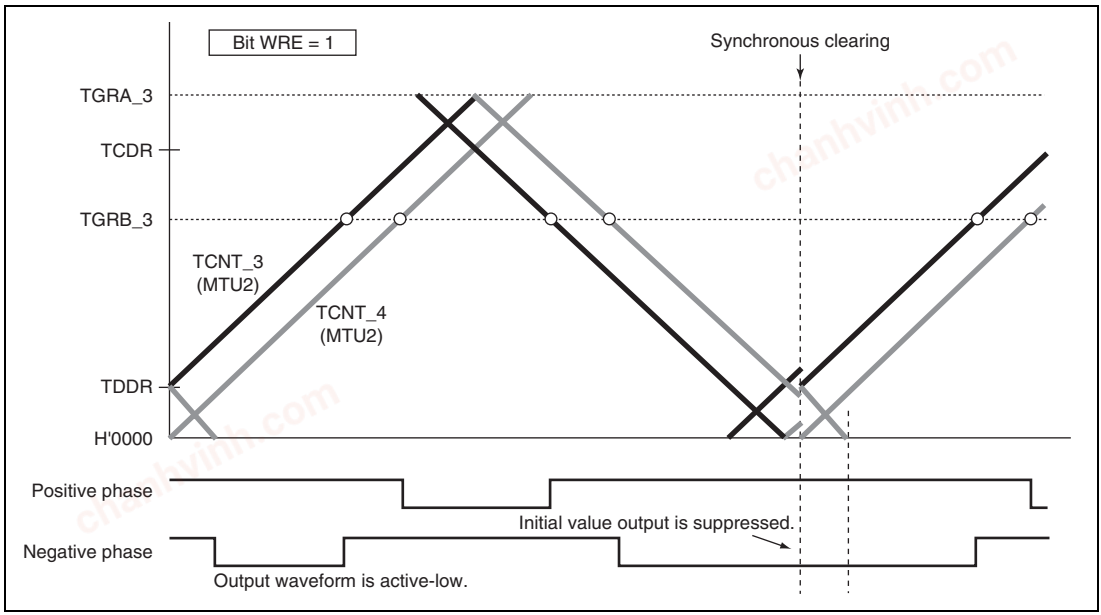


Figure 11.61 Example of Synchronous Clearing in Interval Tb at Trough (Timing (11) in Figure 11.56; Bit WRE of TWCR is 1)

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 11.62 illustrates an operation example.

Notes: 1. Use this function only in complementary PWM mode 1 (transfer at crest)
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).
3. Do not set the PWM duty value to H'0000.
4. Do not set the PSYE bit in timer output control register 1 (TOCR1) to 1.

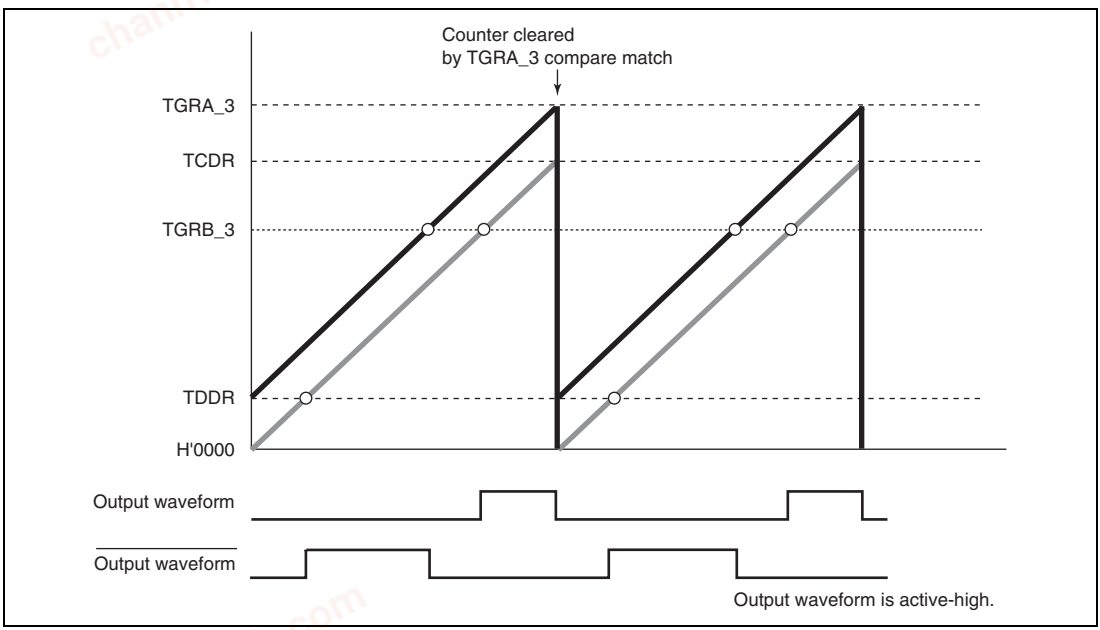


Figure 11.62 Example of Counter Clearing Operation by TGRA_3 Compare Match

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). Figures 11.63 to 11.66 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.

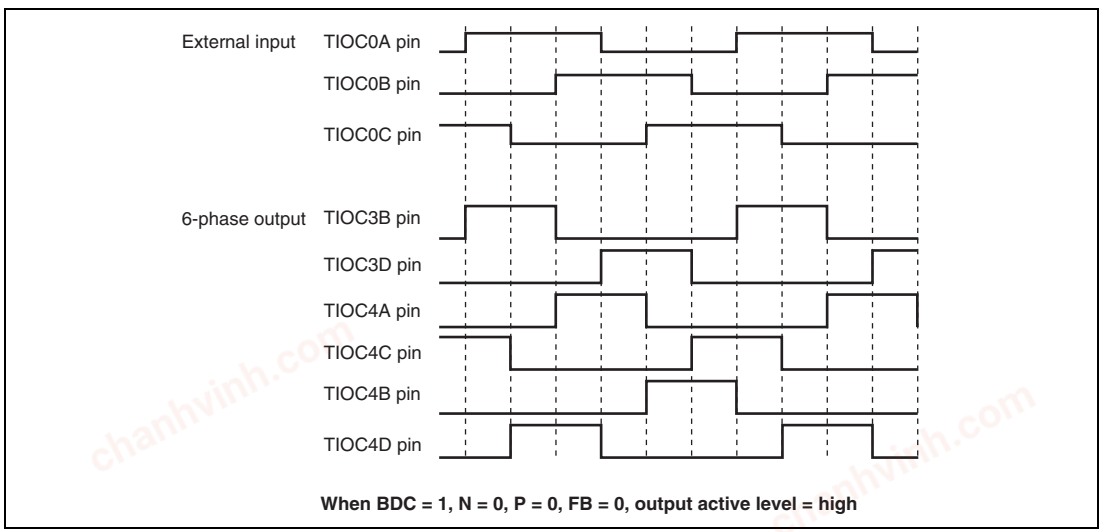


Figure 11.63 Example of Output Phase Switching by External Input (1)

chanhvinh.com

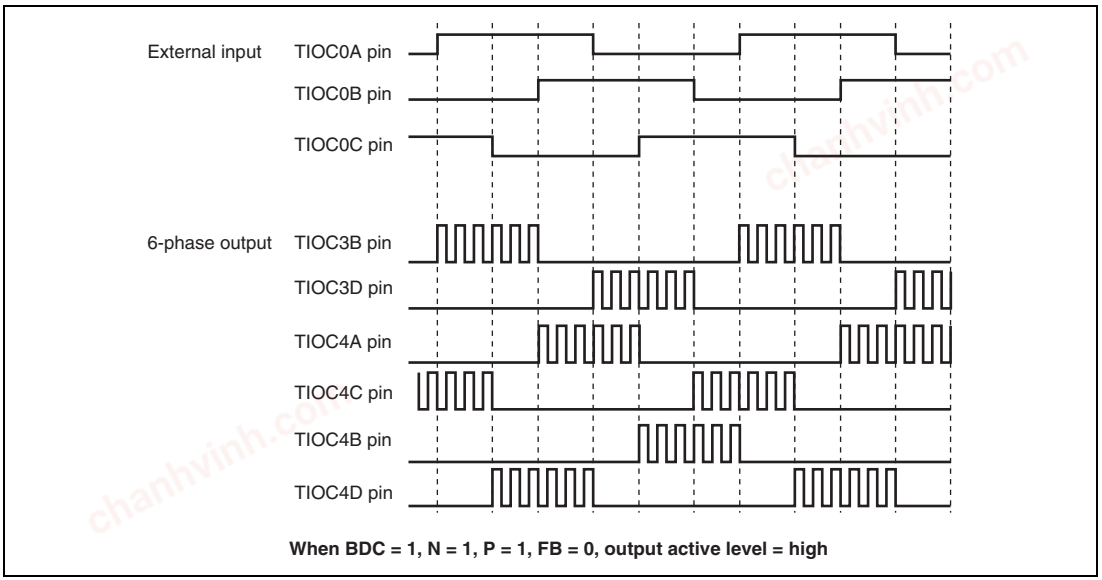


Figure 11.64 Example of Output Phase Switching by External Input (2)

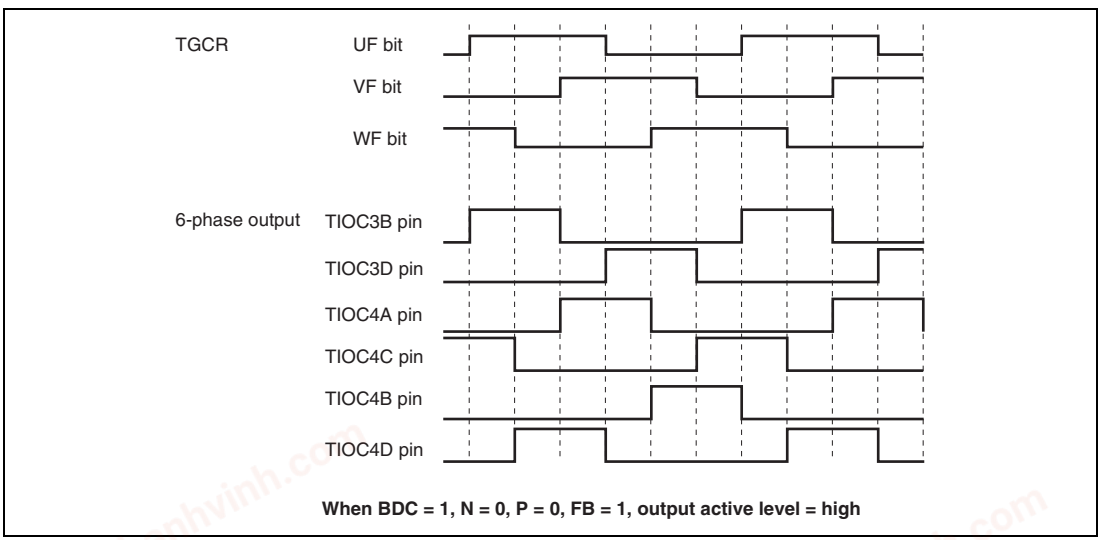


Figure 11.65 Example of Output Phase Switching by Means of UF, VF, WF Bit Settings (1)

chanhvinh.com

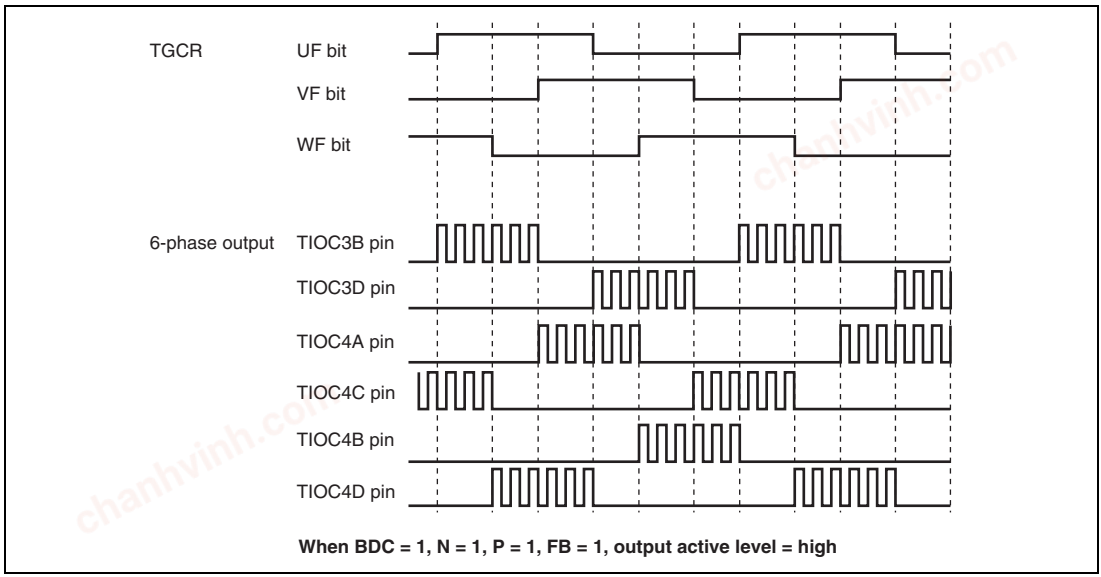


Figure 11.66 Example of Output Phase Switching by Means of UF, VF, WF Bit Settings (2)

(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 11.4.9, 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 11.67 shows an example of the interrupt skipping operation setting procedure. Figure 11.68 shows the periods during which interrupt skipping count can be changed.

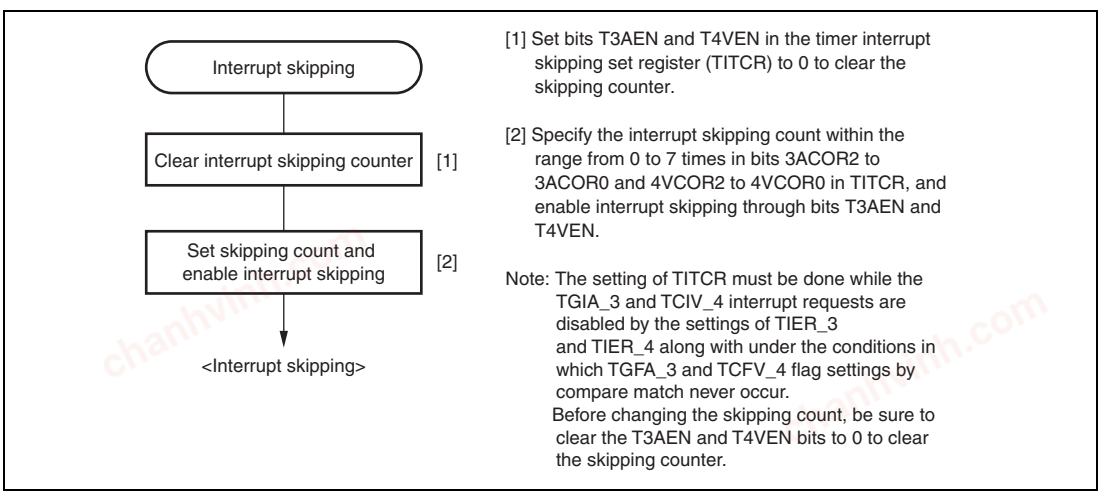


Figure 11.67 Example of Interrupt Skipping Operation Setting Procedure

chanhvinh.com

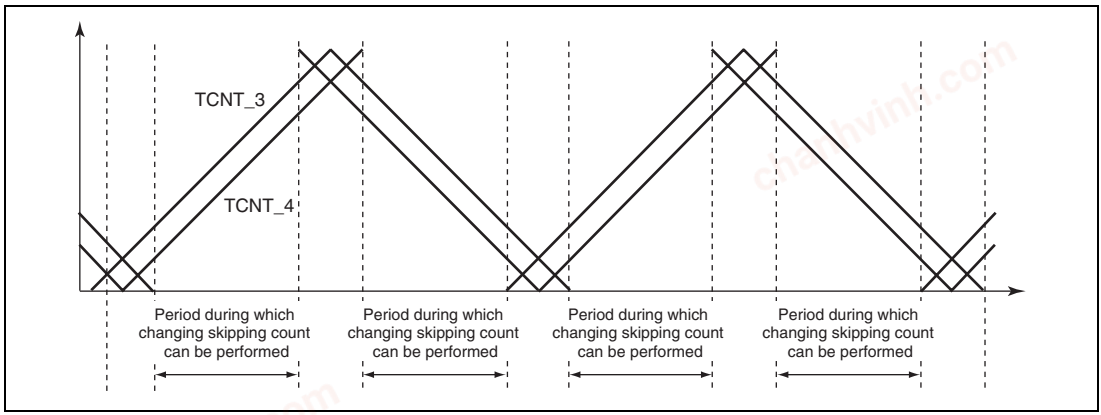


Figure 11.68 Periods during which Interrupt Skipping Count can be Changed

(b) Example of Interrupt Skipping Operation

Figure 11.69 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).

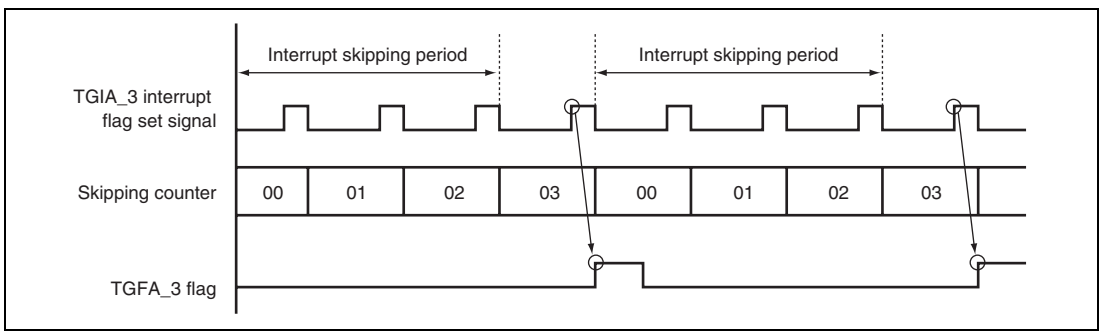


Figure 11.69 Example of Interrupt Skipping Operation

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).

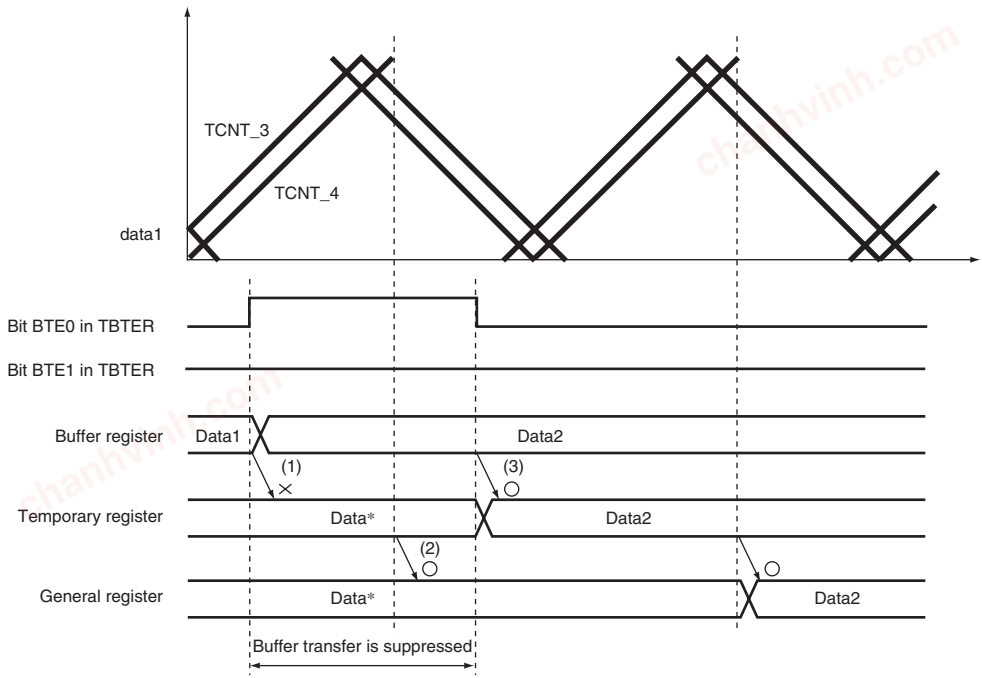
Figure 11.70 shows an example of operation when buffer transfer is suppressed (BTE1 = 0 and BTE0 = 1). While this setting is valid, data is not transferred from the buffer register to the temporary register.

Figure 11.71 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 11.72 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.

chanhvinh.com



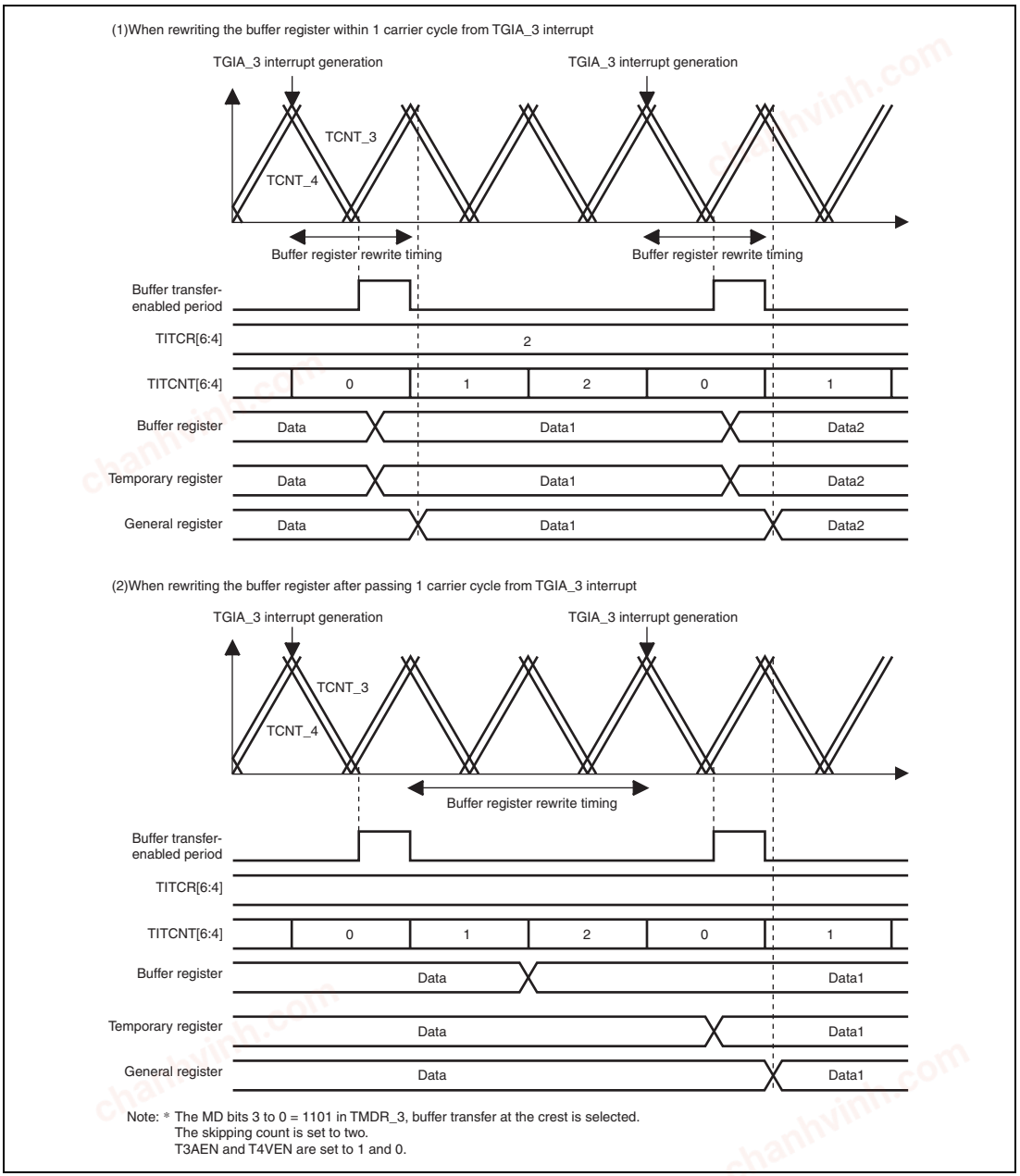
[Legend]

(1) No data is transferred from the buffer register to the temporary register in the buffer transfer-disabled period (bits BTE1 and BTE0 in TBTER are set to 0 and 1, respectively).
(2) Data is transferred from the temporary register to the general register even in the buffer transfer-disabled period.
(3) After buffer transfer is enabled, data is transferred from the buffer register to the temporary register.

Note: * When buffer transfer at the crest is selected.

Figure 11.70 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 11.71 Example of Operation when Buffer Transfer is Linked with Interrupt Skipping (BTE1 = 1 and BTE0 = 0)

chanhvinh.com

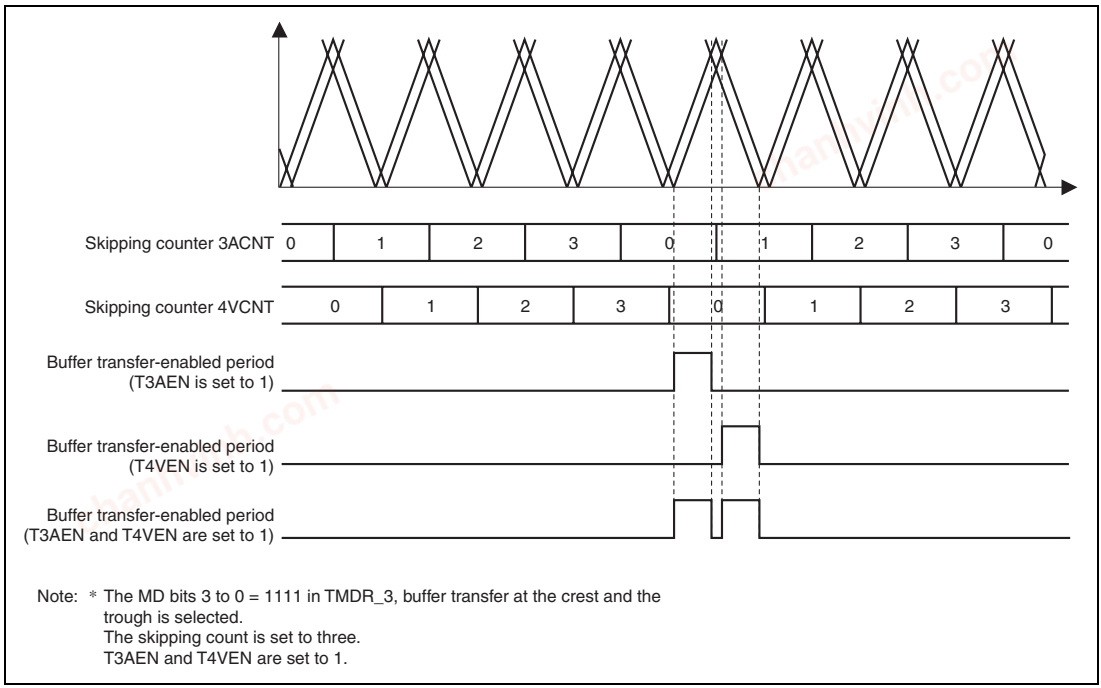


Figure 11.72 Relationship between Bits T3AEN and T4VEN in TITCR and Buffer Transfer-Enabled Period

(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

11.4.9 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 11.73 shows an example of procedure for specifying the A/D converter start request delaying function.

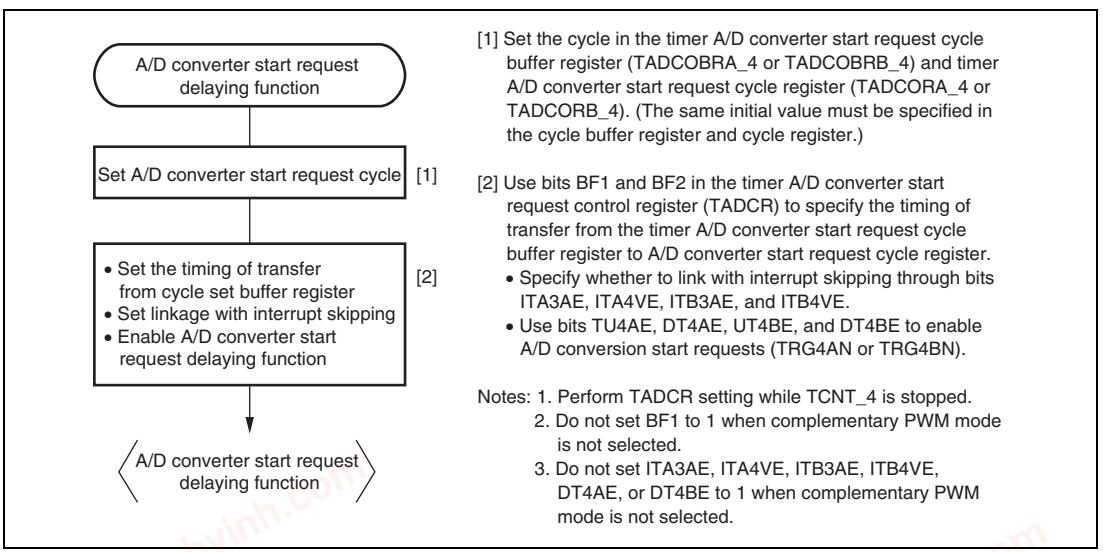


Figure 11.73 Example of Procedure for Specifying A/D Converter Start Request Delaying Function

chanhvinh.com

- Basic Operation Example of A/D Converter Start Request Delaying Function

  Figure 11.74 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.

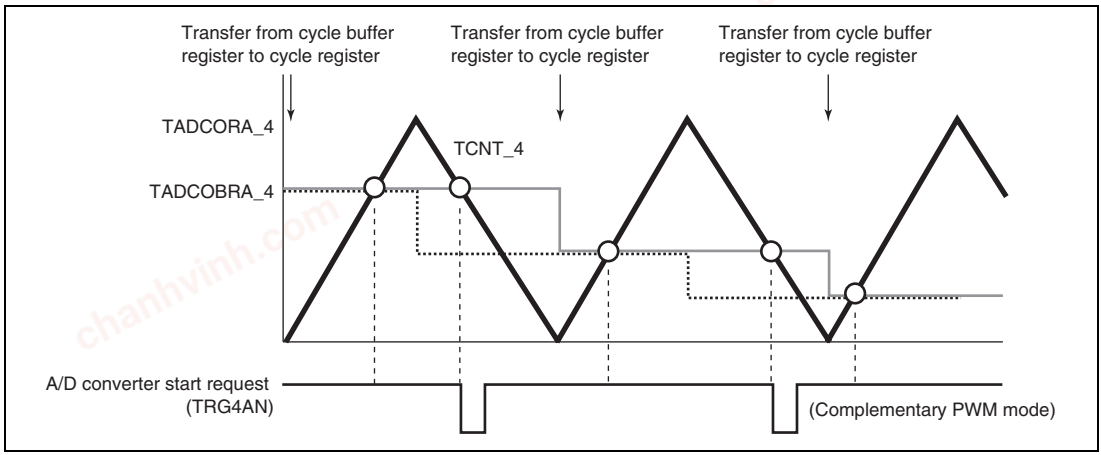


Figure 11.74 Basic Example of A/D Converter Start Request Signal (TRG4AN) Operation

- 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 11.75 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 11.76 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.

chanhvinh.com

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).

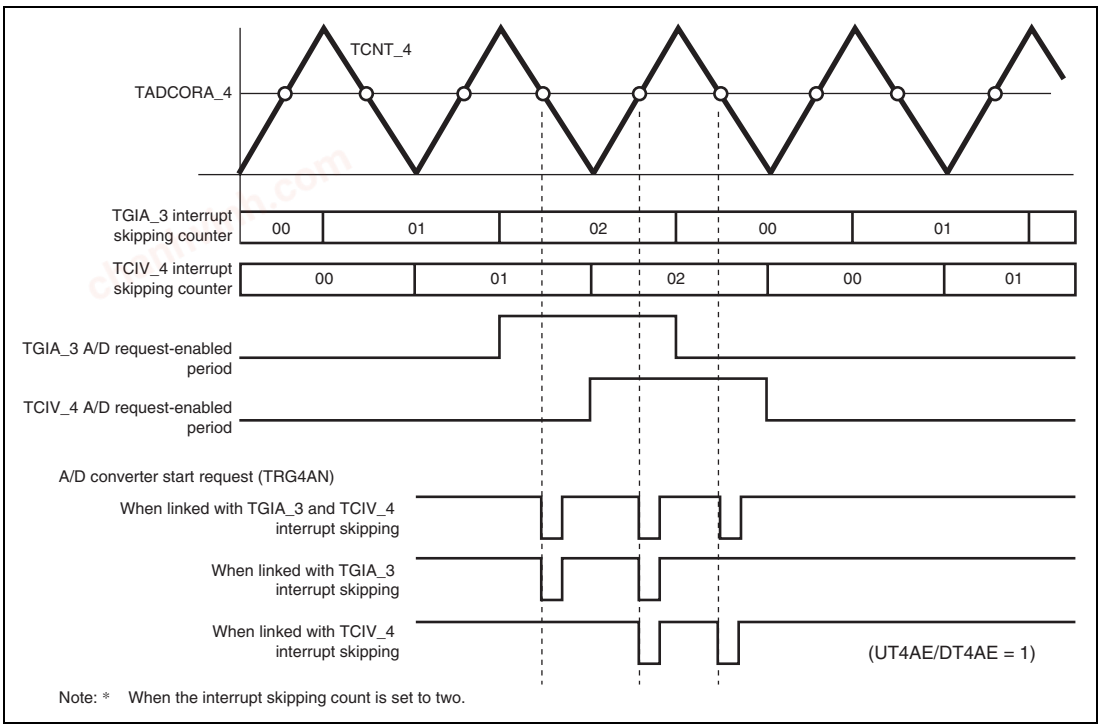


Figure 11.75 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping

chanhvinh.com

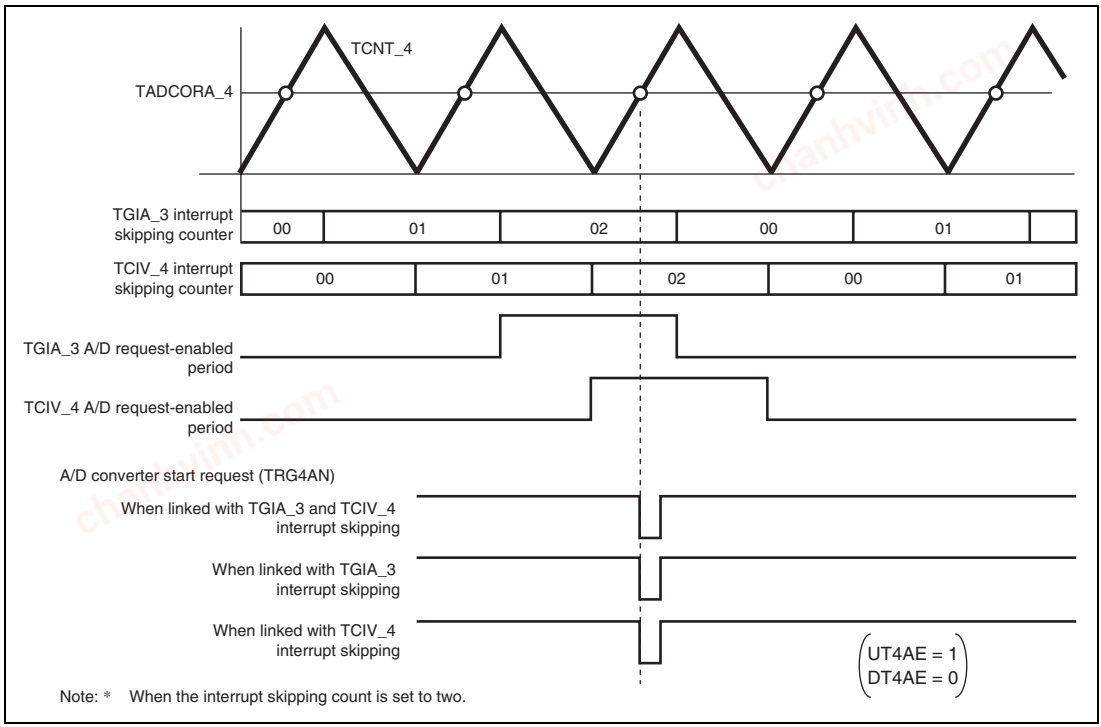


Figure 11.76 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping

chanhvinh.com

11.4.10 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 11.77 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).

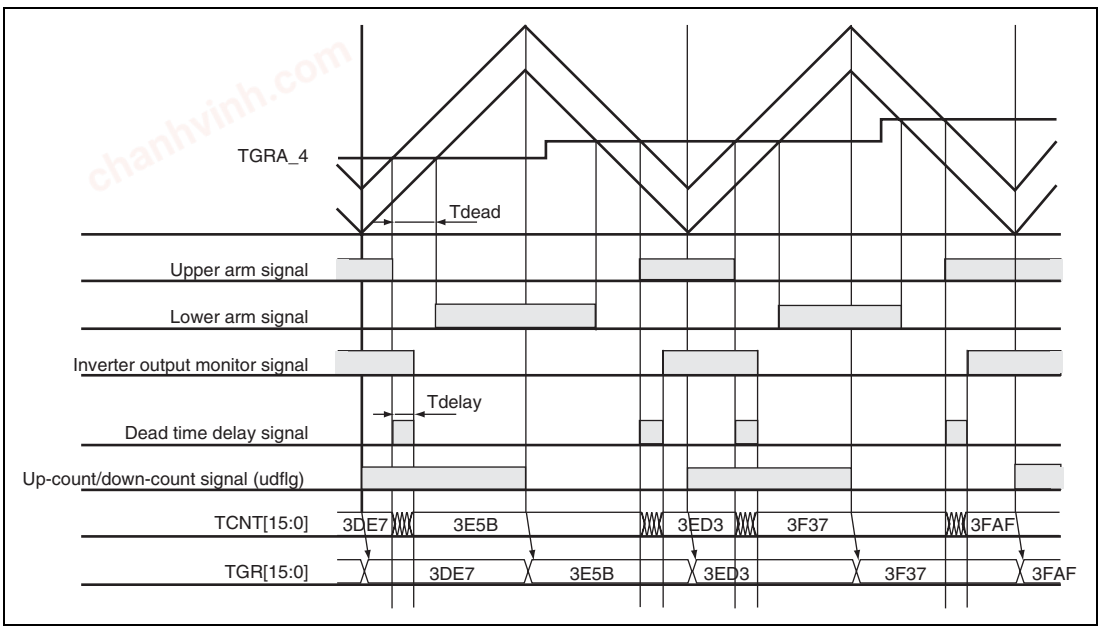


Figure 11.77 TCNT Capturing at Crest and/or Trough in Complementary PWM Operation

chanhvinh.com

11.5 Interrupt Sources

11.5.1 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 11.55 lists the interrupt sources of this module.

Table 11.55 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	Low

chanhvinh.com

Channel	Name	Interrupt Source	Interrupt Flag	Activation of Direct Memory Access Controller	Priority
2	TGIA_2	TGRA_2 input capture/compare match	TGFA_2	Possible	High
	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.

(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.

chanhvinh.com

(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.

11.5.2 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 10, 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.

11.5.3 A/D Converter Activation

The A/D converter can be activated by one of the following three methods in this module. Table 11.56 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 11.4.9, 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 11.56 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

11.6 Operation Timing

11.6.1 Input/Output Timing

(1) TCNT Count Timing

Figure 11.78 shows TCNT count timing in internal clock operation, and Figure 11.79 shows TCNT count timing in external clock operation (normal mode), and Figure 11.80 shows TCNT count timing in external clock operation (phase counting mode).

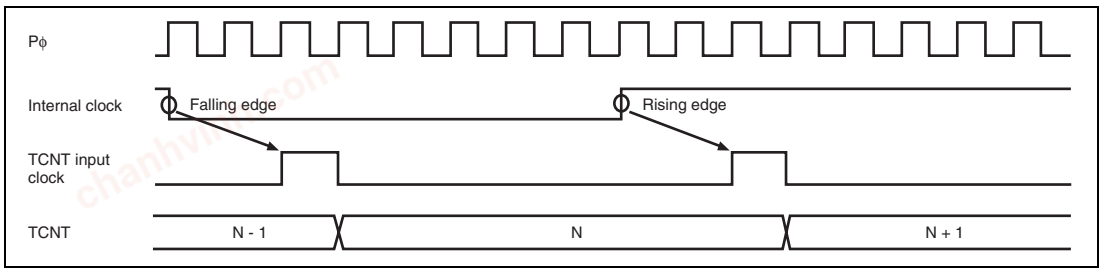


Figure 11.78 Count Timing in Internal Clock Operation

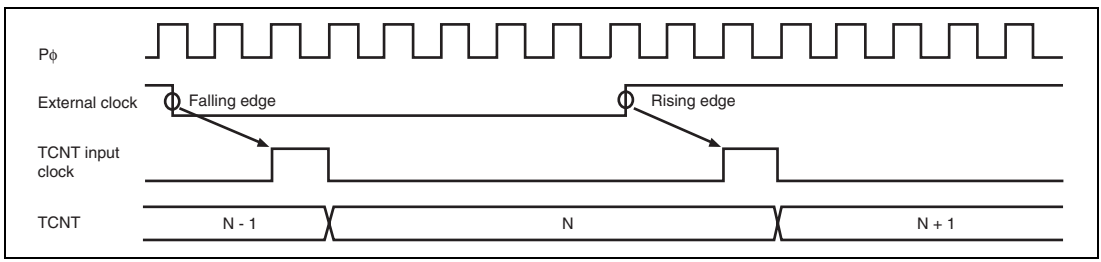


Figure 11.79 Count Timing in External Clock Operation

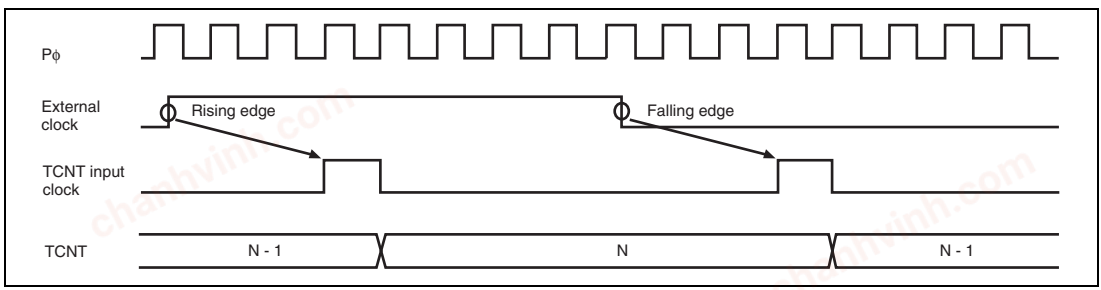


Figure 11.80 Count Timing in External Clock Operation (Phase Counting Mode)

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 11.81 shows output compare output timing (normal mode and PWM mode) and Figure 11.82 shows output compare output timing (complementary PWM mode and reset synchronous PWM mode).

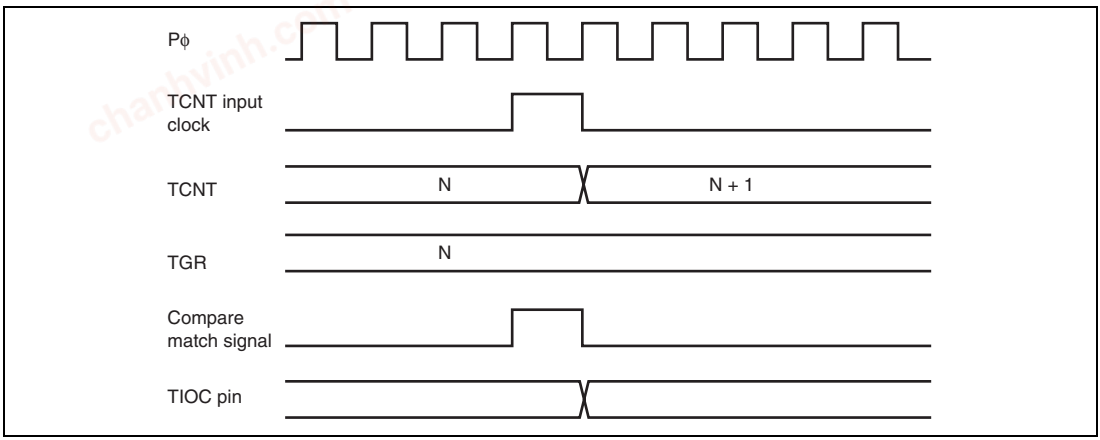


Figure 11.81 Output Compare Output Timing (Normal Mode/PWM Mode)

chanhvinh.com

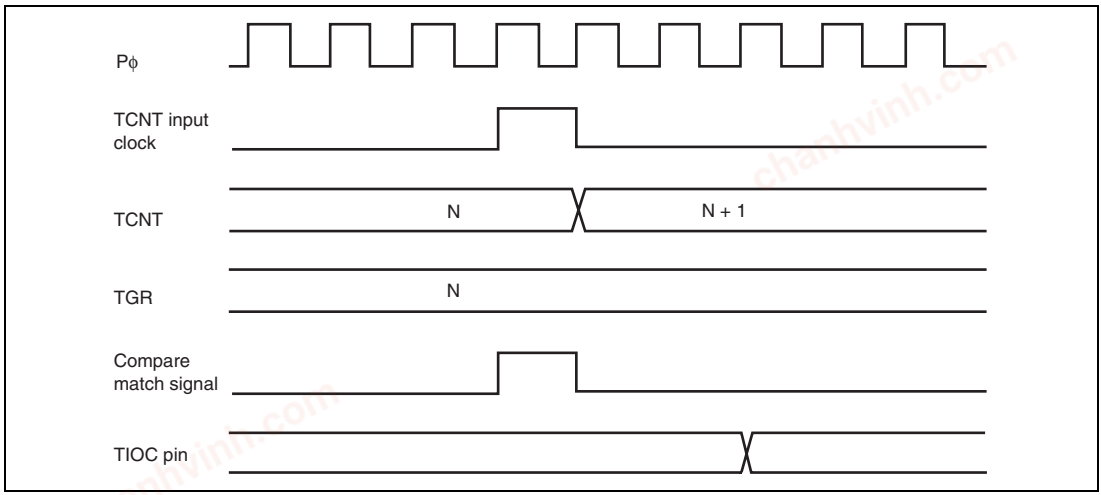


Figure 11.82 Output Compare Output Timing (Complementary PWM Mode/Reset Synchronous PWM Mode)

(3) Input Capture Signal Timing

Figure 11.83 shows input capture signal timing.

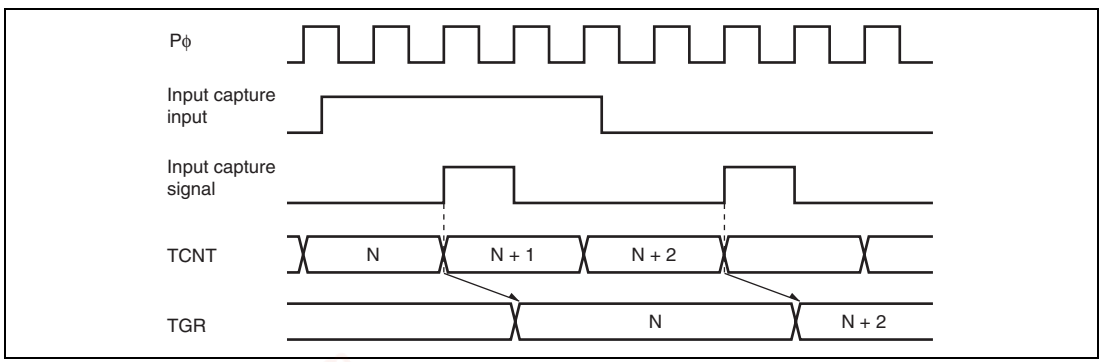


Figure 11.83 Input Capture Input Signal Timing

chanhvinh.com

(4) Timing for Counter Clearing by Compare Match/Input Capture

Figure 11.84 shows the timing when counter clearing on compare match is specified, and Figure 11.85 shows the timing when counter clearing on input capture is specified.

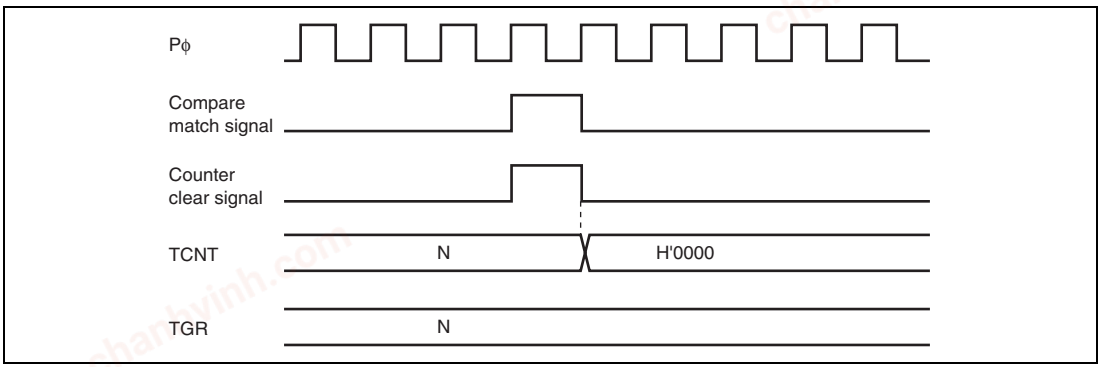


Figure 11.84 Counter Clear Timing (Compare Match)

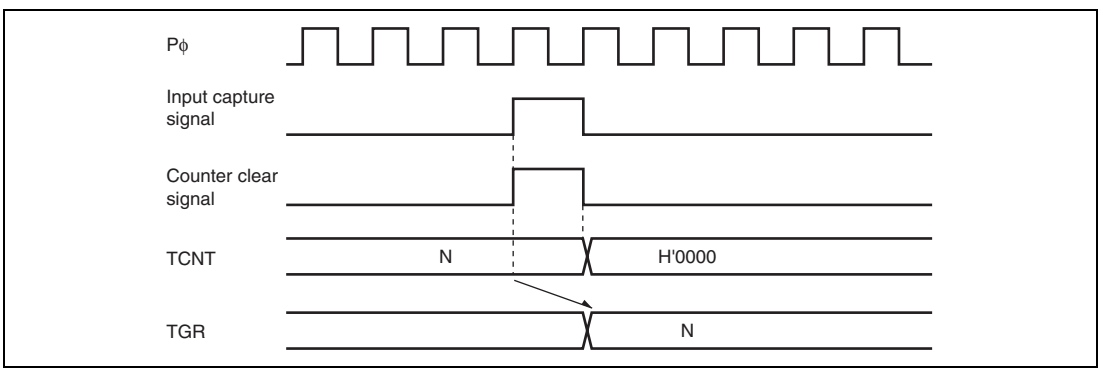


Figure 11.85 Counter Clear Timing (Input Capture)

chanhvinh.com

(5) Buffer Operation Timing

Figures 11.86 to 11.88 show the timing in buffer operation.

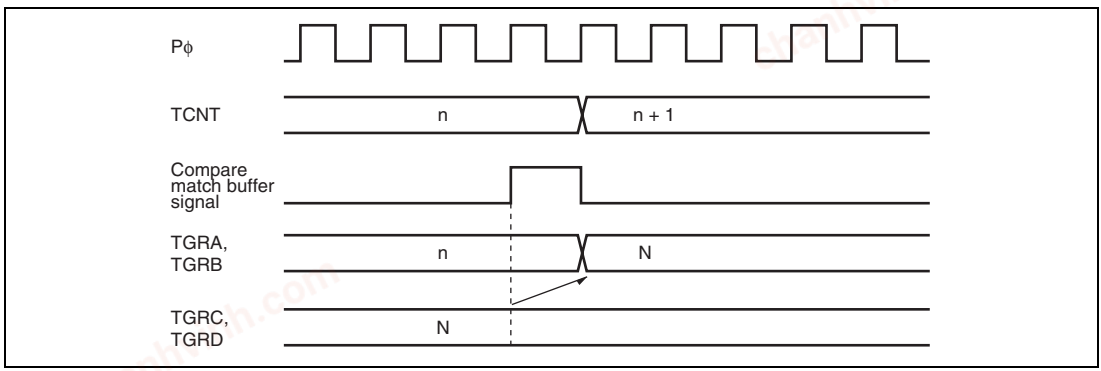


Figure 11.86 Buffer Operation Timing (Compare Match)

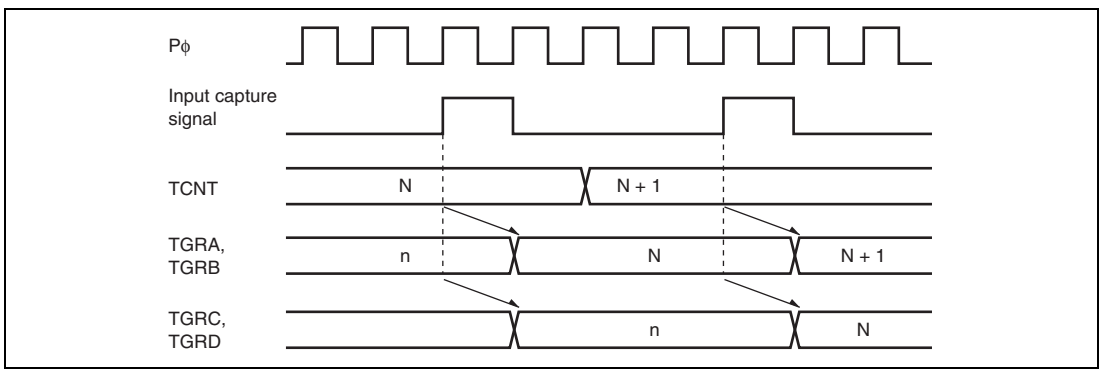


Figure 11.87 Buffer Operation Timing (Input Capture)

chanhvinh.com

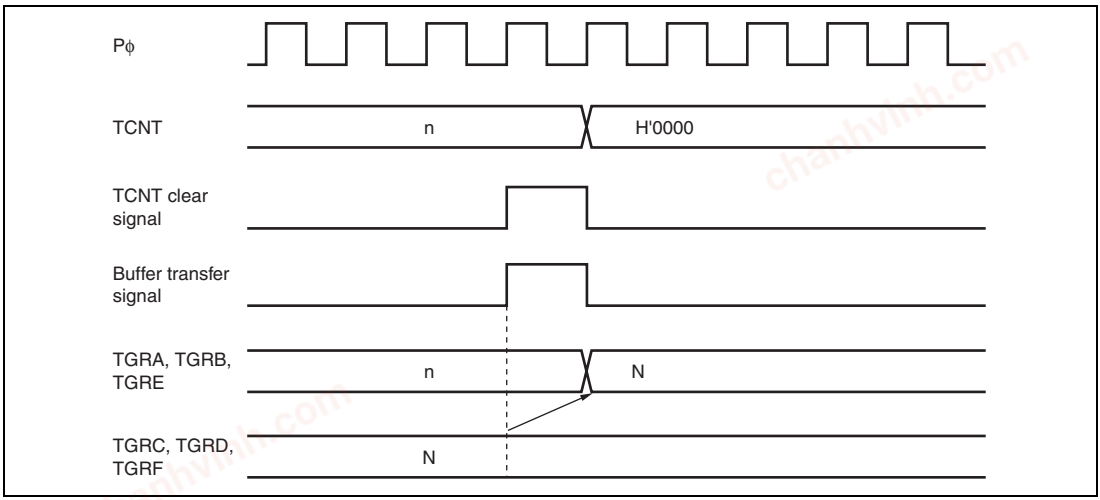


Figure 11.88 Buffer Transfer Timing (when TCNT Cleared)

(6) Buffer Transfer Timing (Complementary PWM Mode)

Figures 11.89 to 11.91 show the buffer transfer timing in complementary PWM mode.

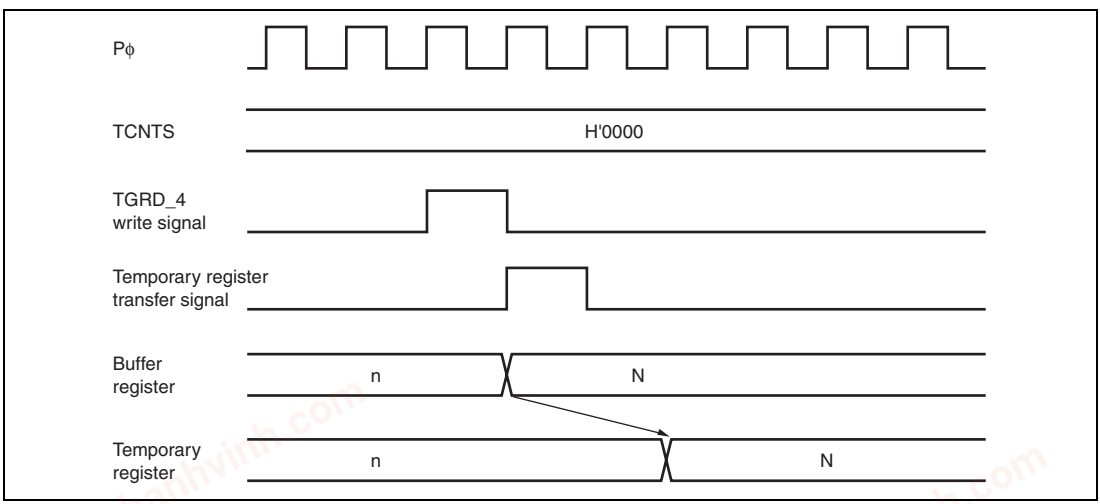


Figure 11.89 Transfer Timing from Buffer Register to Temporary Register (TCNTS Stop)

chanhvinh.com

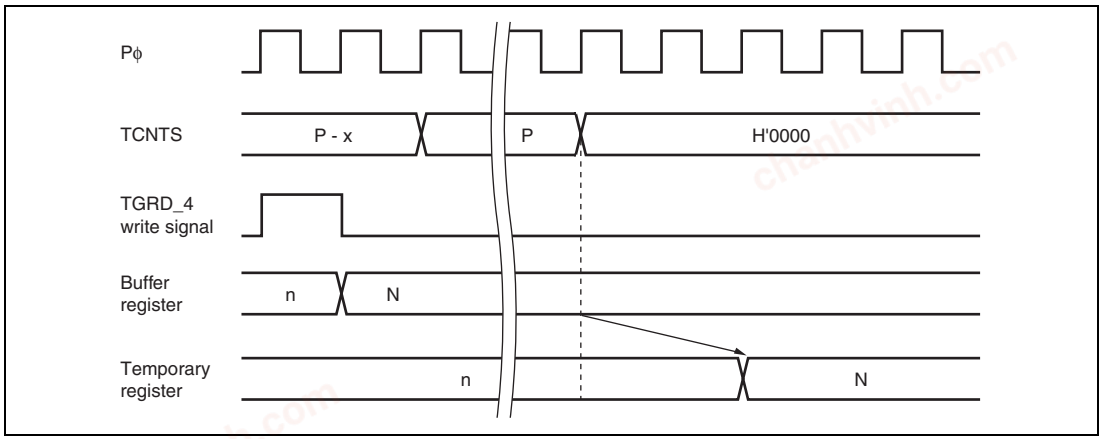


Figure 11.90 Transfer Timing from Buffer Register to Temporary Register (TCNTS Operating)

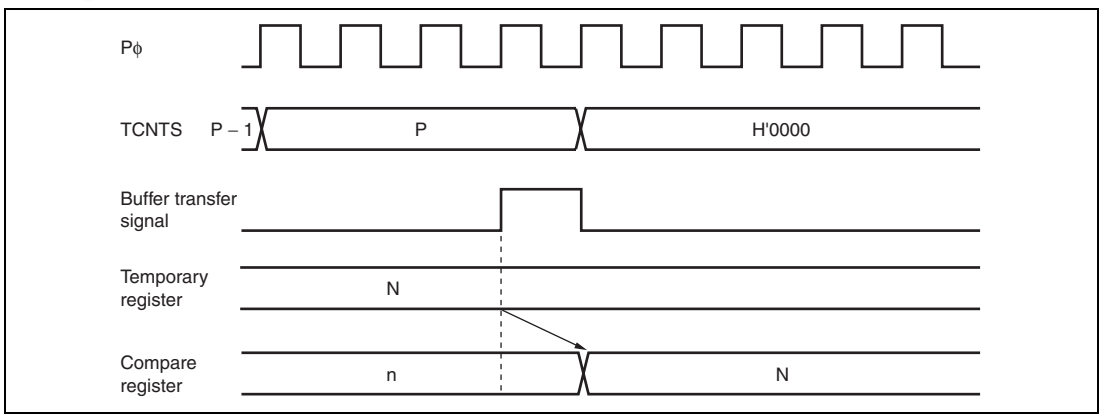


Figure 11.91 Transfer Timing from Temporary Register to Compare Register

chanhvinh.com

11.6.2 Interrupt Signal Timing

(1) TGF Flag Setting Timing in Case of Compare Match

Figure 11.92 shows the timing for setting of the TGF flag in TSR on compare match, and TGI interrupt request signal timing.

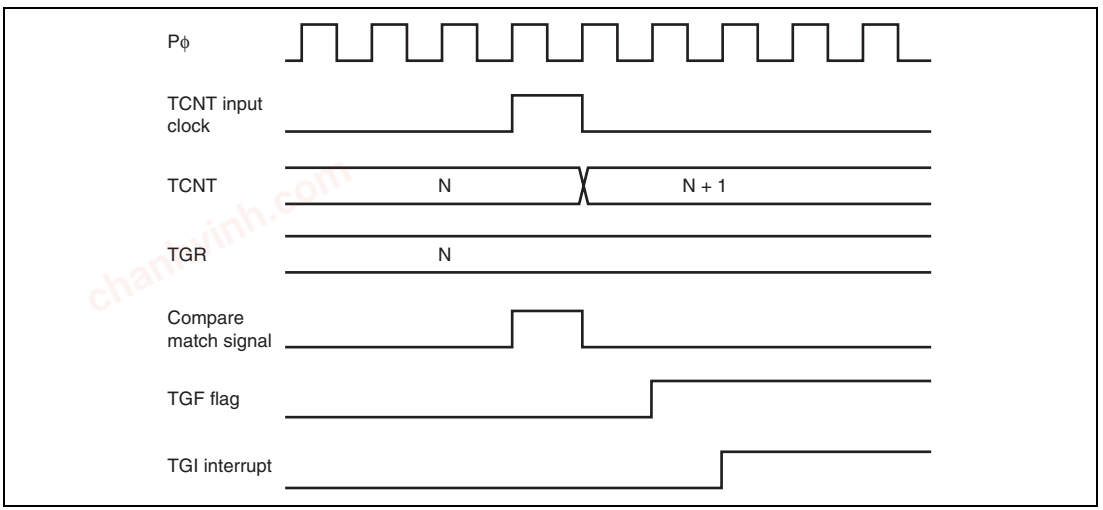


Figure 11.92 TGI Interrupt Timing (Compare Match)

chanhvinh.com

(2) TGF Flag Setting Timing in Case of Input Capture

Figure 11.93 shows the timing for setting of the TGF flag in TSR on input capture, and TGI interrupt request signal timing.

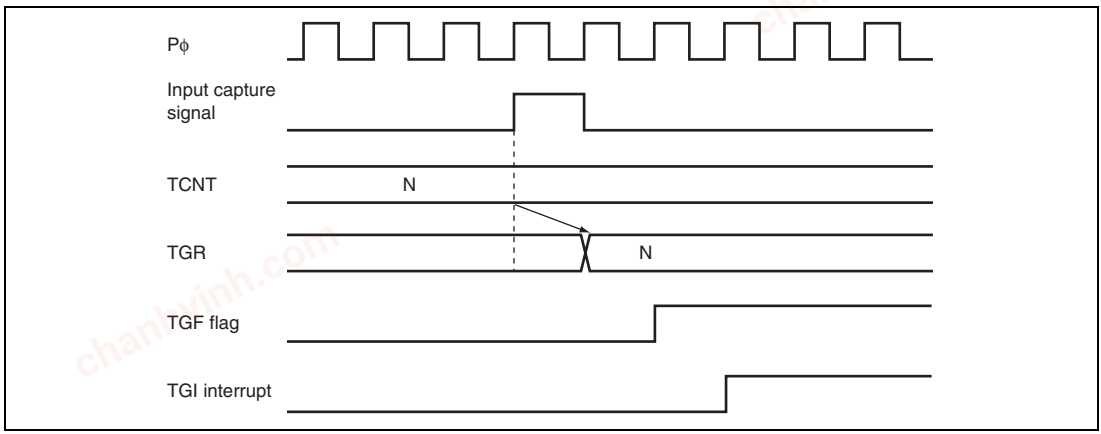


Figure 11.93 TGI Interrupt Timing (Input Capture)

chanhvinh.com

(3) TCFV Flag/TCFU Flag Setting Timing

Figure 11.94 shows the timing for setting of the TCFV flag in TSR on overflow, and TCIV interrupt request signal timing.

Figure 11.95 shows the timing for setting of the TCFU flag in TSR on underflow, and TCIU interrupt request signal timing.

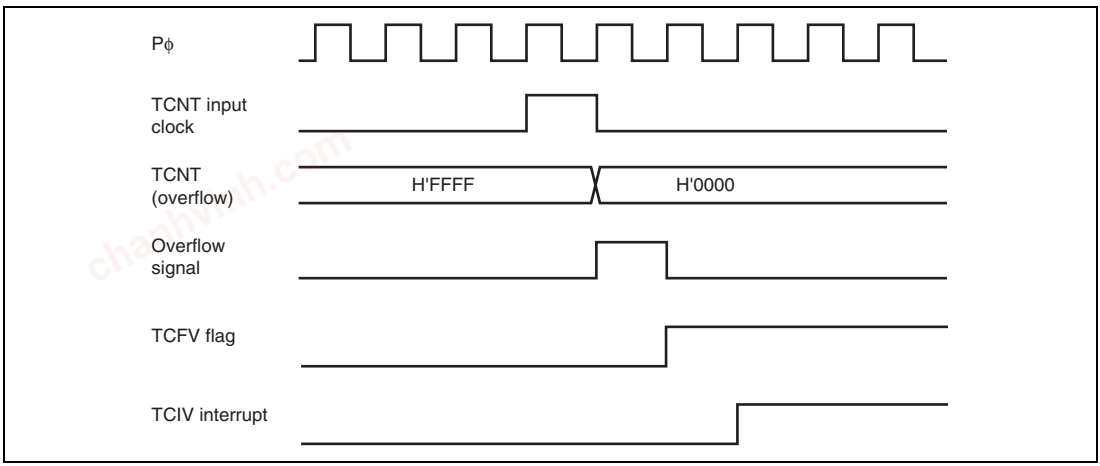


Figure 11.94 TCIV Interrupt Setting Timing

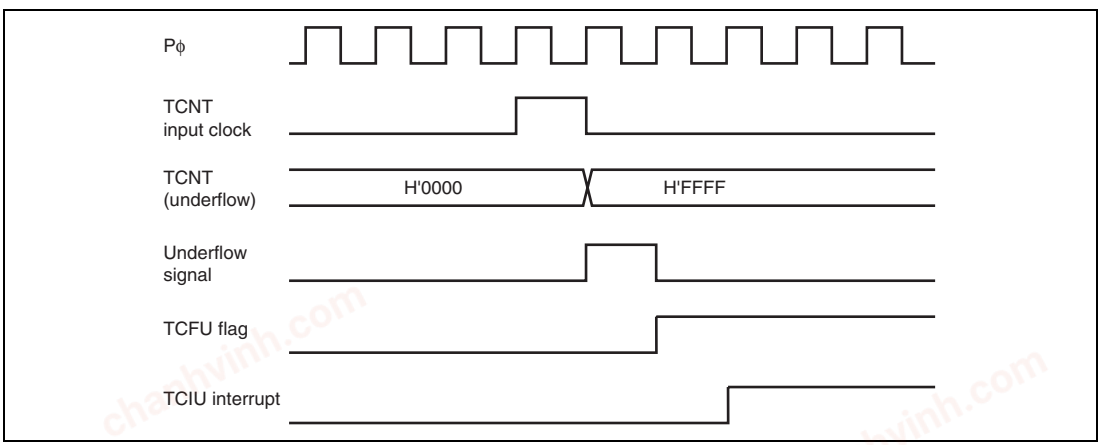


Figure 11.95 TCIU Interrupt Setting Timing

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 11.96 shows the timing for status flag clearing by the CPU, and Figure 11.97 shows the timing for status flag clearing by the direct memory access controller.

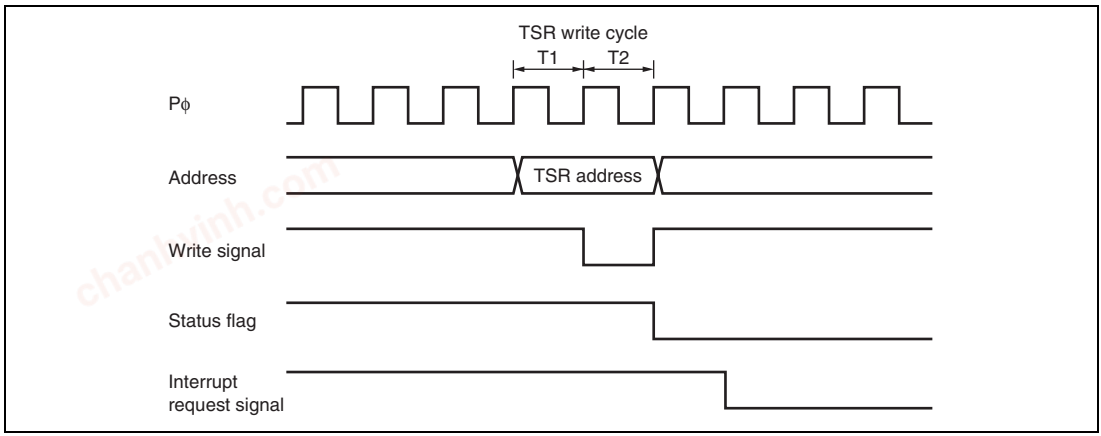


Figure 11.96 Timing for Status Flag Clearing by CPU

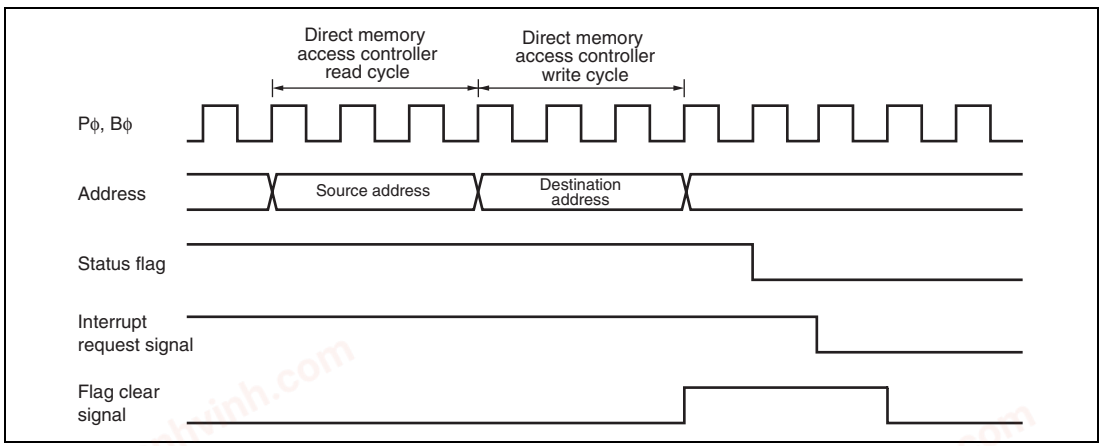


Figure 11.97 Timing for Status Flag Clearing by Direct Memory Access Controller Activation

chanhvinh.com

11.7 Usage Notes

11.7.1 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 33, Power-Down Modes.

11.7.2 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 11.98 shows the input clock conditions in phase counting mode.

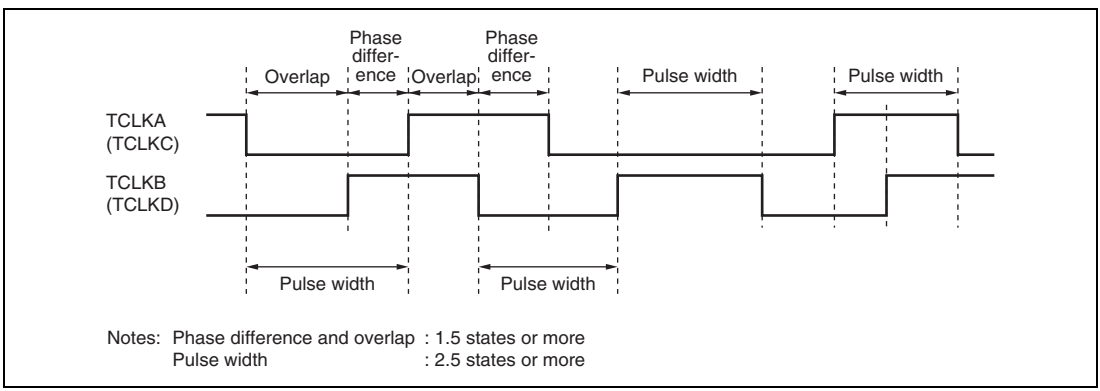


Figure 11.98 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

chanhvinh.com

11.7.3 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:

$$f = \frac{P\phi}{(N + 1)}$$

Where f: Counter frequency
Pϕ: Peripheral clock operating frequency
N: TGR set value

11.7.4 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 11.99 shows the timing in this case.

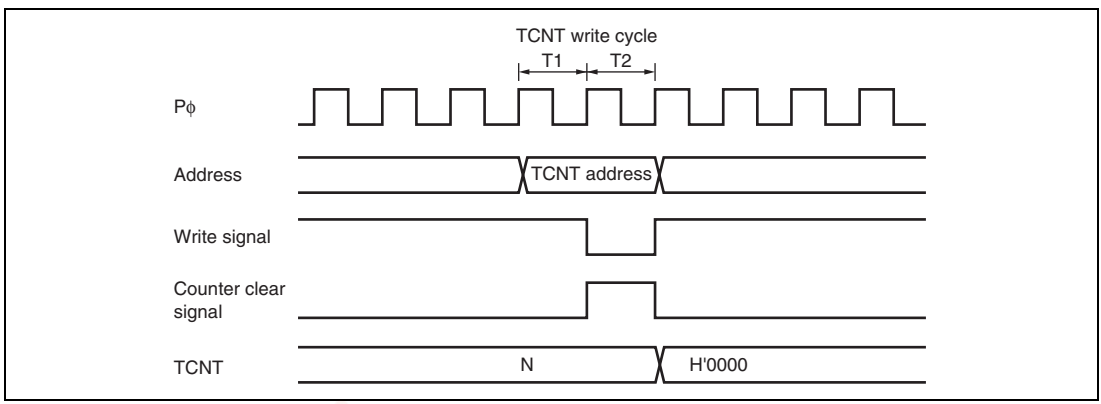


Figure 11.99 Contention between TCNT Write and Clear Operations

chanhvinh.com

11.7.5 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 11.100 shows the timing in this case.

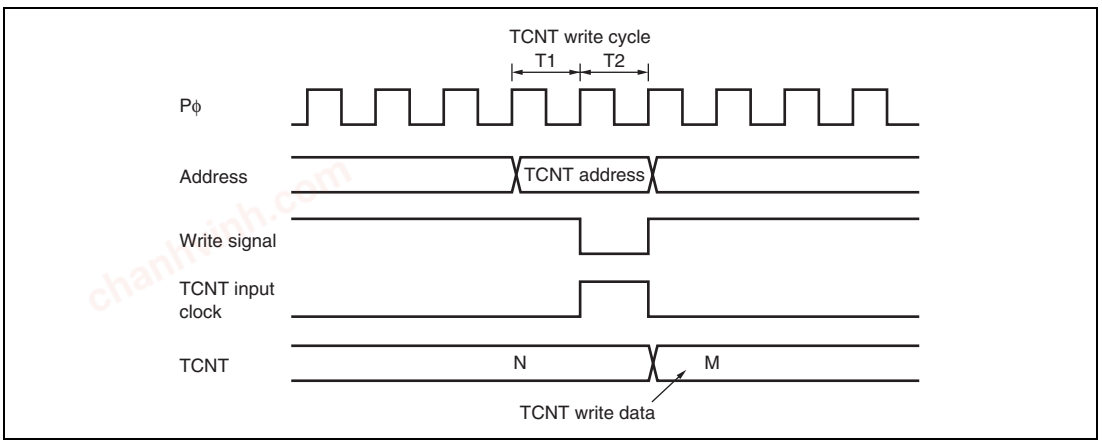


Figure 11.100 Contention between TCNT Write and Increment Operations

chanhvinh.com

11.7.6 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 11.101 shows the timing in this case.

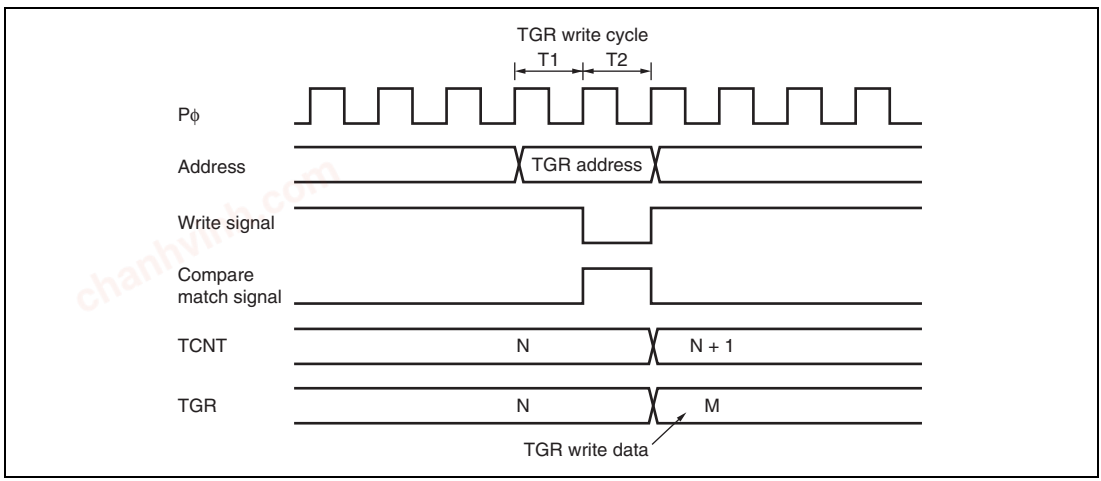


Figure 11.101 Contention between TGR Write and Compare Match

chanhvinh.com

11.7.7 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 11.102 shows the timing in this case.

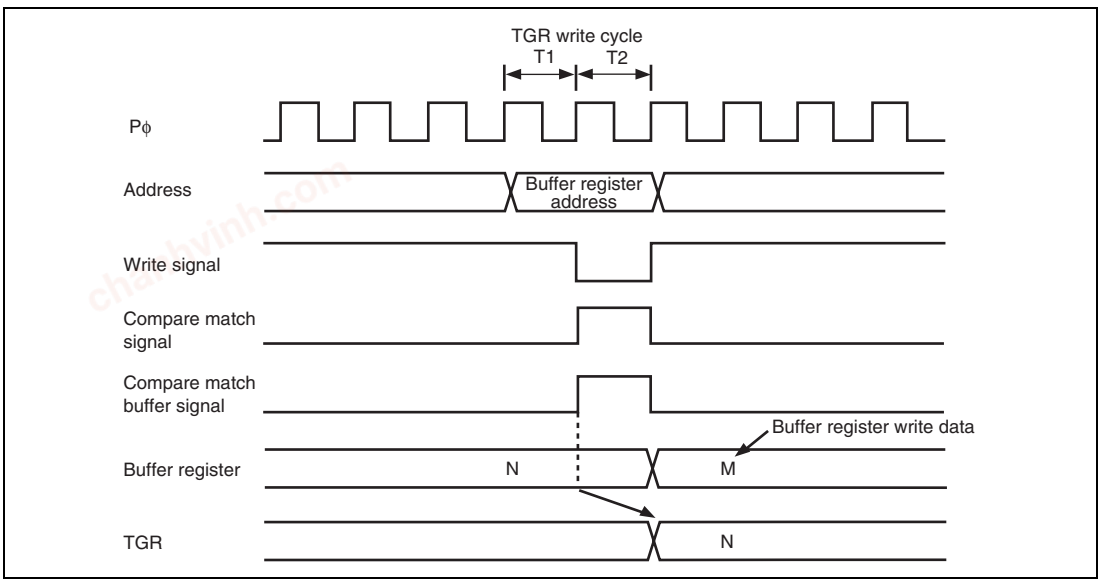


Figure 11.102 Contention between Buffer Register Write and Compare Match

chanhvinh.com

11.7.8 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 11.103 shows the timing in this case.

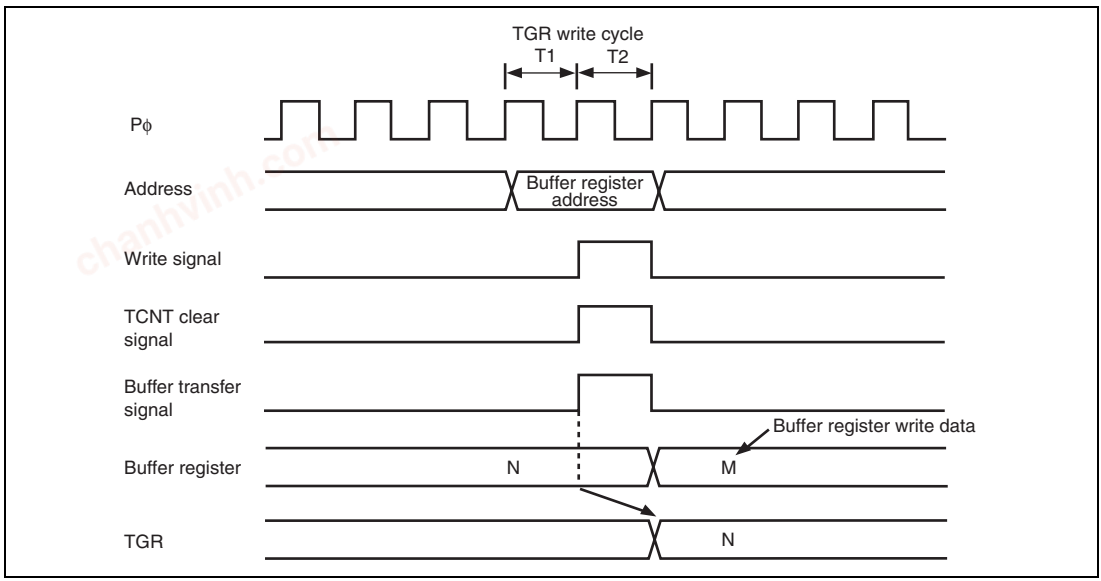


Figure 11.103 Contention between Buffer Register Write and TCNT Clear

chanhvinh.com

11.7.9 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 11.104 shows the timing in this case.

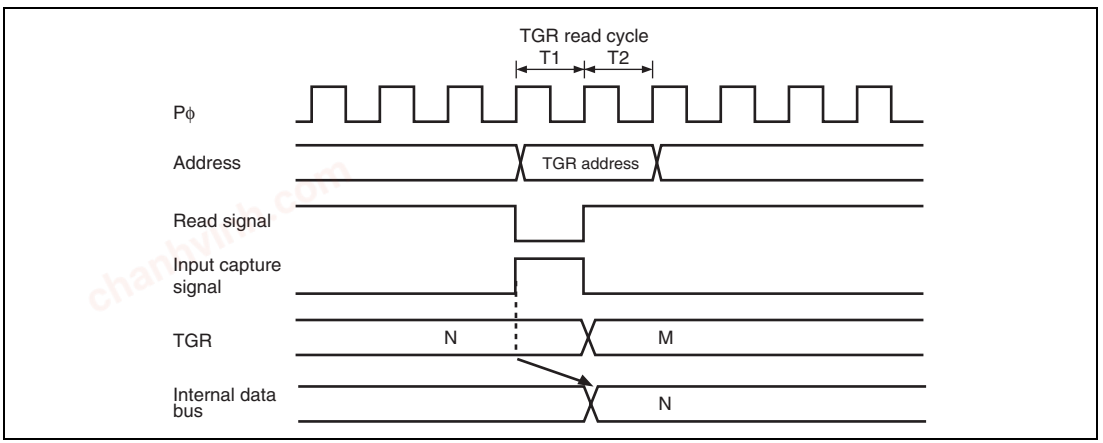


Figure 11.104 Contention between TGR Read and Input Capture

chanhvinh.com

11.7.10 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 11.105 shows the timing in this case.

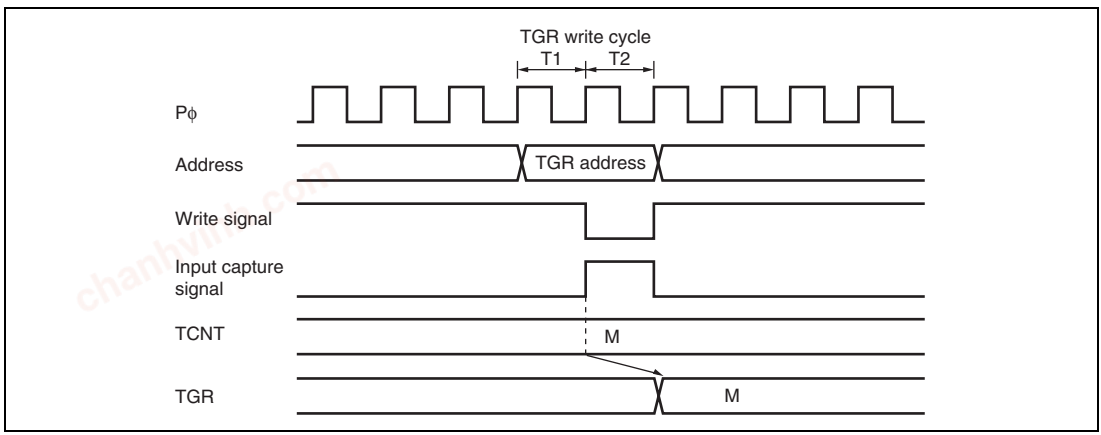


Figure 11.105 Contention between TGR Write and Input Capture

chanhvinh.com

11.7.11 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 11.106 shows the timing in this case.

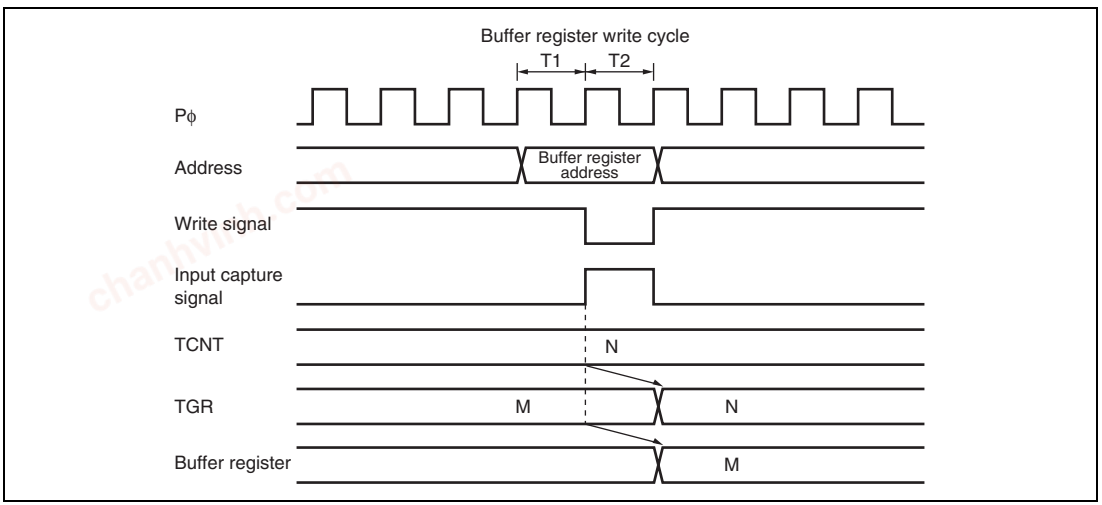


Figure 11.106 Contention between Buffer Register Write and Input Capture

11.7.12 TCNT2 Write and Overflow/Underflow Contention in Cascade Connection

With timer counters TCNT1 and TCNT2 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 11.107.

For cascade connections, be sure to synchronize settings for channels 1 and 2 when setting TCNT clearing.

chanhvinh.com

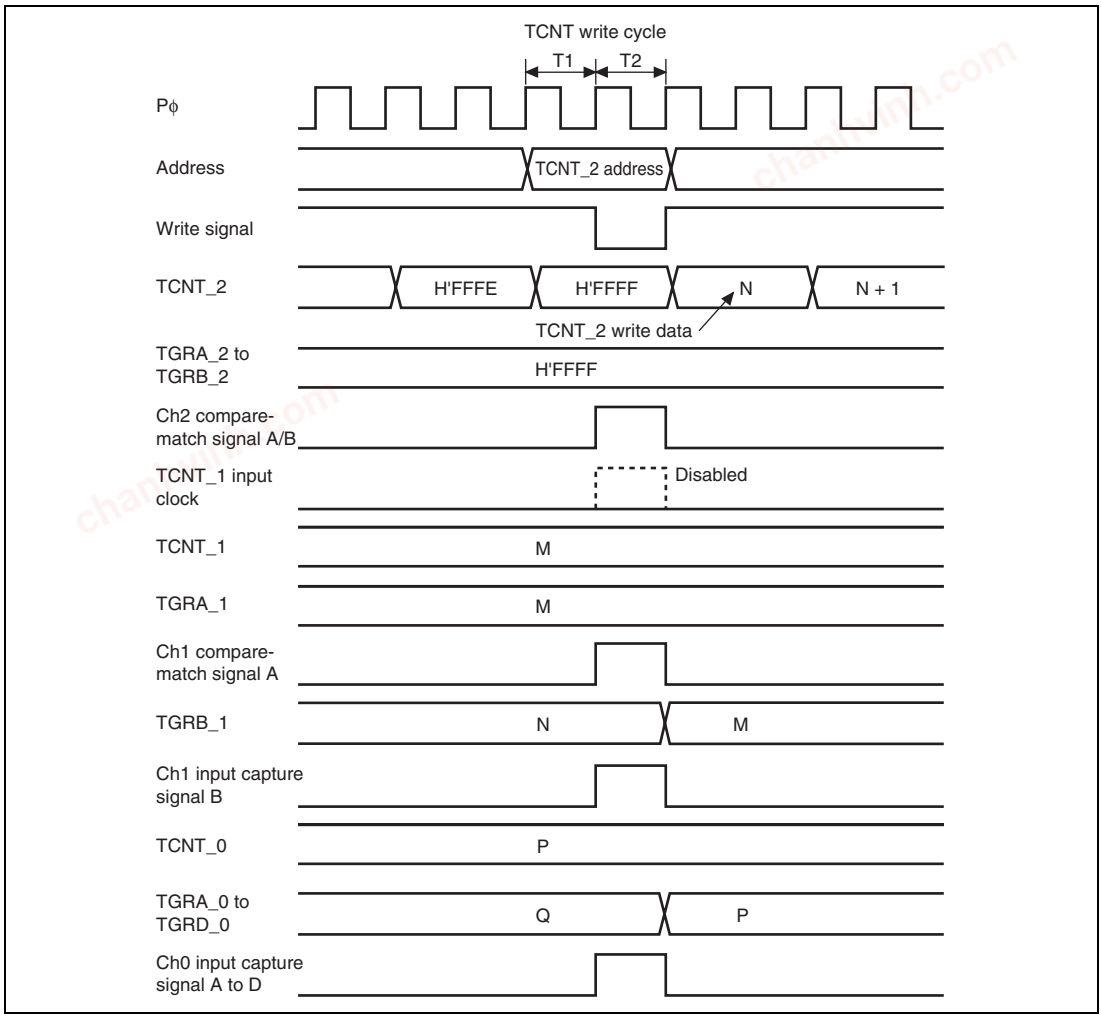


Figure 11.107 TCNT_2 Write and Overflow/Underflow Contention with Cascade Connection

chanhvinh.com

11.7.13 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 11.108.

When counting begins in another operating mode, be sure that TCNT_3 and TCNT_4 are set to the initial values.

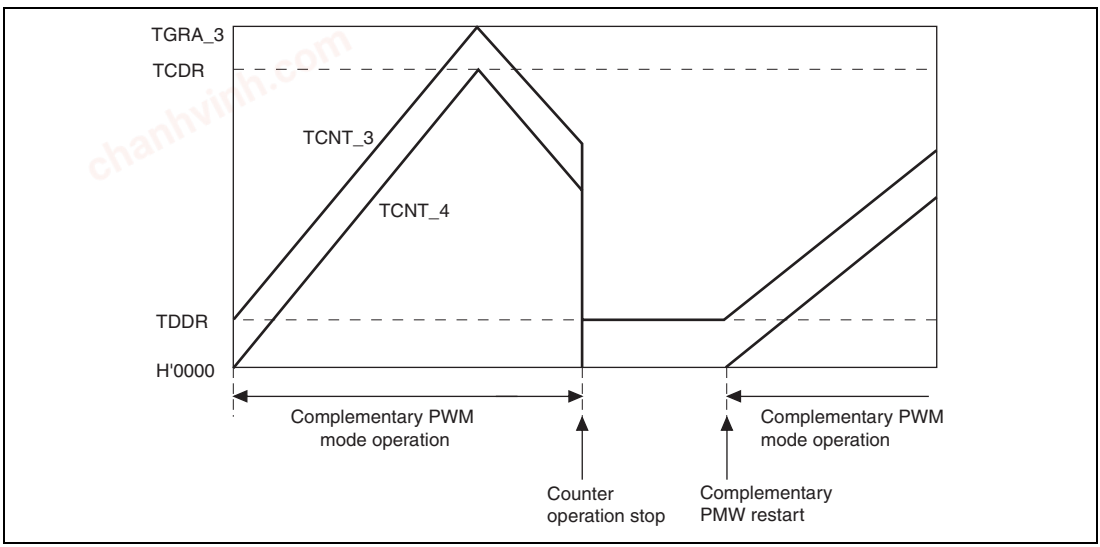


Figure 11.108 Counter Value during Complementary PWM Mode Stop

11.7.14 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

11.7.15 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 11.109 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.

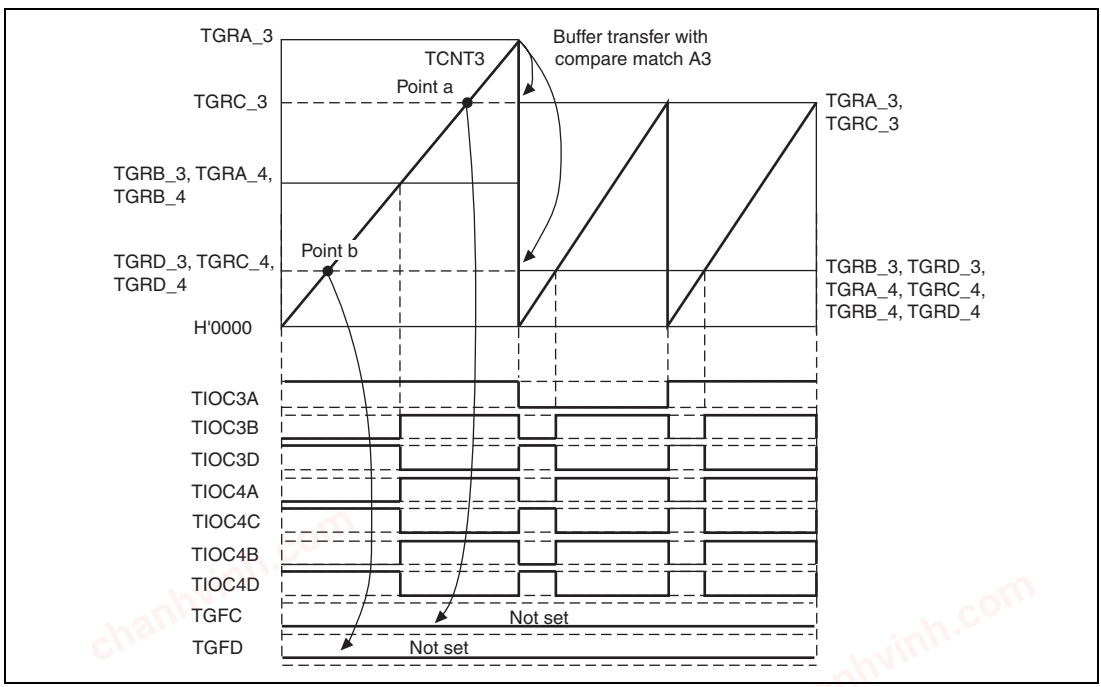


Figure 11.109 Buffer Operation and Compare-Match Flags in Reset Synchronous PWM Mode

chanhvinh.com

11.7.16 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 11.110 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.

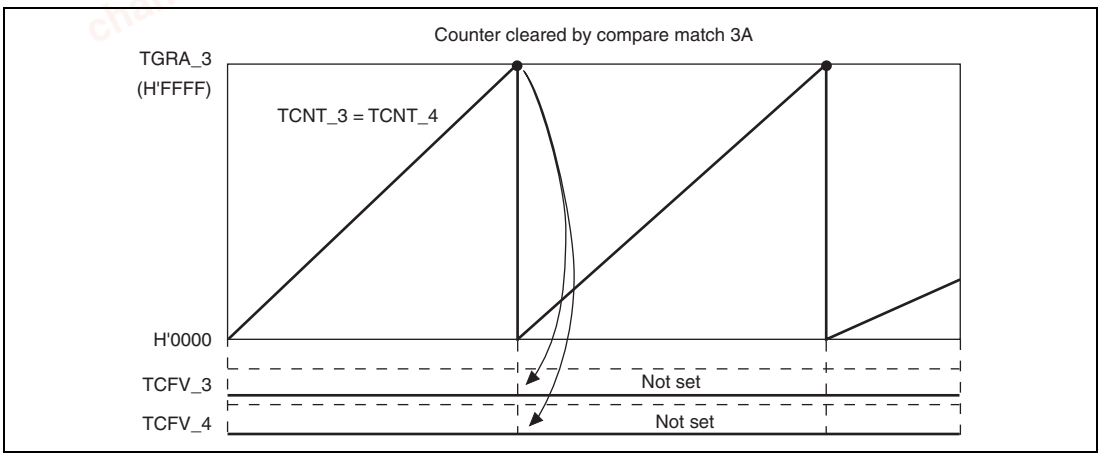


Figure 11.110 Reset Synchronous PWM Mode Overflow Flag

chanhvinh.com

11.7.17 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 11.111 shows the operation timing when a TGR compare match is specified as the clearing source, and when H'FFFF is set in TGR.

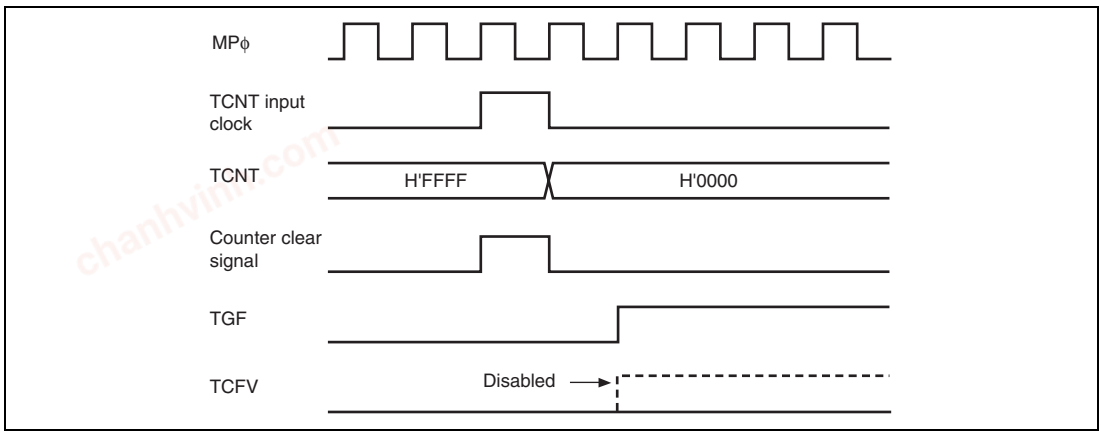


Figure 11.111 Contention between Overflow and Counter Clearing

chanhvinh.com

11.7.18 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 11.112 shows the operation timing when there is contention between TCNT write and overflow.

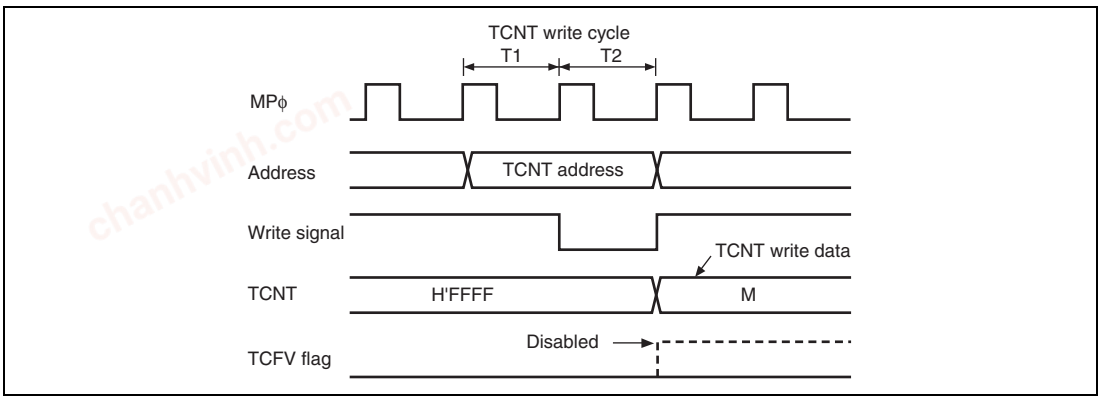


Figure 11.112 Contention between TCNT Write and Overflow

11.7.19 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.

chanhvinh.com

11.7.20 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.

11.7.21 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.

11.7.22 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.

chanhvinh.com

11.7.23 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 11.113).

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 11.114)

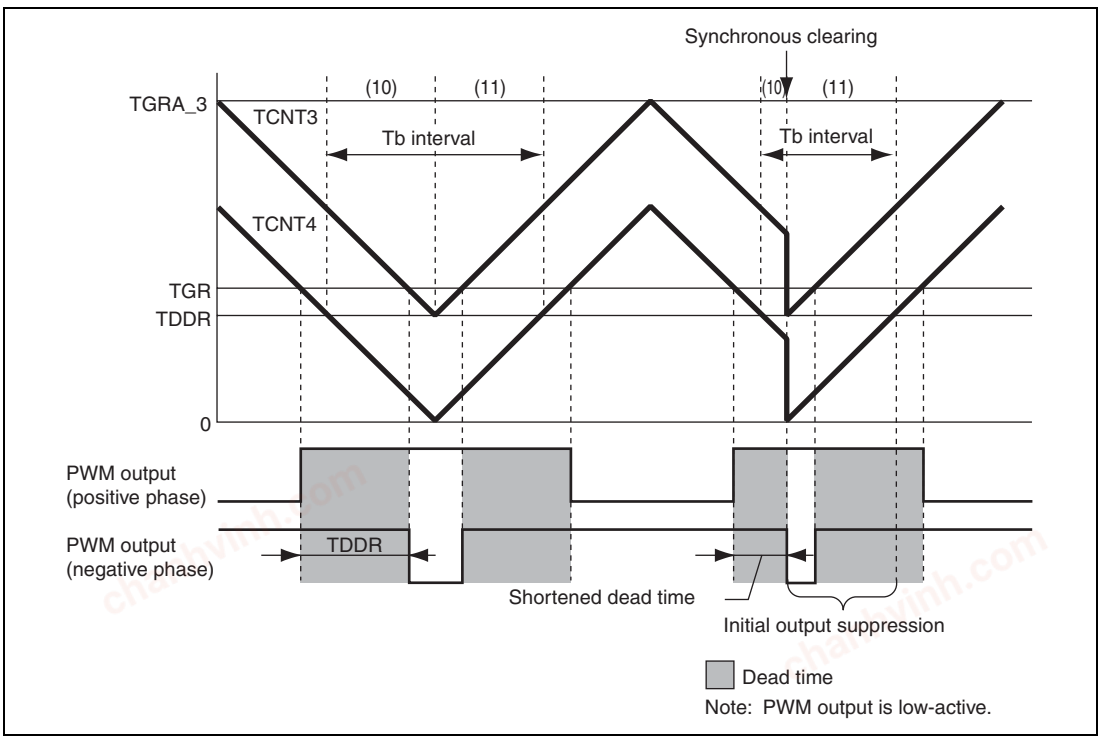


Figure 11.113 Condition (1) Synchronous Clearing Example

chanhvinh.com

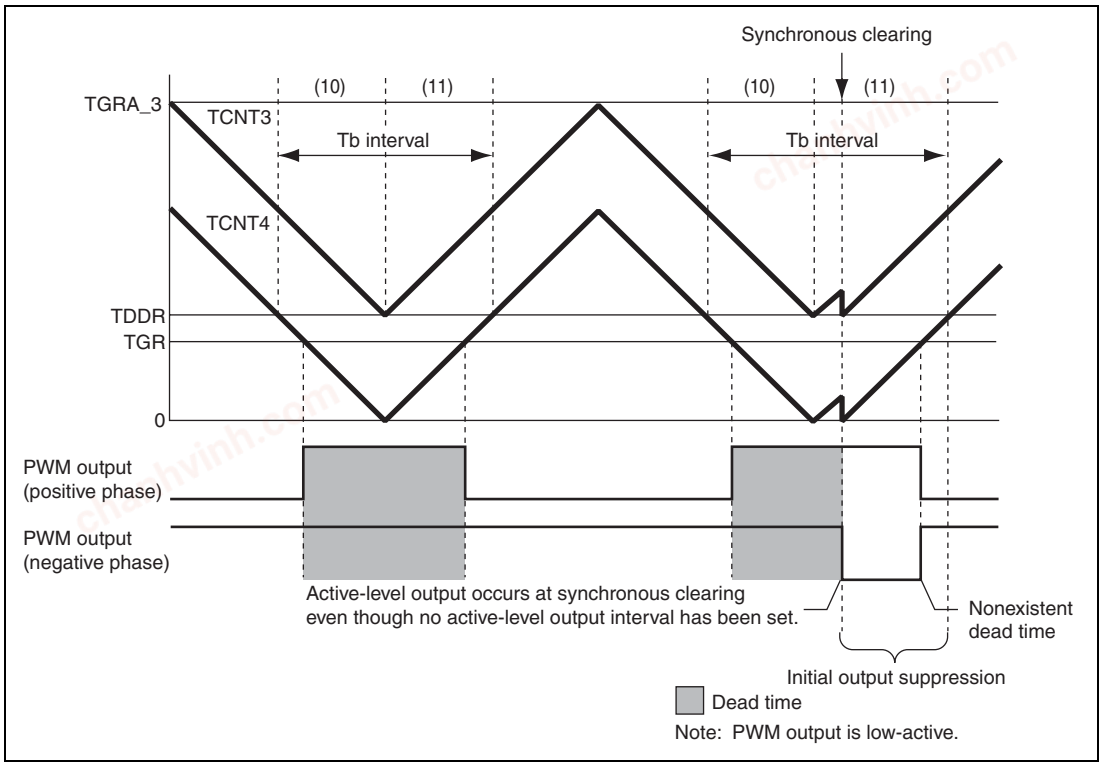


Figure 11.114 Condition (2) Synchronous Clearing Example

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

11.8 Output Pin Initialization for Multi-Function Timer Pulse Unit 2

11.8.1 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.

11.8.2 Reset Start Operation

The output pins of this module (TIOC*) are initialized low by a reset and in 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

11.8.3 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 11.57.

Table 11.57 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

11.8.4 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 11.57. 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 11.115 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in normal mode after re-setting.

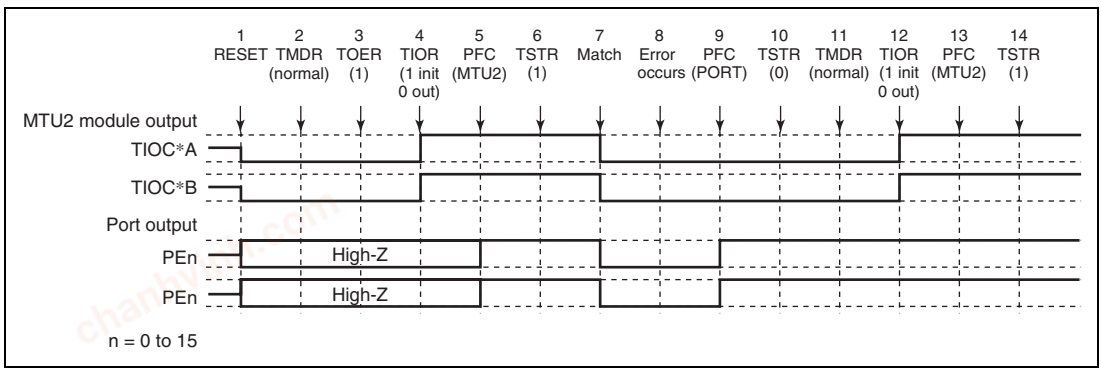


Figure 11.115 Error Occurrence in Normal Mode, Recovery in Normal Mode

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 11.116 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.

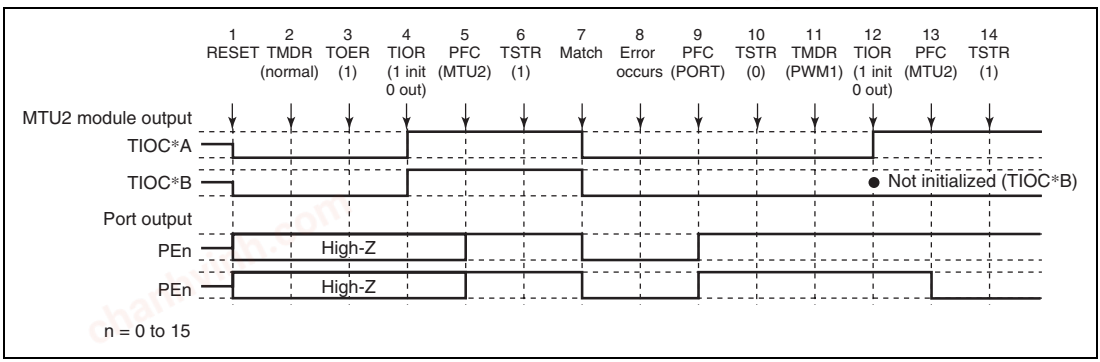


Figure 11.116 Error Occurrence in Normal Mode, Recovery in PWM Mode 1

1 to 10 are the same as in figure 11.115.

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 11.117 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.

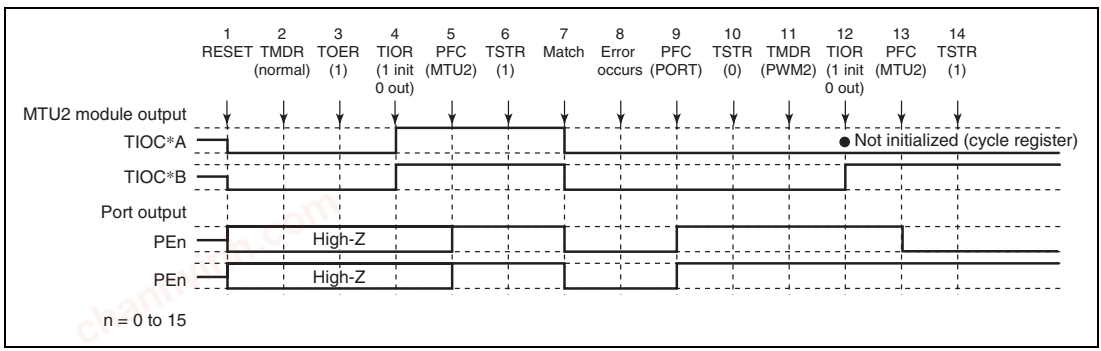


Figure 11.117 Error Occurrence in Normal Mode, Recovery in PWM Mode 2

1 to 10 are the same as in figure 11.115.

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 11.118 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.

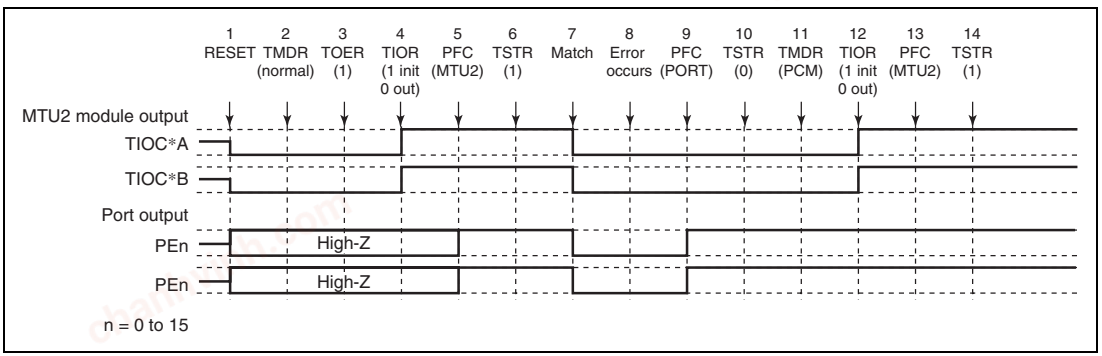


Figure 11.118 Error Occurrence in Normal Mode, Recovery in Phase Counting Mode

1 to 10 are the same as in figure 11.115.

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 11.119 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.

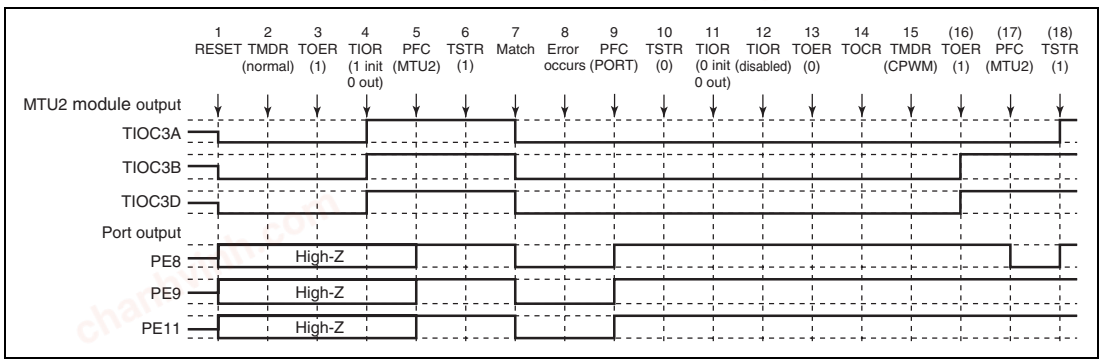


Figure 11.119 Error Occurrence in Normal Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 11.115.

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 11.120 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.

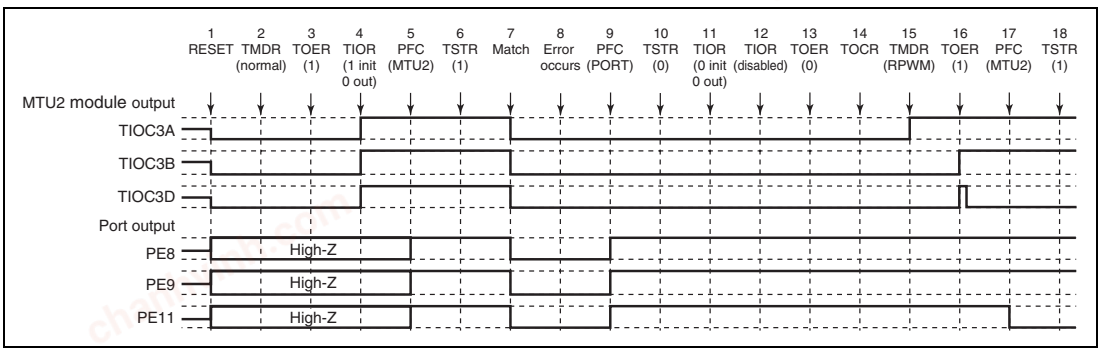


Figure 11.120 Error Occurrence in Normal Mode, Recovery in Reset-Synchronized PWM Mode

1 to 13 are the same as in figure 11.115.

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 11.121 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.

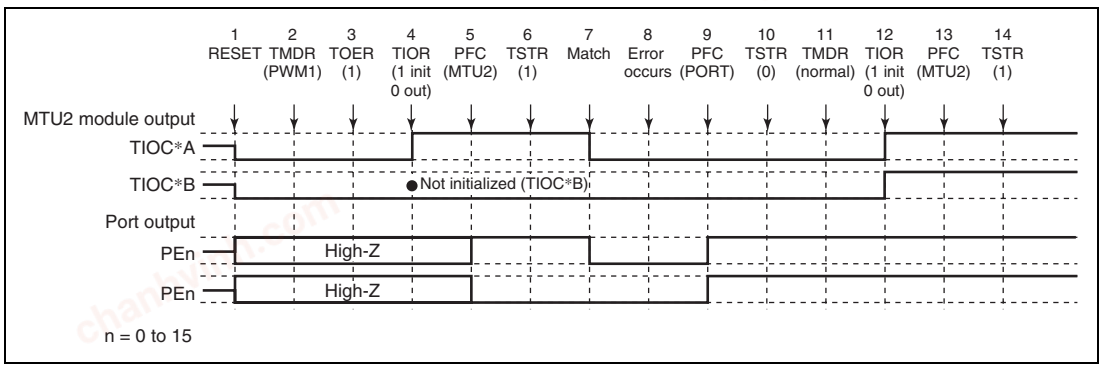


Figure 11.121 Error Occurrence in PWM Mode 1, Recovery in Normal Mode

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 11.122 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.

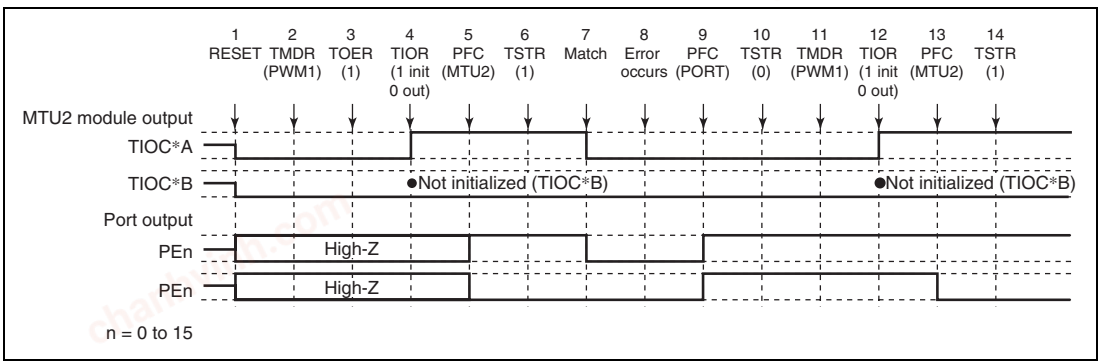


Figure 11.122 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 1

1 to 10 are the same as in figure 11.121.

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 11.123 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.

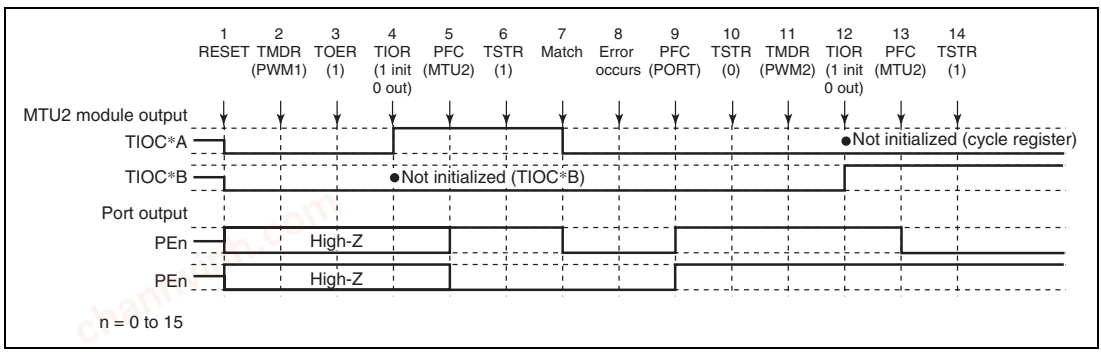


Figure 11.123 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 2

1 to 10 are the same as in figure 11.121.

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 11.124 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.

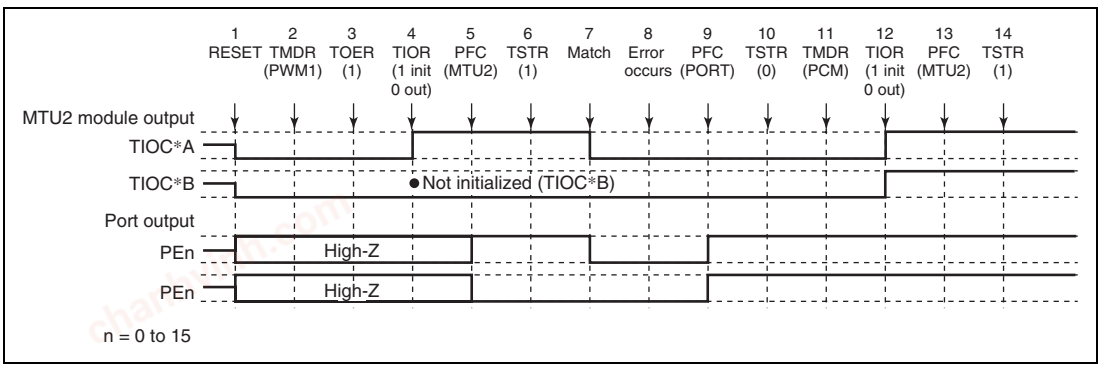


Figure 11.124 Error Occurrence in PWM Mode 1, Recovery in Phase Counting Mode

1 to 10 are the same as in figure 11.121.

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 11.125 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.

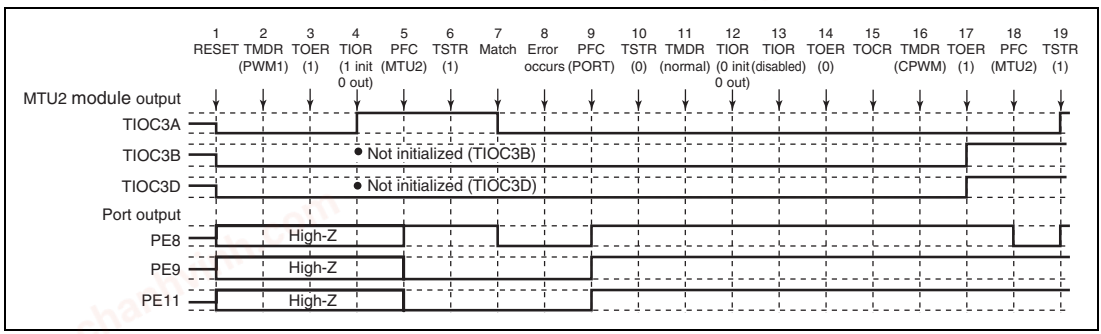


Figure 11.125 Error Occurrence in PWM Mode 1, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 11.121.

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 11.126 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.

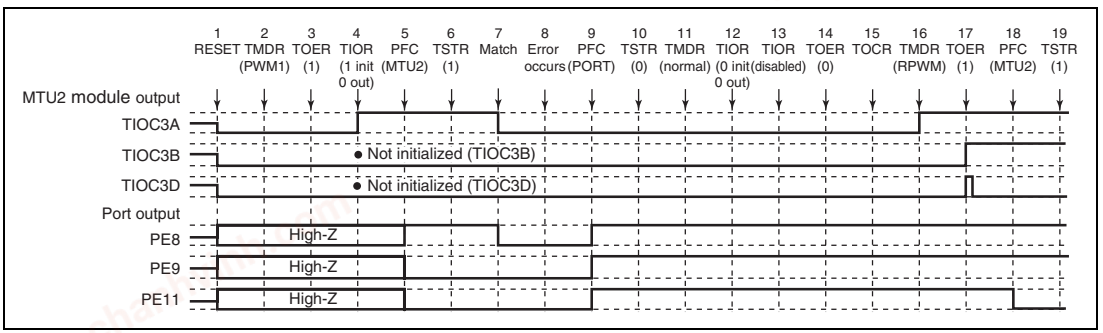


Figure 11.126 Error Occurrence in PWM Mode 1, Recovery in Reset-Synchronized PWM Mode

1 to 14 are the same as in figure 11.125.

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 11.127 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.

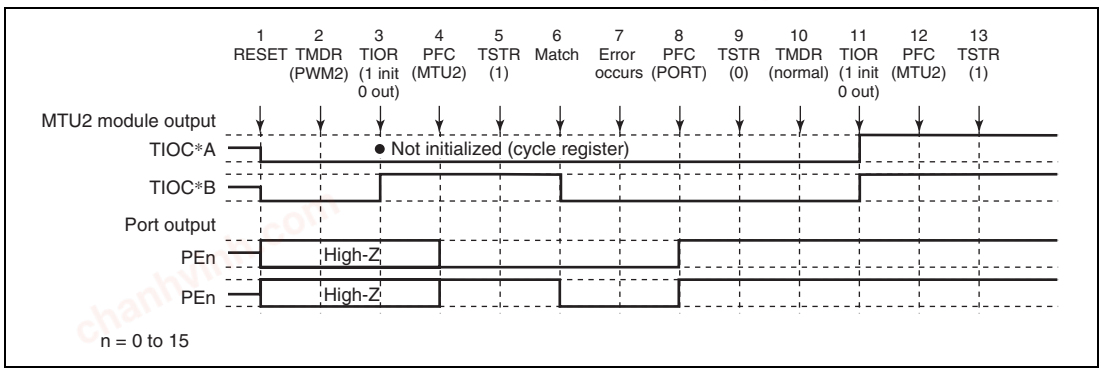


Figure 11.127 Error Occurrence in PWM Mode 2, Recovery in Normal Mode

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 11.128 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.

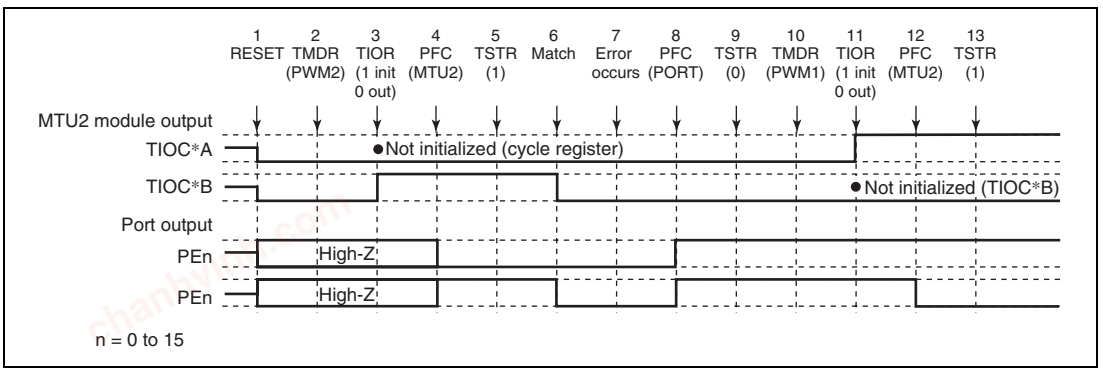


Figure 11.128 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 1

1 to 9 are the same as in figure 11.127.

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 11.129 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.

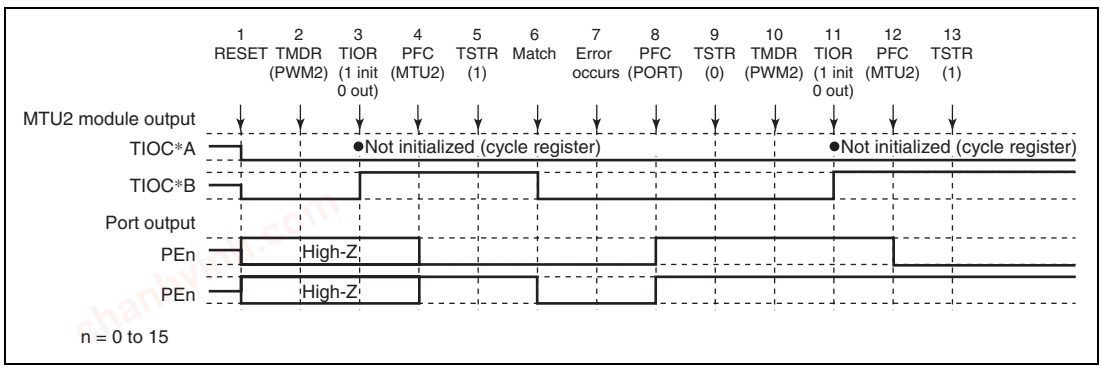


Figure 11.129 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 2

1 to 9 are the same as in figure 11.127.

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 11.130 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.

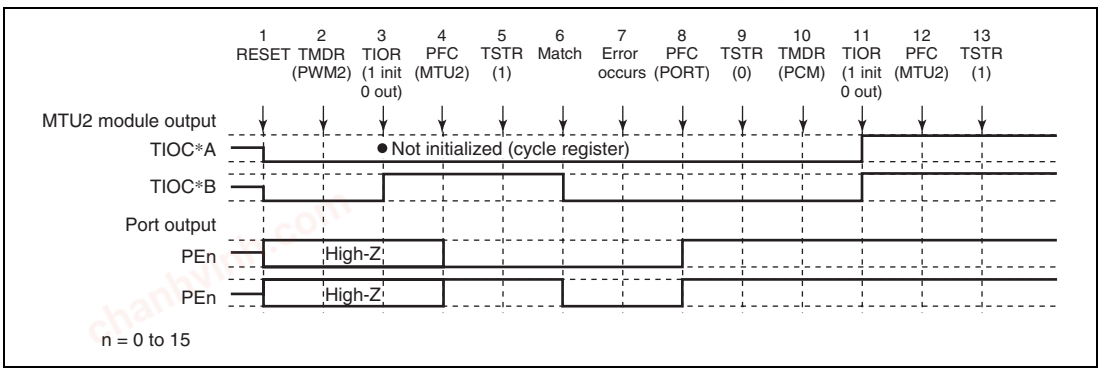


Figure 11.130 Error Occurrence in PWM Mode 2, Recovery in Phase Counting Mode

1 to 9 are the same as in figure 11.127.

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 11.131 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.

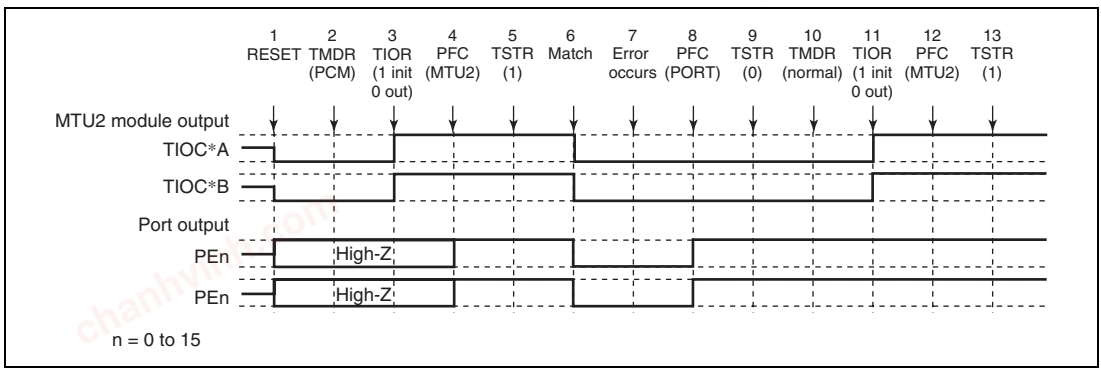


Figure 11.131 Error Occurrence in Phase Counting Mode, Recovery in Normal Mode

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 11.132 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.

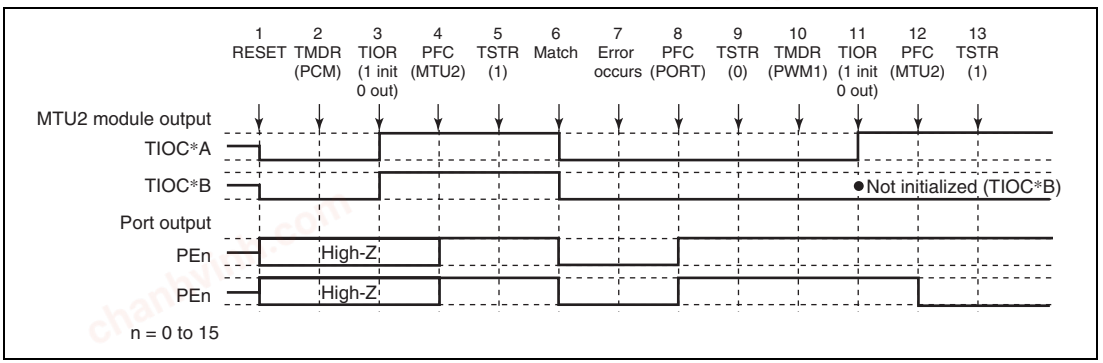


Figure 11.132 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 1

1 to 9 are the same as in figure 11.131.

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 11.133 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.

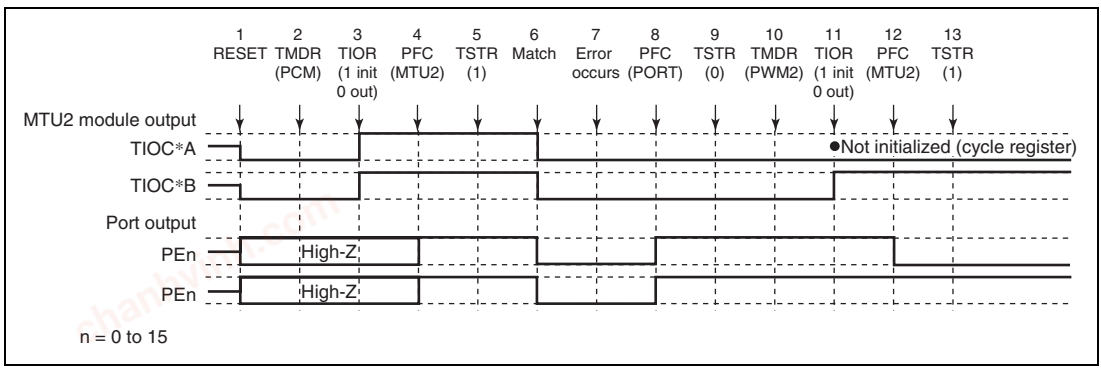


Figure 11.133 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 2

1 to 9 are the same as in figure 11.131.

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 11.134 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.

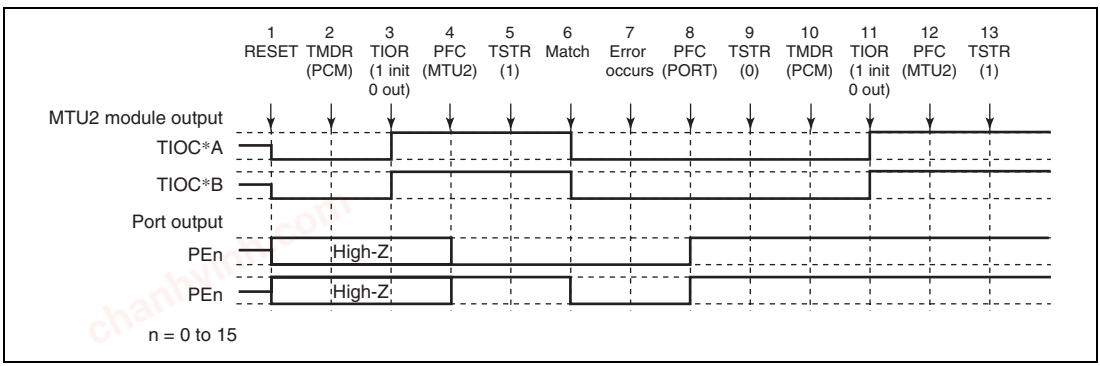


Figure 11.134 Error Occurrence in Phase Counting Mode, Recovery in Phase Counting Mode

1 to 9 are the same as in figure 11.131.

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 11.135 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.

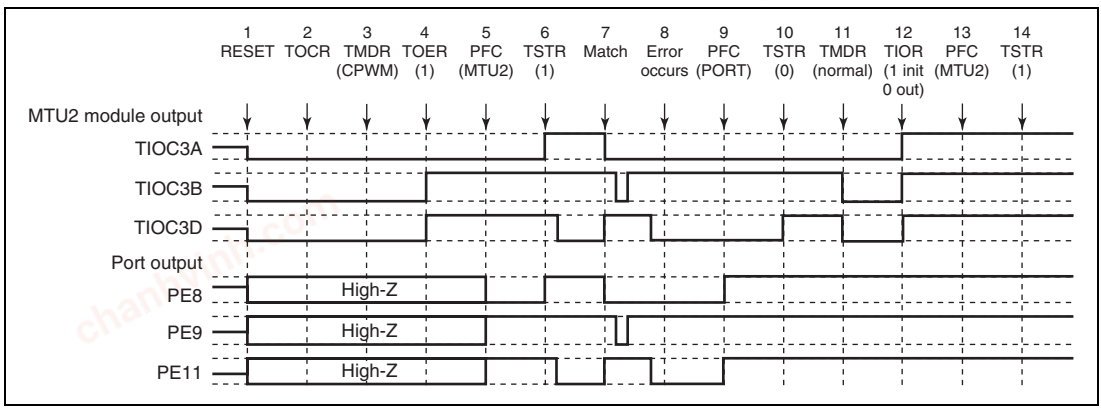


Figure 11.135 Error Occurrence in Complementary PWM Mode, Recovery in Normal Mode

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 11.136 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.

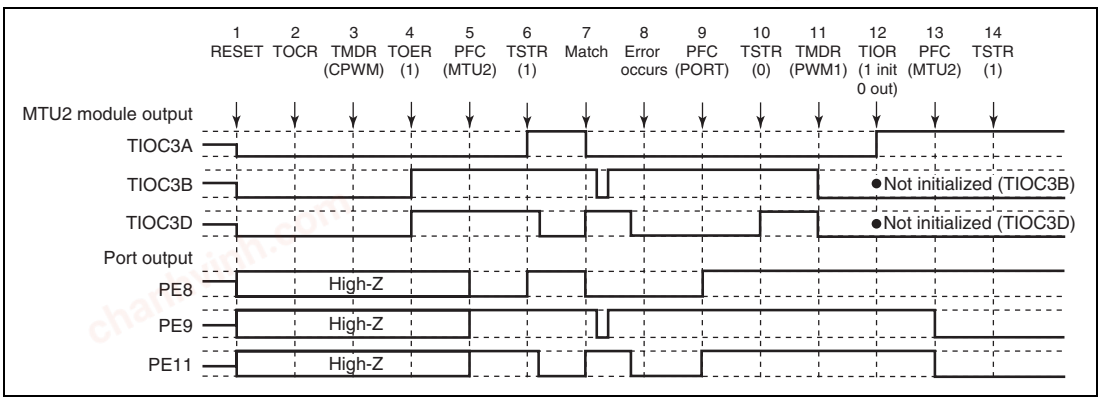


Figure 11.136 Error Occurrence in Complementary PWM Mode, Recovery in PWM Mode 1

1 to 10 are the same as in figure 11.135.

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 11.137 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).

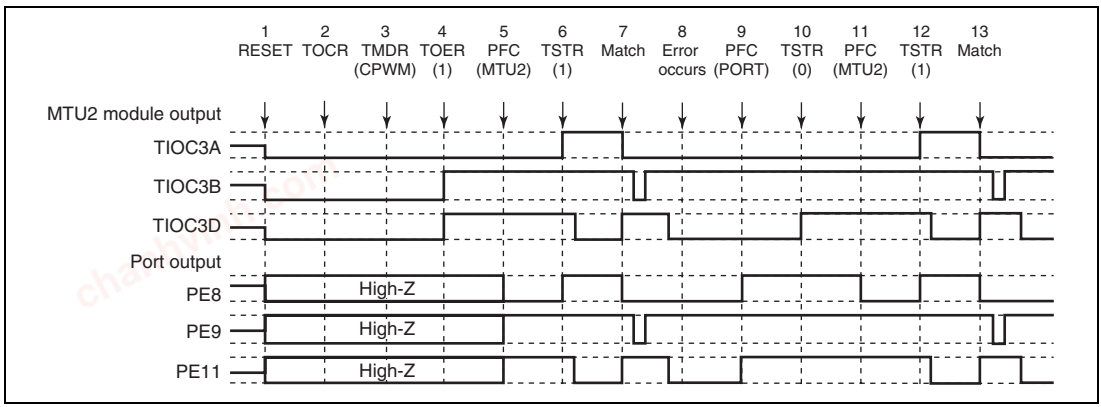


Figure 11.137 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 11.135.

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 11.138 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).

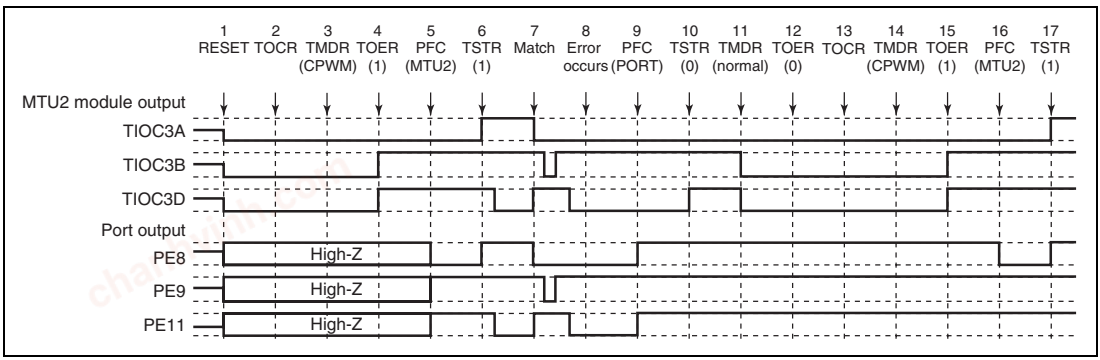


Figure 11.138 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 11.135.

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 11.139 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in reset-synchronized PWM mode.

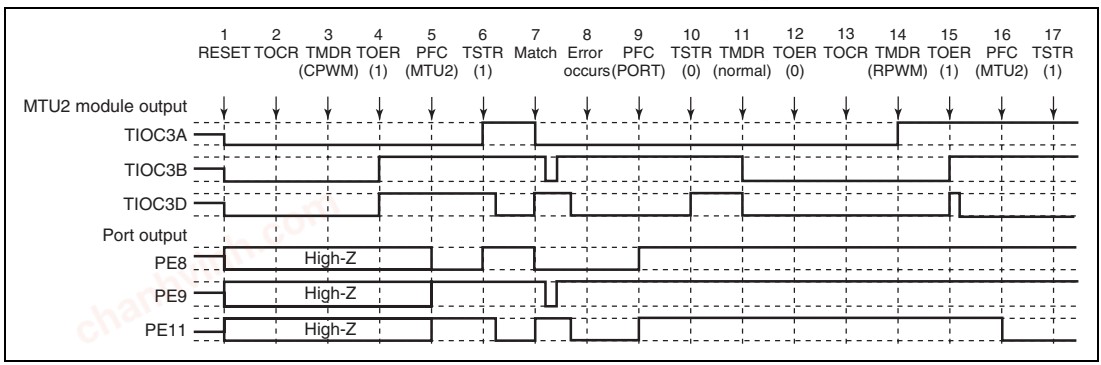


Figure 11.139 Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronized PWM Mode

1 to 10 are the same as in figure 11.135.

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 11.140 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.

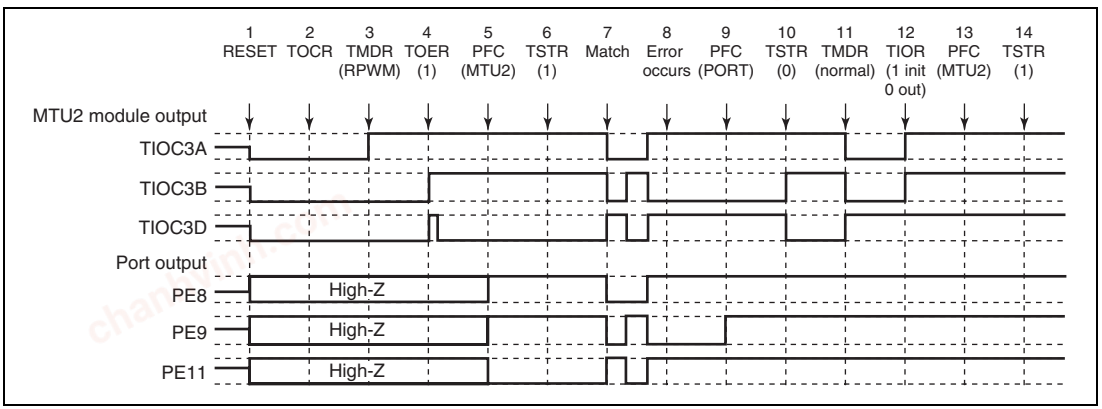


Figure 11.140 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Normal Mode

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 11.141 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.

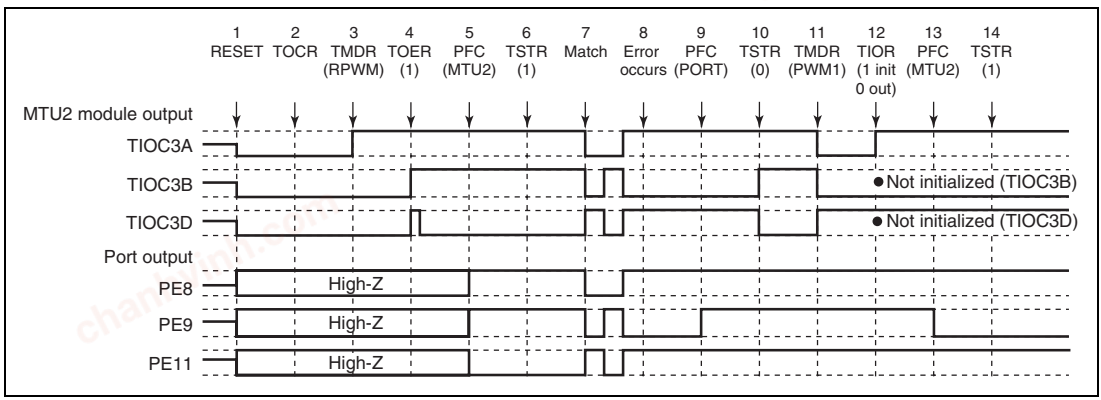


Figure 11.141 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in PWM Mode 1

1 to 10 are the same as in figure 11.140.

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 11.142 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.

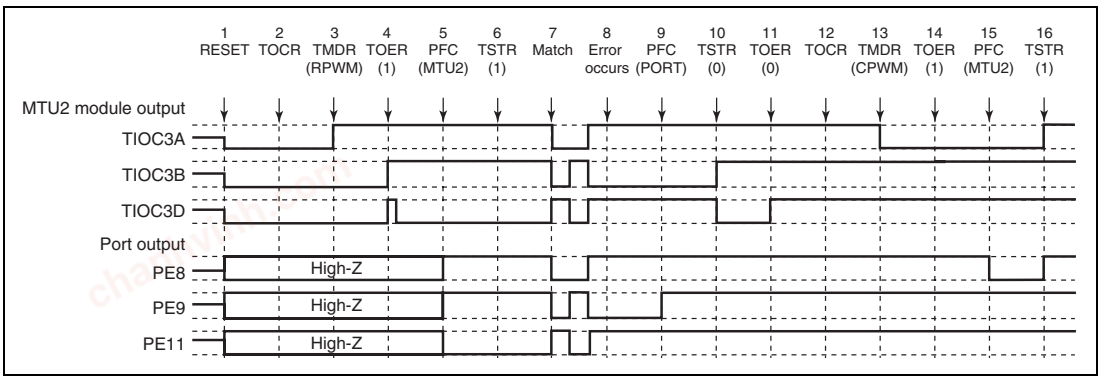


Figure 11.142 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 11.140.

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 11.143 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.

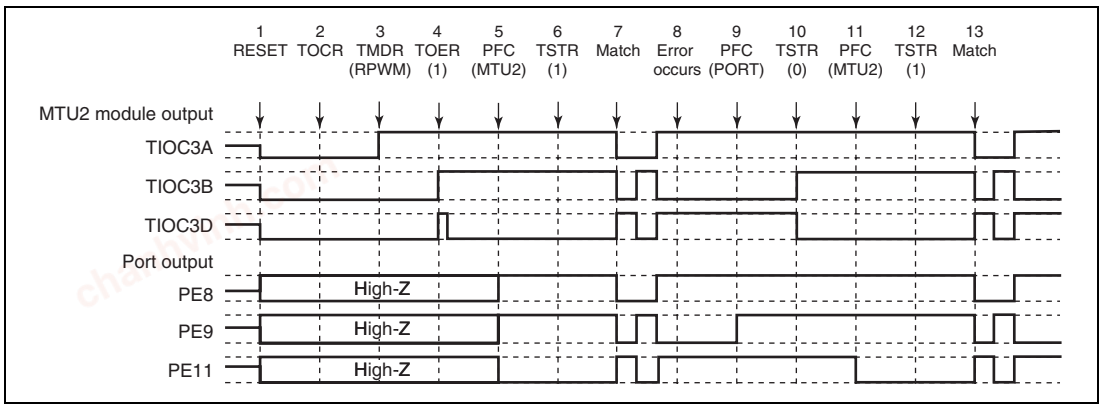


Figure 11.143 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Reset-Synchronized PWM Mode

1 to 10 are the same as in figure 11.140.

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

chanhvinh.com
chanhvinh.com
chanhvinh.com
chanhvinh.com

chanhvinh.com

Section 12 Compare Match Timer

This LSI has an on-chip compare match timer module consisting of two-channel 16-bit timers. This module has a 16-bit counter, and can generate interrupts at set intervals.

12.1 Features

- Independent selection of four counter input clocks at two channels
  Any of four internal clocks (Pϕ/8, Pϕ/32, Pϕ/128, and Pϕ/512) can be selected.
- Selection of DMA transfer request or interrupt request generation on compare match by direct memory access controller setting
- When not in use, this module can be stopped by halting its clock supply to reduce power consumption.

Figure 12.1 shows a block diagram.

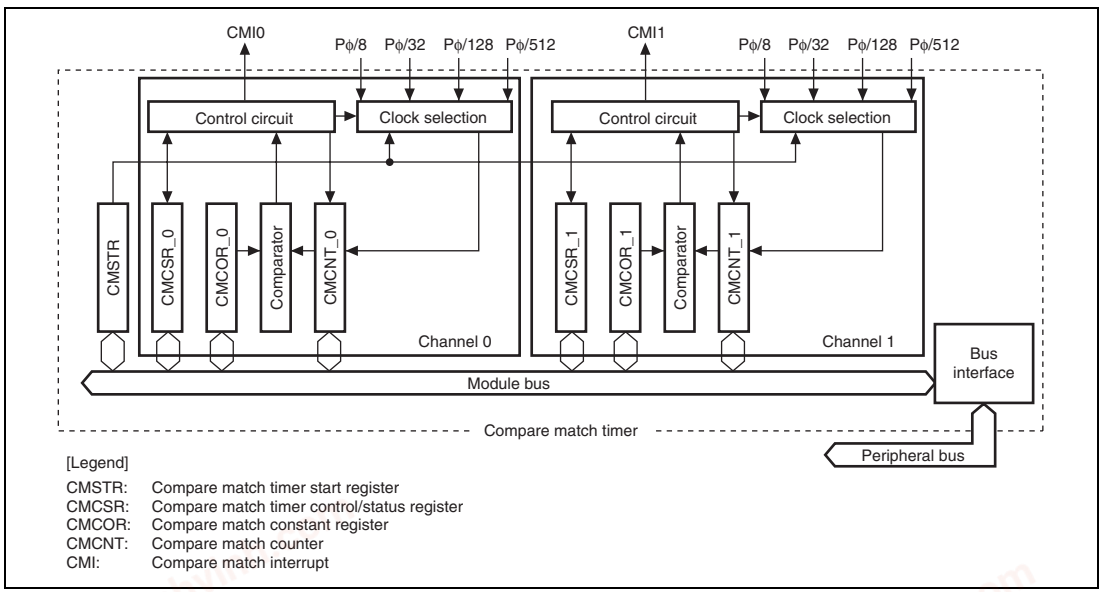


Figure 12.1 Block Diagram

chanhvinh.com

12.2 Register Descriptions

Table 12.1 shows the register configuration.

Table 12.1 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Common	Compare match timer start register	CMSTR	R/W	H'0000	H'FFFEC000	16
0	Compare match timer control/ status register_0	CMCSR_0	R/W	H'0000	H'FFFEC002	16
	Compare match counter_0	CMCNT_0	R/W	H'0000	H'FFFEC004	8, 16
	Compare match constant register_0	CMCOR_0	R/W	H'FFFF	H'FFFEC006	8, 16
1	Compare match timer control/ status register_1	CMCSR_1	R/W	H'0000	H'FFFEC008	16
	Compare match counter_1	CMCNT_1	R/W	H'0000	H'FFFEC00A	8, 16
	Compare match constant register_1	CMCOR_1	R/W	H'FFFF	H'FFFEC00C	8, 16

chanhvinh.com

12.2.1 Compare Match Timer Start Register (CMSTR)

CMSTR is a 16-bit register that selects whether compare match counter (CMCNT) operates or is stopped.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	STR1	STR0
Initial value:	0	0	0	0	0	0	0	0	0	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
15 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	STR1	0	R/W	Count Start 1 Specifies whether compare match counter_1 operates or is stopped. 0: Counting by CMCNT_1 is stopped 1: Counting by CMCNT_1 is started
0	STR0	0	R/W	Count Start 0 Specifies whether compare match counter_0 operates or is stopped. 0: Counting by CMCNT_0 is stopped 1: Counting by CMCNT_0 is started

chanhvinh.com

12.2.2 Compare Match Timer Control/Status Register (CMCSR)

CMCSR is a 16-bit register that indicates compare match generation, enables or disables interrupts, and selects the counter input clock.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	CMF	CMIE	-	-	-	-	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	R	R	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	CMF	0	R/(W)*	Compare Match Flag Indicates whether or not the values of CMCNT and CMCOR match. 0: CMCNT and CMCOR values do not match [Clearing condition] • When 0 is written to CMF after reading CMF = 1 1: CMCNT and CMCOR values match
6	CMIE	0	R/W	Compare Match Interrupt Enable Enables or disables compare match interrupt (CMI) generation when CMCNT and CMCOR values match (CMF = 1). 0: Compare match interrupt (CMI) disabled 1: Compare match interrupt (CMI) enabled
5 to 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	CKS[1:0]	00	R/W	Clock Select These bits select the clock to be input to CMCNT from four internal clocks obtained by dividing the peripheral clock (Pϕ). When the STR bit in CMSTR is set to 1, CMCNT starts counting on the clock selected with bits CKS[1:0]. 00: Pϕ/8 01: Pϕ/32 10: Pϕ/128 11: Pϕ/512

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

chanhvinh.com

12.2.3 Compare Match Counter (CMCNT)

CMCNT is a 16-bit register used as an up-counter. When the counter input clock is selected with bits CKS[1:0] in CMCSR, and the STR bit in CMSTR is set to 1, CMCNT starts counting using the selected clock. When the value in CMCNT and the value in compare match constant register (CMCOR) match, CMCNT is cleared to H'0000 and the CMF flag in CMCSR is set to 1.

CMCNT is initialized to H'0000 by clearing any channels of the counter start bit from 1 to 0 in the compare match timer start register (CMSTR).

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

12.2.4 Compare Match Constant Register (CMCOR)

CMCOR is a 16-bit register that sets the interval up to a compare match with CMCNT.

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

chanhvinh.com

12.3 Operation

12.3.1 Interval Count Operation

When an internal clock is selected with the CKS[1:0] bits in CMCSR and the STR bit in CMSTR is set to 1, CMCNT starts incrementing using the selected clock. When the values in CMCNT and CMCOR match, CMCNT is cleared to H'0000 and the CMF flag in CMCSR is set to 1. When the CMIE bit in CMCSR is set to 1 at this time, a compare match interrupt (CMI) is requested. CMCNT then starts counting up again from H'0000.

Figure 12.2 shows the operation of the compare match counter.

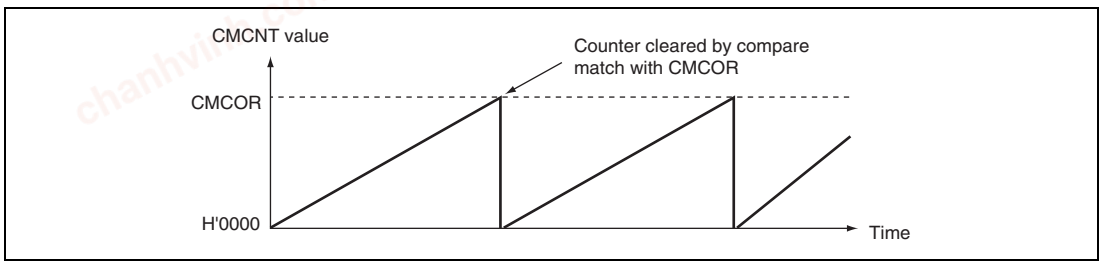


Figure 12.2 Counter Operation

12.3.2 CMCNT Count Timing

One of four clocks (Pϕ/8, Pϕ/32, Pϕ/128, and Pϕ/512) obtained by dividing the peripheral clock (Pϕ) can be selected with the CKS1 and CKS0 bits in CMCSR. Figure 12.3 shows the timing.

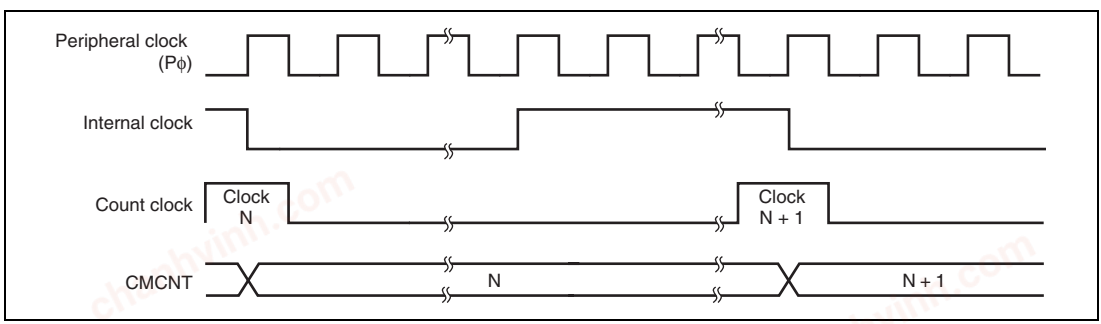


Figure 12.3 Count Timing

chanhvinh.com

12.4 Interrupts

12.4.1 Interrupt Sources and DMA Transfer Requests

This module has channels and each of them to which a different vector address is allocated has a compare match interrupt. When both the compare match flag (CMF) and the interrupt enable bit (CMIE) are set to 1, the corresponding interrupt request is output. When the interrupt is used to activate a CPU interrupt, the priority of channels can be changed by the interrupt controller settings. For details, see section 7, Interrupt Controller.

Clear the CMF bit to 0 by the user exception handling routine. If this operation is not carried out, another interrupt will be generated. By setting the interrupt controller, the direct memory access controller can be activated when a compare match interrupt is requested. 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. The CMF bit is automatically cleared to 0 when data is transferred by the direct memory access controller.

12.4.2 Timing of Compare Match Flag Setting

When CMCOR and CMCNT match, a compare match signal is generated at the last state in which the values match (the timing when the CMCNT value is updated to H'0000) and the CMF bit in CMCSR is set to 1. That is, after a match between CMCOR and CMCNT, the compare match signal is not generated until the next CMCNT counter clock input. Figure 12.4 shows the timing of CMF bit setting.

chanhvinh.com

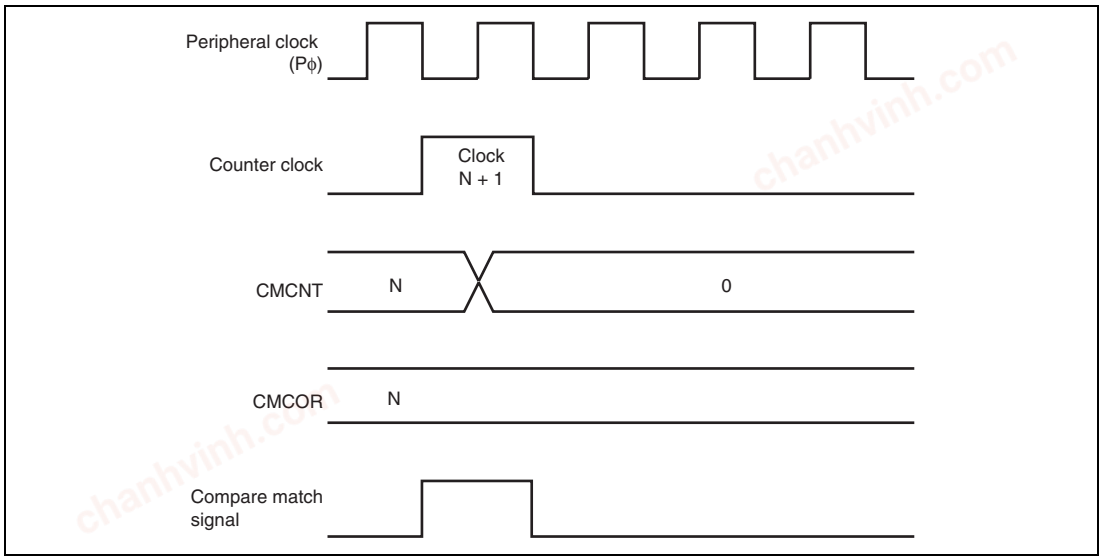


Figure 12.4 Timing of CMF Setting

12.4.3 Timing of Compare Match Flag Clearing

The CMF bit in CMCSR is cleared by first, reading as 1 then writing to 0. However, in the case of the direct memory access controller being activated, the CMF bit is automatically cleared to 0 when data is transferred by the direct memory access controller.

chanhvinh.com

12.5 Usage Notes

12.5.1 Conflict between Write and Compare-Match Processes of CMCNT

When the compare match signal is generated in the T2 cycle while writing to CMCNT, clearing CMCNT has priority over writing to it. In this case, CMCNT is not written to. Figure 12.5 shows the timing to clear the CMCNT counter.

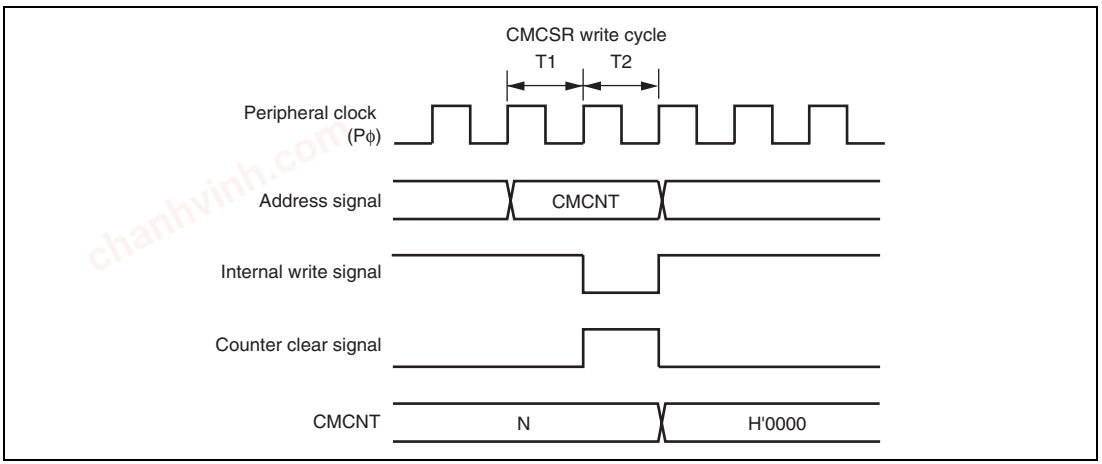


Figure 12.5 Conflict between Write and Compare Match Processes of CMCNT

chanhvinh.com

12.5.2 Conflict between Word-Write and Count-Up Processes of CMCNT

Even when the count-up occurs in the T2 cycle while writing to CMCNT in words, the writing has priority over the count-up. In this case, the count-up is not performed. Figure 12.6 shows the timing to write to CMCNT in words.

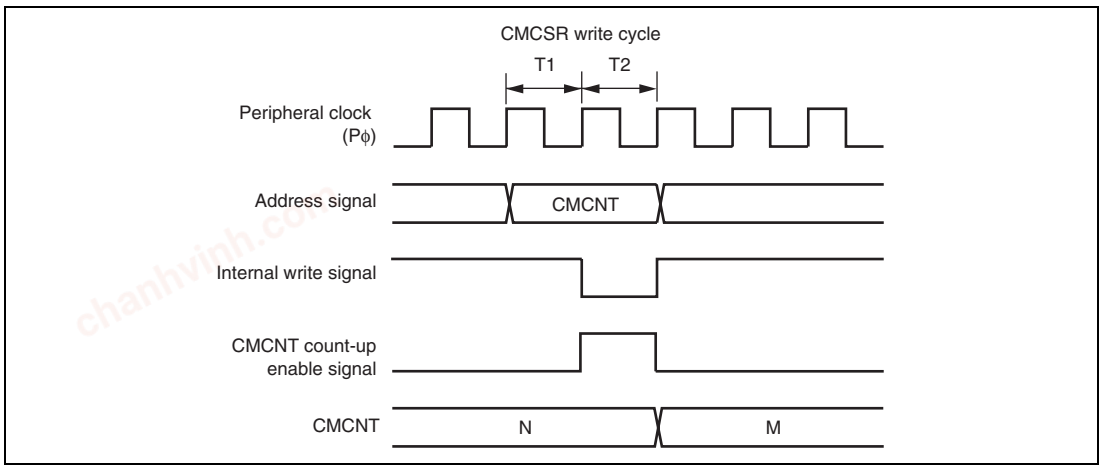


Figure 12.6 Conflict between Word-Write and Count-Up Processes of CMCNT

chanhvinh.com

12.5.3 Conflict between Byte-Write and Count-Up Processes of CMCNT

Even when the count-up occurs in the T2 cycle while writing to CMCNT in bytes, the writing has priority over the count-up. In this case, the count-up is not performed. The byte data on the other side, which is not written to, is also not counted and the previous contents are retained.

Figure 12.7 shows the timing when the count-up occurs in the T2 cycle while writing to CMCNTH in bytes.

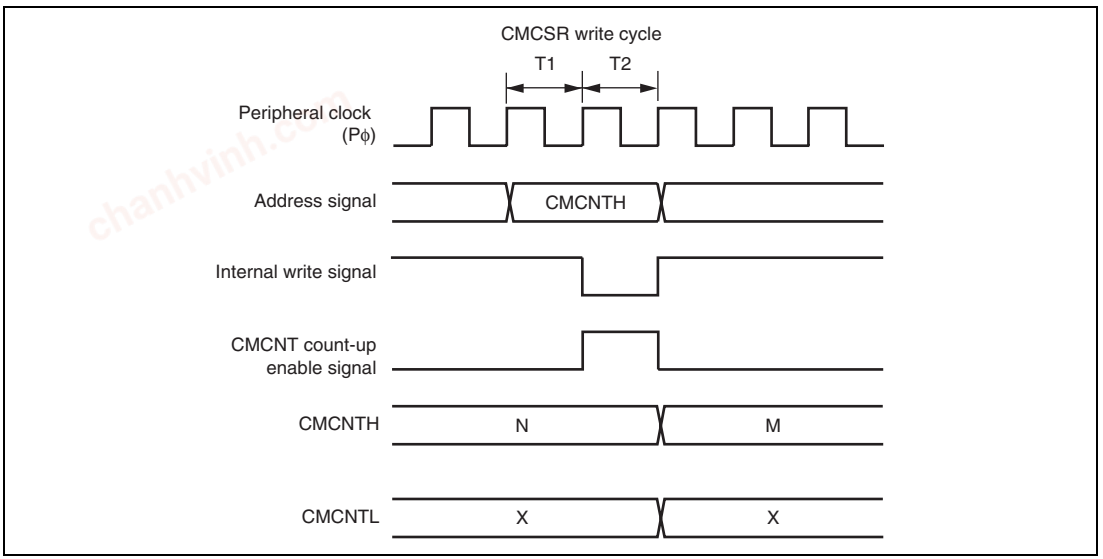


Figure 12.7 Conflict between Byte-Write and Count-Up Processes of CMCNT

12.5.4 Compare Match between CMCNT and CMCOR

Do not set the same value in CMCNT and CMCOR while CMCNT is not counting.

chanhvinh.com

Section 13 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.

13.1 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. Either the power-on reset or manual reset signal can be selected as the internal reset type.
- Interrupt generation in interval timer mode
  An interval timer interrupt is generated when the counter overflows.
- Choice of eight counter input clocks
  Eight clocks (Pϕ × 1 to Pϕ × 1/16384) that are obtained by dividing the peripheral clock can be selected.

chanhvinh.com

Figure 13.1 shows a block diagram.

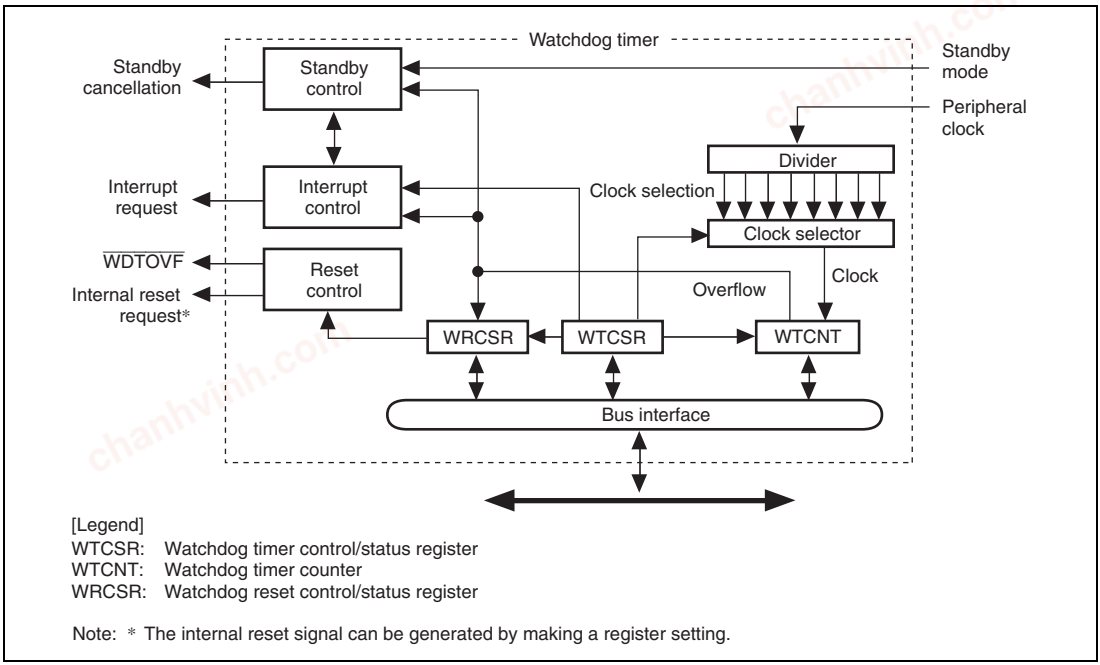


Figure 13.1 Block Diagram

13.2 Input/Output Pin

Table 13.1 shows the pin configuration.

Table 13.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Watchdog timer overflow	$\overline{\text{WDTOVF}}$	Output	Outputs the counter overflow signal in watchdog timer mode

chanhvinh.com

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
Watchdog timer counter	WTCNT	R/W	H'00	H'FFFE0002	16*
Watchdog timer control/status register	WTCSR	R/W	H'18	H'FFFE0000	16*
Watchdog reset control/status register	WRCSR	R/W	H'1F	H'FFFE0004	16*

Note: * For the access size, see section 13.3.4, Notes on Register Access.

13.3.1 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 word access to write to WTCNT, writing H'5A in the upper byte. Use byte access to read from WTCNT.

Note: The method for writing to WTCNT differs from that for other registers to prevent erroneous writes. See section 13.3.4, 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

chanhvinh.com

13.3.2 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 word access to write to WTCSR, writing H'A5 in the upper byte. Use byte access to read from WTCSR.

Note: The method for writing to WTCSR differs from that for other registers to prevent erroneous writes. See section 13.3.4, Notes on Register Access, for details.

Bit:	7	6	5	4	3	2	1	0
	IOVF	WT/$\overline{IT}$	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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
6	WT/$\overline{IT}$	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 $\overline{WDTOVF}$ signal is output externally. If this bit is modified when this module is running, the up-count may not be performed correctly.
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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 (Pϕ). The overflow period that is shown inside the parenthesis in the table is the value when the peripheral clock (Pϕ) is 24 MHz.

Bits 2 to 0	Clock Ratio	Overflow Cycle
000:	1 × Pϕ	10.6 μs
001:	1/64 × Pϕ	680 μs
010:	1/128 × Pϕ	1.4 ms
011:	1/256 × Pϕ	2.7 ms
100:	1/512 × Pϕ	5.4 ms
101:	1/1024 × Pϕ	10.9 ms
110:	1/4096 × Pϕ	44 ms
111:	1/16384 × Pϕ	174 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

13.3.3 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 13.3.4, Notes on Register Access, for details.

Bit:	7	6	5	4	3	2	1	0
	WOVF	RSTE	RSTS	-	-	-	-	-
Initial value:	0	0	0	1	1	1	1	1
R/W:	R/(W)	R/W	R/W	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	RSTS	0	R/W	Reset Select Selects the type of reset when the WTCNT overflows in watchdog timer mode. In interval timer mode, this setting is ignored. 0: Power-on reset 1: Manual reset
4 to 0	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.

chanhvinh.com

13.3.4 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 word transfer instruction. They cannot be written by a byte or longword 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 13.2. 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.

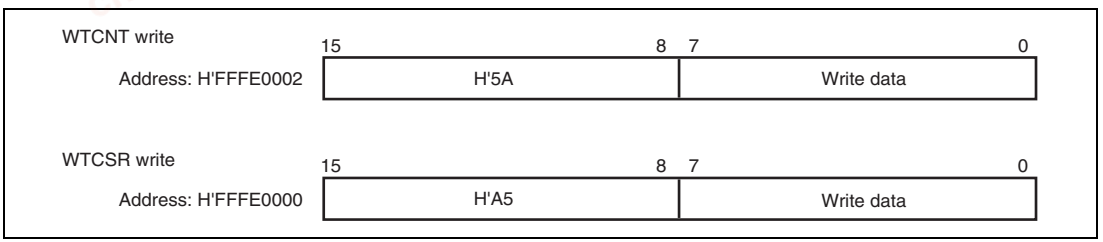


Figure 13.2 Writing to WTCNT and WTCSR

chanhvinh.com

(2) Writing to WRCSR

WRCSR must be written by a word access to address H'FFFE0004. It cannot be written by byte transfer or longword transfer instructions.

Procedures for writing 0 to WOVF (bit 7) and for writing to RSTE (bit 6) and RSTS (bit 5) are different, as shown in figure 13.3.

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 and RSTS bits are not affected. To write to the RSTE and RSTS bits, the upper byte must be H'5A and the lower byte must be the write data. The values of bits 6 and 5 of the lower byte are transferred to the RSTE and RSTS bits, respectively. The WOVF bit is not affected.

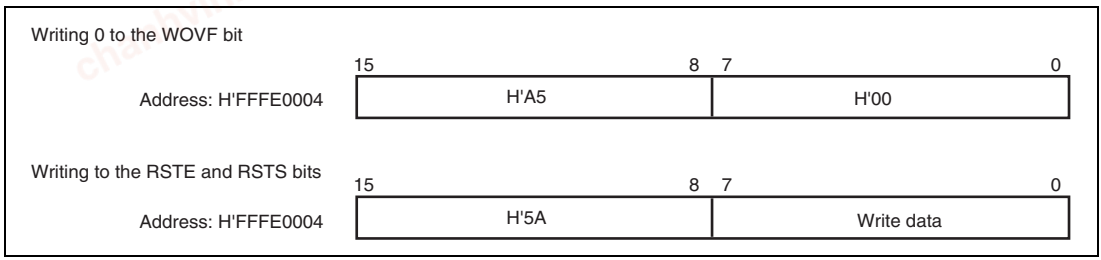


Figure 13.3 Writing to WRCSR

(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'FFFE0000, WTCNT to address H'FFFE0002, and WRCSR to address H'FFFE0004. Byte transfer instructions must be used for reading from these registers.

chanhvinh.com

13.4 Usage

13.4.1 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 $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ 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 longer than the clock oscillation settling time.
3. After setting the STBY and DEEP bits of the standby control register 1 (STBCR1: see section 33, Power-Down Modes) to 1 and 0 respectively, the execution of a SLEEP 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.

13.4.2 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, the reset type if it is generated in the RSTS 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 13.4). The $\overline{\text{WDTOVF}}$ signal can be used to reset the system. The $\overline{\text{WDTOVF}}$ signal is output for 64 × Pϕ 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. Either power-on reset or manual reset can be selected for this interrupt by the RSTS bit in WRCSR. The internal reset signal is output for 128 × Pϕ 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

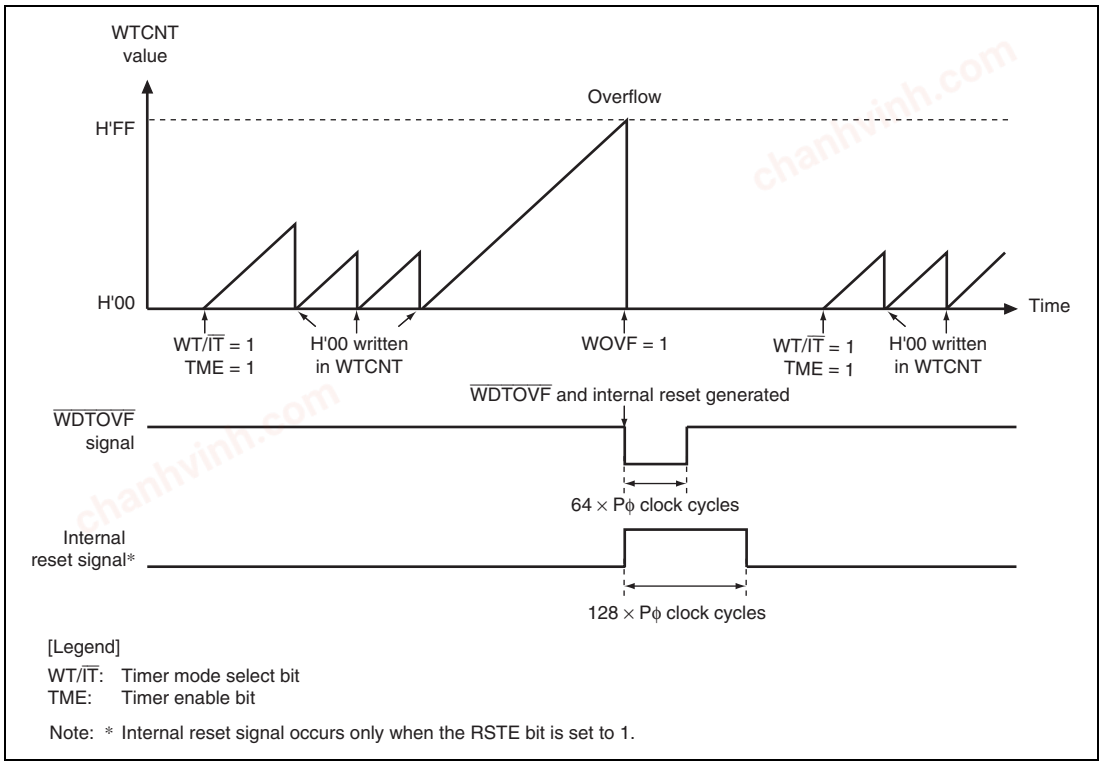


Figure 13.4 Operation in Watchdog Timer Mode

chanhvinh.com

13.4.3 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.

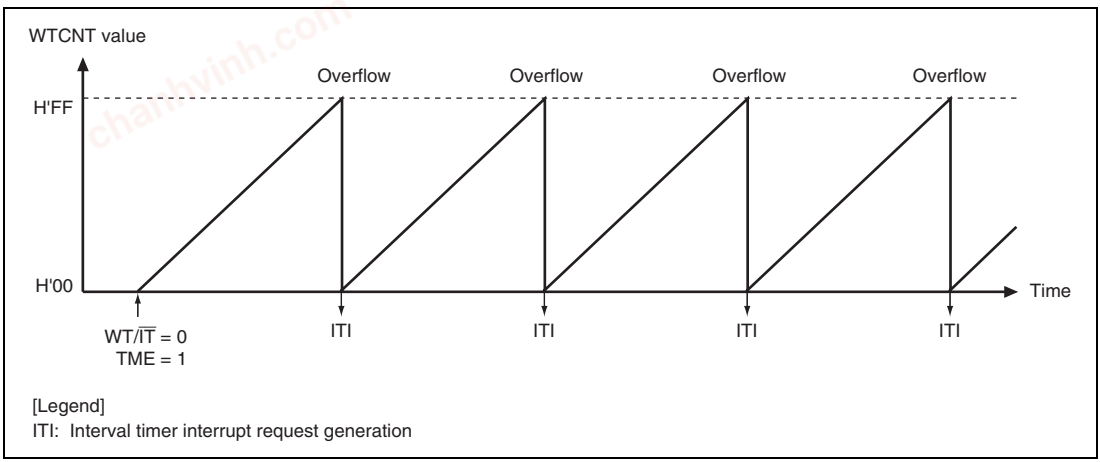


Figure 13.5 Operation in Interval Timer Mode

chanhvinh.com

13.5 Usage Notes

Pay attention to the following points when using this module in either the interval timer or watchdog timer mode.

13.5.1 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, Pϕ, 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.

13.5.2 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.

13.5.3 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.

chanhvinh.com

13.5.4 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 13.6.

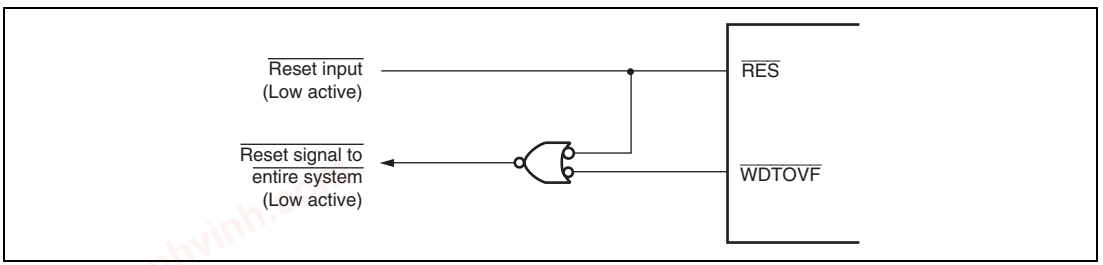


Figure 13.6 Example of System Reset Circuit Using $\overline{\text{WDTOVF}}$ Signal

13.5.5 Manual Reset in Watchdog Timer Mode

When a manual reset occurs in watchdog timer mode, the bus cycle is continued. If a manual reset occurs while the bus is released or during burst transfer by the direct memory access controller, manual reset exception handling will be pended until the CPU acquires the bus mastership.

chanhvinh.com

Section 14 Realtime Clock

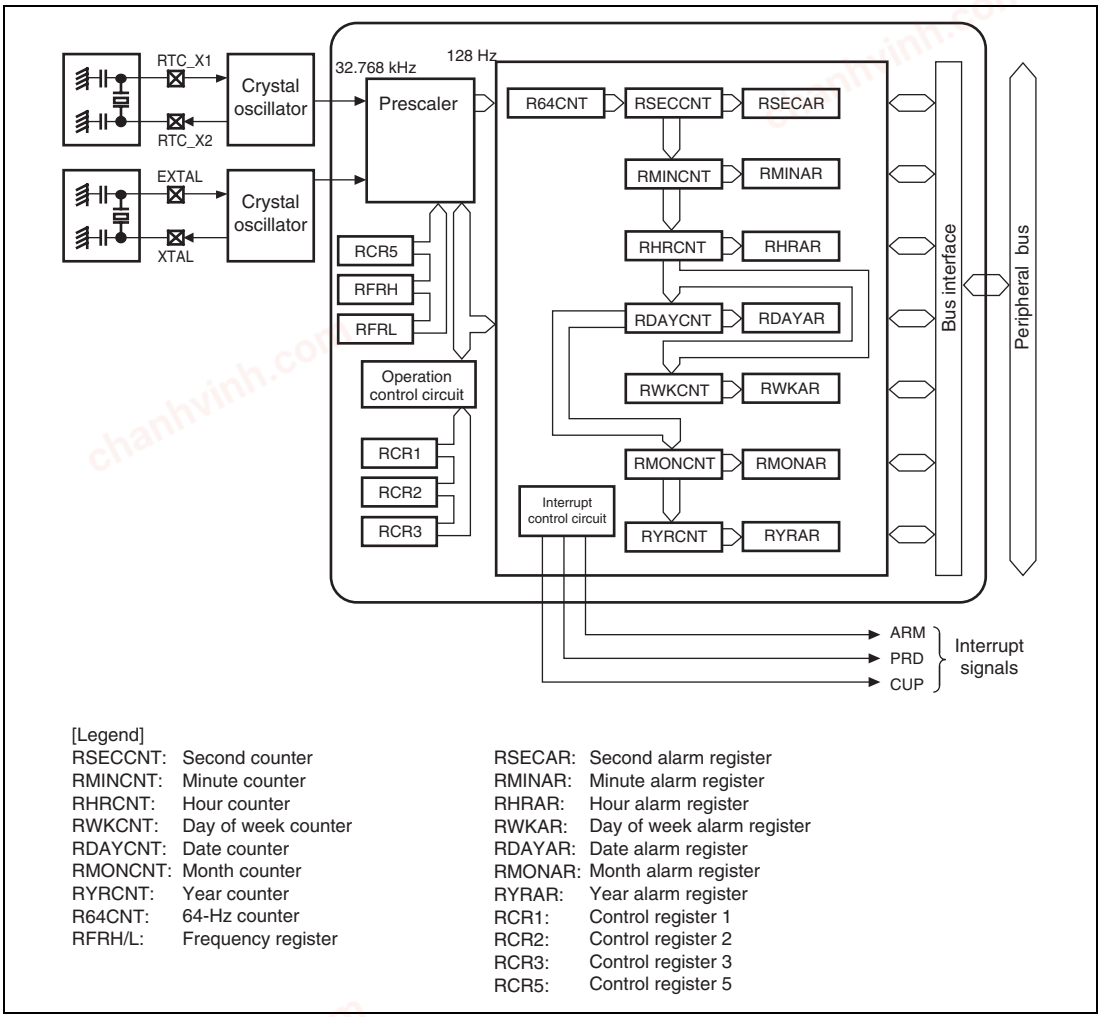
This LSI has a realtime clock and a 32.768-kHz crystal oscillator.

14.1 Features

- Clock and calendar functions (BCD format): Seconds, minutes, hours, date, day of the week, 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: Frame comparison of seconds, minutes, hours, date, day of the week, month, and year can be used as conditions for the alarm interrupt
- Periodic interrupts: the interrupt cycle may be 1/256 second, 1/64 second, 1/16 second, 1/4 second, 1/2 second, 1 second, or 2 seconds
- Carry interrupt: a carry interrupt indicates when a carry occurs during a counter read
- Automatic leap year adjustment
- Either the external clock signal dedicated for the clock function or the internal signal 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 14.1 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:	Date counter	RDAYAR:	Date 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 14.1 Block Diagram

chanhvinh.com

14.2 Input/Output Pin

Table 14.1 shows the pin configuration.

Table 14.1 Pin Configuration

Pin Name	Symbol	I/O	Description
Realtime clock resonator crystal pin/ external clock	RTC_X1	Input	Connects 32.768-kHz crystal resonator for this module, and enables to input the external clock to the RTC_X1 pin.
	RTC_X2	Output	
Internal clock resonator crystal/ external clock	EXTAL	Input	Connects crystal resonator used for internal operation. For details, see section 5, Clock Pulse Generator.
	XTAL	Output	

14.3 Register Descriptions

Table 14.2 shows the register configuration.

Table 14.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
64-Hz counter	R64CNT	R	H'xx	H'FFFE6000	8
Second counter	RSECCNT	R/W	H'xx	H'FFFE6002	8
Minute counter	RMINCNT	R/W	H'xx	H'FFFE6004	8
Hour counter	RHRCNT	R/W	H'xx	H'FFFE6006	8
Day of week counter	RWKCNT	R/W	H'xx	H'FFFE6008	8
Date counter	RDAYCNT	R/W	H'xx	H'FFFE600A	8
Month counter	RMONCNT	R/W	H'xx	H'FFFE600C	8
Year counter	RYRCNT	R/W	H'xxxx	H'FFFE600E	16
Second alarm register	RSECAR	R/W	H'xx	H'FFFE6010	8
Minute alarm register	RMINAR	R/W	H'xx	H'FFFE6012	8
Hour alarm register	RHRAR	R/W	H'xx	H'FFFE6014	8
Day of week alarm register	RWKAR	R/W	H'xx	H'FFFE6016	8

chanhvinh.com

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Date alarm register	RDAYAR	R/W	H'xx	H'FFFE6018	8
Month alarm register	RMONAR	R/W	H'xx	H'FFFE601A	8
Year alarm register	RYRAR	R/W	H'xxxx	H'FFFE6020	16
Control register 1	RCR1	R/W	H'xx	H'FFFE601C	8
Control register 2	RCR2	R/W	H'09	H'FFFE601E	8
Control register 3	RCR3	R/W	H'x0	H'FFFE6024	8
Control register 5	RCR5	R/W	H'xx	H'FFFE6026	8
Frequency register H	RFRH	R/W	H'xxxx	H'FFFE602A	16
Frequency register L	RFRL	R/W	H'xxxx	H'FFFE602C	16

14.3.1 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 so that the carrying and reading 64 Hz counter are performed at the same time is indicated. 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.

After the RESET bit or ADJ bit in the control register 2 (RCR2) is set to 1, the divider circuit is initialized and R64CNT is initialized.

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

chanhvinh.com

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	

14.3.2 Second Counter (RSECCNT)

RSECCNT is used for setting/counting in the BCD-coded second section. 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 operation errors 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 seconds			1 second			
Initial value:	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
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 to 4	10 seconds	Undefined	R/W	Counting Ten's Position of Seconds Counts on 0 to 5 for 60-seconds counting.
3 to 0	1 second	Undefined	R/W	Counting One's Position of Seconds Counts on 0 to 9 once per second. When a carry is generated, 1 is added to the ten's position.

chanhvinh.com

14.3.3 Minute Counter (RMINCNT)

RMINCNT is used for setting/counting in the BCD-coded minute section. 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 operation errors 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 minutes			1 minute			
Initial value:	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
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 to 4	10 minutes	Undefined	R/W	Counting Ten's Position of Minutes Counts on 0 to 5 for 60-minutes counting.
3 to 0	1 minute	Undefined	R/W	Counting One's Position of Minutes Counts on 0 to 9 once per second. When a carry is generated, 1 is added to the ten's position.

chanhvinh.com

14.3.4 Hour Counter (RHRCNT)

RHRCNT is used for setting/counting in the BCD-coded hour section. 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 operation errors 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 on 0 to 2 for ten's position of hours.
3 to 0	1 hour	Undefined	R/W	Counting One's Position of Hours Counts on 0 to 9 once per hour. When a carry is generated, 1 is added to the ten's position.

chanhvinh.com

14.3.5 Day of Week Counter (RWKCNT)

RWKCNT is used for setting/counting day of week section. The count operation is performed by a carry for each day of the date counter.

The assignable range is from 0 through 6 (practically in BCD), otherwise operation errors 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

14.3.6 Date Counter (RDAYCNT)

RDAYCNT is used for setting/counting in the BCD-coded date section. 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 operation errors occur. Carry out write processing after stopping the count operation through the setting of the START bit in RCR2.

The range of date changes with each month and in leap years. Confirm the correct setting. Leap years are recognized by dividing the year counter (RYRCNT) values by 400, 100, and 4 and obtaining a fractional result of 0.

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 Dates
3 to 0	1 day	Undefined	R/W	Counting One's Position of Dates Counts on 0 to 9 once per date. When a carry is generated, 1 is added to the ten's position.

chanhvinh.com

14.3.7 Month Counter (RMONCNT)

RMONCNT is used for setting/counting in the BCD-coded month section. 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 operation errors 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

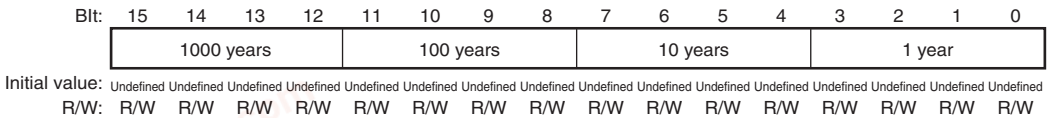
Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	10 months	Undefined	R/W	Counting Ten's Position of Months
3 to 0	1 month	Undefined	R/W	Counting One's Position of Months Counts on 0 to 9 once per month. When a carry is generated, 1 is added to the ten's position.

chanhvinh.com

14.3.8 Year Counter (RYRCNT)

RYRCNT is used for setting/counting in the BCD-coded year section. 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 operation errors occur. Carry out write processing after stopping the count operation through the setting of the START bit in RCR2.



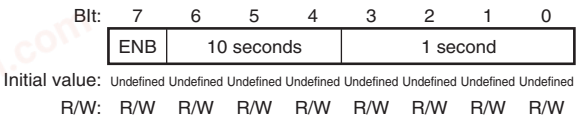
Bit	Bit Name	Initial Value	R/W	Description
15 to 12	1000 years	Undefined	R/W	Counting Thousand's Position of Years
11 to 8	100 years	Undefined	R/W	Counting Hundred's Position of Years
7 to 4	10 years	Undefined	R/W	Counting Ten's Position of Years
3 to 0	1 year	Undefined	R/W	Counting One's Position of Years

chanhvinh.com

14.3.9 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 operation errors 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

14.3.10 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 operation errors 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

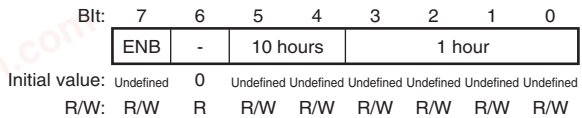
Bit	Bit Name	Initial Value	R/W	Description
7	ENB	Undefined	R/W	When this bit is set to 1, a comparison with the RMINCNT value is performed.
6 to 4	10 minutes	Undefined	R/W	Ten's position of minutes setting value
3 to 0	1 minute	Undefined	R/W	One's position of minutes setting value

chanhvinh.com

14.3.11 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 operation errors 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

14.3.12 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 operation errors 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)

chanhvinh.com

14.3.13 Date Alarm Register (RDAYAR)

RDAYAR is an alarm register corresponding to the BCD-coded date 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 operation errors 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 dates setting value
3 to 0	1 day	Undefined	R/W	One's position of dates setting value

chanhvinh.com

14.3.14 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 operation errors occur.

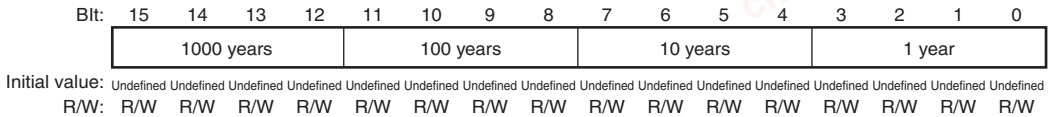
Bit:	7	6	5	4	3	2	1	0
	ENB	-	-	10 months	1 month			
Initial value:	Undefined	0	0	Undefined	Undefined	Undefined	Undefined	Undefined
R/W:	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	Undefined	R/W	When this bit is set to 1, a comparison with the 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

chanhvinh.com

14.3.15 Year Alarm Register (RYRAR)

RYRAR is an alarm register corresponding to the year counter RYRCNT. The assignable range is from 0000 through 9999 (practically in BCD), otherwise operation errors 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

14.3.16 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 count-up to 64-Hz occurs at the second counter carry or 64-Hz counter read. A count register value read at this time cannot be guaranteed; another read is required. 0: No carry of 64-Hz counter by second counter or 64-Hz counter [Clearing condition] When 0 is written to CF 1: Carry of 64-Hz counter by second counter or 64 Hz counter [Setting condition] When the second counter or 64-Hz counter is read during a carry occurrence by 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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
4	CIE	0	R/W	Carry Interrupt Enable Flag When the carry flag (CF) is set to 1, the CIE bit enables interrupts. 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 allows interrupts. 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 an alarm register (ENB bit in RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, or RYRAR is set to 1), and counter match. 0: Alarm register and counter not match [Clearing condition] When 0 is written to AF. 1: Alarm register and counter match* [Setting condition] When alarm register (only a register with ENB bit set to 1) and counter match Note: * Writing 1 holds previous value.

chanhvinh.com

14.3.17 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. Bits other than the RTCEN and START bits are initialized by a manual reset.

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 interrupt generation 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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	PES[2:0]	000	R/W	Interrupt Enable Flags These bits specify 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 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.
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 to 00 seconds and 30 seconds or more to 1 minute. The divider circuit (prescaler and R64CNT) will be simultaneously reset. This bit always reads 0. 0: Runs normally. 1: 30-second adjustment.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 1 to this bit. This bit is always read as 0. 0: Runs normally. 1: Divider circuit is reset.
0	START	1	R/W	Start Halts and restarts the counter (clock). 0: Second/minute/hour/day/week/month/year counter halts. 1: Second/minute/hour/day/week/month/year counter runs normally.

14.3.18 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.

chanhvinh.com

14.3.19 Control Register 5 (RCR5)

When the RCKSEL bit is set to 0, the RTC_X1 clock pulses are counted; and when the RCKSEL bit is set to 1, the EXTAL clock pulses are counted to implement the clock function.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	RCKSEL
Initial value:	0	0	0	0	0	0	0	Undefined
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	RCKSEL	Undefined	R/W	Operation clock select Operation clock can be selected from either RTC_X1 or EXTAL. The setting of this bit should not be switched during operation. 0: Selects RTC_X1. 1: Selects EXTAL.

chanhvinh.com

14.3.20 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.

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 operating clock that the EXTAL clock frequency is dividable by 64-Hz and not dividable by 128-Hz. 0: EXTAL clock frequency is dividable by 128-Hz. 1: EXTAL clock frequency is dividable by 64-Hz and not dividable by 128-Hz. Note: For 1-Mbyte version, this bit is reserved and always read as 0. The write value should always be 0.
30 to 19	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
18, 17	RFC[18:17]	Undefined	R/W	Frequency comparison value Sets the comparison value to generate operation clock from the EXTAL clock frequency. Note: For 1-Mbyte version, these bits are reserved and always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
16 to 0	RFC[16:0]	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".

- 1-Mbyte version
  - EXTAL clock frequency is dividable by 128-Hz
    RFC[16:0] = (EXTAL clock frequency) / 128 – 2
    Note: EXTAL clock frequency must be less than 16.778 MHz in the 1-Mbyte version.
- 640-Kbyte version
  - EXTAL clock frequency is dividable by 128-Hz
    RFC[18:0] = (EXTAL clock frequency) / 128 – 2
    Clear the SEL64 bit to 0 in this case.
  - EXTAL clock frequency is dividable by 64-Hz and not dividable by 128-Hz
    RFC[18:0] = (EXTAL clock frequency) / 64 – 2
    Set the SEL64 bit to 1 in this case.

(2) Setting Example

Table 14.3 Setting Example

EXTAL Clock Frequency	1-Mbyte Version	640-Kbyte Version	
	RFC Setting Value	SEL64 Setting Value	RFC Setting Value
10 MHz	H'1312B	0	H'1312B
12 MHz	H'16E34	0	H'16E34
15 MHz	No setting possible	1	H'39385
16 MHz	H'1E846	0	H'1E846

chanhvinh.com

14.4 Operation

Usage of this module is shown below.

14.4.1 Initial Settings of Registers after Power-On

All the registers should be set after the power is turned on.

14.4.2 Setting Time

Figure 14.2 shows how to set the time when the clock is stopped.

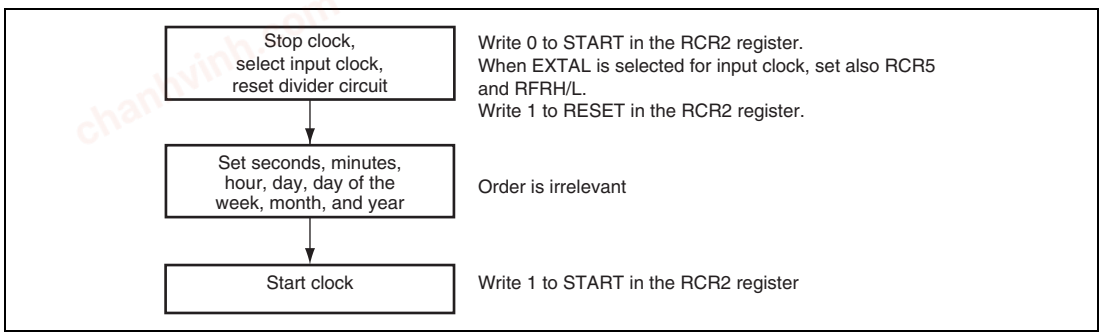


Figure 14.2 Setting Time

chanhvinh.com

14.4.3 Reading Time

Figure 14.3 shows how to read the time.

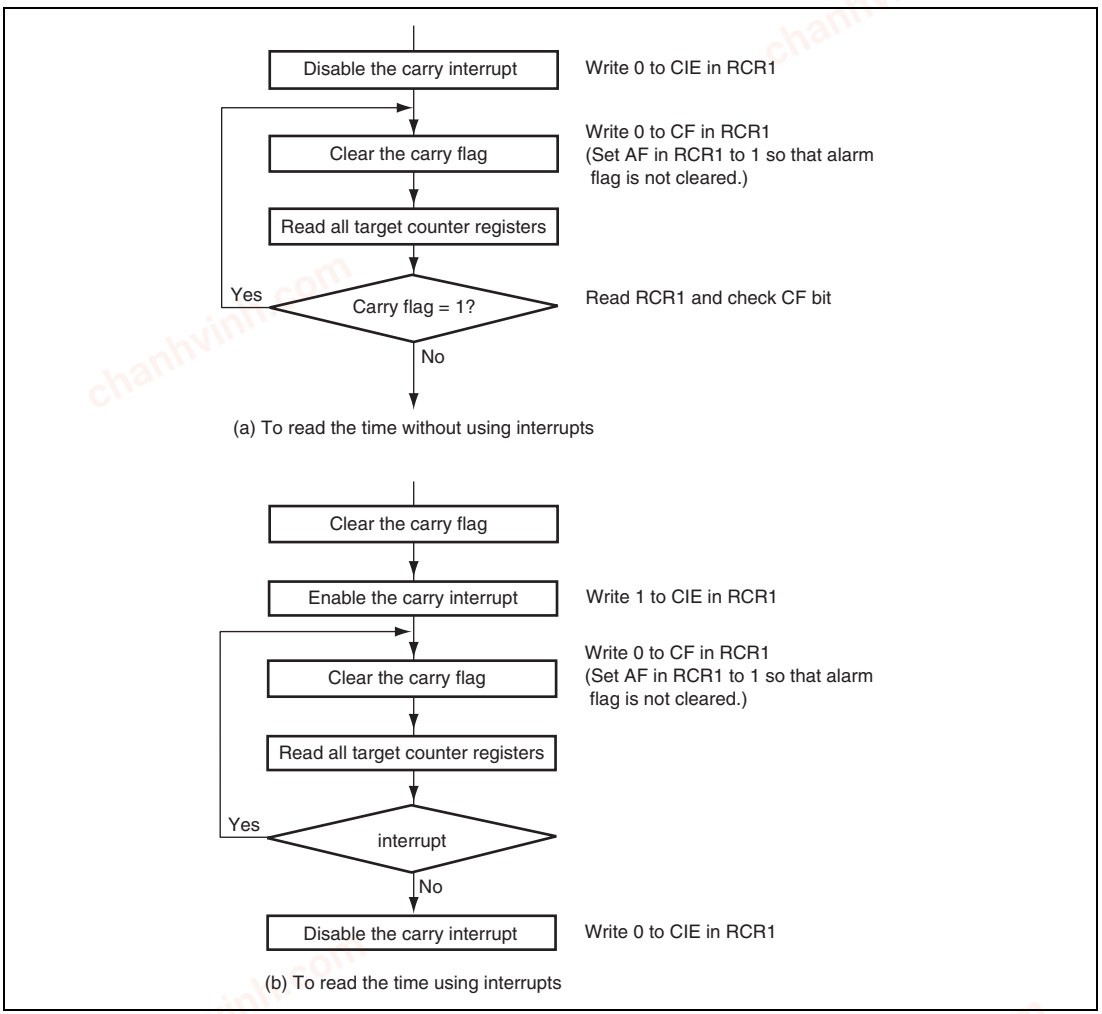


Figure 14.3 Reading Time

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 14.3 shows the method of reading the time without using interrupts; part (b) in figure 14.3 shows the method using carry interrupts. To keep programming simple, method (a) should normally be used.

chanhvinh.com

14.4.4 Alarm Function

Figure 14.4 shows how to use the alarm function.

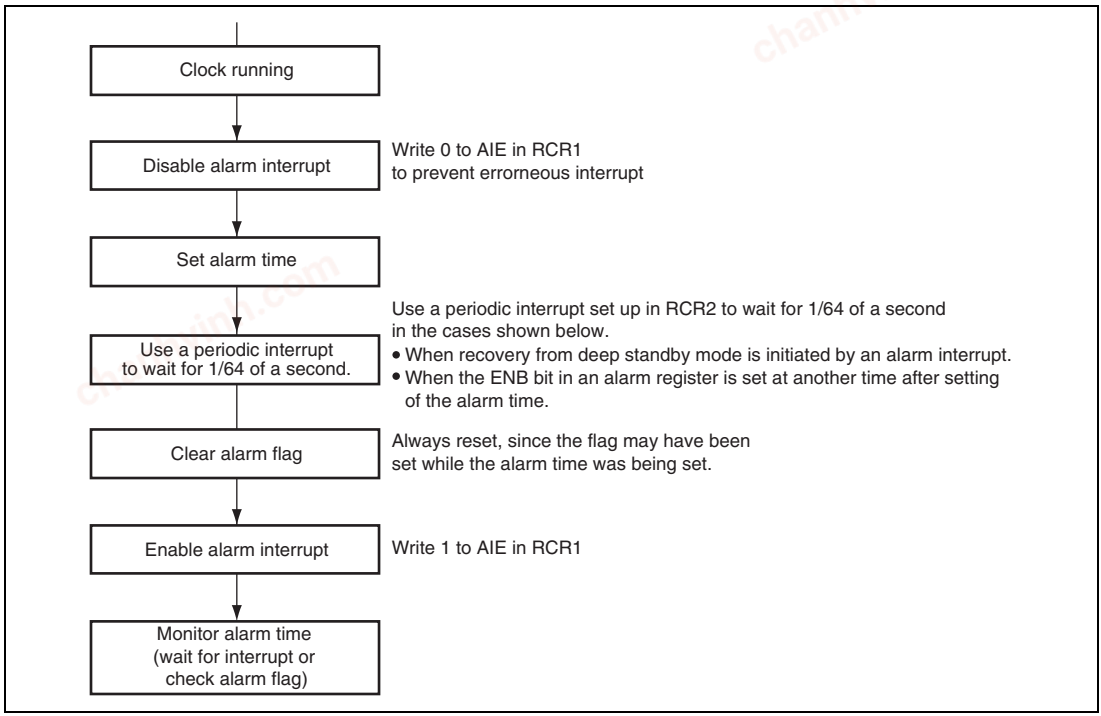


Figure 14.4 Using Alarm Function

Alarms can be generated using seconds, minutes, hours, day of the week, date, 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

14.5 Usage Notes

14.5.1 Register Writing during Count

The following registers cannot be written to during a count (while bit 0 = 1 in RCR2).

RSECCNT, RMINCNT, RHRCNT, RDAYCNT, RWKCNT, RMONCNT, RYRCONT

The count must be stopped before writing to any of the above registers.

14.5.2 Use of Realtime Clock Periodic Interrupts

The method of using the periodic interrupt function is shown in figure 14.5.

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.

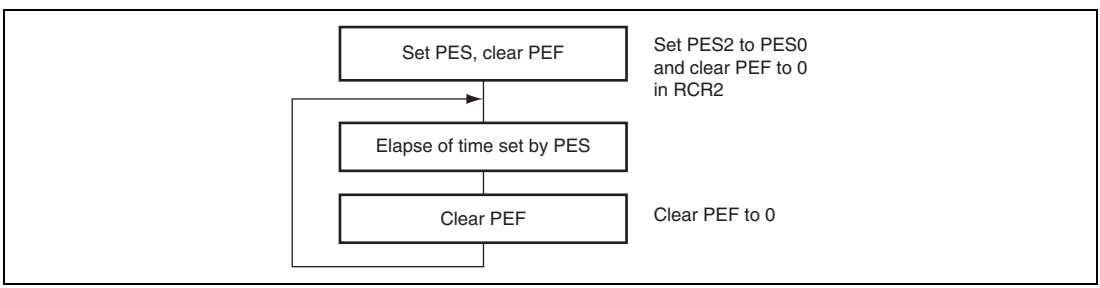


Figure 14.5 Using Periodic Interrupt Function

14.5.3 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 making register settings, do not fail to perform a dummy read of the registers to which settings were made before entering the standby state.

chanhvinh.com

14.5.4 Usage Notes when Writing to and Reading the Register

- After writing to a count register such as the second counter or to the RCR2 register, perform two dummy reads before reading from the register. The register value before the write is returned by the two dummy reads. The value newly written to the register takes effect only on the third read operation.
- Registers other than the above can be read immediately after a write and the written value is reflected.

chanhvinh.com

chanhvinh.com

Section 15 Serial Communication Interface with FIFO

This LSI has an eight-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.

15.1 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 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)

chanhvinh.com

- Four types of interrupts: Transmit-FIFO-data-empty interrupt, break interrupt, receive-FIFO-data-full interrupt, and receive-error interrupts 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 1 and 3).
- 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 15.1 shows a block diagram.

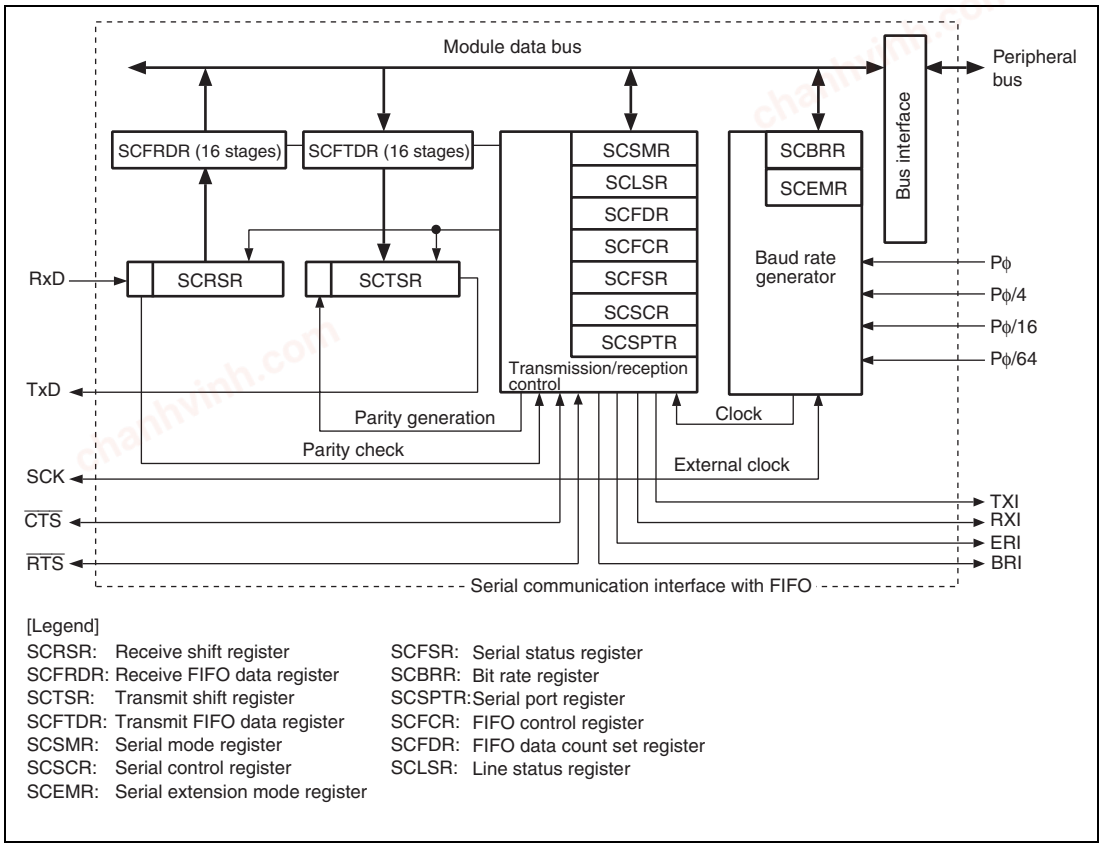


Figure 15.1 Block Diagram

chanhvinh.com

15.2 Input/Output Pins

Table 15.1 shows the pin configuration.

Table 15.1 Pin Configuration

Channel	Pin Name	Symbol	I/O	Function
0 to 3	Serial clock pins	SCK0 to SCK3*	I/O	Clock I/O
	Receive data pins	RxD0 to RxD3	Input	Receive data input
	Transmit data pins	TxD0 to TxD3	Output	Transmit data output
4 to 7	Receive data pins	RxD4 to RxD7	Input	Receive data input
	Transmit data pins	TxD4 to TxD7	Output	Transmit data output
1, 3	Request to send pin	$\overline{\text{RTS1}}$, $\overline{\text{RTS3}}$*	I/O	Request to send
	Clear to send pin	$\overline{\text{CTS1}}$, $\overline{\text{CTS3}}$*	I/O	Clear to send

Note: * Pins SCK3, $\overline{\text{RTS3}}$ and $\overline{\text{CTS3}}$ cannot be used in the SH7262 Group.

chanhvinh.com

15.3 Register Descriptions

This module has the following registers.

Table 15.2 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'FFFE8000	16
	Bit rate register_0	SCBRR_0	R/W	H'FF	H'FFFE8004	8
	Serial control register_0	SCSCR_0	R/W	H'0000	H'FFFE8008	16
	Transmit FIFO data register_0	SCFTDR_0	W	Undefined	H'FFFE800C	8
	Serial status register_0	SCFSR_0	R/(W)*[1]	H'0060	H'FFFE8010	16
	Receive FIFO data register_0	SCFRDR_0	R	Undefined	H'FFFE8014	8
	FIFO control register_0	SCFCR_0	R/W	H'0000	H'FFFE8018	16
	FIFO data count register_0	SCFDR_0	R	H'0000	H'FFFE801C	16
	Serial port register_0	SCSPTR_0	R/W	H'0050	H'FFFE8020	16
	Line status register_0	SCLSR_0	R/(W)*[2]	H'0000	H'FFFE8024	16
	Serial extension mode register_0	SCEMR_0	R/W	H'0000	H'FFFE8028	16
1	Serial mode register_1	SCSMR_1	R/W	H'0000	H'FFFE8800	16
	Bit rate register_1	SCBRR_1	R/W	H'FF	H'FFFE8804	8
	Serial control register_1	SCSCR_1	R/W	H'0000	H'FFFE8808	16
	Transmit FIFO data register_1	SCFTDR_1	W	Undefined	H'FFFE880C	8
	Serial status register_1	SCFSR_1	R/(W)*[1]	H'0060	H'FFFE8810	16
	Receive FIFO data register_1	SCFRDR_1	R	Undefined	H'FFFE8814	8
	FIFO control register_1	SCFCR_1	R/W	H'0000	H'FFFE8818	16
	FIFO data count register_1	SCFDR_1	R	H'0000	H'FFFE881C	16
	Serial port register_1	SCSPTR_1	R/W	H'0050	H'FFFE8820	16
	Line status register_1	SCLSR_1	R/(W)*[2]	H'0000	H'FFFE8824	16
	Serial extension mode register_1	SCEMR_1	R/W	H'0000	H'FFFE8828	16

chanhvinh.com

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
2	Serial mode register_2	SCSMR_2	R/W	H'0000	H'FFFE9000	16
	Bit rate register_2	SCBRR_2	R/W	H'FF	H'FFFE9004	8
	Serial control register_2	SCSCR_2	R/W	H'0000	H'FFFE9008	16
	Transmit FIFO data register_2	SCFTDR_2	W	Undefined	H'FFFE900C	8
	Serial status register_2	SCFSR_2	R/(W)*[1]	H'0060	H'FFFE9010	16
	Receive FIFO data register_2	SCFRDR_2	R	Undefined	H'FFFE9014	8
	FIFO control register_2	SCFCR_2	R/W	H'0000	H'FFFE9018	16
	FIFO data count register_2	SCFDR_2	R	H'0000	H'FFFE901C	16
	Serial port register_2	SCSPTR_2	R/W	H'0050	H'FFFE9020	16
	Line status register_2	SCLSR_2	R/(W)*[2]	H'0000	H'FFFE9024	16
	Serial extension mode register_2	SCEMR_2	R/W	H'0000	H'FFFE9028	16
3	Serial mode register_3	SCSMR_3	R/W	H'0000	H'FFFE9800	16
	Bit rate register_3	SCBRR_3	R/W	H'FF	H'FFFE9804	8
	Serial control register_3	SCSCR_3	R/W	H'0000	H'FFFE9808	16
	Transmit FIFO data register_3	SCFTDR_3	W	Undefined	H'FFFE980C	8
	Serial status register_3	SCFSR_3	R/(W)*[1]	H'0060	H'FFFE9810	16
	Receive FIFO data register_3	SCFRDR_3	R	Undefined	H'FFFE9814	8
	FIFO control register_3	SCFCR_3	R/W	H'0000	H'FFFE9818	16
	FIFO data count register_3	SCFDR_3	R	H'0000	H'FFFE981C	16
	Serial port register_3	SCSPTR_3	R/W	H'0050	H'FFFE9820	16
	Line status register_3	SCLSR_3	R/(W)*[2]	H'0000	H'FFFE9824	16
	Serial extension mode register_3	SCEMR_3	R/W	H'0000	H'FFFE9828	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'FFFEA000	16
	Bit rate register_4	SCBRR_4	R/W	H'FF	H'FFFEA004	8
	Serial control register_4	SCSCR_4	R/W	H'0000	H'FFFEA008	16
	Transmit FIFO data register_4	SCFTDR_4	W	Undefined	H'FFFEA00C	8
	Serial status register_4	SCFSR_4	R/(W)*[1]	H'0060	H'FFFEA010	16
	Receive FIFO data register_4	SCFRDR_4	R	Undefined	H'FFFEA014	8
	FIFO control register_4	SCFCR_4	R/W	H'0000	H'FFFEA018	16
	FIFO data count register_4	SCFDR_4	R	H'0000	H'FFFEA01C	16
	Serial port register_4	SCSPTR_4	R/W	H'0050	H'FFFEA020	16
	Line status register_4	SCLSR_4	R/(W)*[2]	H'0000	H'FFFEA024	16
	Serial extension mode register_4	SCEMR_4	R/W	H'0000	H'FFFEA028	16
5	Serial mode register_5	SCSMR_5	R/W	H'0000	H'FFFEA800	16
	Bit rate register_5	SCBRR_5	R/W	H'FF	H'FFFEA804	8
	Serial control register_5	SCSCR_5	R/W	H'0000	H'FFFEA808	16
	Transmit FIFO data register_5	SCFTDR_5	W	Undefined	H'FFFEA80C	8
	Serial status register_5	SCFSR_5	R/(W)*[1]	H'0060	H'FFFEA810	16
	Receive FIFO data register_5	SCFRDR_5	R	Undefined	H'FFFEA814	8
	FIFO control register_5	SCFCR_5	R/W	H'0000	H'FFFEA818	16
	FIFO data count register_5	SCFDR_5	R	H'0000	H'FFFEA81C	16
	Serial port register_5	SCSPTR_5	R/W	H'0050	H'FFFEA820	16
	Line status register_5	SCLSR_5	R/(W)*[2]	H'0000	H'FFFEA824	16
	Serial extension mode register_5	SCEMR_5	R/W	H'0000	H'FFFEA828	16

chanhvinh.com

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
6	Serial mode register_6	SCSMR_6	R/W	H'0000	H'FFFEB000	16
	Bit rate register_6	SCBRR_6	R/W	H'FF	H'FFFEB004	8
	Serial control register_6	SCSCR_6	R/W	H'0000	H'FFFEB008	16
	Transmit FIFO data register_6	SCFTDR_6	W	Undefined	H'FFFEB00C	8
	Serial status register_6	SCFSR_6	R/(W)*1	H'0060	H'FFFEB010	16
	Receive FIFO data register_6	SCFRDR_6	R	Undefined	H'FFFEB014	8
	FIFO control register_6	SCFCR_6	R/W	H'0000	H'FFFEB018	16
	FIFO data count register_6	SCFDR_6	R	H'0000	H'FFFEB01C	16
	Serial port register_6	SCSPTR_6	R/W	H'0050	H'FFFEB020	16
	Line status register_6	SCLSR_6	R/(W)*2	H'0000	H'FFFEB024	16
	Serial extension mode register_6	SCEMR_6	R/W	H'0000	H'FFFEB028	16
7	Serial mode register_7	SCSMR_7	R/W	H'0000	H'FFFEB800	16
	Bit rate register_7	SCBRR_7	R/W	H'FF	H'FFFEB804	8
	Serial control register_7	SCSCR_7	R/W	H'0000	H'FFFEB808	16
	Transmit FIFO data register_7	SCFTDR_7	W	Undefined	H'FFFEB80C	8
	Serial status register_7	SCFSR_7	R/(W)*1	H'0060	H'FFFEB810	16
	Receive FIFO data register_7	SCFRDR_7	R	Undefined	H'FFFEB814	8
	FIFO control register_7	SCFCR_7	R/W	H'0000	H'FFFEB818	16
	FIFO data count register_7	SCFDR_7	R	H'0000	H'FFFEB81C	16
	Serial port register_7	SCSPTR_7	R/W	H'0050	H'FFFEB820	16
	Line status register_7	SCLSR_7	R/(W)*2	H'0000	H'FFFEB824	16
	Serial extension mode register_7	SCEMR_7	R/W	H'0000	H'FFFEB828	16

Notes: 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.

2. Only 0 can be written to clear the flag. Bits 15 to 1 are read-only bits that cannot be modified.

chanhvinh.com

15.3.1 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, converting the data to parallel form. When one byte has been received, it is automatically transferred to the receive FIFO data register (SCFRDR).

The CPU cannot read or write to SCRSR directly.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-

15.3.2 Receive FIFO Data Register (SCFRDR)

SCFRDR is a 16-byte 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

15.3.3 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:	-	-	-	-	-	-	-	-

15.3.4 Transmit FIFO Data Register (SCFTDR)

SCFTDR is a 16-byte 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 new 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

15.3.5 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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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/$\overline{E}$) setting. Receive data parity is checked according to the even/odd (O/$\overline{E}$) 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] Notes: 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 15.3.8, Bit Rate Register (SCBRR). 00: Pϕ 01: Pϕ/4 10: Pϕ/16 11: Pϕ/64 Note: Pϕ: Peripheral clock

chanhvinh.com

15.3.6 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), when the quantity of data in the transmit FIFO register becomes less than the specified number of transmission triggers, and when the TDFE flag in the serial status register (SCFSR) is set to1. 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 by clearing TDFE to 0 after reading 1 from TDFE, or can be cleared by clearing TIE to 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 to1, receive-error (ERI) interrupts requested when the ER flag in SCFSR is set to1, and break (BRI) interrupts requested when the BRK flag in SCFSR or the ORER flag in line status register (SCLSR) is set to1. 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, BR 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 the serial transmitter. 0: Transmitter disabled 1: Transmitter enabled* Note: * Serial transmission starts after writing of transmit data into SCFTDR. Select the transmit format in SCSMR and SCFCR and reset the transmit FIFO before setting TE to 1.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
4	RE	0	R/W	Receive Enable Enables or disables the serial receiver. 0: Receiver disabled*[1] 1: Receiver 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. Serial reception starts when a start bit is detected in asynchronous mode, or synchronous clock is detected in clock synchronous mode. Select the receive format in SCSMR and SCFCR and reset the receive FIFO before setting RE to 1.
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, BR 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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 serial 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 pin (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 serial clock output 01: Internal clock, SCK pin used for serial clock output 10: External clock, SCK pin used for serial clock input 11: Setting prohibited

chanhvinh.com

15.3.7 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 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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 a power-on reset • ER is cleared to 0 when the chip is when 0 is written 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/$\overline{E}$ bit in SCSMR Notes: 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 when the chip is 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 receive 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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 • TDFE is cleared to 0 when direct memory access controller is activated by transmit FIFO data empty interrupt (TXI) and write data exceeding the specified transmission trigger number 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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 when the chip is 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, and a framing error occurs with space 0 in the subsequent receive data 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.
3	FER	0	R	Framing Error Indication Indicates a framing error in the data read from the next 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 when the chip undergoes 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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
2	PER	0	R	Parity Error Indication Indicates a parity error in the data read from the next 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 when the chip undergoes 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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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, standby mode • 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 read SCFRDR until the quantity of receive data in SCFRDR becomes less than the specified receive trigger number 1: The quantity of receive data in SCFRDR is more than the specified receive trigger number [Setting condition] • RDF is set to 1 when a quantity of receive data more than the specified receive trigger number is stored in SCFRDR*[1] Note: 1. As SCFTDR is a 16-byte FIFO register, the maximum quantity of data that can be read when RDF is 1 becomes 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 when the chip undergoes a power-on reset • DR is cleared to 0 when all receive data are read after 1 is read from DR and then 0 is written. • DR is cleared to 0 when all receive data are read 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

15.3.8 Bit Rate Register (SCBRR)

SCBRR is an 8-bit register that is used with the CKS1 and CKS0 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 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):

$$N = \frac{P\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$ (Operation on a base clock with a frequency of 16 times the bit rate)

$$N = \frac{P\phi}{32 \times 2^{2n-1} \times B} \times 10^6 - 1$$ (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):

$$N = \frac{P\phi}{32 \times 2^{2n-1} \times B} \times 10^6 - 1$$ (Operation on a base clock with a frequency of 16 times the bit rate)

$$N = \frac{P\phi}{16 \times 2^{2n-1} \times B} \times 10^6 - 1$$ (Operation on a base clock with a frequency of 8 times the bit rate)

chanhvinh.com

- Clock synchronous mode:

$$N = \frac{P\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B: Bit rate (bits/s)
N: SCBRR setting for baud rate generator ($0 \leq N \leq 255$)
(The setting must satisfy the electrical characteristics.)
Pϕ: 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 15.3.)

Table 15.3 SCSMR Settings

		SCSMR Settings	
n	**Clock Source**	**CKS[1]**	**CKS[0]**
0	Pϕ	0	0
1	Pϕ/4	0	1
2	Pϕ/16	1	0
3	Pϕ/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):

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N+1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$ (Operation on a base clock with a frequency of 16 times the bit rate)

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N+1) \times B \times 32 \times 2^{2n-1}} - 1 \right\} \times 100$$ (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):

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N+1) \times B \times 32 \times 2^{2n-1}} - 1 \right\} \times 100$$ (Operation on a base clock with a frequency of 16 times the bit rate)

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N+1) \times B \times 16 \times 2^{2n-1}} - 1 \right\} \times 100$$ (Operation on a base clock with a frequency of 8 times the bit rate)

chanhvinh.com

Table 15.4 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 1), and table 15.5 lists the sample SCBRR settings in clock synchronous mode.

Table 15.4 Bit Rates and SCBRR Settings (Asynchronous Mode, BGDM = 0, ABCS = 0)

Bit Rate	Pϕ (MHz)											
	24			28			32			36		
(bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	106	–0.44	3	123	0.23	3	141	0.03	3	159	–0.12
150	3	77	0.16	3	90	0.16	3	103	0.16	3	116	0.16
300	2	155	0.16	3	45	–0.93	3	51	0.16	2	233	0.16
600	2	77	0.16	3	22	–0.93	3	25	0.16	2	116	0.16
1200	1	155	0.16	2	45	–0.93	2	51	0.16	1	233	0.16
2400	1	77	0.16	2	22	–0.93	2	25	0.16	1	116	0.16
4800	0	155	0.16	1	45	–0.93	1	51	0.16	0	233	0.16
9600	0	77	0.16	1	22	–0.93	1	25	0.16	0	116	0.16
19200	0	38	0.16	0	45	–0.93	0	51	0.16	0	58	–0.69
31250	0	23	0.00	0	27	0.00	0	31	0.00	0	35	0.00
38400	0	19	–2.34	0	22	–0.93	0	25	0.16	0	28	1.02

Note: The error rate should be ≤ 1 %.

chanhvinh.com

Table 15.5 Bit Rates and SCBRR Settings (Clock Synchronous Mode)

Bit Rate (bits/s)	Pϕ (MHz)							
	24		28		32		36	
	n	N	n	N	n	N	n	N
500	3	187	3	218	3	249	—	—
1000	3	93	3	108	3	124	3	140
2500	2	149	2	174	2	199	2	224
5000	2	74	2	87	2	99	2	112
10000	1	149	1	174	1	199	1	224
25000	1	59	1	69	1	79	1	89
50000	1	29	0	139	0	159	0	179
100000	0	59	0	69	0	79	0	89
250000	0	23	0	27	0	31	0	35
500000	0	11	0	13	0	15	0	17
1000000	0	5	0	6	0	7	0	8
2000000	0	2	—	—	0	3	—	—

[Legend]

—: Setting possible, but error occurs

chanhvinh.com

Table 15.6 indicates the maximum bit rates in asynchronous mode when the baud rate generator is used. Table 15.7 lists the maximum bit rates in asynchronous mode when the external clock input is used. Table 15.8 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 15.6 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode)

	Settings				
Pϕ (MHz)	**BGDM**	**ABCS**	**n**	**N**	**Maximum Bit Rate (bits/s)**
24	0	0	0	0	750000
		1	0	0	1500000
	1	0	0	0	1500000
		1	0	0	3000000
28	0	0	0	0	875000
		1	0	0	1750000
	1	0	0	0	1750000
		1	0	0	3500000
32	0	0	0	0	1000000
		1	0	0	2000000
	1	0	0	0	2000000
		1	0	0	4000000
36	0	0	0	0	1125000
		1	0	0	2250000
	1	0	0	0	2250000
		1	0	0	4500000

chanhvinh.com

Table 15.7 Maximum Bit Rates with External Clock Input (Asynchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Settings ABCS	Maximum Bit Rate (bits/s)
24	6.0000	0	375000
		1	750000
28	7.0000	0	437500
		1	875000
32	8.0000	0	500000
		1	1000000
36	9.0000	0	562500
		1	1125000

Table 15.8 Maximum Bit Rates with External Clock Input (Clock Synchronous Mode, $t_{Scyc} = 12\ t_{pcyc}$)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
24	2.0000	2000000.0
28	2.3333	2333333.3
32	2.6667	2666666.7
36	3.0000	3000000.0

chanhvinh.com

15.3.9 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 more than the 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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 register (SCFRDR) is increased more than the set trigger number shown below. • Asynchronous mode • Clock synchronous mode 00: 1 00: 1 01: 4 01: 2 10: 8 10: 8 11: 14 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 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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
3	MCE	0	R/W	Modem Control Enable Enables modem control signals $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$. For channels 0 to 2 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.
2	TFRST	0	R/W	Transmit FIFO Data Register Reset Disables the transmit data in the transmit FIFO data register and resets the data 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 data 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

15.3.10 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 SCFTDR is full of transmit data.
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 SCFRDR full of receive data.

chanhvinh.com

15.3.11 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 Indicates input or output of 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 Indicates 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 PFC. 0: Input/output data is low level 1: Input/output data is high level

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
5	CTSIO	0	R/W	$\overline{\text{CTS}}$ Port Input/Output Indicates input or output of 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 Indicates 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 PFC. 0: Input/output data is low level 1: Input/output data is high level
3	SCKIO	0	R/W	SCK Port Input/Output Indicates input or output of 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 Indicates 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 PFC. 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 Indicates input or output of 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 Indicates 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 PFC. 0: Input/output data is low level 1: Input/output data is high level

chanhvinh.com

15.3.12 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. 0: Receiving is in progress or has ended normally*[1] [Clearing conditions] • ORER is cleared to 0 when the chip is 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*[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. Notes: 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

15.3.13 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 operates 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).

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, this bit is invalid (the baud rate generator operates in normal mode regardless of the BGDM setting). 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

15.4 Operation

15.4.1 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 receptions, reducing the overhead of the CPU, and enabling continuous high-speed communication. Furthermore, channel 3 has $\overline{\text{RTS}}$ and $\overline{\text{CTS}}$ signals to be used as modem control signals.

The transmission format is selected in the serial mode register (SCSMR), as shown in table 15.9. The clock source is selected by the combination of the CKE1 and CKE0 bits in the serial control register (SCSCR), as shown in table 15.10.

(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 15.9 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

chanhvinh.com

Table 15.10 SCSMR and SCSCR Settings and Clock Source Selection

SCSMR	SCSCR		Transmit/Receive Clock	
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 serial clock
	10		External	Inputs the serial 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/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

15.4.2 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-byte FIFO buffered, so data can be written and read while transmitting and receiving are in progress, enabling continuous transmitting and receiving.

Figure 15.2 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.

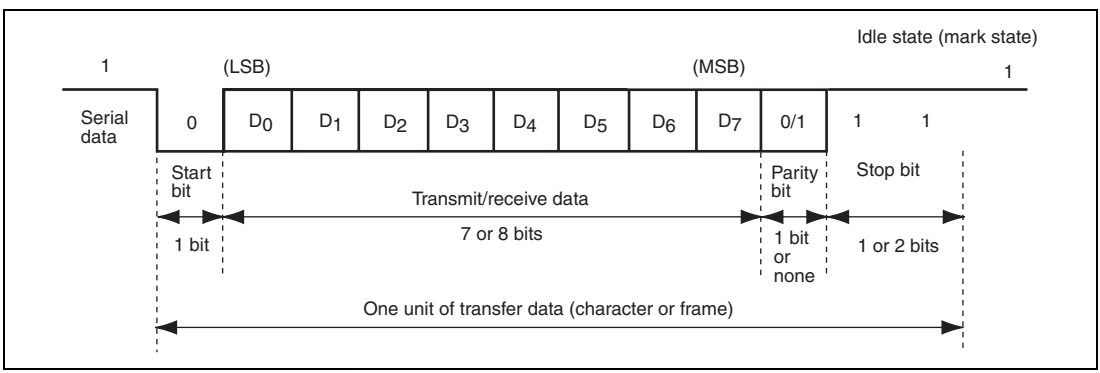


Figure 15.2 Example of Data Format in Asynchronous Communication (8-Bit Data with Parity and Two Stop Bits)

chanhvinh.com

(1) Transmit/Receive Formats

Table 15.11 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 15.11 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 CKE1 and CKE0 bits in the serial control register (SCSCR). For clock source selection, refer to table 15.10, 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 15.3 shows a sample flowchart for initialization.

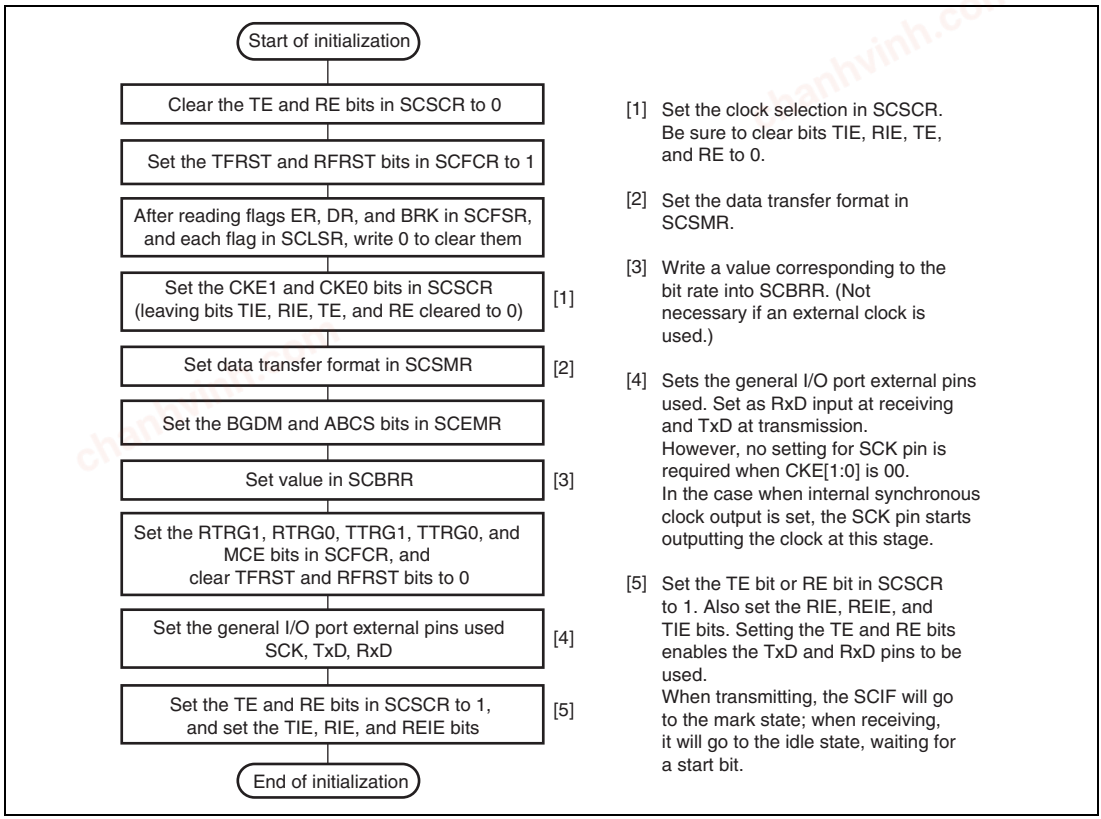


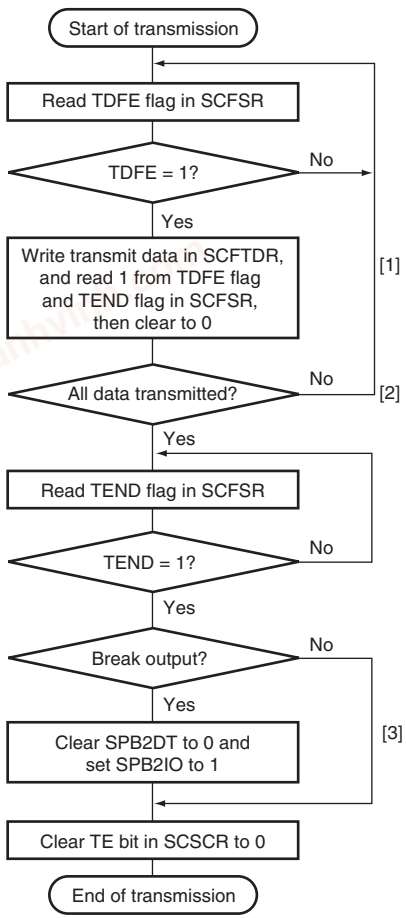
Figure 15.3 Sample Flowchart for Initialization

chanhvinh.com

- Transmitting Serial Data (Asynchronous Mode)

  Figure 15.4 shows a sample flowchart for serial transmission.

  Use the following procedure for serial data transmission after enabling transmission.



[1] Status check and transmit data write:
Read SCFSR and check that the TDFE flag is set to 1, then write transmit data to SCFTDR, and read 1 from the TDFE and TEND flags, then clear to 0.
The quantity of transmit data that can be written is 16 - (transmit trigger set number).

[2] Serial transmission continuation procedure:
To continue serial transmission, read 1 from the TDFE flag to confirm that writing is possible, then write data to SCFTDR, and then clear the TDFE flag to 0.

[3] Break output during serial transmission:
To output a break in serial transmission, clear the SPB2DT bit to 0 and set the SPB2IO bit to 1 in SCSPTR, then clear the TE bit in SCSCR to 0.

In [1] and [2], it is possible to ascertain the number of data bytes that can be written from the number of transmit data bytes in SCFTDR indicated by the upper 8 bits of SCFDR.

Figure 15.4 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 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 (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 15.5 shows an example of the operation for transmission.

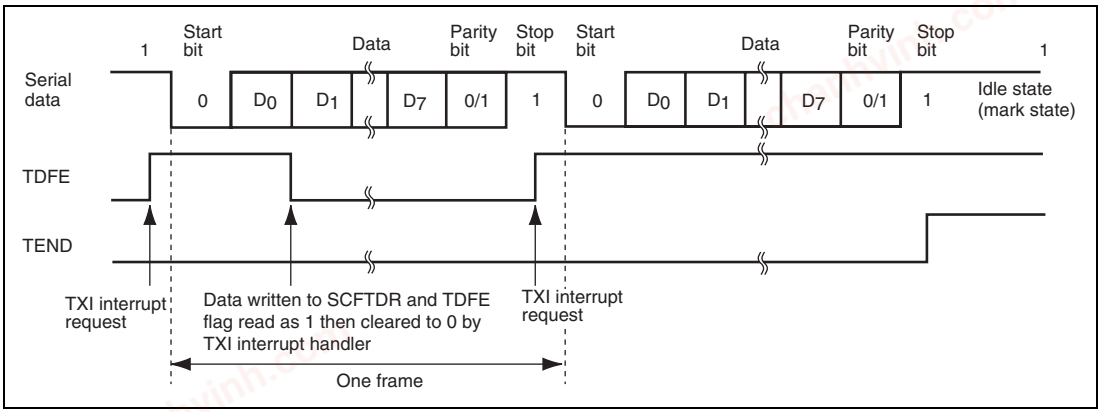


Figure 15.5 Example of Transmit Operation (8-Bit Data, Parity, 1 Stop Bit)

4. When modem control is enabled in channel 3, 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 15.6 shows an example of the operation when modem control is used.

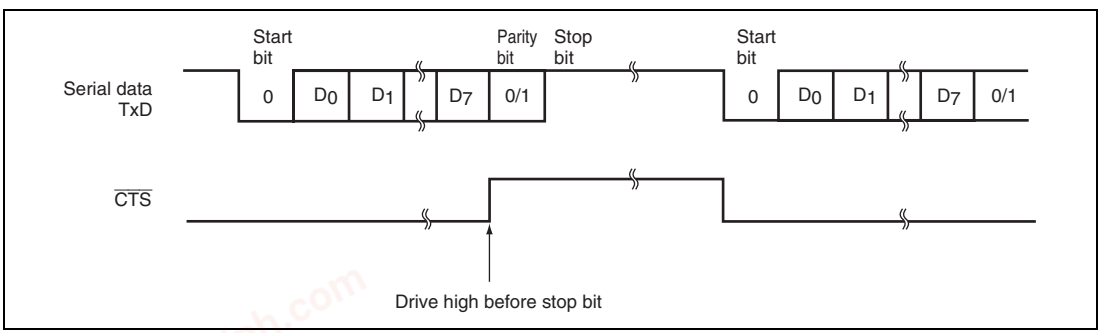


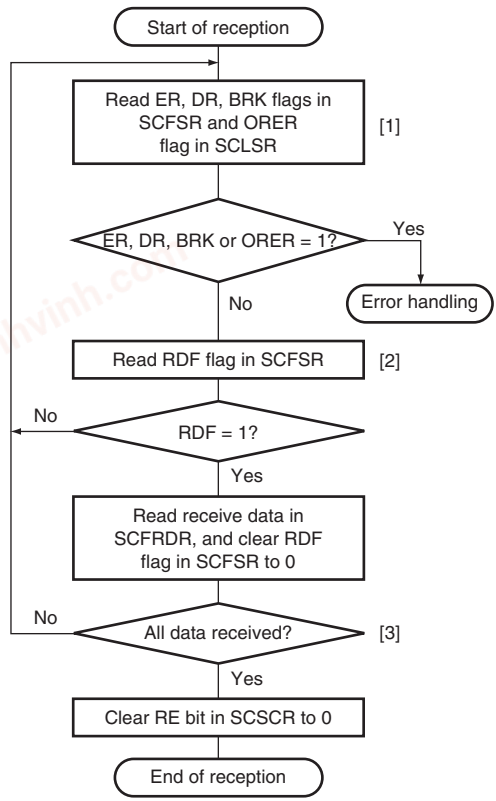
Figure 15.6 Example of Operation Using Modem Control ($\overline{\text{CTS}}$)

chanhvinh.com

- Receiving Serial Data (Asynchronous Mode)

  Figures 15.7 and 15.8 show sample flowcharts for serial reception.

  Use the following procedure for serial data reception after enabling reception.



[1] Receive error handling and break detection:

Read the DR, ER, and BRK flags in SCFSR, and the ORER flag in SCLSR, to identify any error, perform the appropriate error handling, then clear the DR, ER, BRK, and ORER flags to 0. In the case of a framing error, a break can also be detected by reading the value of the RxD pin.

[2] Status check and receive data read:

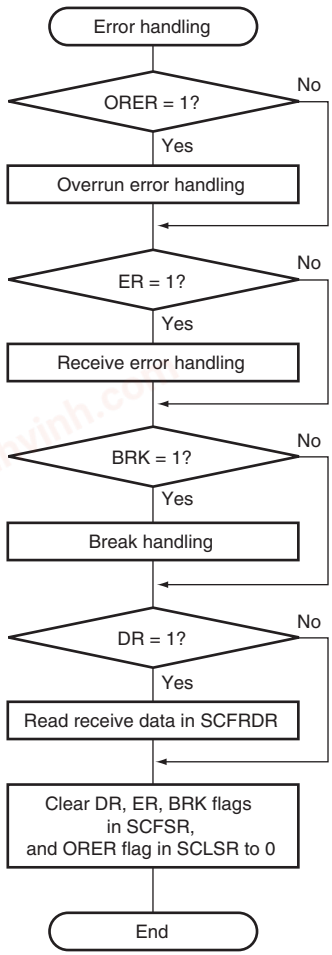
Read SCFSR and check that RDF flag = 1, then read the receive data in SCFRDR, read 1 from the RDF flag, and then clear the RDF flag to 0. The transition of the RDF flag from 0 to 1 can also be identified by a receive FIFO data full interrupt (RXI).

[3] Serial reception continuation procedure:

To continue serial reception, read at least the receive trigger set number of receive data bytes from SCFRDR, read 1 from the RDF flag, then clear the RDF flag to 0. The number of receive data bytes in SCFRDR can be ascertained by reading from SCRFDR.

Figure 15.7 Sample Flowchart for Receiving Serial Data

chanhvinh.com



- Whether a framing error or parity error has occurred in the receive data that is to be read from the receive FIFO data register (SCFRDR) can be ascertained from the FER and PER bits in the serial status register (SCFSR).
- When a break signal is received, receive data is not transferred to SCFRDR while the BRK flag is set. However, note that the last data in SCFRDR is H'00, and the break data in which a framing error occurred is stored.

Figure 15.8 Sample Flowchart for Receiving Serial Data (cont)

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 15.9 shows an example of the operation for reception.

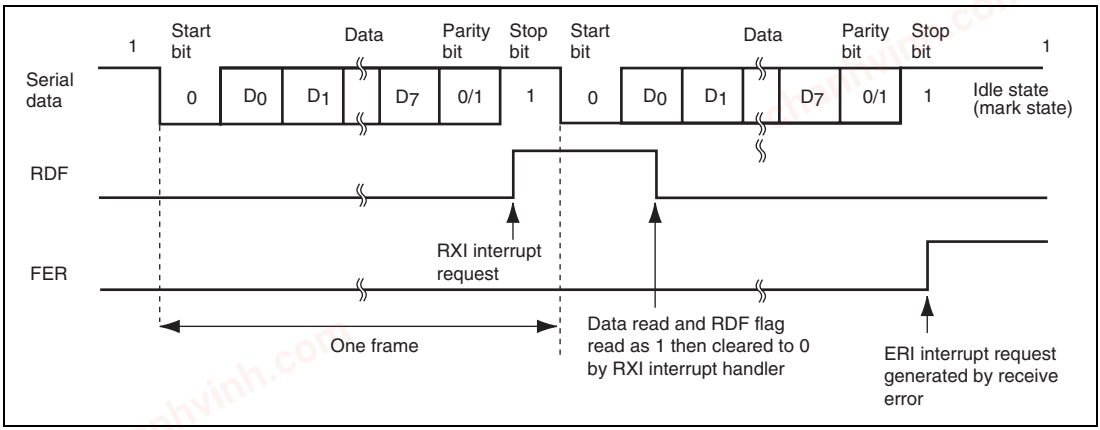


Figure 15.9 Example of Receive Operation (8-Bit Data, Parity, 1 Stop Bit)

5. When modem control is enabled in channel 3, 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 SCFRDR exceeds the number set for the RTS output active trigger.

Figure 15.10 shows an example of the operation when modem control is used.

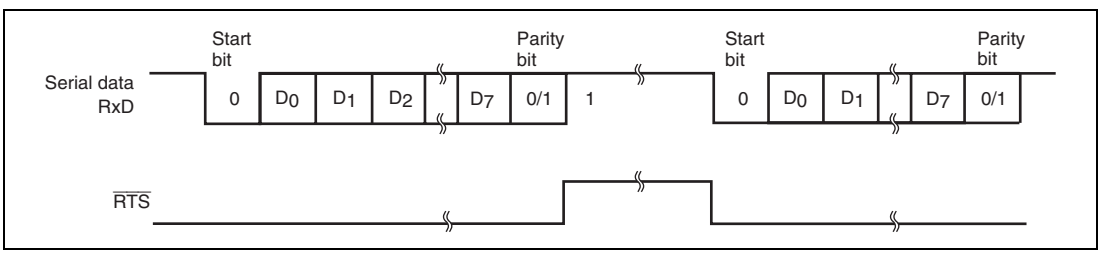


Figure 15.10 Example of Operation Using Modem Control ($\overline{\text{RTS}}$)

chanhvinh.com

15.4.3 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-byte FIFO buffered, so continuous transmitting or receiving is possible by reading or writing data while transmitting or receiving is in progress.

Figure 15.11 shows the general format in clock synchronous serial communication.

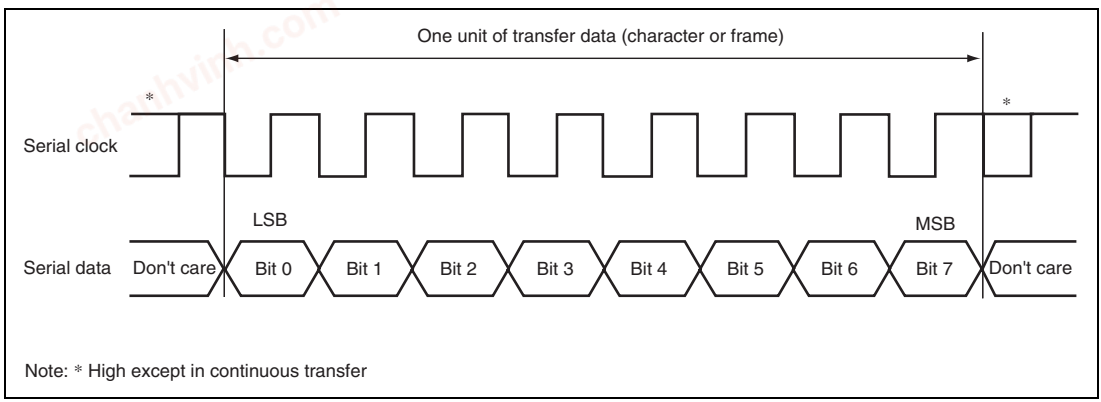


Figure 15.11 Data Format in Clock Synchronous Communication

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 serial 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 by the setting of the C/A bit in SCSMR and CKE[1:0] in SCSCR, or an external clock input from the SCK pin can be selected as the transmit/receive clock.

When this module operates on an internal clock, it outputs the clock signal at the SCK pin. Eight clock pulses are output per transmitted or received character. When transmission or reception is not performed, the clock signal remains in the high state. When only receiving, the clock signal outputs 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), then initialize this module. 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 15.12 shows a sample flowchart for initialization.

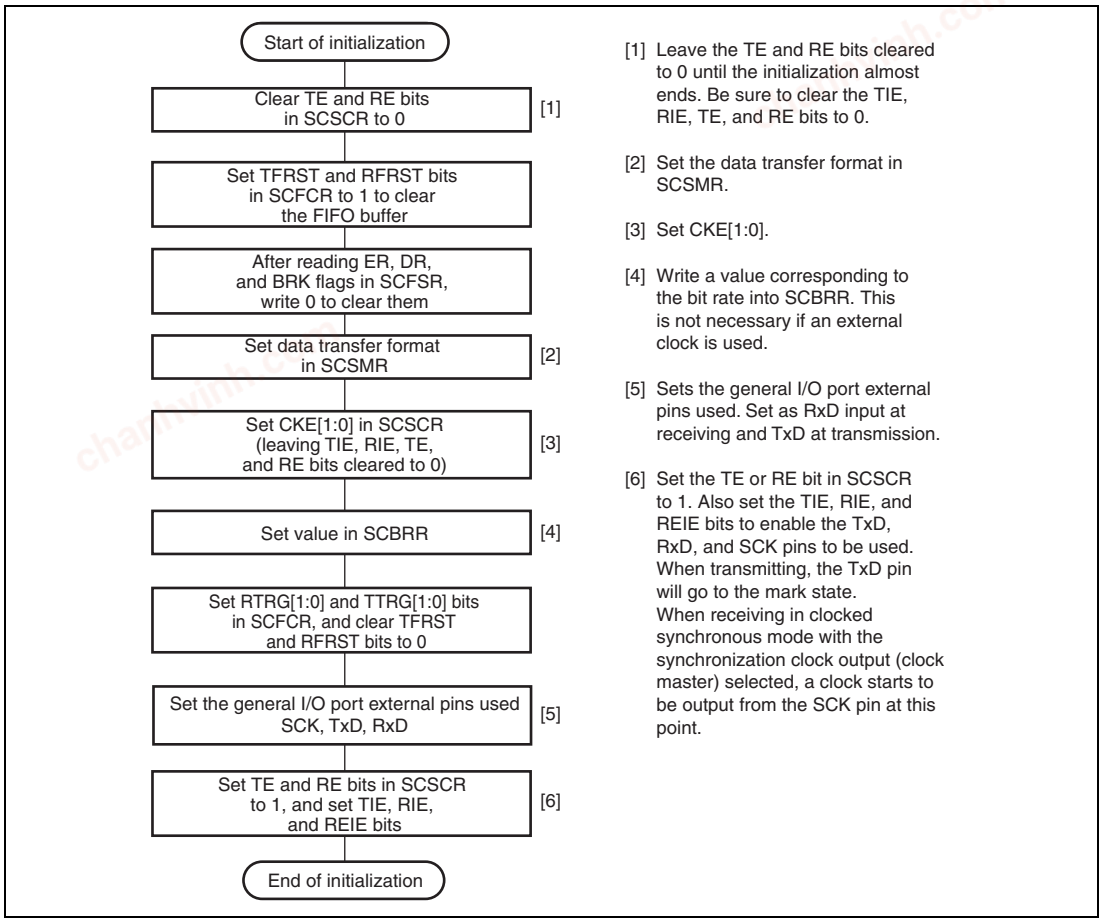


Figure 15.12 Sample Flowchart for Initialization

chanhvinh.com

- Transmitting Serial Data (Clock Synchronous Mode)

  Figure 15.13 shows a sample flowchart for transmitting serial data.

  Use the following procedure for serial data transmission after enabling transmit operation.

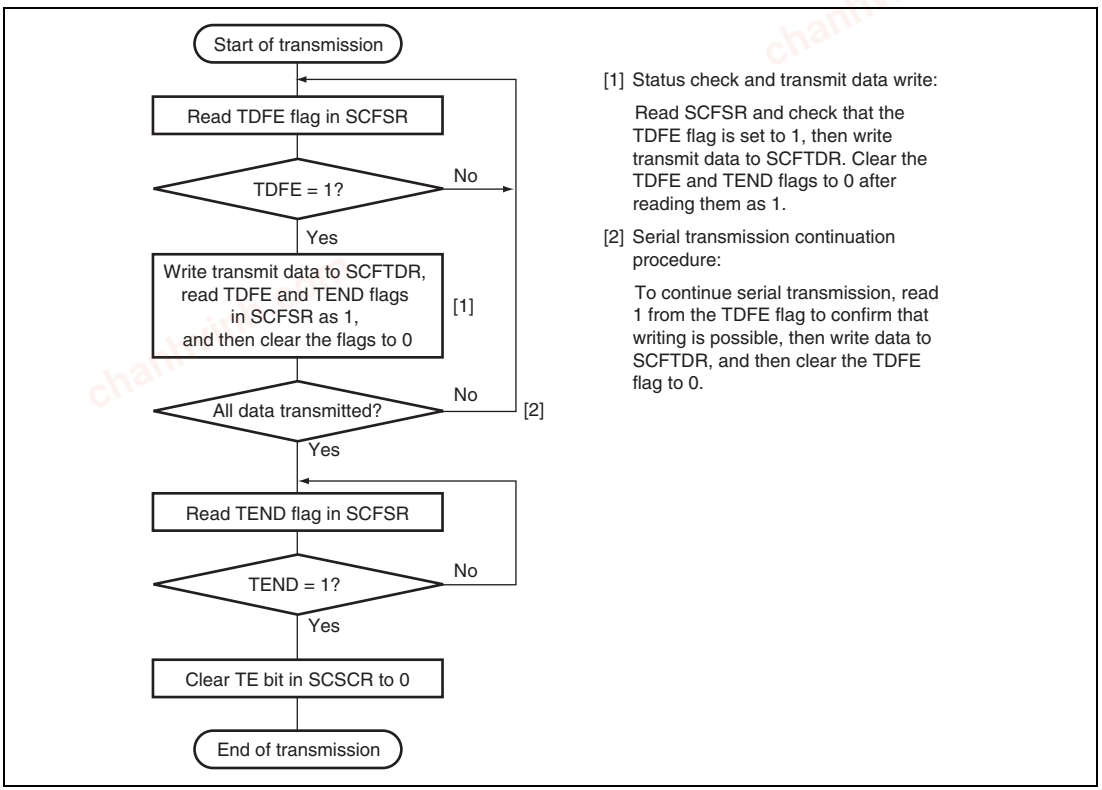


Figure 15.13 Sample Flowchart for Transmitting Serial Data

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 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 (SCSR) 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 15.14 shows an example of transmit operation.

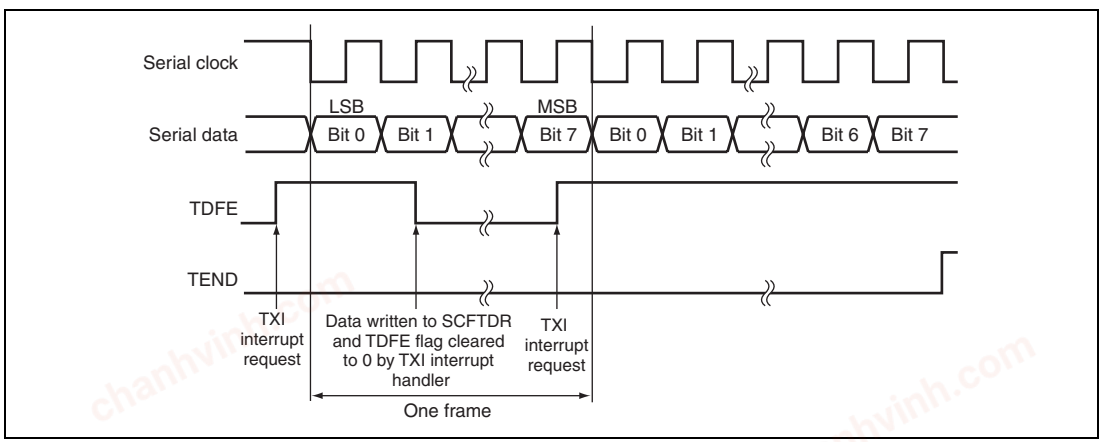


Figure 15.14 Example of Transmit Operation

chanhvinh.com

- Receiving Serial Data (Clock Synchronous Mode)

  Figures 15.15 and 15.16 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.

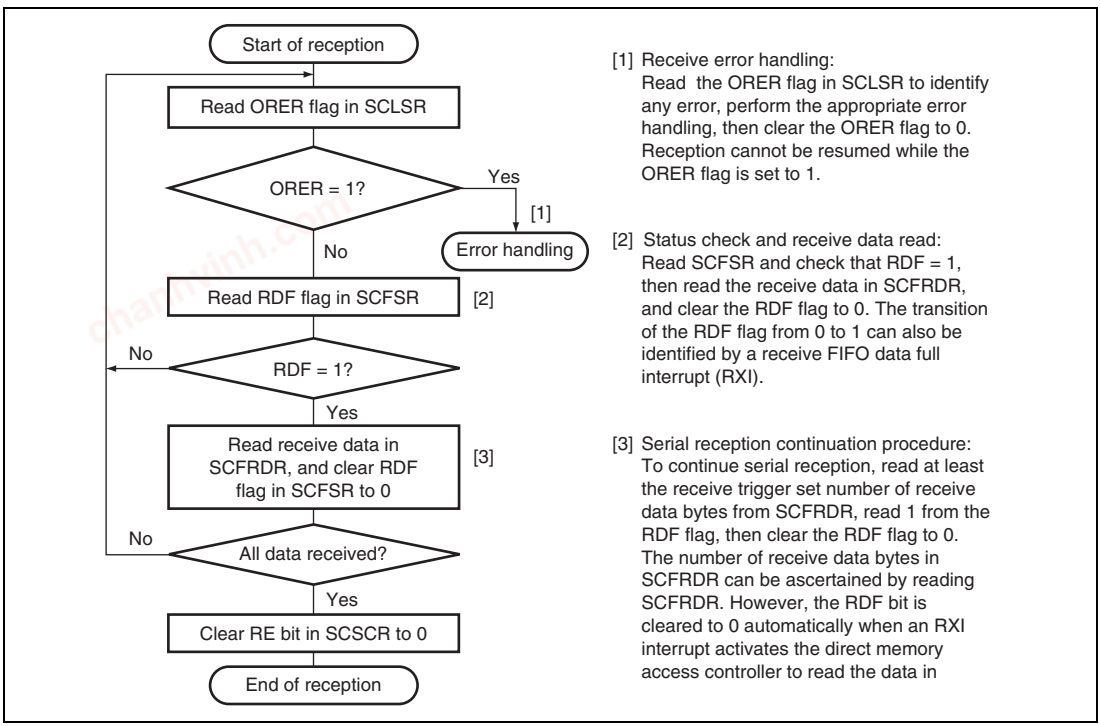


Figure 15.15 Sample Flowchart for Receiving Serial Data (1)

chanhvinh.com

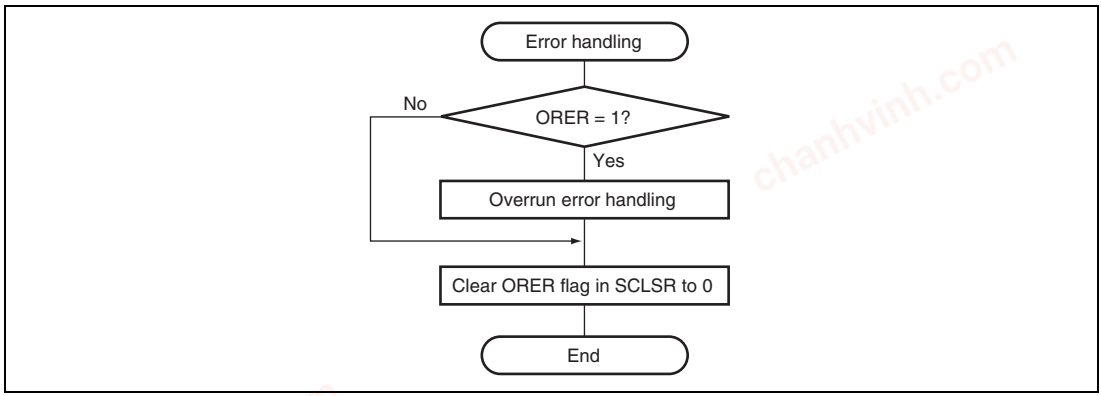


Figure 15.16 Sample Flowchart for Receiving Serial Data (2)

In serial reception, this module operates as described below.

1. Reception is started in synchronization with serial 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 FIFO data full interrupt enable bit (RIE) is set to 1 in SCSCR, a receive-data-full interrupt (RXI) request is generated. If the ORER bit is set to 1 and the receive-data-full 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.

chanhvinh.com

Figure 15.17 shows an example of receive operation.

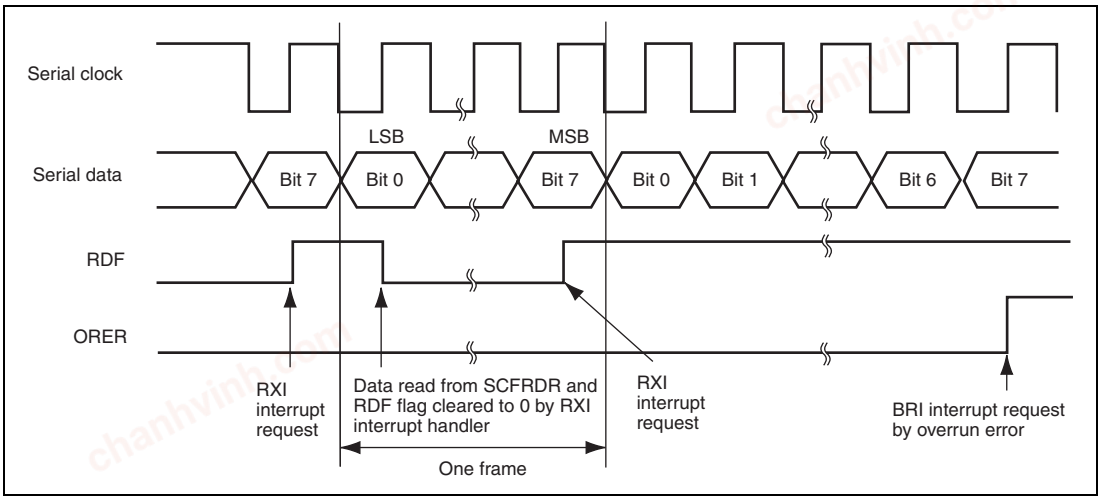


Figure 15.17 Example of Receive Operation

chanhvinh.com

- Transmitting and Receiving Serial Data Simultaneously (Clock Synchronous Mode)

  Figure 15.18 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.

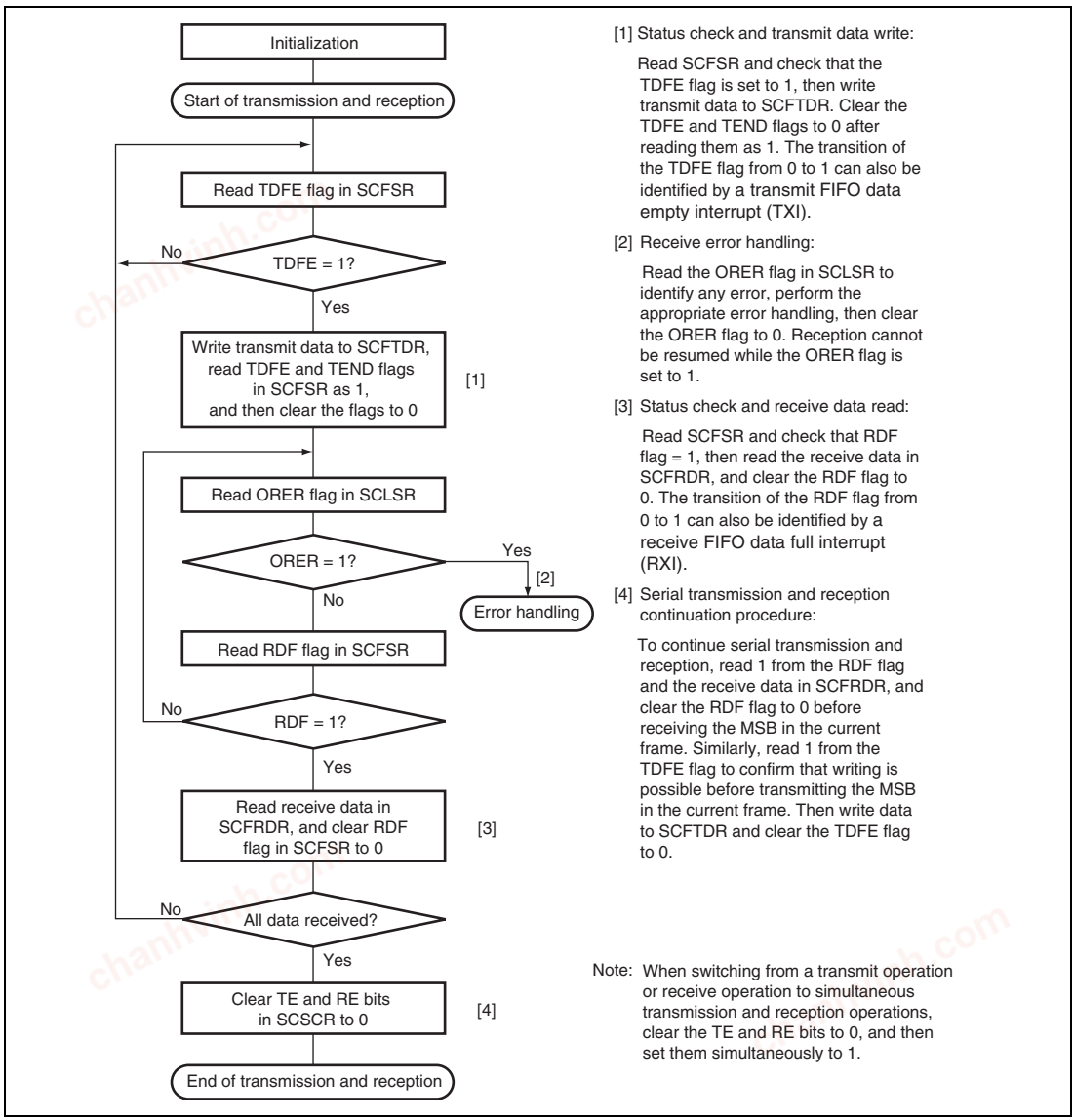


Figure 15.18 Sample Flowchart for Transmitting/Receiving Serial Data

chanhvinh.com

15.5 Interrupts

This module has four interrupt sources: transmit-FIFO-data-empty (TXI), receive-error (ERI), receive FIFO data full (RXI), and break (BRI).

Table 15.12 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.

It is possible to generate only the ERI or BRI interrupt without generating the RXI interrupt by setting the RIE bit to 0 and the REIE bit to 1 in SCSCR.

The TXI indicates that transmit data can be written, and the RXI indicates that there is receive data in SCFRDR.

Table 15.12 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

15.6 Usage Notes

Note the following when using this module.

15.6.1 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. TDFE flag clearing should therefore be carried out 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).

15.6.2 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 if it is 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).

chanhvinh.com

15.6.3 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.

15.6.4 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.

15.6.5 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.

15.6.6 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 15.19.

chanhvinh.com

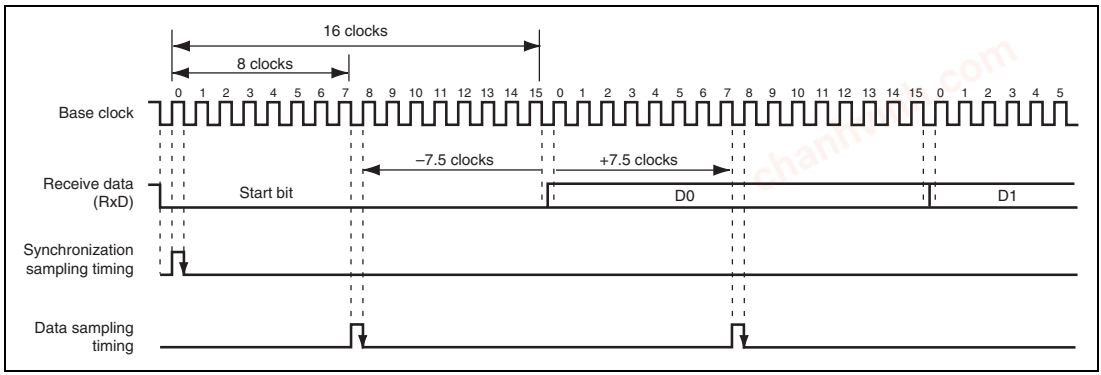


Figure 15.19 Receive Data Sampling Timing in Asynchronous Mode (Operation on a Base Clock with a Frequency 16 Times the Bit Rate)

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.

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%.

chanhvinh.com

15.6.7 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 15.6.6, Receive Data Sampling Timing and Receive Margin (Asynchronous Mode).

If the desired bit rate can be set simply by setting SCBRR and the CKS1and CKS0 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

Section 16 Renesas Serial Peripheral Interface

This LSI includes two-channel Renesas serial peripheral interfaces.

This module is capable of full-duplex 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 allow 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 Bϕ 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.

chanhvinh.com

- 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)
- Others
  Provides loop back mode
  Provides a function for disabling (initializing) this module

chanhvinh.com

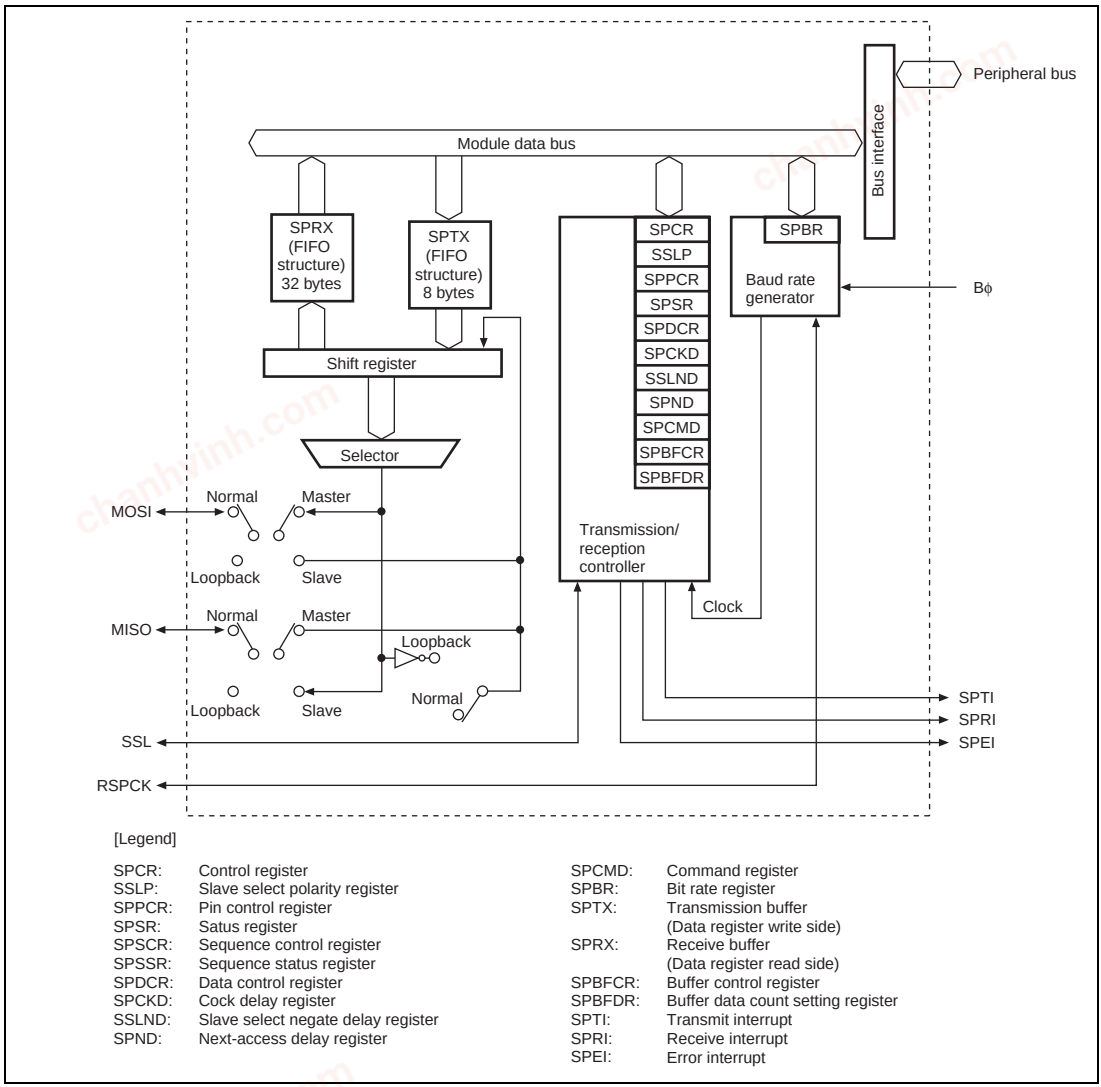


Figure 16.1 Block Diagram (for One Channel)

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

Note: In the description of the pins, the channel is omitted and pin names are described as RSPICK, 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'FFFF8000	8, 16
	Slave select polarity register_0	SSLP_0	R/W	H'00	H'FFFF8001	8, 16
	Pin control register_0	SPPCR_0	R/W	H'00	H'FFFF8002	8, 16
	Status register_0	SPSR_0	R/(W)*[2]	H'60	H'FFFF8003	8, 16
	Data register_0	SPDR_0	R/W	Undefined	H'FFFF8004	8,16,32
	Sequence control register_0	SPSCR_0	R/W	H'00	H'FFFF8008	8, 16
	Sequence status register_0	SPSSR_0	R	H'00	H'FFFF8009	8, 16
	Bit rate register_0	SPBR_0	R/W	H'FF	H'FFFF800A	8, 16
	Data control register_0	SPDCR_0	R/W	H'20	H'FFFF800B	8, 16
	Clock delay register_0	SPCKD_0	R/W	H'00	H'FFFF800C	8, 16
	Slave select negation delay register_0	SSLND_0	R/W	H'00	H'FFFF800D	8, 16
	Next-access delay register_0	SPND_0	R/W	H'00	H'FFFF800E	8
	Command register_00	SPCMD_00	R/W	H'070D	H'FFFF8010	16
	Command register_01	SPCMD_01	R/W	H'070D	H'FFFF8012	16
	Command register_02	SPCMD_02	R/W	H'070D	H'FFFF8014	16
	Command register_03	SPCMD_03	R/W	H'070D	H'FFFF8016	16
	Buffer control register_0	SPBFCR_0	R/W	H'00	H'FFFF8020	8, 16
	Buffer data count setting register_0	SPBFDR_0	R/W	H'0000	H'FFFF8022	16

chanhvinh.com

Channel	Register Name	Abbreviation*[1]	R/W	Initial Value	Address	Access Size
1	Control register_1	SPCR_1	R/W	H'00	H'FFFF8800	8, 16
	Slave select polarity register_1	SSLP_1	R/W	H'00	H'FFFF8801	8, 16
	Pin control register_1	SPPCR_1	R/W	H'00	H'FFFF8802	8, 16
	Status register_1	SPSR_1	R/(W)*[2]	H'60	H'FFF8803	8, 16
	Data register_1	SPDR_1	R/W	Undefined	H'FFFF8804	16, 32
	Sequence control register_1	SPSCR_1	R/W	H'00	H'FFFF8808	8, 16
	Sequence status register_1	SPSSR_1	R	H'00	H'FFFF8809	8, 16
	Bit rate register_1	SPBR_1	R/W	H'FF	H'FFFF880A	8, 16
	Data control register_1	SPDCR_1	R/W	H'20	H'FFFF880B	8, 16
	Clock delay register_1	SPCKD_1	R/W	H'00	H'FFFF880C	8, 16
	Slave select negation delay register_1	SSLND_1	R/W	H'00	H'FFFF880D	8, 16
	Next-access delay register_1	SPND_1	R/W	H'00	H'FFFF880E	8
	Command register_10	SPCMD_10	R/W	H'070D	H'FFFF8810	16
	Command register_11	SPCMD_11	R/W	H'070D	H'FFFF8812	16
	Command register_12	SPCMD_12	R/W	H'070D	H'FFFF8814	16
	Command register_13	SPCMD_13	R/W	H'070D	H'FFFF8816	16
	Buffer control register_1	SPBFCR_1	R/W	H'00	H'FFFF8820	8, 16
	Buffer data count setting register_1	SPBFDR_1	R/W	H'0000	H'FFFF8822	8, 16

Notes: 1. In the description of the register names, the channel is omitted.
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	MOD FEN	—	—
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	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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 pins 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 mode fault errors (see section 16.4.6, Error Detection). 0: Disables the detection of mode fault errors 1: Enables the detection of mode fault errors 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	Description
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

chanhvinh.com

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	Description
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 data from the previous serial transfer during the SSL negation period. When MOIFE is 1, this module outputs the fixed value set in the MOIFV bit to the MOSI bit. 0: MOSI output value equals final data from previous transfer 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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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	Description
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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 moved 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.
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	Description
2	MODF	0	R/(W)*	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)*	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: * 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).

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 the following bits. If data is written to the other bits, the data is not guaranteed.

- Longword: Bits 31 to 0
- Word: Bits 31 to 16
- Byte: Bits 31 to 24

When SPDR is read with the longword-, word-, or byte-access width, the receive data should be read from the following bits. If data is read from the other bits, the data is not guaranteed.

- Longword: Bits 31 to 0
- Word: Bits 31 to 16
- Byte: Bits 31 to 24

chanhvinh.com

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 controlled 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	Description
7 to 2	—	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	Description
1	SPSLN1	0	R/W	Sequence Length Specification
0	SPSLN0	0	R/W	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. 00: Sequence Length 1 — Referenced SPCMD # 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	Description
7 to 2	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
1	SPCP1	0	R	Command Pointer
0	SPCP0	0	R	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

chanhvinh.com

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 bit settings in the bits BRDV1 and BRDV0 (0, 1, 2, 3).

$$\text{Bit rate} = \frac{f(B\phi)}{2 \times (n + 1) \times 2^{N}}$$

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**	**Bϕ = 40 MHz**	**Bϕ = 48 MHz**	**Bϕ = 72 MHz**
0	0	2	20.0 Mbps*	24.0 Mbps	36.0 Mbps
1	0	4	10.0 Mbps	12.0 Mbps	18.0 Mbps
2	0	6	6.67 Mbps	8.00 Mbps	12.0 Mbps
3	0	8	5.00 Mbps	6.00 Mbps	9.0 Mbps
4	0	10	4.00 Mbps	4.80 Mbps	7.2 Mbps
5	0	12	3.33 Mbps	6.67 Mbps	6.0 Mbps
5	1	24	1.67 Mbps	2.00 Mbps	3.0 Mbps
5	2	48	833 kbps	1.00 Mbps	1.5 Mbps
5	3	96	417 kbps	500 kbps	750 kbps
255	3	4096	9.77 kbps	11.72 kbps	17.58 kbps

chanhvinh.com

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 bits other than the TXDMY 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	Description
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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
6	SPLW1	0	R/W	Access Width Specification
5	SPLW0	1	R/W	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. 10: SPDR is accessed in words. 1:1 SPDR is accessed in longwords.
4 to 0	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.

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	Description
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
1	SCKDL1	0	R/W	These bits set an RSPCK delay value when the SCKDEN bit in SPCMD is 1.
0	SCKDL0	0	R/W	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

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	Description
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
1	SLNDL1	0	R/W	These bits set an SSL negation delay when the SLNDEN bit in SPCMD is 1.
0	SLNDL0	0	R/W	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

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	Description
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 Bϕ 001: 2 RSPCK + 2 Bϕ 010: 3 RSPCK + 2 Bϕ 011: 4 RSPCK + 2 Bϕ 100: 5 RSPCK + 2 Bϕ 101: 6 RSPCK + 2 Bϕ 110: 7 RSPCK + 2 Bϕ 111: 8 RSPCK + 2 Bϕ

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 master 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	Description
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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 + 2Bϕ. 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 Bϕ 1: A next-access delay equal to SPND settings.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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
11	SPB3	0	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
10	SPB2	1	R/W	
9	SPB1	1	R/W	
8	SPB0	1	R/W	
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 all SSL signals upon completion of transfer. 1: Keeps the SSL signal level from the end of the transfer until the beginning of the next access.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
3	BRDV1	1	R/W	Bit Rate Division Setting
2	BRDV0	1	R/W	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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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	Description
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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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	Description
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/W	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/W	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 Bϕ/8	Up to Bϕ/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 to 32 bits	8 to 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

chanhvinh.com

Mode	Slave (SPI Operation)	Master (SPI Operation)
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

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*	CMOS output/Hi-Z

Note: When SSL is at the non-active level or the SPE bit in SPCR is clear 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	Final data from previous transfer
1	0	Always 0
1	1	Always 1

chanhvinh.com

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.

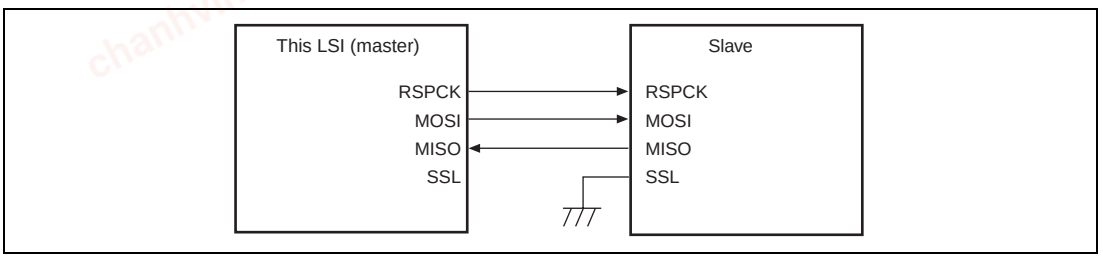


Figure 16.2 Master/Slave Configuration Example (This LSI = Master)

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).

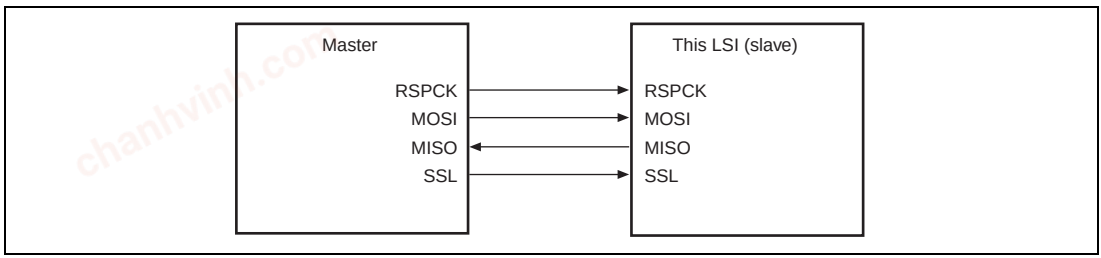


Figure 16.3 Master/Slave Configuration Example (This LSI = Slave)

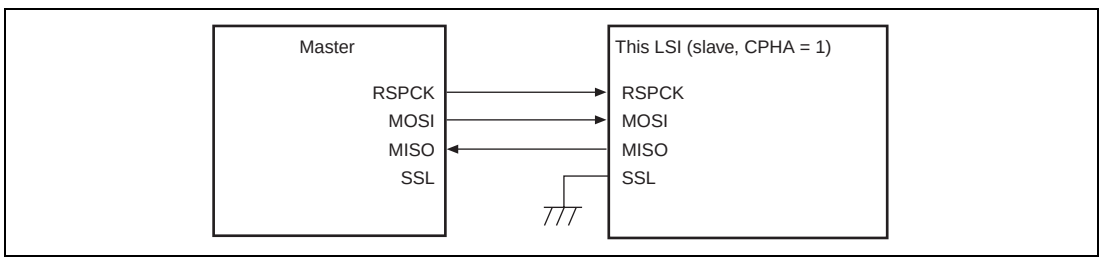


Figure 16.4 Master/Slave Configuration Example (This LSI = Slave, CPHA = 1)

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 an 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.

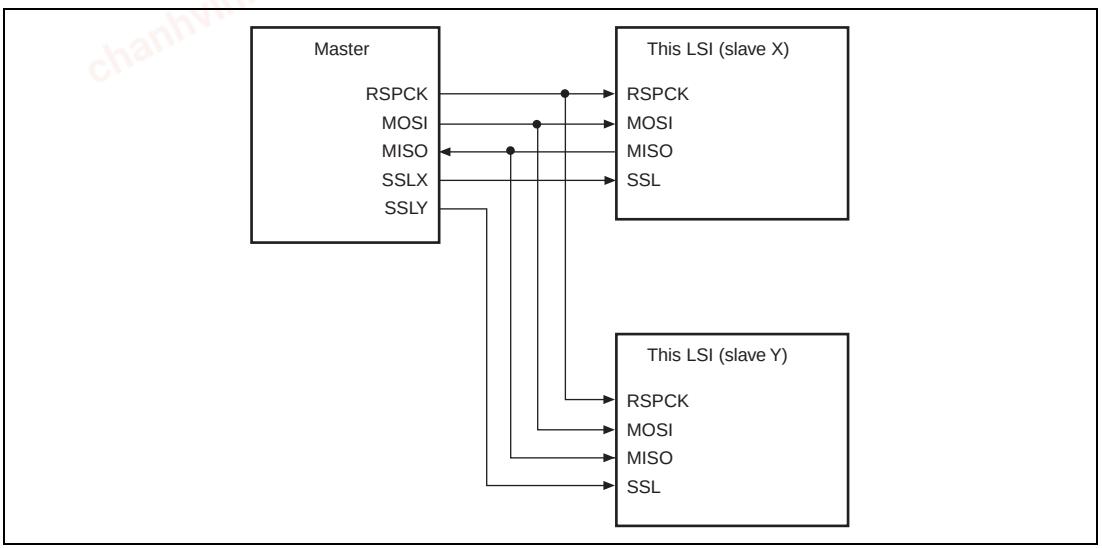


Figure 16.5 Master/Multi-Slave Configuration Example (This LSI = Slave)

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).

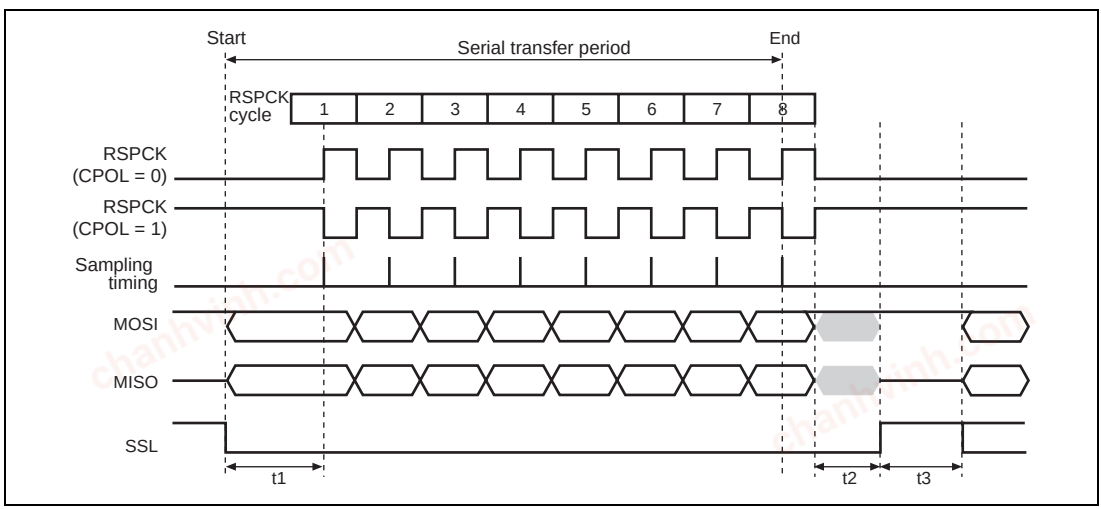


Figure 16.6 Transfer Format (CPHA = 0)

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).

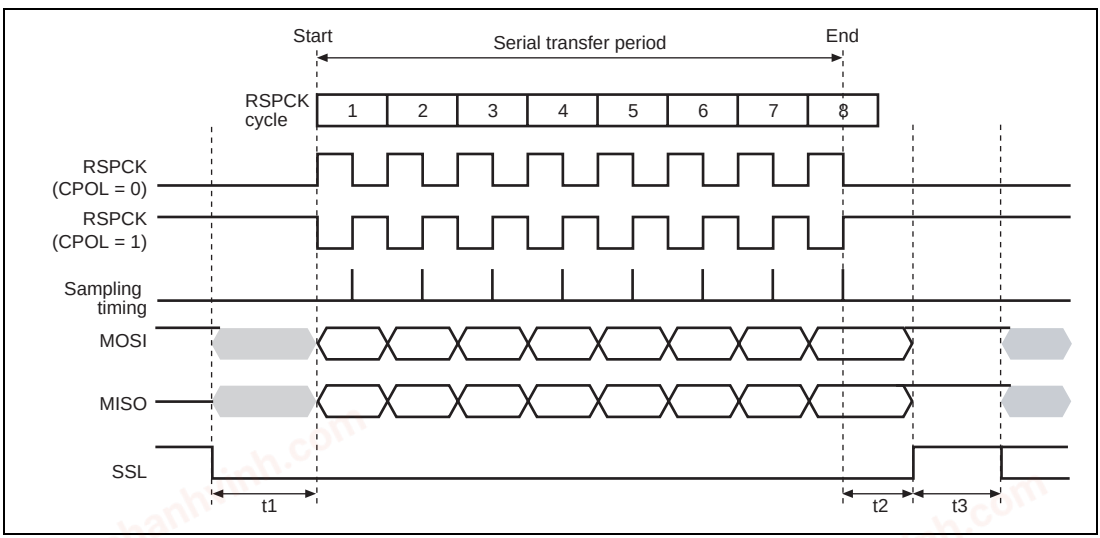


Figure 16.7 Transfer Format (CPHA = 1)

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) of the shift register, and shifts in the data from the LSB (bit 0) of 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.

chanhvinh.com

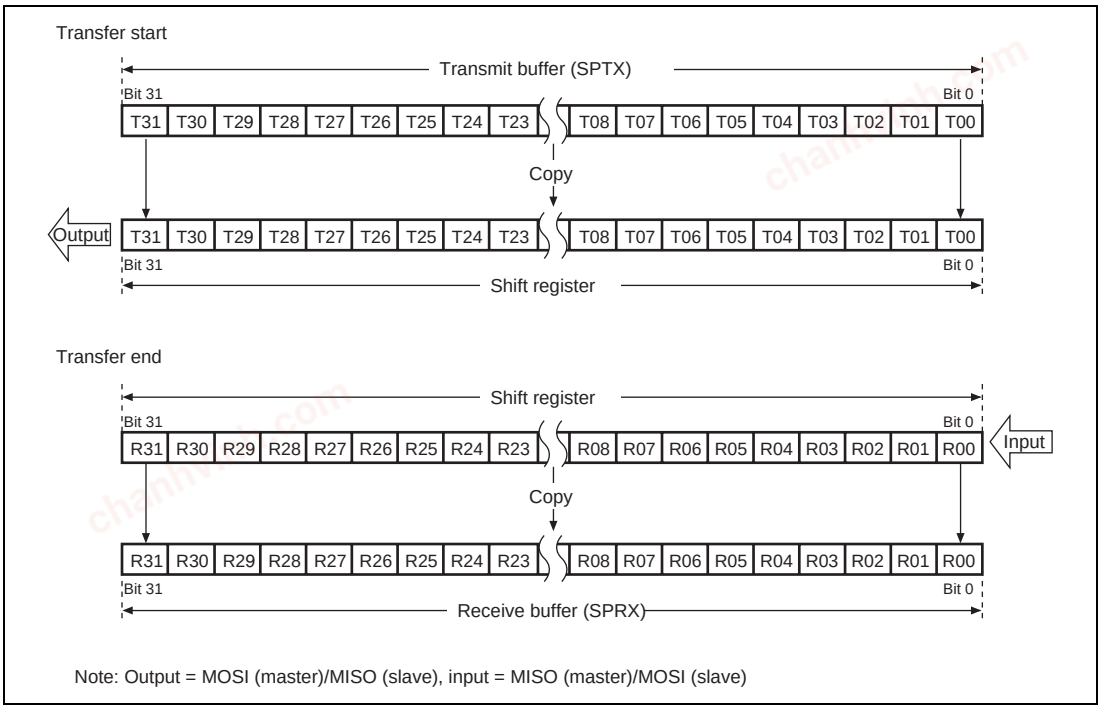


Figure 16.8 MSB First Transfer (32-Bit Data)

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 of the shift register, and shifts in the data from the LSB (bit 0) of 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 of 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.

chanhvinh.com

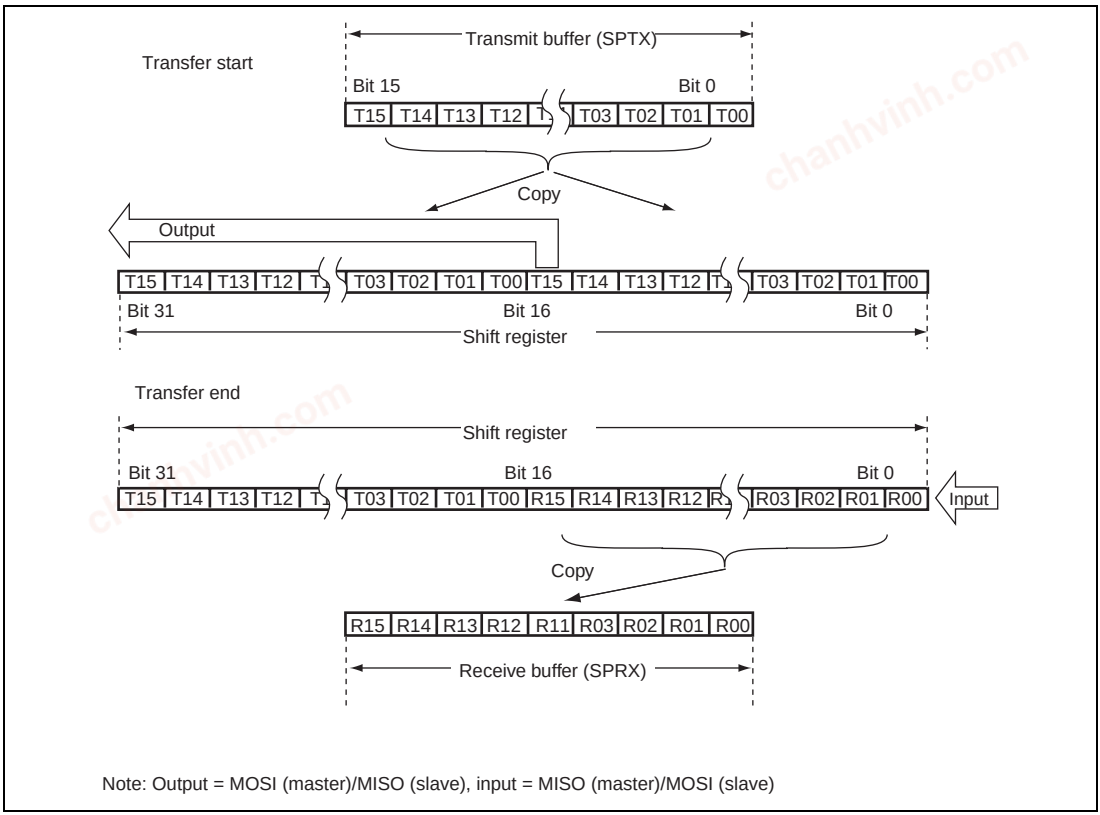


Figure 16.9 MSB First Transfer (16-Bit Data)

chanhvinh.com

(3) MSB First Transfer (8-Bit Data)

Figure 16.10 shows the operation of the transmit buffer (SPDR) 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 of the shift register, and shifts in the data from the LSB (bit 0) of 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 of 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.

chanhvinh.com

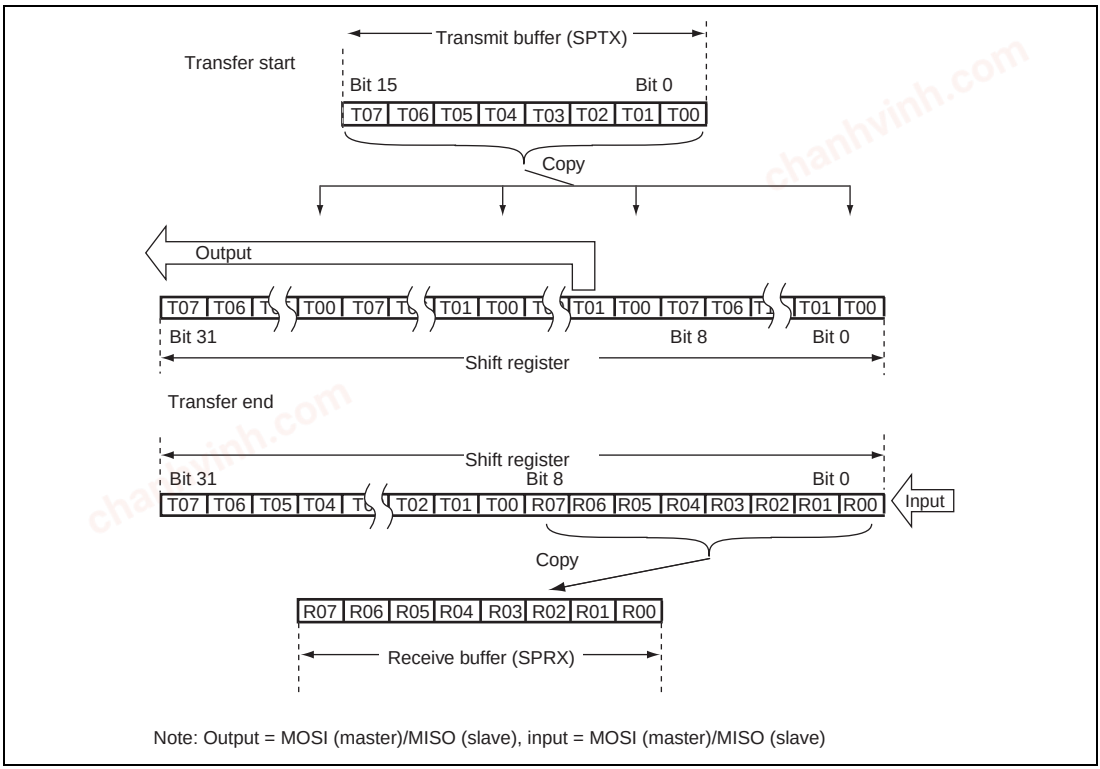


Figure 16.10 MSB First Transfer (8-Bit Data)

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) of the shift register, and shifts in the data from the LSB (bit 0) of 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.

chanhvinh.com

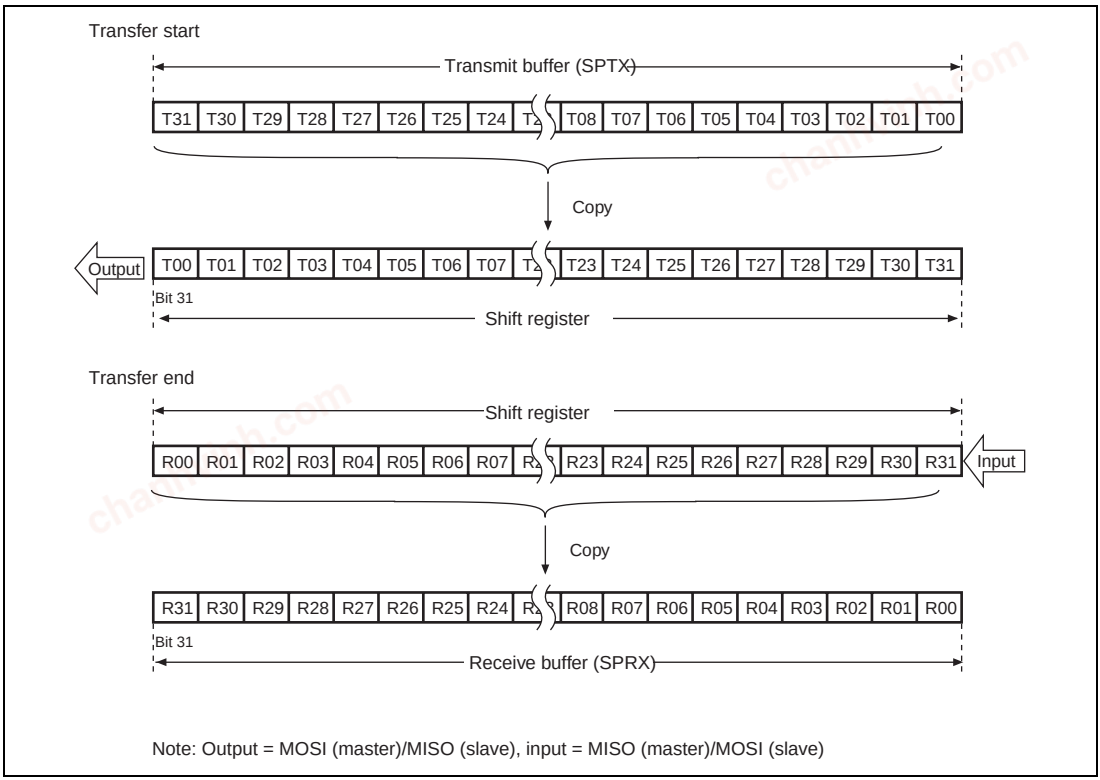


Figure 16.11 LSB First Transfer (32-Bit Data)

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) of the shift register, and shifts in the data from bit 16 of 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 of the shift register. After completion of the serial transfer, data that existed before the transfer is retained in bits 15 to 0 of 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.

chanhvinh.com

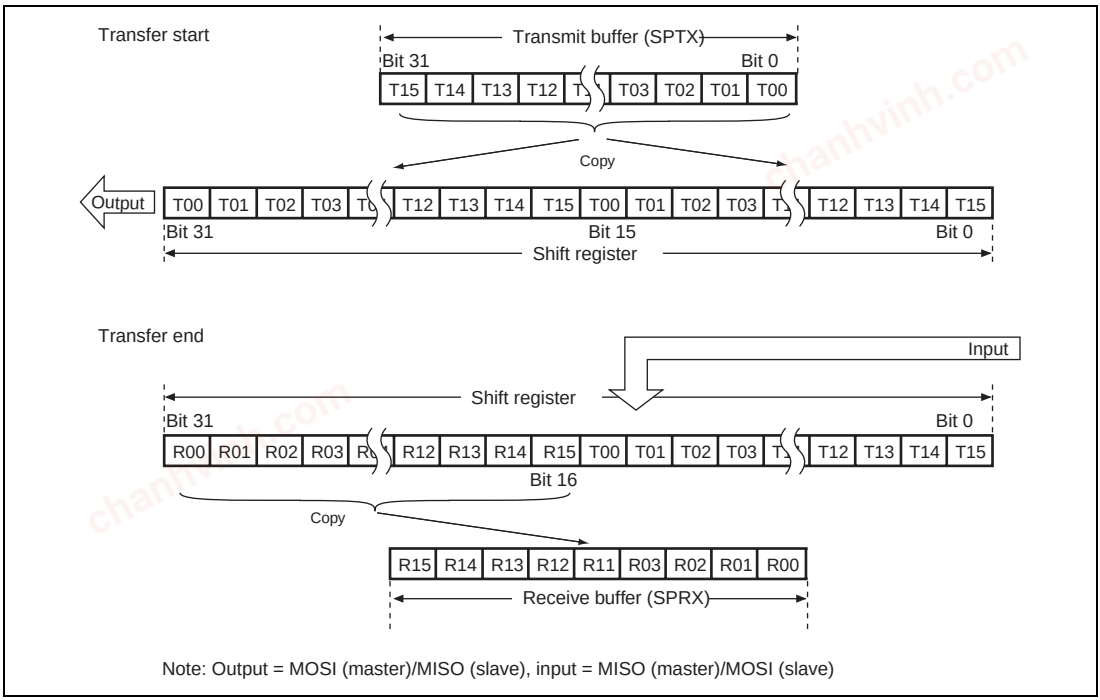


Figure 16.12 LSB First Transfer (16-Bit Data)

chanhvinh.com

(6) LSB First Transfer (8-Bit Data)

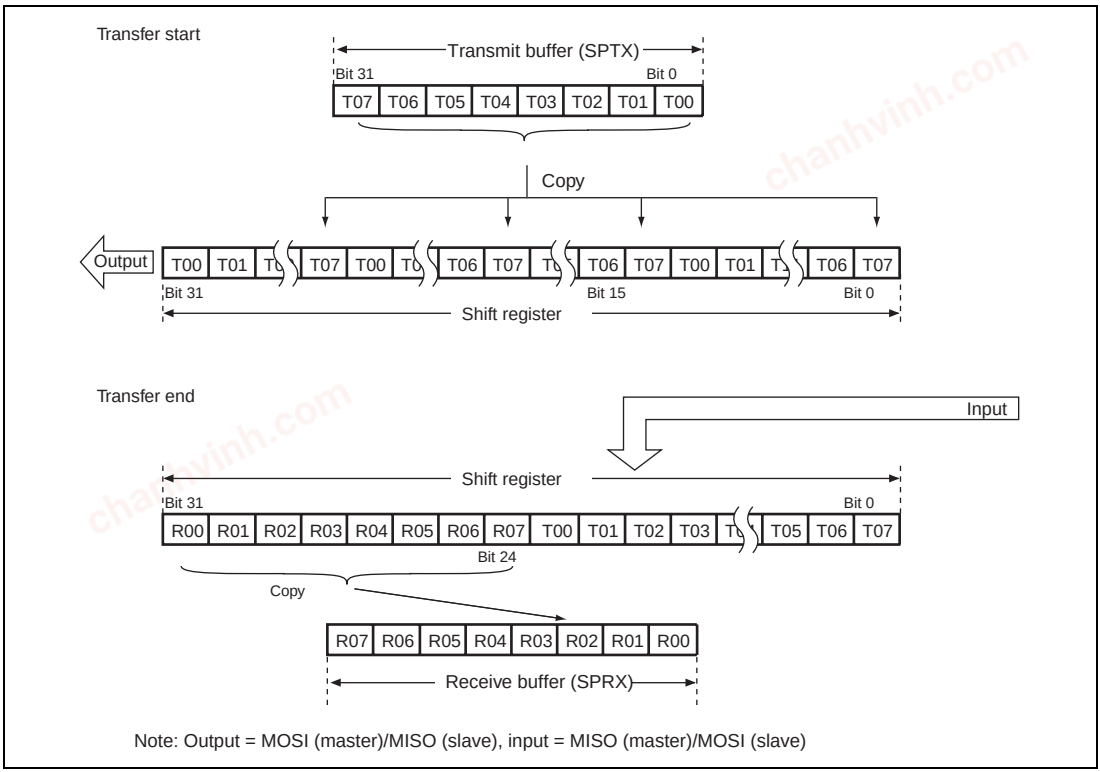
Figure 16.13 shows the operation of the transmit buffer (SPTX) and the shift register when this module performs an 8-bit data length LSB-first data transfer.

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) of the shift register, and shifts in the data from bit 24 of 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 of 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.

chanhvinh.com



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

Figure 16.13 LSB First Transfer (8-Bit Data)

chanhvinh.com

16.4.6 Error Detection

In the normal serial transfer, the data written from the data register (SPDR) to the transmit buffer 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

chanhvinh.com

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 indicates 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).

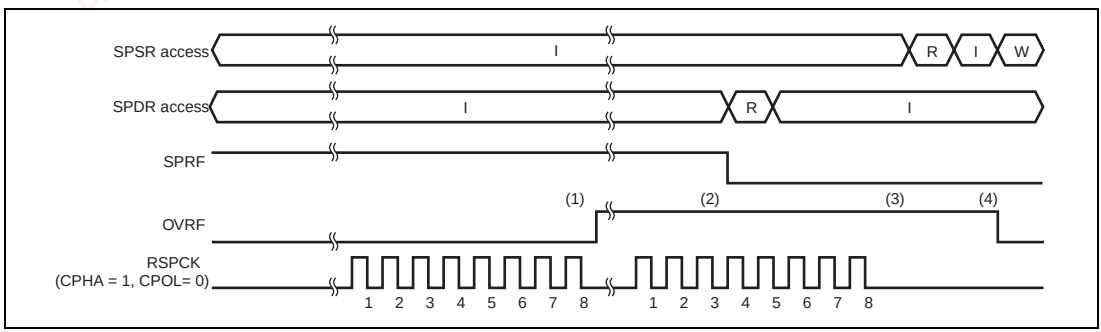


Figure 16.14 SPRF and OVRF Bit Operation Example

chanhvinh.com

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

chanhvinh.com

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 and initialization by a power-on reset.

(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.

chanhvinh.com

(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.

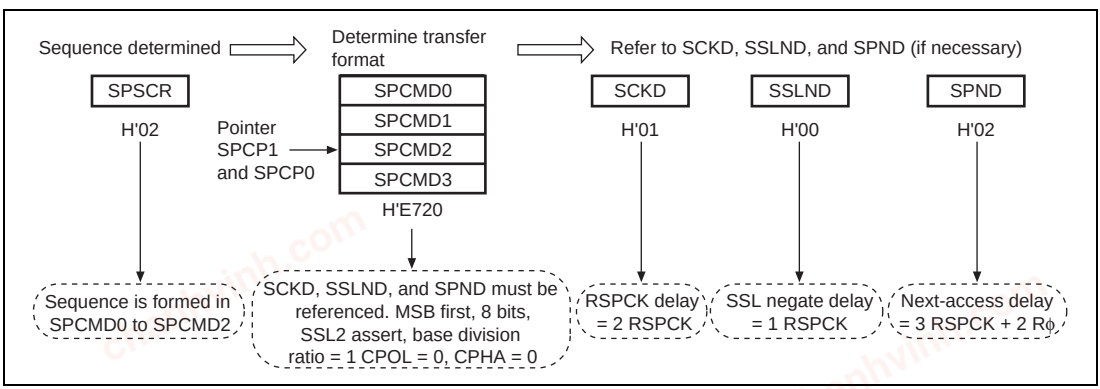


Figure 16.15 Determination Procedure of Serial Transfer Mode in Master Mode

chanhvinh.com

(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.

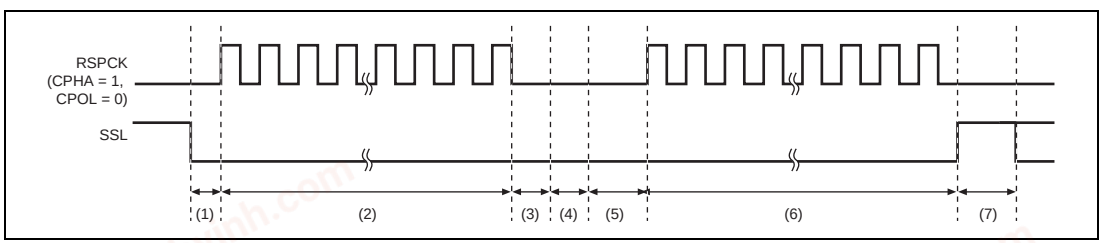


Figure 16.16 Example of Burst Transfer Operation using SSLKP Bit

chanhvinh.com

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 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

chanhvinh.com

(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 Bϕ
1	000	1 RSPCK + 2 Bϕ
	001	2 RSPCK + 2 Bϕ
	010	3 RSPCK + 2 Bϕ
	011	4 RSPCK + 2 Bϕ
	100	5 RSPCK + 2 Bϕ
	101	6 RSPCK + 2 Bϕ
	110	7 RSPCK + 2 Bϕ
	111	8 RSPCK + 2 Bϕ

(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.

chanhvinh.com

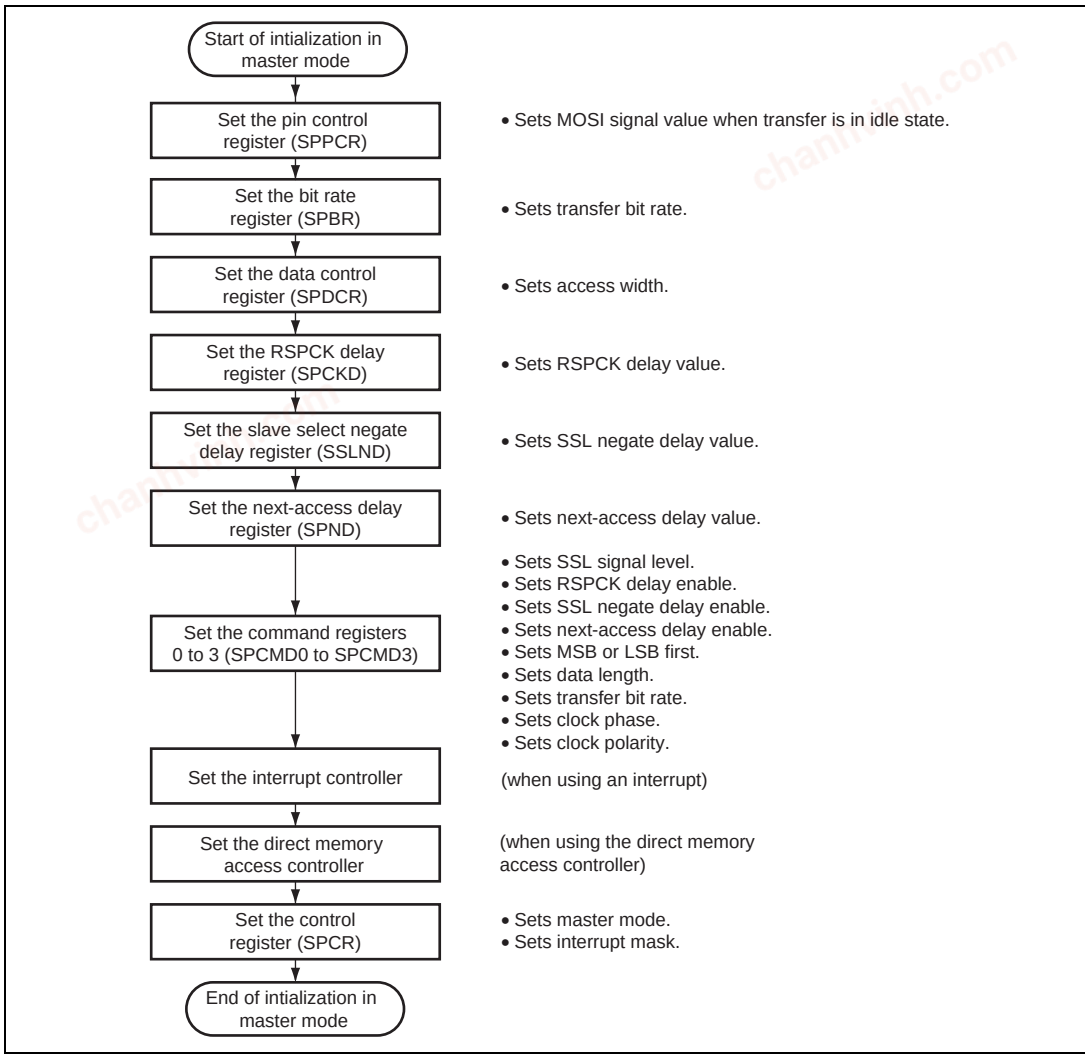


Figure 16.17 Example of Initialization Flowchart in Master Mode

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.

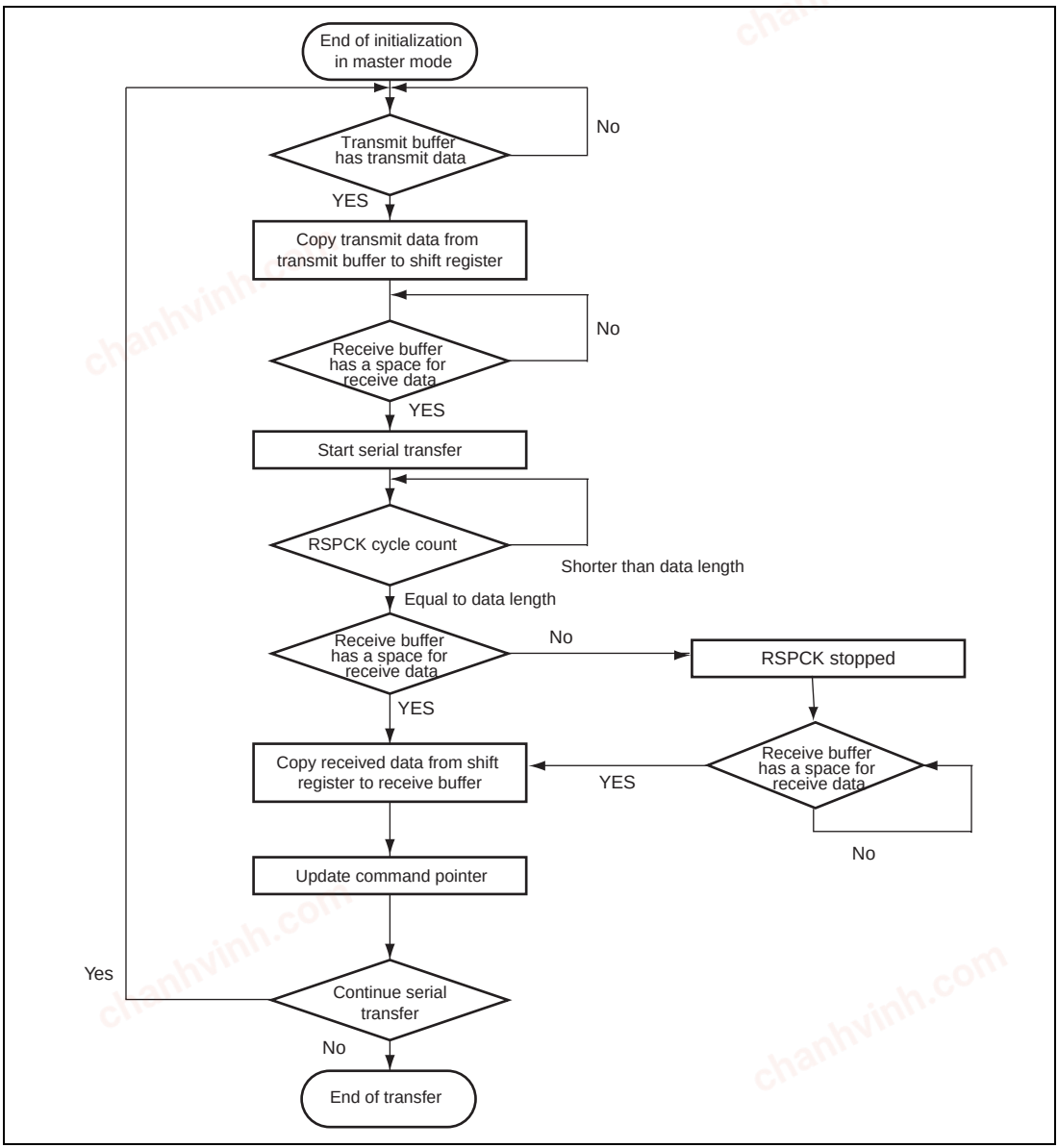


Figure 16.18 Transfer Operation Flowchart in Master Mode

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).

chanhvinh.com

(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.

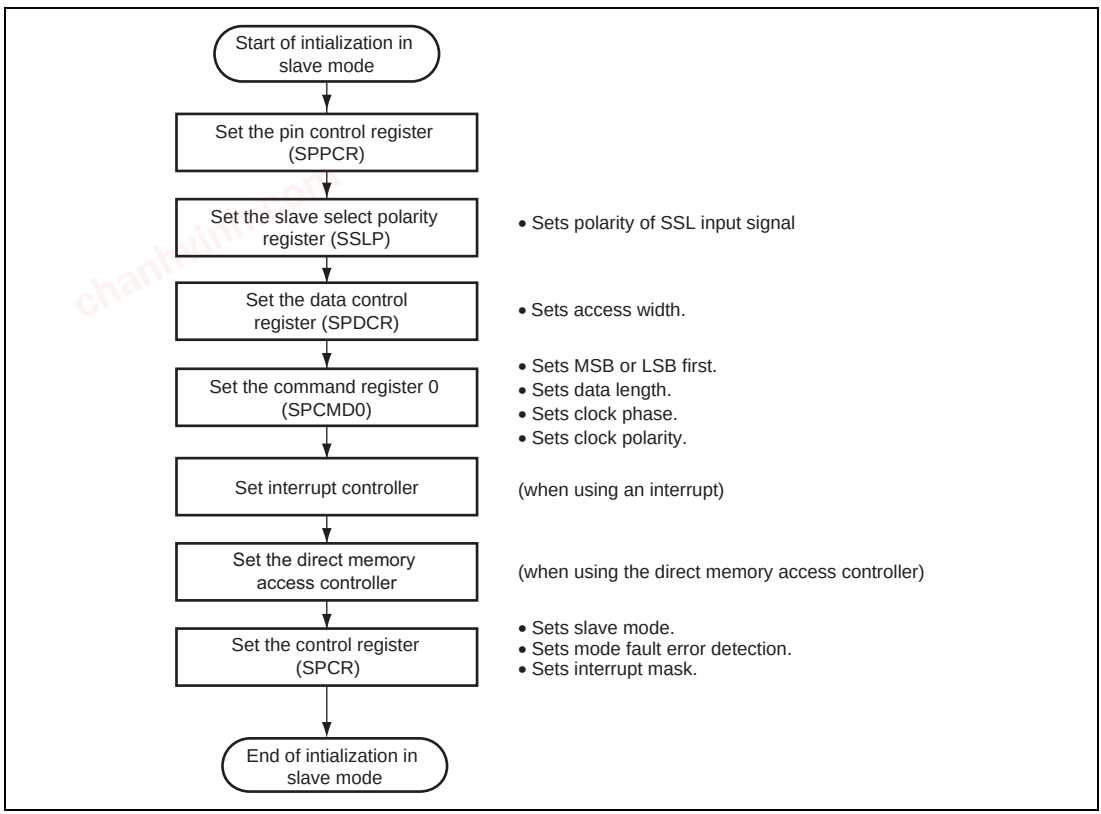


Figure 16.19 Example of Initialization Flowchart in Slave Mode

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.

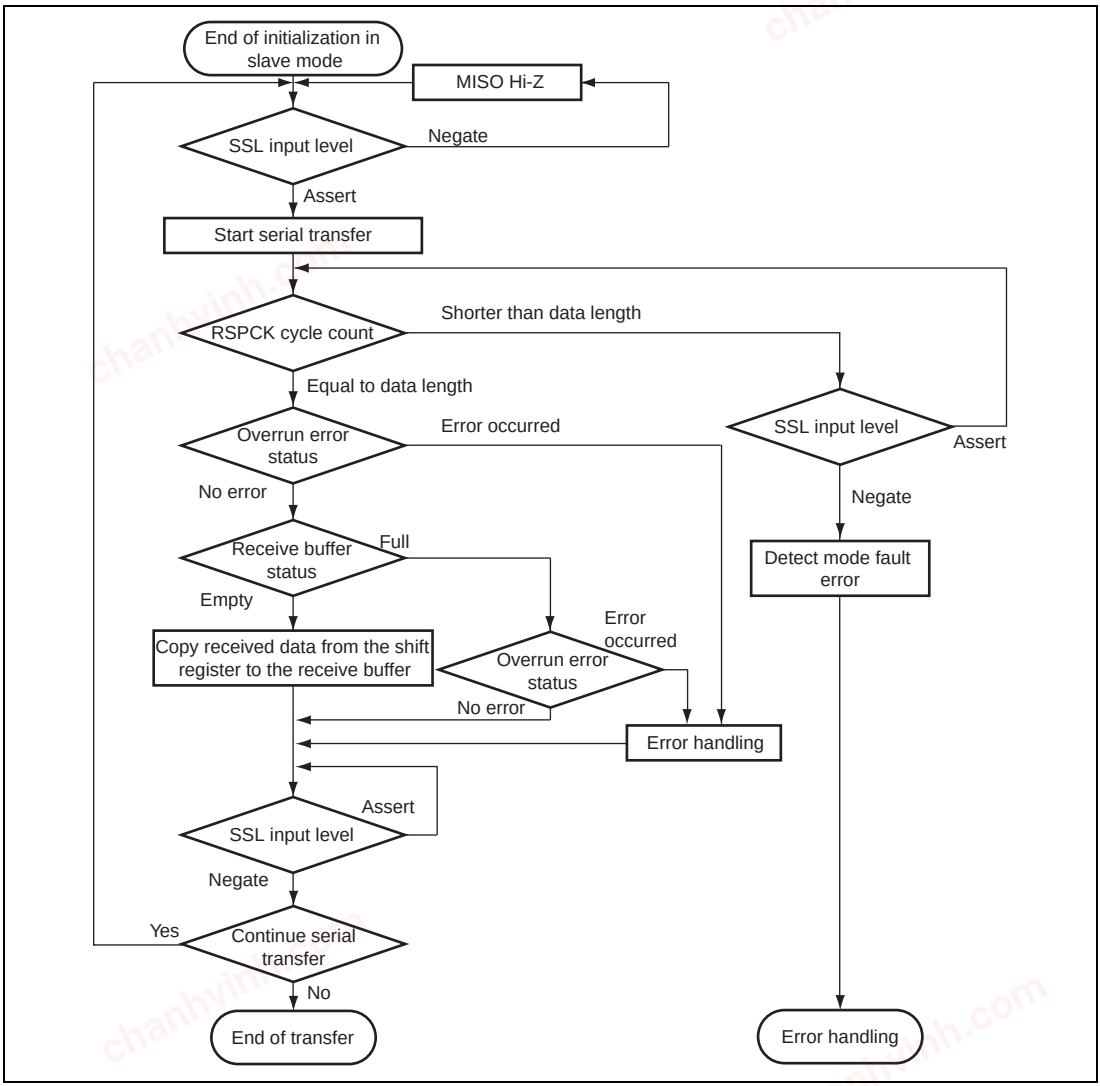


Figure 16.20 Transfer Operation Flowchart in Slave Mode (CPHA = 0)

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.

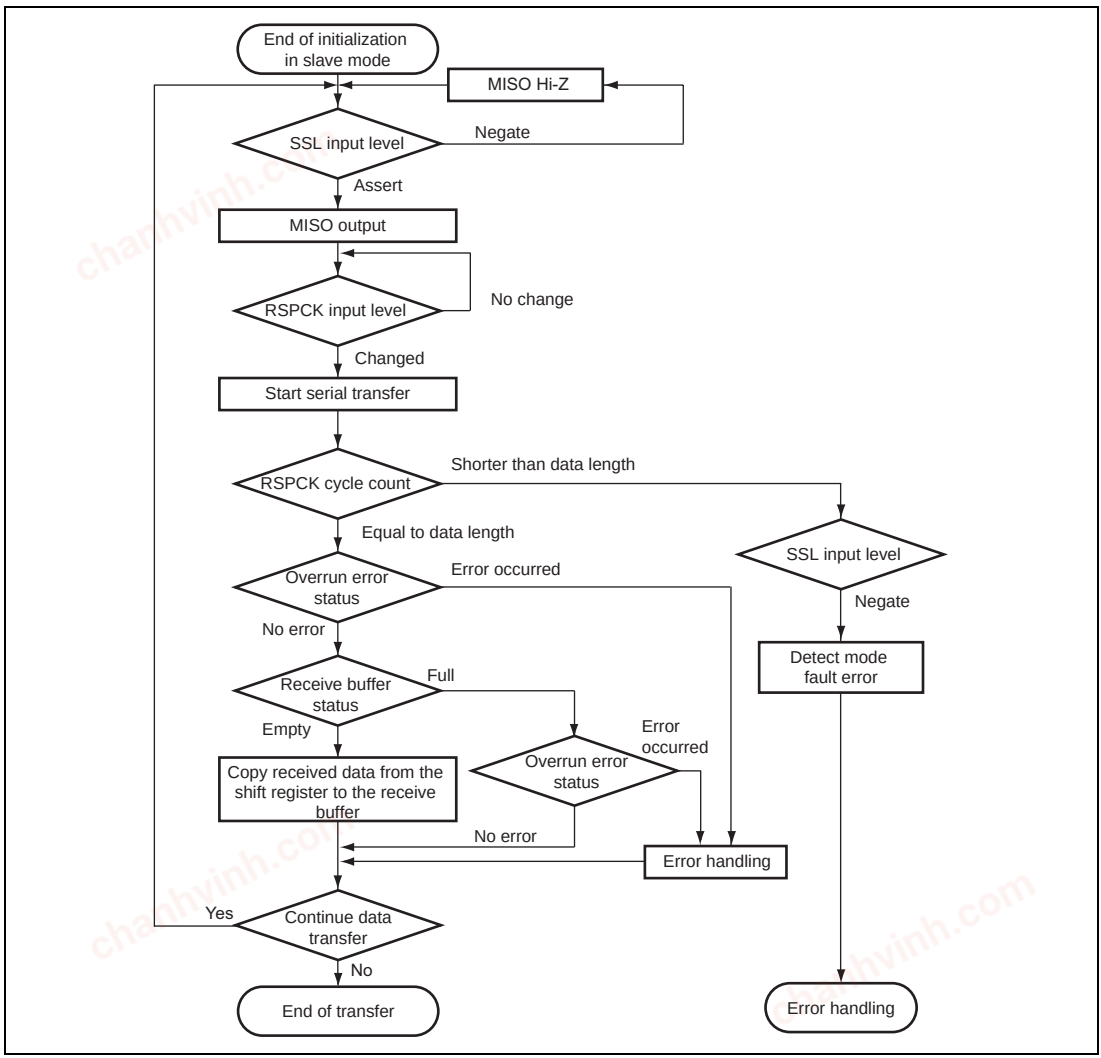


Figure 16.21 Transfer Operation Flowchart in Slave Mode (CPHA = 1)

chanhvinh.com

16.4.9 Error Handling

Figures 16.22 and 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.

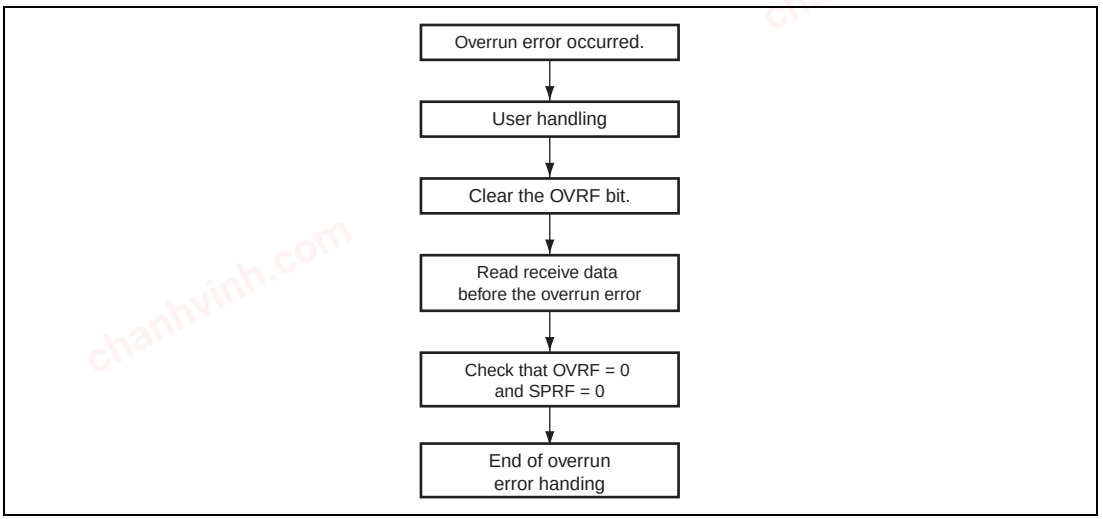


Figure 16.22 Error Handling (Overrun Error)

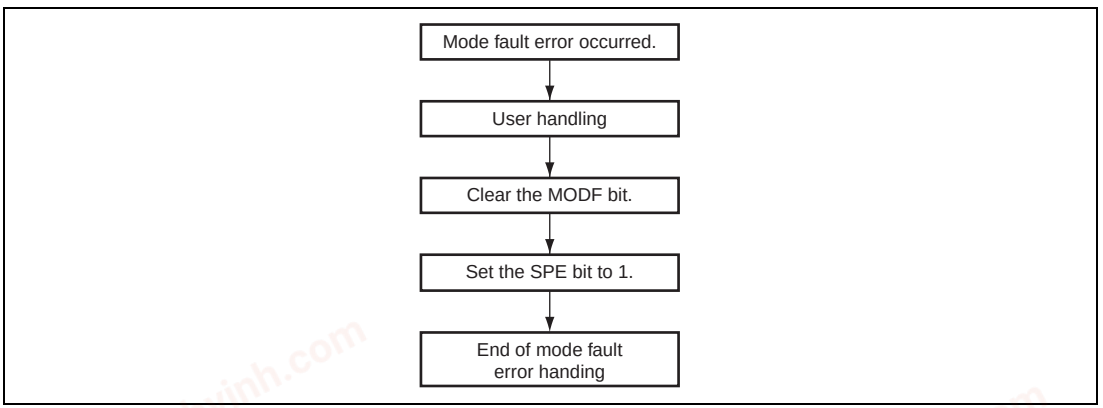


Figure 16.23 Error Handling (Mode Fault Error)

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.

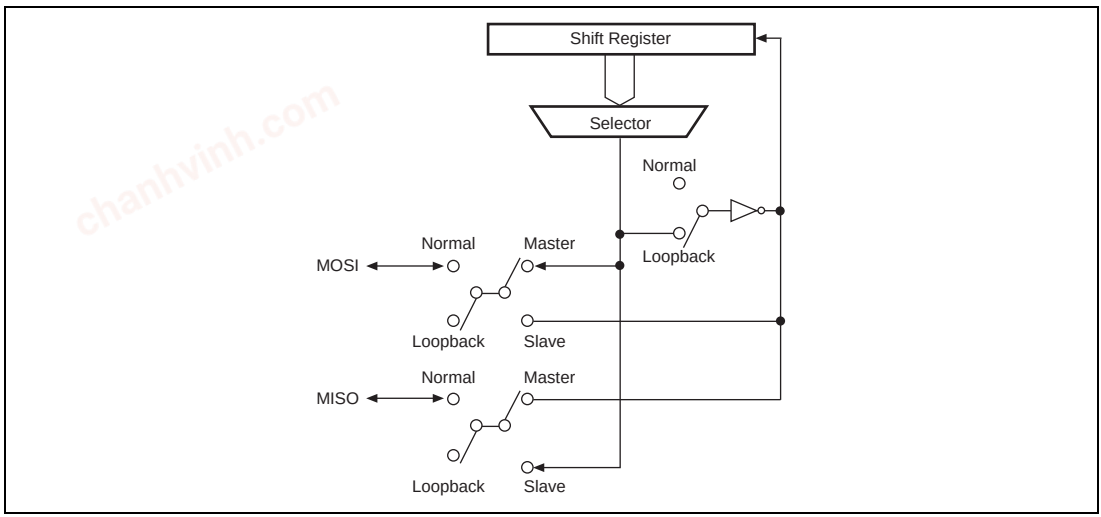


Figure 16.24 Configuration of Shift Register Input/Output Paths in Loopback Mode (Master Mode)

chanhvinh.com

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

Section 17 I²C Bus Interface 3

The I²C bus interface 3 conforms to and provides a subset of the Philips I²C (Inter-IC) bus interface functions. However, the configuration of the registers that control the I²C bus differs partly from the Philips register configuration.

The I²C bus interface 3 has three channels.

17.1 Features

- Selection of I²C format or clocked synchronous serial format
- Continuous transmission/reception

  Since the shift register, transmit data register, and receive data register are independent from each other, the continuous transmission/reception can be performed.

I²C bus format:

- Start and stop conditions generated automatically in master mode
- Selection of acknowledge output levels when receiving
- Automatic loading of acknowledge bit when transmitting
- Bit synchronization function

  In master mode, the state of SCL is monitored per bit, and the timing is synchronized automatically. If transmission/reception is not yet possible, set the SCL to low until preparations are completed.
- Six interrupt sources

  Transmit data empty (including slave-address match), transmit end, receive data full (including slave-address match), arbitration lost, NACK detection, and stop condition detection
- The direct memory access controller can be activated by a transmit-data-empty request or receive-data-full request to transfer data.
- Direct bus drive

  Two pins, SCL and SDA pins, function as NMOS open-drain outputs when the bus drive function is selected.

Clocked synchronous serial format:

- Four interrupt sources

  Transmit-data-empty, transmit-end, receive-data-full, and overrun error
- The direct memory access controller can be activated by a transmit-data-empty request or receive-data-full request to transfer data.

chanhvinh.com

Figure 17.1 shows a block diagram.

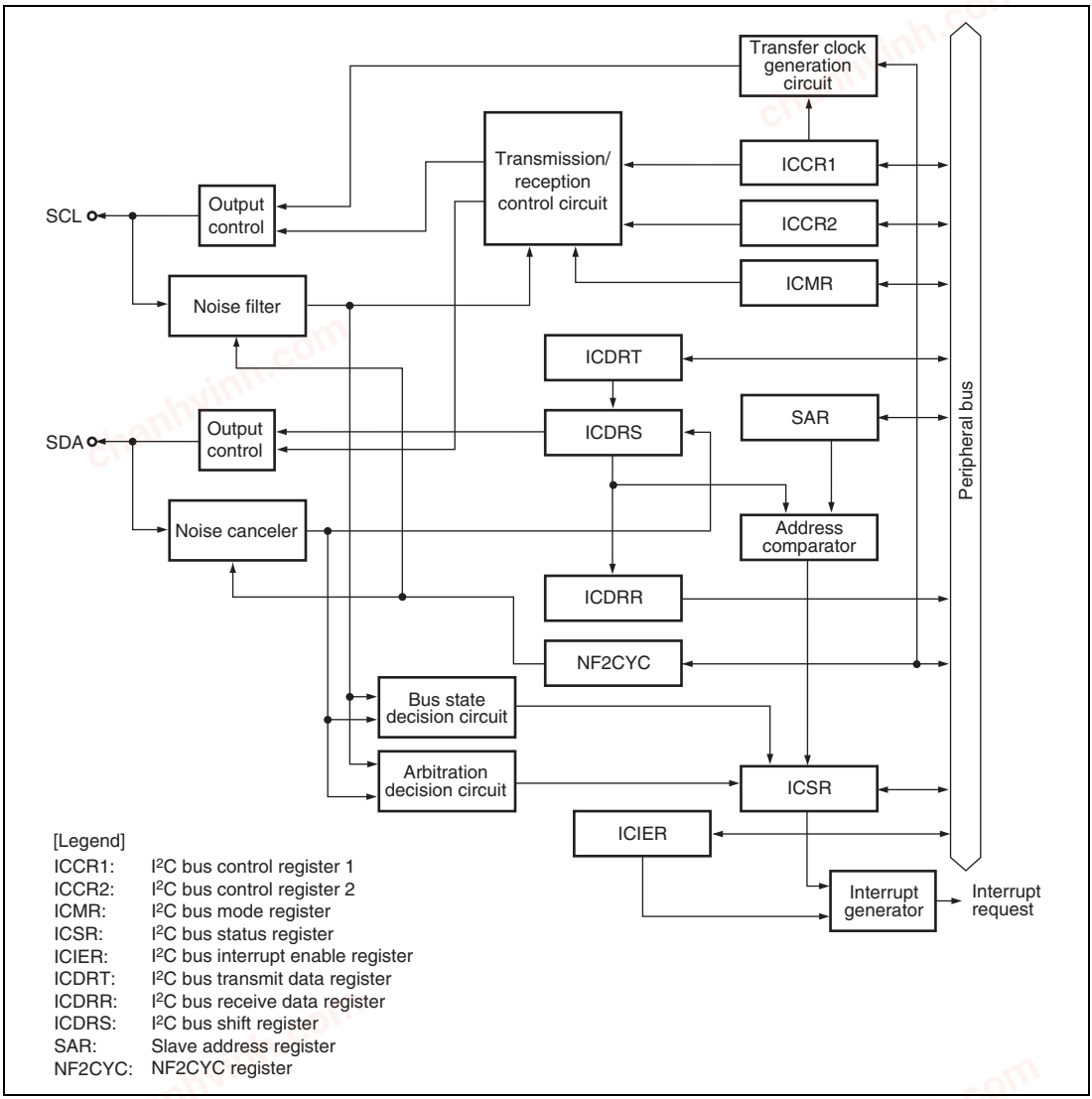


Figure 17.1 Block Diagram

chanhvinh.com

17.2 Input/Output Pins

Table 17.1 shows the pin configuration.

Table 17.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Serial clock	SCL0 to SCL2	I/O	I²C serial clock input/output
Serial data	SDA0 to SDA2	I/O	I²C serial data input/output

Figure 17.2 shows an example of I/O pin connections to external circuits.

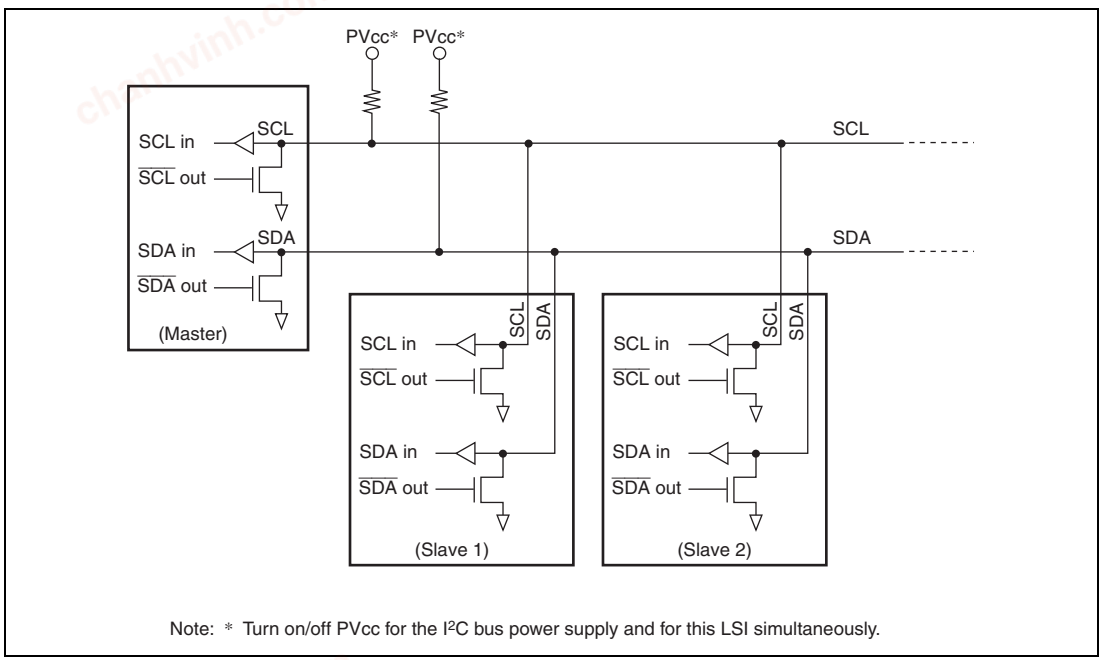


Figure 17.2 External Circuit Connections of I/O Pins

chanhvinh.com

17.3 Register Descriptions

Table 17.2 shows the register configuration.

Table 17.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
0	I²C bus control register 1	ICCR1_0	R/W	H'00	H'FFFEE000	8
	I²C bus control register 2	ICCR2_0	R/W	H'7D	H'FFFEE001	8
	I²C bus mode register	ICMR_0	R/W	H'38	H'FFFEE002	8
	I²C bus interrupt enable register	ICIER_0	R/W	H'00	H'FFFEE003	8
	I²C bus status register	ICSR_0	R/W	H'00	H'FFFEE004	8
	Slave address register	SAR_0	R/W	H'00	H'FFFEE005	8
	I²C bus transmit data register	ICDRT_0	R/W	H'FF	H'FFFEE006	8
	I²C bus receive data register	ICDRR_0	R/W	H'FF	H'FFFEE007	8
	NF2CYC register	NF2CYC_0	R/W	H'00	H'FFFEE008	8
1	I²C bus control register 1	ICCR1_1	R/W	H'00	H'FFFEE400	8
	I²C bus control register 2	ICCR2_1	R/W	H'7D	H'FFFEE401	8
	I²C bus mode register	ICMR_1	R/W	H'38	H'FFFEE402	8
	I²C bus interrupt enable register	ICIER_1	R/W	H'00	H'FFFEE403	8
	I²C bus status register	ICSR_1	R/W	H'00	H'FFFEE404	8
	Slave address register	SAR_1	R/W	H'00	H'FFFEE405	8
	I²C bus transmit data register	ICDRT_1	R/W	H'FF	H'FFFEE406	8
	I²C bus receive data register	ICDRR_1	R/W	H'FF	H'FFFEE407	8
	NF2CYC register	NF2CYC_1	R/W	H'00	H'FFFEE408	8
2	I²C bus control register 1	ICCR1_2	R/W	H'00	H'FFFEE800	8
	I²C bus control register 2	ICCR2_2	R/W	H'7D	H'FFFEE801	8
	I²C bus mode register	ICMR_2	R/W	H'38	H'FFFEE802	8
	I²C bus interrupt enable register	ICIER_2	R/W	H'00	H'FFFEE803	8
	I²C bus status register	ICSR_2	R/W	H'00	H'FFFEE804	8
	Slave address register	SAR_2	R/W	H'00	H'FFFEE805	8
	I²C bus transmit data register	ICDRT_2	R/W	H'FF	H'FFFEE806	8
	I²C bus receive data register	ICDRR_2	R/W	H'FF	H'FFFEE807	8
	NF2CYC register	NF2CYC_2	R/W	H'00	H'FFFEE808	8

chanhvinh.com

17.3.1 I²C Bus Control Register 1 (ICCR1)

ICCR1 is an 8-bit readable/writable register that enables or disables the I²C bus interface 3, controls transmission or reception, and selects master or slave mode, transmission or reception, and transfer clock frequency in master mode.

Bit:	7	6	5	4	3	2	1	0
	ICE	RCVD	MST	TRS	CKS[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	ICE	0	R/W	I²C Bus Interface 3 Enable 0: SCL and SDA output is disabled. (Input to SCL and SDA is enabled.) 1: This bit is enabled for transfer operations. (SCL and SDA pins are bus drive state.)
6	RCVD	0	R/W	Reception Disable Enables or disables the next operation when TRS is 0 and ICDRR is read. 0: Enables next reception 1: Disables next reception

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
5	MST	0	R/W	Master/Slave Select
4	TRS	0	R/W	Transmit/Receive Select In master mode with the I²C bus format, when arbitration is lost, MST and TRS are both reset by hardware, causing a transition to slave receive mode. Modification of the TRS bit should be made between transfer frames. When seven bits after the start condition is issued in slave receive mode match the slave address set to SAR and the 8th bit is set to 1, TRS is automatically set to 1. If an overrun error occurs in master receive mode with the clocked synchronous serial format, MST is cleared and the mode changes to slave receive mode. Operating modes are described below according to MST and TRS combination. When clocked synchronous serial format is selected and MST = 1, clock is output. 00: Slave receive mode 01: Slave transmit mode 10: Master receive mode 11: Master transmit mode
3 to 0	CKS[3:0]	0000	R/W	Transfer Clock Select These bits should be set according to the necessary transfer rate (table 17.3) in master mode.

chanhvinh.com

Table 17.3 Transfer Rate

NF2CYC	ICCR1					Transfer Rate (kHz)				
Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		Pϕ = 20.0 MHz	Pϕ = 24.0 MHz	Pϕ = 32.0 MHz	Pϕ = 36.0 MHz	
CKS4*[1]	CKS3	CKS2	CKS1	CKS0	Clock					Remarks
0	0	0	0	0	Pϕ/44	455	545	727	818	
				1	Pϕ/52	385	462	615	692	
			1	0	Pϕ/64	313	375	500	563	
				1	Pϕ/72	278	333	444	500	
		1	0	0	Pϕ/84	238	286	381	429	
				1	Pϕ/92	217	261	348	391	
			1	0	Pϕ/100	200	240	320	360	
				1	Pϕ/108	185	222	296	333	
	1	0	0	0	Pϕ/176	114	136	182	205	
				1	Pϕ/208	96.2	115	154	173	
			1	0	Pϕ/256	78.1	93.8	125	141	
				1	Pϕ/288	69.4	83.3	111	125	
		1	0	0	Pϕ/336	59.5	71.4	95.2	107	
				1	Pϕ/368	54.3	65.2	87.0	97.8	
			1	0	Pϕ/400	50.0	60.0	80.0	90.0	
				1	Pϕ/432	46.3	55.6	74.1	83.3	
1	0	0	0	0	Pϕ/352	56.8	68.2	90.9	102	*[2]
				1	Pϕ/416	48.1	57.7	76.9	86.5	
			1	0	Pϕ/512	39.1	46.9	62.5	70.3	
				1	Pϕ/576	34.7	41.7	55.6	62.5	
		1	0	0	Pϕ/672	29.8	35.7	47.6	53.6	
				1	Pϕ/736	27.2	32.6	43.5	48.9	
			1	0	Pϕ/800	25.0	30.0	40.0	45.0	
				1	Pϕ/864	23.1	27.8	37.0	41.7	
	1	0	0	0	Pϕ/704	28.4	34.1	45.5	51.1	
				1	Pϕ/832	24.0	28.8	38.5	43.3	
			1	0	Pϕ/1024	19.5	23.4	31.3	35.2	
				1	Pϕ/1152	17.4	20.8	27.8	31.3	
		1	0	0	Pϕ/1344	14.9	17.9	23.8	26.8	
				1	Pϕ/1472	13.6	16.3	21.7	24.5	
			1	0	Pϕ/1600	12.5	15.0	20.0	22.5	
				1	Pϕ/1728	11.6	13.9	18.5	20.8	

Notes: The settings should satisfy external specifications.

1. This bit is reserved for 1-Mbyte version.
2. These settings are valid only in 640-Kbyte version.

chanhvinh.com

17.3.2 I²C Bus Control Register 2 (ICCR2)

ICCR2 is an 8-bit readable/writable register that issues start/stop conditions, manipulates the SDA pin, monitors the SCL pin, and controls reset in the control part of the I²C bus.

Bit:	7	6	5	4	3	2	1	0
	BBSY	SCP	SDAO	SDAOP	SCLO	-	IICRST	-
Initial value:	0	1	1	1	1	1	0	1
R/W:	R/W	R/W	R/W	R/W	R	R	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
7	BBSY	0	R/W	Bus Busy Enables to confirm whether the I²C bus is occupied or released and to issue start/stop conditions in master mode. With the clocked synchronous serial format, this bit is always read as 0. With the I²C bus format, this bit is set to 1 when the SDA level changes from high to low under the condition of SCL = high, assuming that the start condition has been issued. This bit is cleared to 0 when the SDA level changes from low to high under the condition of SCL = high, assuming that the stop condition has been issued. Write 1 to BBSY and 0 to SCP to issue a start condition. Follow this procedure when also re-transmitting a start condition. Write 0 in BBSY and 0 in SCP to issue a stop condition.
6	SCP	1	R/W	Start/Stop Issue Condition Disable Controls the issue of start/stop conditions in master mode. To issue a start condition, write 1 in BBSY and 0 in SCP. A retransmit start condition is issued in the same way. To issue a stop condition, write 0 in BBSY and 0 in SCP. This bit is always read as 1. Even if 1 is written to this bit, the data will not be stored.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
5	SDAO	1	R/W	SDA Output Value Control This bit is used with SDAOP when modifying output level of SDA. This bit should not be manipulated during transfer. 0: When reading, SDA pin outputs low. When writing, SDA pin is changed to output low. 1: When reading, SDA pin outputs high. When writing, SDA pin is changed to output Hi-Z (outputs high by external pull-up resistance).
4	SDAOP	1	R/W	SDAO Write Protect Controls change of output level of the SDA pin by modifying the SDAO bit. To change the output level, clear SDAO and SDAOP to 0 or set SDAO to 1 and clear SDAOP to 0. This bit is always read as 1.
3	SCLO	1	R	SCL Output Level Monitors SCL output level. When SCLO is 1, SCL pin outputs high. When SCLO is 0, SCL pin outputs low.
2	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
1	IICRST	0	R/W	Control Part Reset Resets bits BC[2:0] in ICMR and internal circuits. If this bit is set to 1 when hang-up occurs because of communication failure during I²C bus operation, bits BC[2:0] in ICMR and internal circuits can be reset.
0	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.

chanhvinh.com

17.3.3 I²C Bus Mode Register (ICMR)

ICMR is an 8-bit readable/writable register that selects whether the MSB or LSB is transferred first, performs master mode wait control, and selects the transfer bit count.

Bits BC[2:0] are initialized to H'0 by the IICRST bit in ICCR2.

Bit:	7	6	5	4	3	2	1	0
	MLS	-	-	-	BCWP	BC[2:0]		
Initial value:	0	0	1	1	1	0	0	0
R/W:	R/W	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	MLS	0	R/W	MSB-First/LSB-First Select 0: MSB-first 1: LSB-first Set this bit to 0 when the I²C bus format is used.
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5, 4	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
3	BCWP	1	R/W	BC Write Protect Controls the BC[2:0] modifications. When modifying the BC[2:0] bits, this bit should be cleared to 0. In clocked synchronous serial mode, the BC[2:0] bits should not be modified. 0: When writing, values of the BC[2:0] bits are set. 1: When reading, 1 is always read. When writing, settings of the BC[2:0] bits are invalid.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	BC[2:0]	000	R/W	Bit Counter These bits specify the number of bits to be transferred next. When read, the remaining number of transfer bits is indicated. With the I²C bus format, the data is transferred with one addition acknowledge bit. Should be made between transfer frames. If these bits are set to a value other than B'000, the setting should be made while the SCL pin is low. The bit value returns to B'000 automatically at the end of a data transfer including the acknowledge bit. And the value becomes B'111 automatically after the stop condition detection. These bits are cleared by a power-on reset and in software standby mode and module standby mode. These bits are also cleared by setting the IICRST bit of ICCR2 to 1. With the clocked synchronous serial format, these bits should not be modified. I²C Bus Format — Clocked Synchronous Serial Format 000: 9 bits — 000: 8 bits 001: 2 bits — 001: 1 bit 010: 3 bits — 010: 2 bits 011: 4 bits — 011: 3 bits 100: 5 bits — 100: 4 bits 101: 6 bits — 101: 5 bits 110: 7 bits — 110: 6 bits 111: 8 bits — 111: 7 bits

chanhvinh.com

17.3.4 I²C Bus Interrupt Enable Register (ICIER)

ICIER is an 8-bit readable/writable register that enables or disables interrupt sources and acknowledge bits, sets acknowledge bits to be transferred, and confirms acknowledge bits received.

Bit:	7	6	5	4	3	2	1	0
	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT
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/W

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When the TDRE bit in ICSR is set to 1 or 0, this bit enables or disables the transmit data empty interrupt (TXI). 0: Transmit data empty interrupt request (TXI) is disabled. 1: Transmit data empty interrupt request (TXI) is enabled.
6	TEIE	0	R/W	Transmit End Interrupt Enable Enables or disables the transmit end interrupt (TEI) at the rising of the ninth clock while the TDRE bit in ICSR is 1. TEI can be canceled by clearing the TEND bit or the TEIE bit to 0. 0: Transmit end interrupt request (TEI) is disabled. 1: Transmit end interrupt request (TEI) is enabled.
5	RIE	0	R/W	Receive Interrupt Enable Enables or disables the receive data full interrupt (RXI) when receive data is transferred from ICDRS to ICDRR while RDRF in ICSR is set to 1. RXI can be canceled by clearing the RDRF or RIE bit to 0. 0: Receive data full interrupt request (RXI) are disabled. 1: Receive data full interrupt request (RXI) are enabled.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
4	NAKIE	0	R/W	NACK Receive Interrupt Enable Enables or disables the NACK detection and arbitration lost/overrun error interrupt request (NAKI) when NACKF or AL/OVE in ICSR is set to 1. NAKI can be canceled by clearing the NACKF, AL/OVE, or NAKIE bit to 0. 0: NACK receive interrupt request (NAKI) is disabled. 1: NACK receive interrupt request (NAKI) is enabled.
3	STIE	0	R/W	Stop Condition Detection Interrupt Enable Enables or disables the stop condition detection interrupt request (STPI) when the STOP bit in ICSR is set. 0: Stop condition detection interrupt request (STPI) is disabled. 1: Stop condition detection interrupt request (STPI) is enabled.
2	ACKE	0	R/W	Acknowledge Bit Judgment Select 0: The value of the receive acknowledge bit is ignored, and continuous transfer is performed. 1: If the receive acknowledge bit is 1, continuous transfer is halted.
1	ACKBR	0	R	Receive Acknowledge In transmit mode, this bit stores the acknowledge data that are returned by the receive device. This bit cannot be modified. This bit can be canceled by setting the BBSY bit in ICCR2 to 1. 0: Receive acknowledge = 0 1: Receive acknowledge = 1
0	ACKBT	0	R/W	Transmit Acknowledge In receive mode, this bit specifies the bit to be sent at the acknowledge timing. 0: 0 is sent at the acknowledge timing. 1: 1 is sent at the acknowledge timing.

chanhvinh.com

17.3.5 I²C Bus Status Register (ICSR)

ICSR is an 8-bit readable/writable register that confirms interrupt request flags and their status.

Bit:	7	6	5	4	3	2	1	0
	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ
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	TDRE	0	R/W	Transmit Data Register Empty [Clearing conditions] • When 0 is written in TDRE after reading TDRE = 1 • When data is written to ICDRT [Setting conditions] • When data is transferred from ICDRT to ICDRS and ICDRT becomes empty • When TRS is set • When the start condition (including retransmission) is issued • When slave mode is changed from receive mode to transmit mode
6	TEND	0	R/W	Transmit End [Clearing conditions] • When 0 is written in TEND after reading TEND = 1 • When data is written to ICDRT [Setting conditions] • When the ninth clock of SCL rises with the I²C bus format while the TDRE flag is 1 • When the final bit of transmit frame is sent with the clocked synchronous serial format

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
5	RDRF	0	R/W	Receive Data Full [Clearing conditions] • When 0 is written in RDRF after reading RDRF = 1 • When ICDRR is read [Setting condition] • When a receive data is transferred from ICDRS to ICDRR
4	NACKF	0	R/W	No Acknowledge Detection Flag [Clearing condition] • When 0 is written in NACKF after reading NACKF = 1 [Setting condition] • When no acknowledge is detected from the receive device in transmission while the ACKE bit in ICIER is 1
3	STOP	0	R/W	Stop Condition Detection Flag [Clearing condition] • When 0 is written in STOP after reading STOP = 1 [Setting condition] • When a stop condition is detected after frame transfer is completed

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
2	AL/OVE	0	R/W	Arbitration Lost Flag/Overrun Error Flag Indicates that arbitration was lost in master mode with the I²C bus format and that the final bit has been received while RDRF = 1 with the clocked synchronous format. When two or more master devices attempt to seize the bus at nearly the same time, if the I²C bus interface 3 detects data differing from the data it sent, it sets AL to 1 to indicate that the bus has been occupied by another master. [Clearing condition] • When 0 is written in AL/OVE after reading AL/OVE = 1 [Setting conditions] • If the internal SDA and SDA pin disagree at the rise of SCL in master transmit mode • When the SDA pin outputs high in master mode while a start condition is detected • When the final bit is received with the clocked synchronous format while RDRF = 1
1	AAS	0	R/W	Slave Address Recognition Flag In slave receive mode, this flag is set to 1 if the first frame following a start condition matches bits SVA[6:0] in SAR. [Clearing condition] • When 0 is written in AAS after reading AAS = 1 [Setting conditions] • When the slave address is detected in slave receive mode • When the general call address is detected in slave receive mode.
0	ADZ	0	R/W	General Call Address Recognition Flag This bit is valid in slave receive mode with the I²C bus format. [Clearing condition] • When 0 is written in ADZ after reading ADZ = 1 [Setting condition] • When the general call address is detected in slave receive mode

chanhvinh.com

17.3.6 Slave Address Register (SAR)

SAR is an 8-bit readable/writable register that selects the communications format and sets the slave address. In slave mode with the I²C bus format, if the upper seven bits of SAR match the upper seven bits of the first frame received after a start condition, this module operates as the slave device.

Bit:	7	6	5	4	3	2	1	0
	SVA[6:0]							FS
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	SVA[6:0]	0000000	R/W	Slave Address These bits set a unique address in these bits, differing form the addresses of other slave devices connected to the I²C bus.
0	FS	0	R/W	Format Select 0: I²C bus format is selected 1: Clocked synchronous serial format is selected

17.3.7 I²C Bus Transmit Data Register (ICDRT)

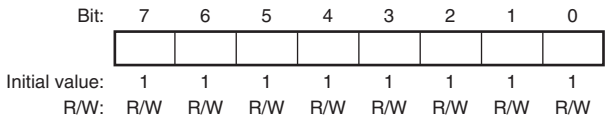
ICDRT is an 8-bit readable/writable register that stores the transmit data. When ICDRT detects the space in the shift register (ICDRS), it transfers the transmit data which is written in ICDRT to ICDRS and starts transferring data. If the next transfer data is written to ICDRT while transferring data of ICDRS, continuous transfer is possible.

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

chanhvinh.com

17.3.8 I²C Bus Receive Data Register (ICDRR)

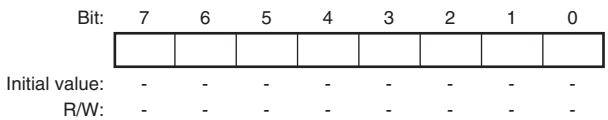
ICDRR is an 8-bit register that stores the receive data. When data of one byte is received, ICDRR transfers the receive data from ICDRS to ICDRR and the next data can be received. ICDRR is a receive-only register, therefore the CPU cannot write to this register.



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

17.3.9 I²C Bus Shift Register (ICDRS)

ICDRS is a register that is used to transfer/receive data. In transmission, data is transferred from ICDRT to ICDRS and the data is sent from the SDA pin. In reception, data is transferred from ICDRS to ICDRR after data of one byte is received. This register cannot be read directly from the CPU.



Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-

chanhvinh.com

17.3.10 NF2CYC Register (NF2CYC)

NF2CYC is an 8-bit readable/writable register that selects a transfer clock and the range of the noise filtering for the SCL and SDA pins. For details of the noise filter, see section 17.4.7, Noise Filter.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	CKS4	-	-	PRS	NF2CYC
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	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	CKS4	0	R/W	Transfer Clock Select This bit should be set according to the necessary transfer rate (table 17.3) in master mode. For 1-Mbyte version, this bit is reserved and always read as 0. The write value should always be 0.
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	PRS	0	R/W	Pulse Width Ratio Select Specifies the ratio of the high-level period to the low-level period for the SCL signal. 0: The ratio of high to low is 0.5 to 0.5. 1: The ratio of high to low is about 0.4 to 0.6.
0	NF2CYC	0	R/W	Noise Filtering Range Select 0: The noise less than one cycle of the peripheral clock can be filtered out 1: The noise less than two cycles of the peripheral clock can be filtered out

chanhvinh.com

17.4 Operation

The I²C bus interface 3 can communicate either in I²C bus mode or clocked synchronous serial mode by setting FS in SAR.

17.4.1 I²C Bus Format

Figure 17.3 shows the I²C bus formats. Figure 17.4 shows the I²C bus timing. The first frame following a start condition always consists of eight bits.

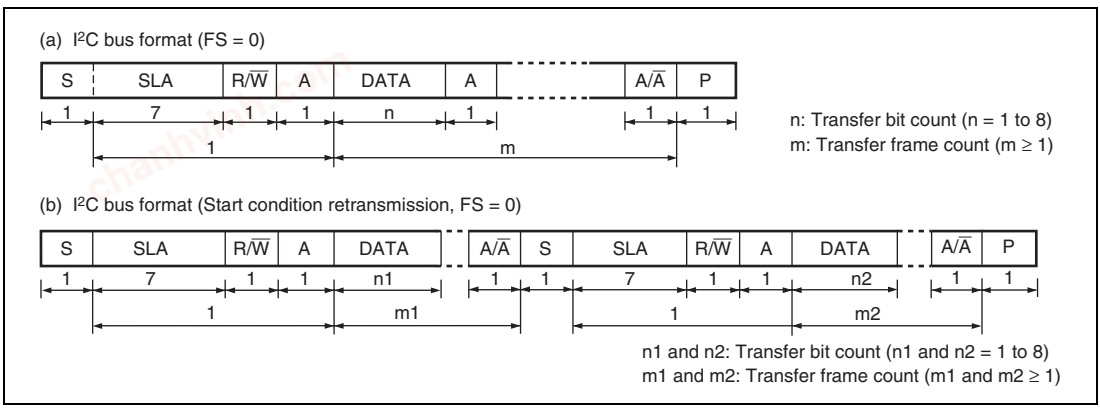


Figure 17.3 I²C Bus Formats

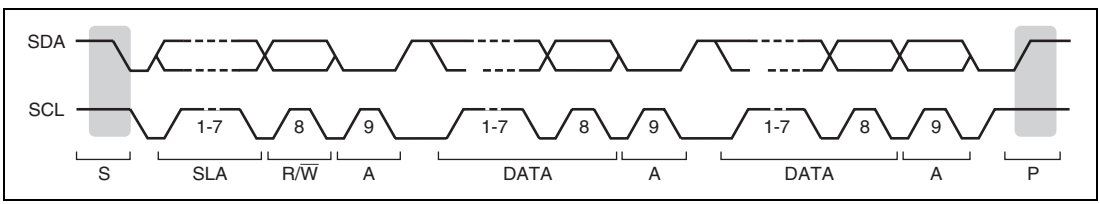


Figure 17.4 I²C Bus Timing

[Legend]

S: Start condition. The master device drives SDA from high to low while SCL is high.

SLA: Slave address

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 SDA to low.

DATA: Transfer data

P: Stop condition. The master device drives SDA from low to high while SCL is high.

chanhvinh.com

17.4.2 Master Transmit Operation

In master transmit mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. For master transmit mode operation timing, refer to figures 17.5 and 17.6. The transmission procedure and operations in master transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Also, set bits CKS[3:0] in ICCR1. (Initial setting)
2. Read the BBSY flag in ICCR2 to confirm that the bus is released. Set the MST and TRS bits in ICCR1 to select master transmit mode. Then, write 1 to BBSY and 0 to SCP. (Start condition issued) This generates the start condition.
3. After confirming that TDRE in ICSR has been set, write the transmit data (the first byte data show the slave address and R/$\overline{W}$) to ICDRT. At this time, TDRE is automatically cleared to 0, and data is transferred from ICDRT to ICDRS. TDRE is set again.
4. When transmission of one byte data is completed while TDRE is 1, TEND in ICSR is set to 1 at the rise of the 9th transmit clock pulse. Read the ACKBR bit in ICIER, and confirm that the slave device has been selected. Then, write second byte data to ICDRT. When ACKBR is 1, the slave device has not been acknowledged, so issue the stop condition. To issue the stop condition, write 0 to BBSY and SCP. SCL is fixed low until the transmit data is prepared or the stop condition is issued.
5. The transmit data after the second byte is written to ICDRT every time TDRE is set.
6. Write the number of bytes to be transmitted to ICDRT. Wait until TEND is set (the end of last byte data transmission) while TDRE is 1, or wait for NACK (NACKF in ICSR = 1) from the receive device while ACKE in ICIER is 1. Then, issue the stop condition to clear TEND or NACKF.
7. When the STOP bit in ICSR is set to 1, the operation returns to the slave receive mode.

chanhvinh.com

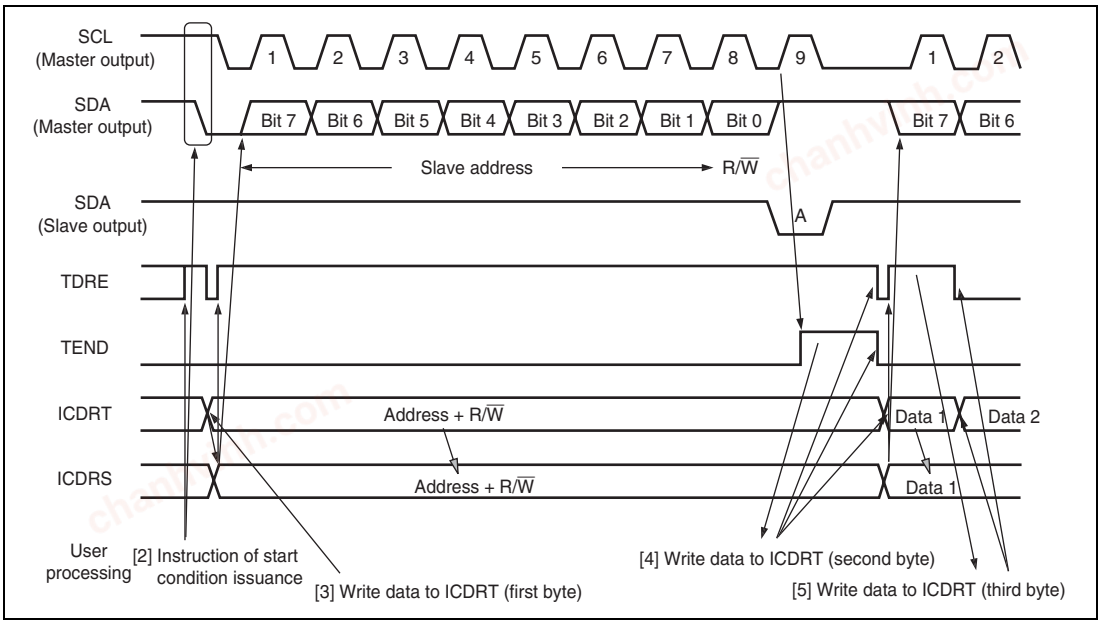


Figure 17.5 Master Transmit Mode Operation Timing (1)

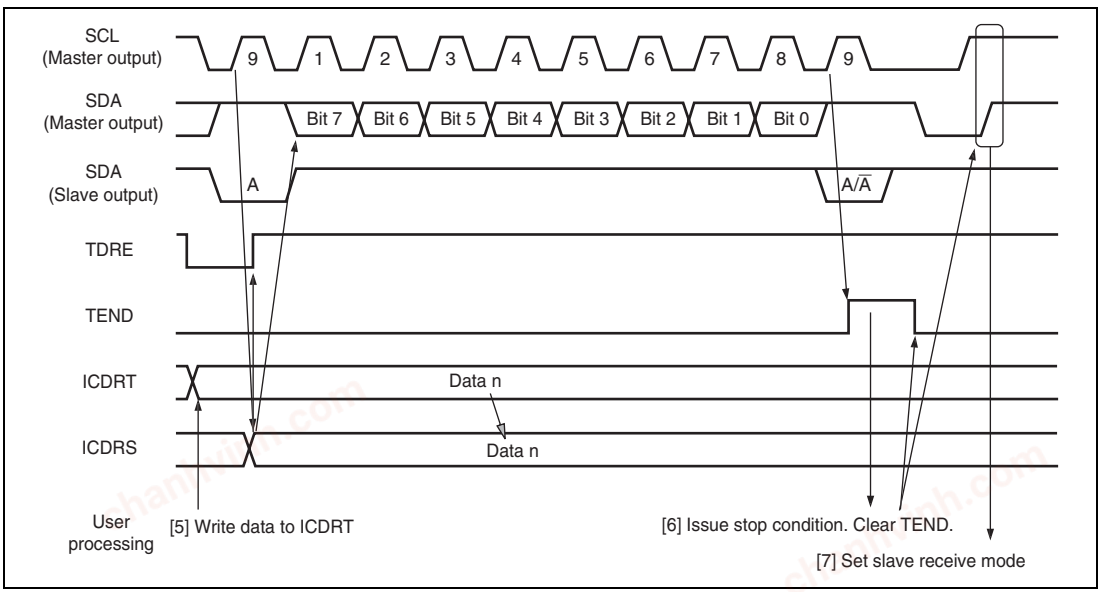


Figure 17.6 Master Transmit Mode Operation Timing (2)

chanhvinh.com

17.4.3 Master Receive Operation

In master receive mode, the master device outputs the receive clock, receives data from the slave device, and returns an acknowledge signal. For master receive mode operation timing, refer to figures 17.7 and 17.8. The reception procedure and operations in master receive mode are shown below.

1. Clear the TEND bit in ICSR to 0, then clear the TRS bit in ICCR1 to 0 to switch from master transmit mode to master receive mode. Then, clear the TDRE bit to 0.
2. When ICDRR is read (dummy data read), reception is started, and the receive clock is output, and data received, in synchronization with the internal clock. The master device outputs the level specified by ACKBT in ICIER to SDA, at the 9th receive clock pulse.
3. After the reception of first frame data is completed, the RDRF bit in ICSR is set to 1 at the rise of 9th receive clock pulse. At this time, the receive data is read by reading ICDRR, and RDRF is cleared to 0.
4. The continuous reception is performed by reading ICDRR every time RDRF is set. If 8th receive clock pulse falls after reading ICDRR by the other processing while RDRF is 1, SCL is fixed low until ICDRR is read.
5. If next frame is the last receive data, set the RCVD bit in ICCR1 to 1 before reading ICDRR. This enables the issuance of the stop condition after the next reception.
6. When the RDRF bit is set to 1 at rise of the 9th receive clock pulse, issue the stage condition.
7. When the STOP bit in ICSR is set to 1, read ICDRR. Then clear the RCVD bit to 0.
8. The operation returns to the slave receive mode.

Note: If only one byte is received, read ICDRR (dummy-read) after the RCVD bit in ICCR1 is set.

chanhvinh.com

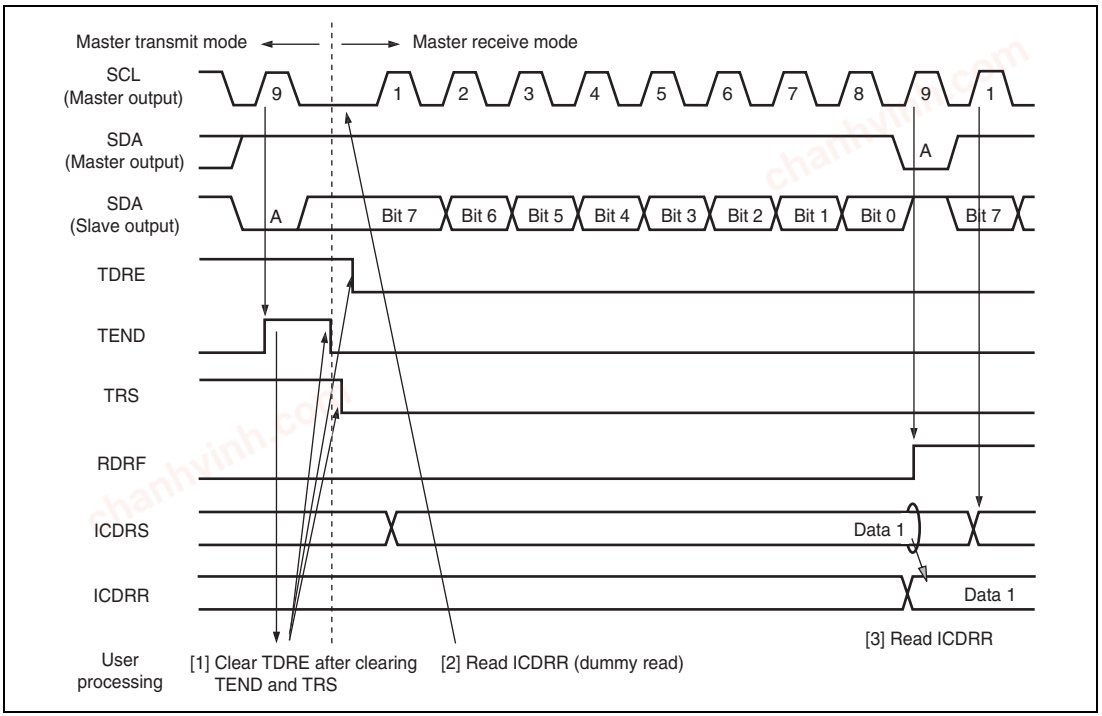


Figure 17.7 Master Receive Mode Operation Timing (1)

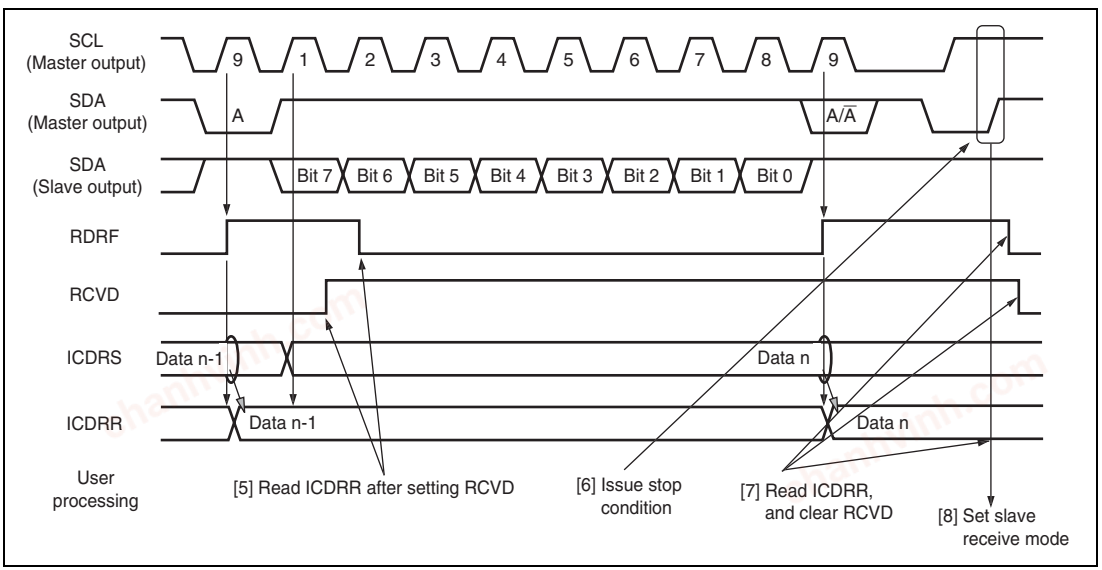


Figure 17.8 Master Receive Mode Operation Timing (2)

chanhvinh.com

17.4.4 Slave Transmit Operation

In slave transmit mode, the slave device outputs the transmit data, while the master device outputs the receive clock and returns an acknowledge signal. For slave transmit mode operation timing, refer to figures 17.9 and 17.10.

The transmission procedure and operations in slave transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set bits CKS[3:0] in ICCR1. (Initial setting) Set the MST and TRS bits in ICCR1 to select slave receive mode, and wait until the slave address matches.
2. When the slave address matches in the first frame following detection of the start condition, the slave device outputs the level specified by ACKBT in ICIER to SDA, at the rise of the 9th clock pulse. At this time, if the 8th bit data (R/W) is 1, the TRS bit in ICCR1 and the TDRE bit in ICSR are set to 1, and the mode changes to slave transmit mode automatically. The continuous transmission is performed by writing transmit data to ICDRT every time TDRE is set.
3. If TDRE is set after writing last transmit data to ICDRT, wait until TEND in ICSR is set to 1, with TDRE = 1. When TEND is set, clear TEND.
4. Clear TRS for the end processing, and read ICDRR (dummy read). SCL is opened.
5. Clear TDRE.

chanhvinh.com

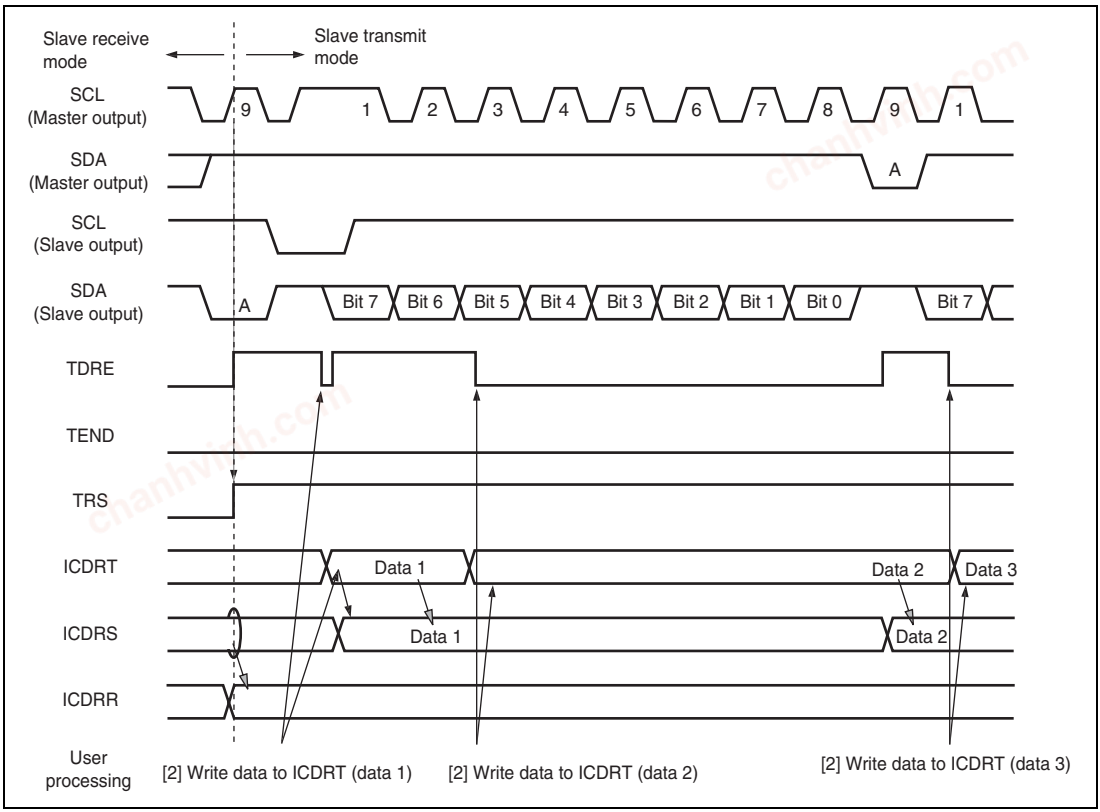


Figure 17.9 Slave Transmit Mode Operation Timing (1)

chanhvinh.com

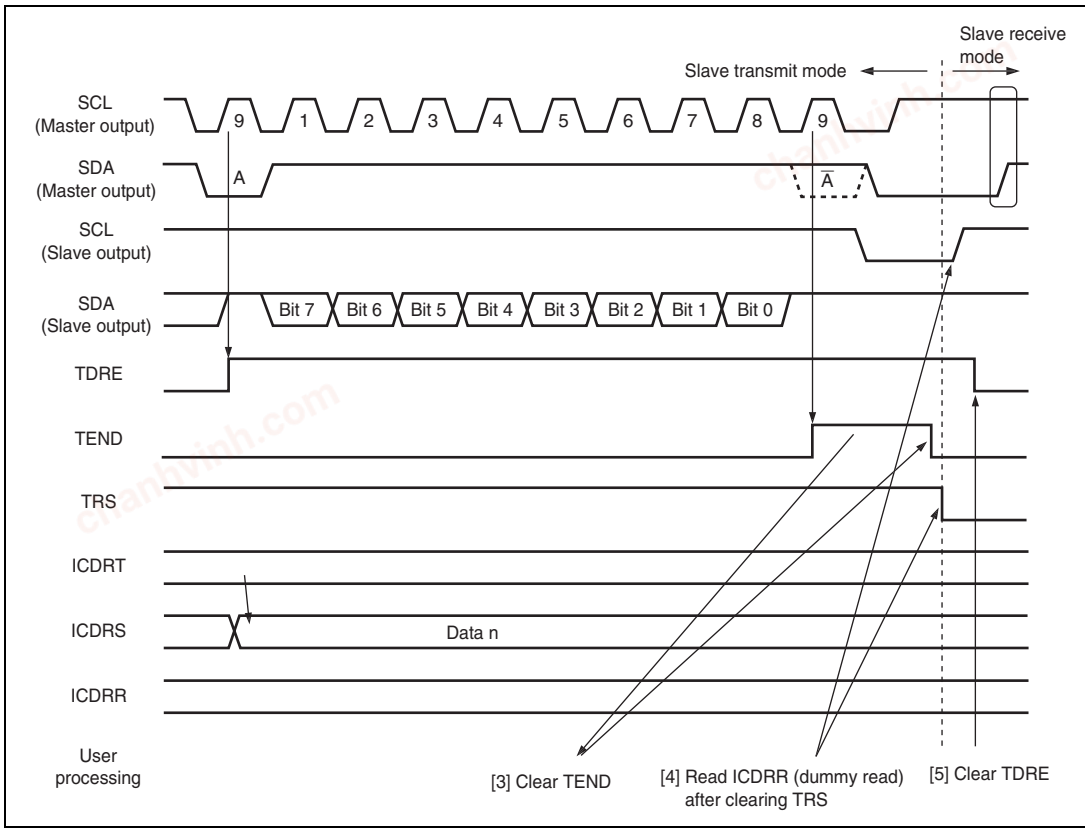


Figure 17.10 Slave Transmit Mode Operation Timing (2)

chanhvinh.com

17.4.5 Slave Receive Operation

In slave receive mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. For slave receive mode operation timing, refer to figures 17.11 and 17.12. The reception procedure and operations in slave receive mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set bits CKS[3:0] in ICCR1. (Initial setting) Set the MST and TRS bits in ICCR1 to select slave receive mode, and wait until the slave address matches.
2. When the slave address matches in the first frame following detection of the start condition, the slave device outputs the level specified by ACKBT in ICIER to SDA, at the rise of the 9th clock pulse. At the same time, RDRF in ICSR is set to read ICDRR (dummy read). (Since the read data show the slave address and R/W, it is not used.)
3. Read ICDRR every time RDRF is set. If 8th receive clock pulse falls while RDRF is 1, SCL is fixed low until ICDRR is read. The change of the acknowledge before reading ICDRR, to be returned to the master device, is reflected to the next transmit frame.
4. The last byte data is read by reading ICDRR.

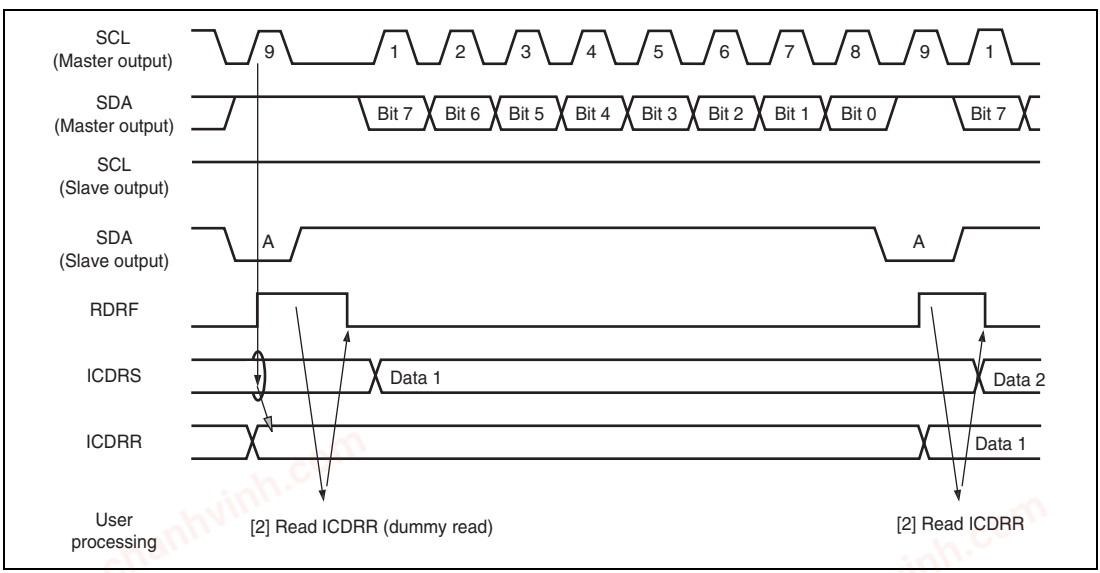


Figure 17.11 Slave Receive Mode Operation Timing (1)

chanhvinh.com

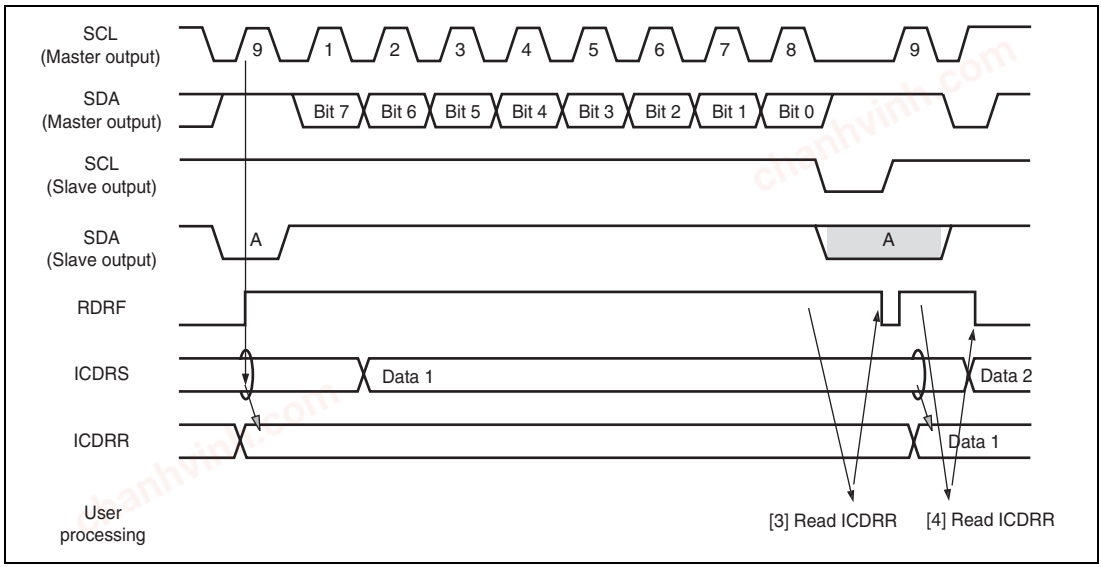


Figure 17.12 Slave Receive Mode Operation Timing (2)

17.4.6 Clocked Synchronous Serial Format

This module can be operated with the clocked synchronous serial format, by setting the FS bit in SAR to 1. When the MST bit in ICCR1 is 1, the transfer clock output from SCL is selected. When MST is 0, the external clock input is selected.

(1) Data Transfer Format

Figure 17.13 shows the clocked synchronous serial transfer format.

The transfer data is output from the fall to the fall of the SCL clock, and the data at the rising edge of the SCL clock is guaranteed. The MLS bit in ICMR sets the order of data transfer, in either the MSB first or LSB first. The output level of SDA can be changed during the transfer wait, by the SDAO bit in ICCR2.

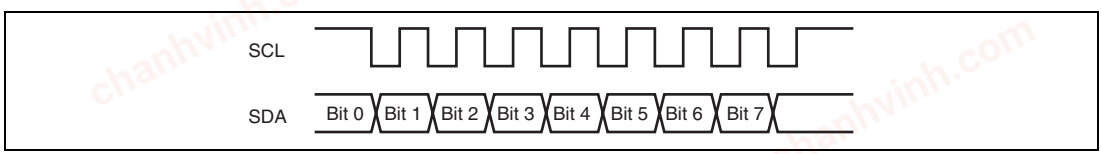


Figure 17.13 Clocked Synchronous Serial Transfer Format

chanhvinh.com

(2) Transmit Operation

In transmit mode, transmit data is output from SDA, in synchronization with the fall of the transfer clock. The transfer clock is output when MST in ICCR1 is 1, and is input when MST is 0. For transmit mode operation timing, refer to figure 17.14. The transmission procedure and operations in transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MST and CKS[3:0] bits in ICCR1. (Initial setting)
2. Set the TRS bit in ICCR1 to select the transmit mode. Then, TDRE in ICSR is set.
3. Confirm that TDRE has been set. Then, write the transmit data to ICDRT. The data is transferred from ICDRT to ICDRS, and TDRE is set automatically. The continuous transmission is performed by writing data to ICDRT every time TDRE is set. When changing from transmit mode to receive mode, clear TRS while TDRE is 1.

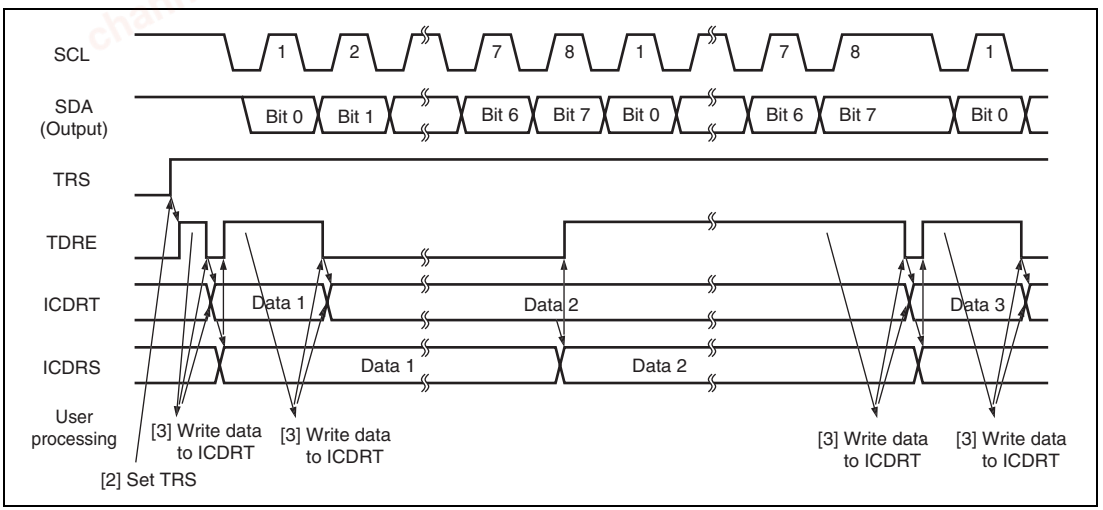


Figure 17.14 Transmit Mode Operation Timing

chanhvinh.com

(3) Receive Operation

In receive mode, data is latched at the rise of the transfer clock. The transfer clock is output when MST in ICCR1 is 1, and is input when MST is 0. For receive mode operation timing, refer to figure 17.15. The reception procedure and operations in receive mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set bits CKS[3:0] in ICCR1. (Initial setting)
2. When the transfer clock is output, set MST to 1 to start outputting the receive clock.
3. When the receive operation is completed, data is transferred from ICDRS to ICDRR and RDRF in ICSR is set. When MST = 1, the next byte can be received, so the clock is continually output. The continuous reception is performed by reading ICDRR every time RDRF is set. When the 8th clock is risen while RDRF is 1, the overrun is detected and AL/OVE in ICSR is set. At this time, the previous reception data is retained in ICDRR.
4. To stop receiving when MST = 1, set RCVD in ICCR1 to 1, then read ICDRR. Then, SCL is fixed high after receiving the next byte data.

Notes: Follow the steps below to receive only one byte with MST = 1 specified. See figure 17.16 for the operation timing.

1. Set the ICE bit in ICCR1 to 1. Set bits CKS[3:0] in ICCR1. (Initial setting)
2. Set MST = 1 while the RCVD bit in ICCR1 is 0. This causes the receive clock to be output.
3. Check if the BC2 bit in ICMR is set to 1 and then set the RCVD bit in ICCR1 to 1. This causes the SCL to be fixed to the high level after outputting one byte of the receive clock.

chanhvinh.com

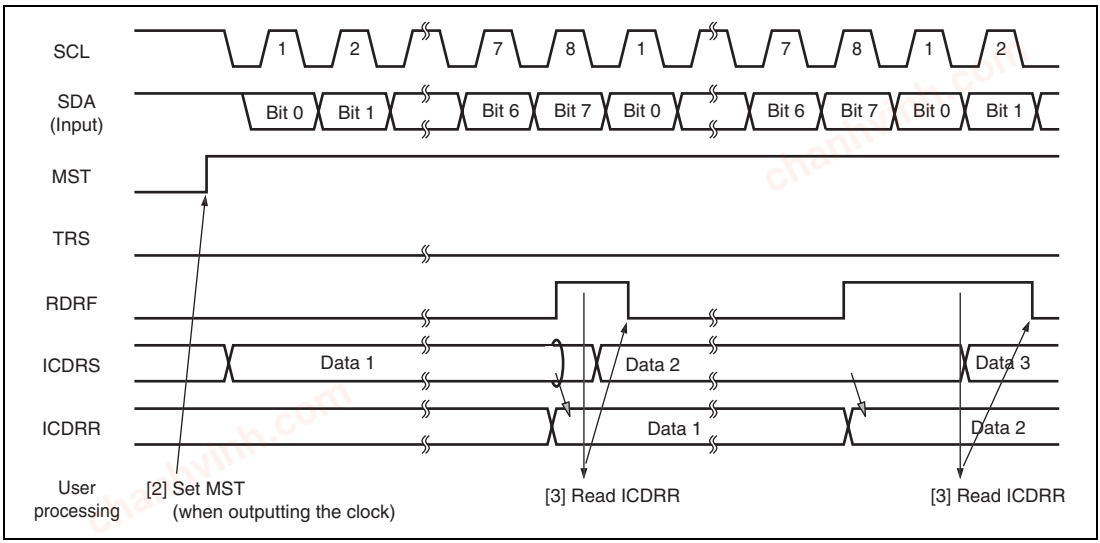


Figure 17.15 Receive Mode Operation Timing

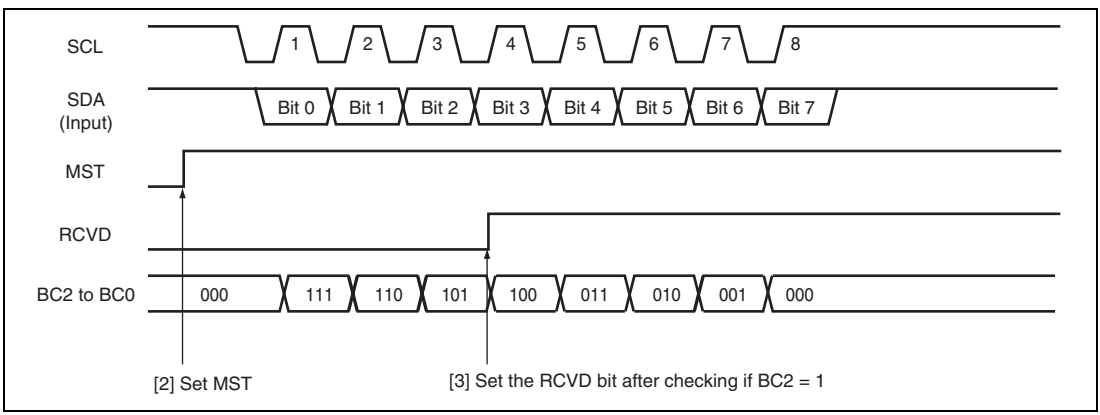


Figure 17.16 Operation Timing For Receiving One Byte (MST = 1)

chanhvinh.com

17.4.7 Noise Filter

The logic levels at the SCL and SDA pins are routed through noise filters before being latched internally. Figure 17.17 shows a block diagram of the noise filter circuit.

The noise filter consists of three cascaded latches and a match detector. The SCL (or SDA) input signal is sampled on the peripheral clock. When NF2CYC is set to 0, this signal is not passed forward to the next circuit unless the outputs of both latches agree. When NF2CYC is set to 1, this signal is not passed forward to the next circuit unless the outputs of three latches agree. If they do not agree, the previous value is held.

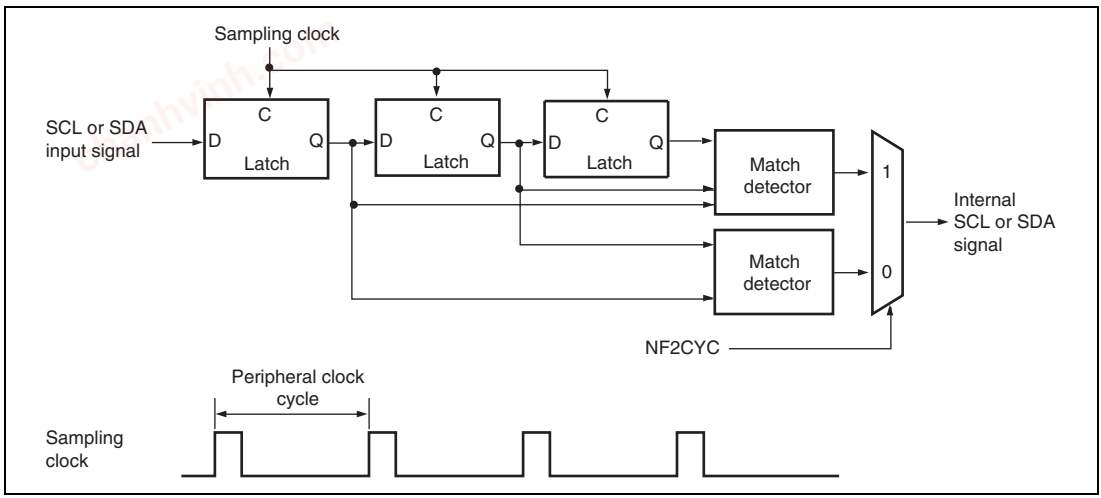


Figure 17.17 Block Diagram of Noise Filter

chanhvinh.com

17.4.8 Example of Use

Flowcharts in respective modes that use the I²C bus interface 3 are shown in figures 17.18 to 17.21.

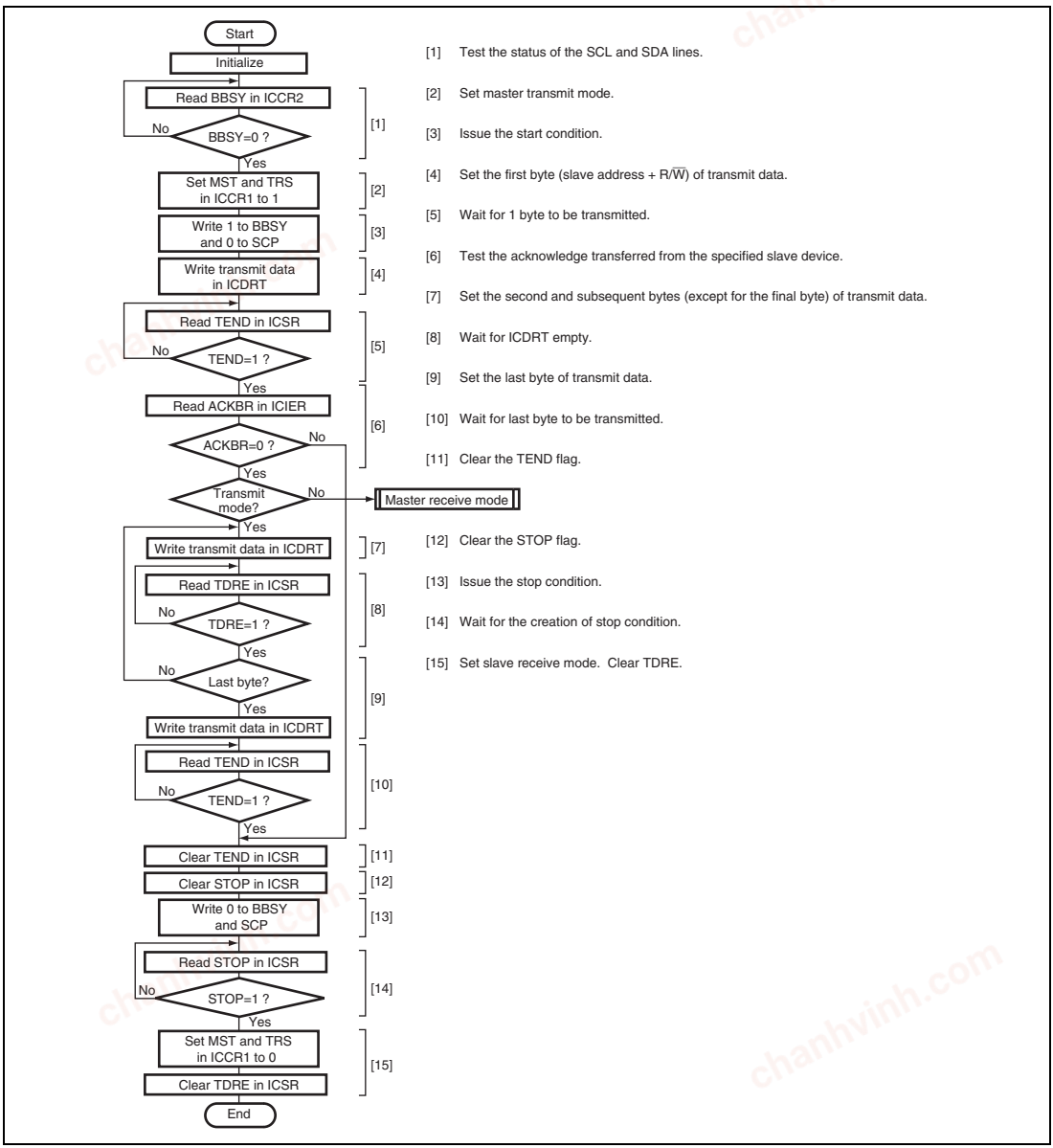


Figure 17.18 Sample Flowchart for Master Transmit Mode

chanhvinh.com

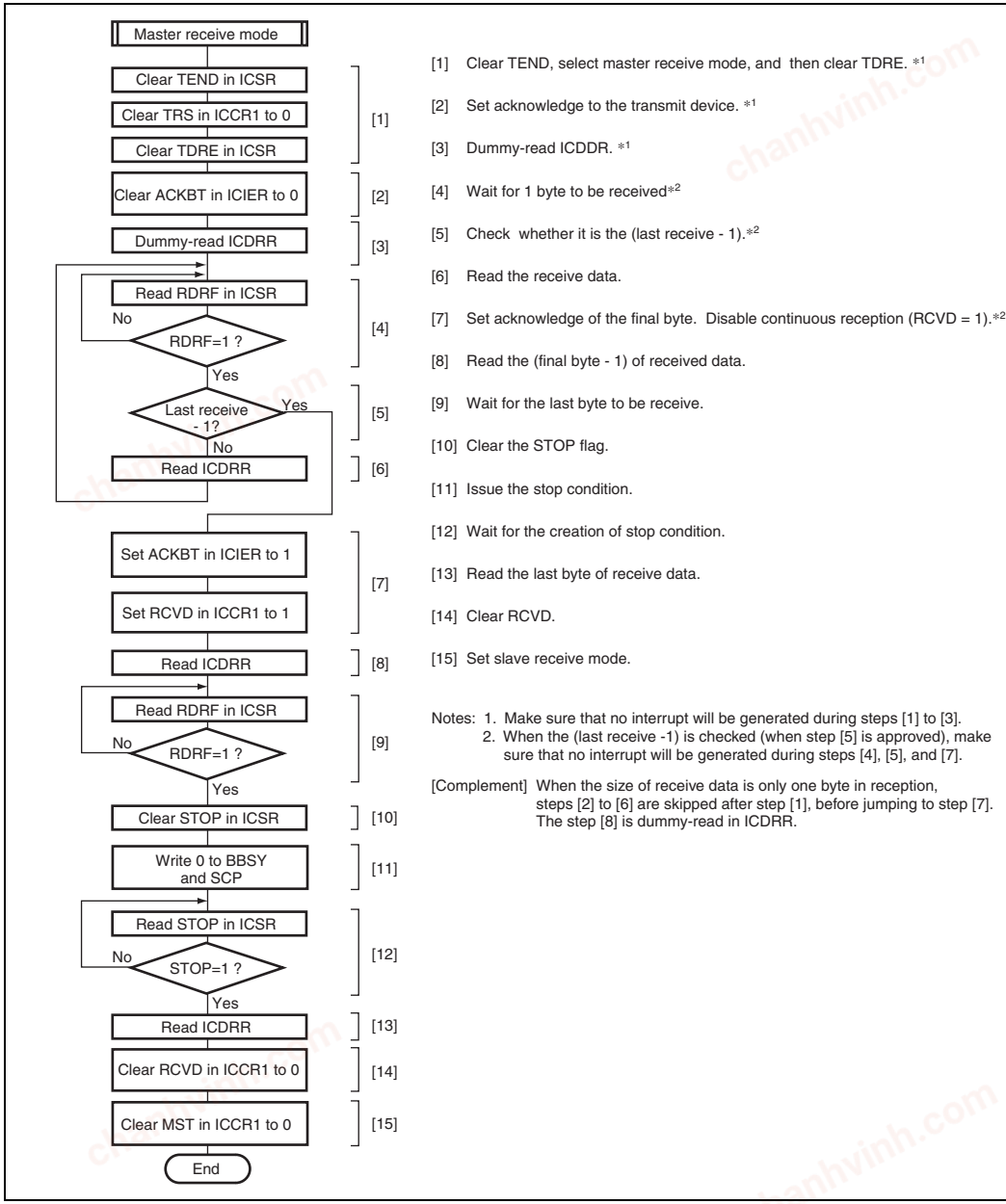


Figure 17.19 Sample Flowchart for Master Receive Mode

chanhvinh.com

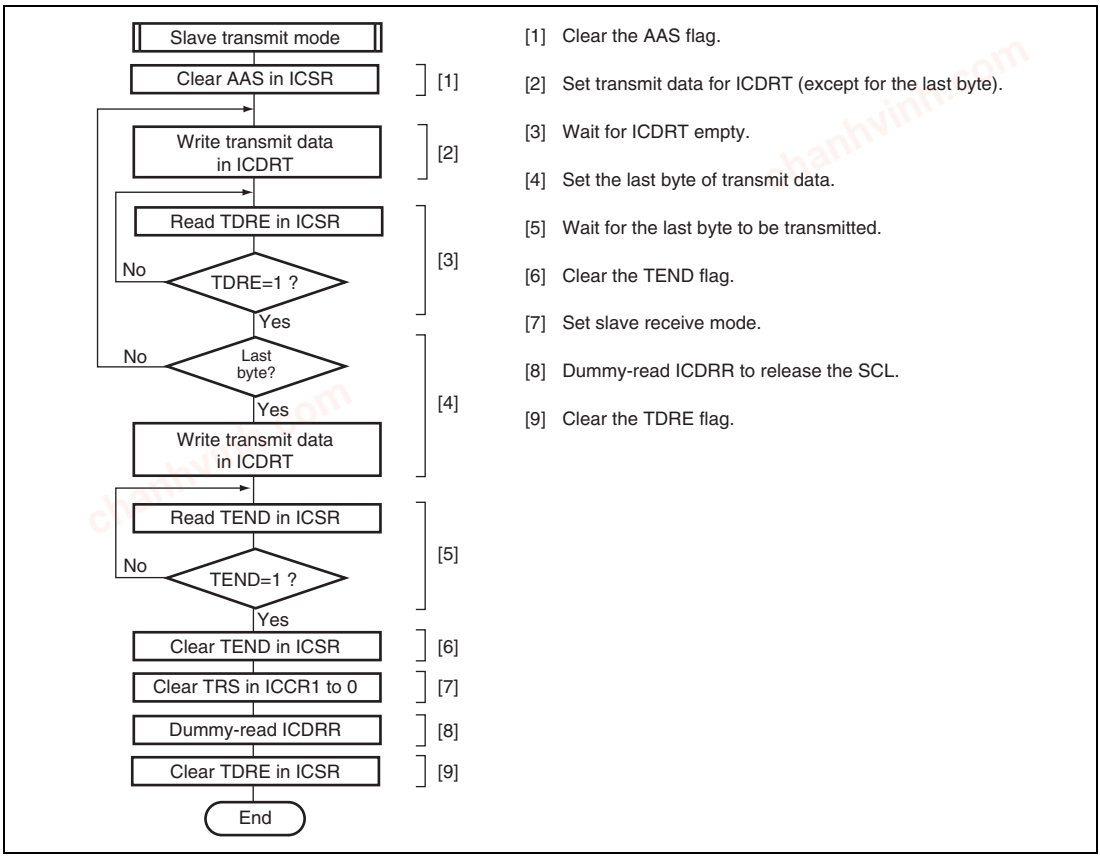


Figure 17.20 Sample Flowchart for Slave Transmit Mode

chanhvinh.com

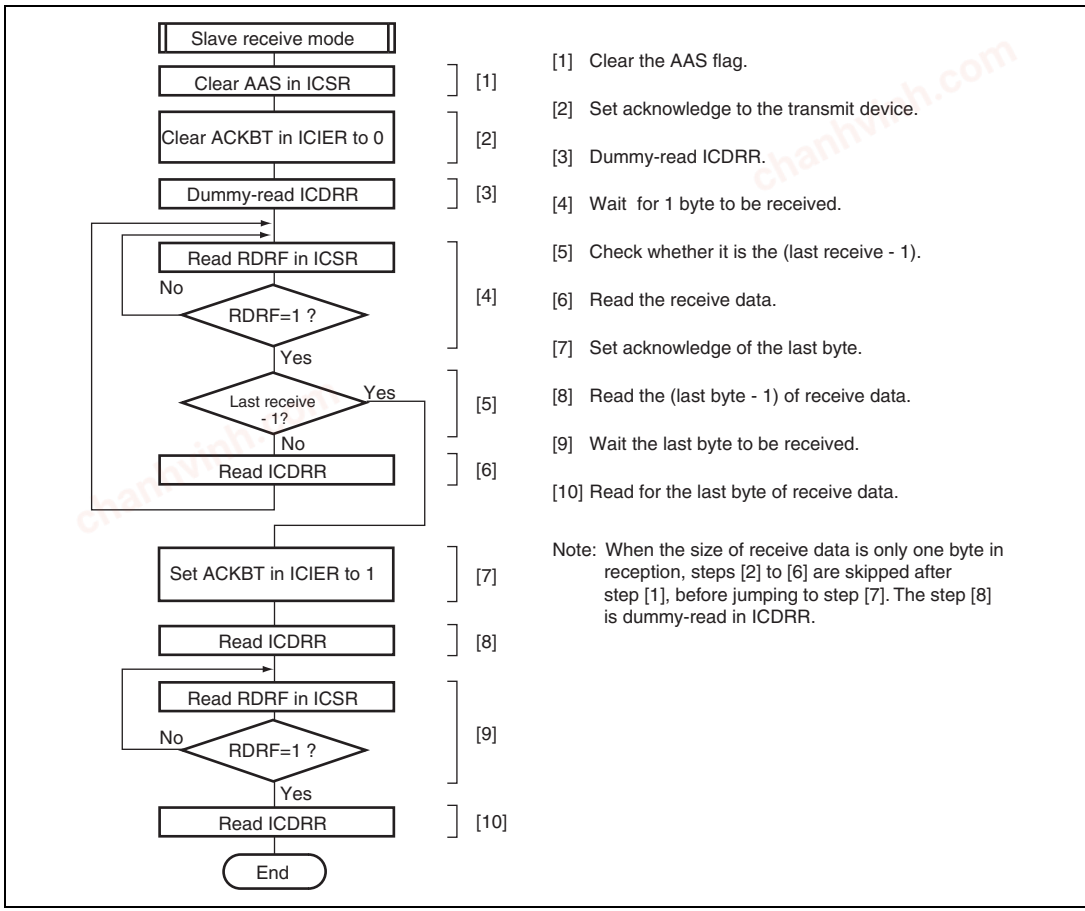


Figure 17.21 Sample Flowchart for Slave Receive Mode

chanhvinh.com

17.5 Interrupt Requests

There are six interrupt requests in this module; transmit data empty, transmit end, receive data full, NACK detection, STOP recognition, and arbitration lost/overrun error. Table 17.4 shows the contents of each interrupt request.

Table 17.4 Interrupt Requests

Interrupt Request	Abbreviation	Interrupt Condition	I²C Bus Format	Clocked Synchronous Serial Format
Transmit data Empty	TXI	(TDRE = 1) • (TIE = 1)	√	√
Transmit end	TEI	(TEND = 1) • (TEIE = 1)	√	√
Receive data full	RXI	(RDRF = 1) • (RIE = 1)	√	√
STOP recognition	STPI	(STOP = 1) • (STIE = 1)	√	—
NACK detection	NAKI	{(NACKF = 1) + (AL = 1)} • (NAKIE = 1)	√	—
Arbitration lost/ overrun error			√	√

When the interrupt condition described in table 17.4 is 1, the CPU executes an interrupt exception handling. Note that a TXI or RXI interrupt can activate the direct memory access controller if the setting for direct memory access controller activation has been made. In such a case, an interrupt request is not sent to the CPU. Interrupt sources should be cleared in the exception handling. The TDRE and TEND bits are automatically cleared to 0 by writing the transmit data to ICDRT. The RDRF bit is automatically cleared to 0 by reading ICDRR. The TDRE bit is set to 1 again at the same time when the transmit data is written to ICDRT. Therefore, when the TDRE bit is cleared to 0, then an excessive data of one byte may be transmitted.

chanhvinh.com

17.6 Bit Synchronous Circuit

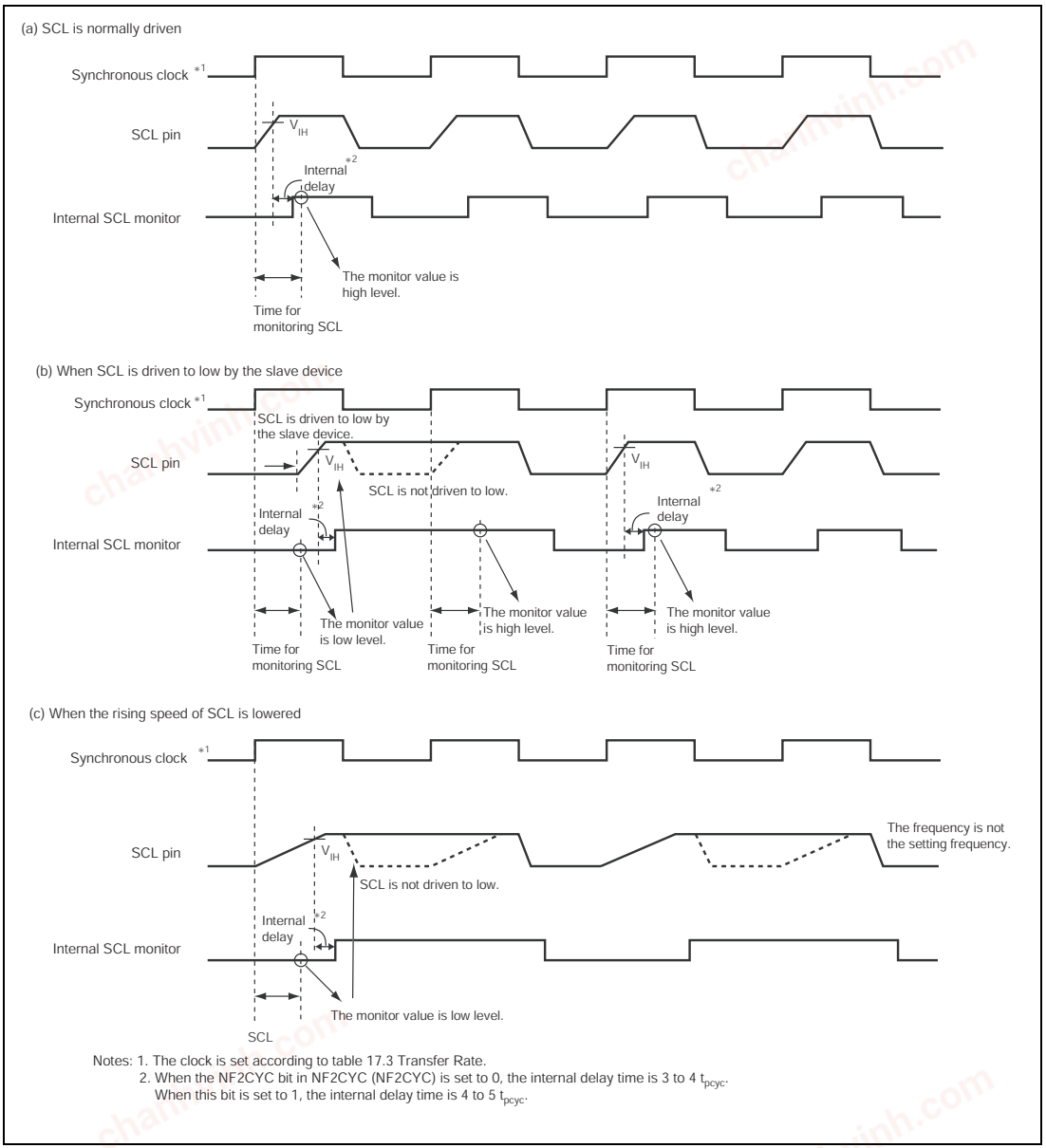
In master mode, this module has a possibility that high level period may be short in the two states described below.

- When SCL is driven to low by the slave device
- When the rising speed of SCL is lowered by the load of the SCL line (load capacitance or pull-up resistance)

Therefore, it monitors SCL and communicates by bit with synchronization.

Figure 17.22 shows the timing of the bit synchronous circuit and table 17.5 shows the time when the SCL output changes from low to Hi-Z then SCL is monitored.

chanhvinh.com



Notes: 1. The clock is set according to table 17.3 Transfer Rate.
2. When the NF2CYC bit in NF2CYC (NF2CYC) is set to 0, the internal delay time is 3 to 4 t_{pcyc}.
When this bit is set to 1, the internal delay time is 4 to 5 t_{pcyc}.

Figure 17.22 Bit Synchronous Circuit Timing

chanhvinh.com

Table 17.5 Time for Monitoring SCL

CKS4*[2]	CKS3	CKS2	Time for Monitoring SCL	Remarks
0	0	0	9 tpcyc*[1]	
		1	21 tpcyc*[1]	
	1	0	39 tpcyc*[1]	
		1	87 tpcyc*[1]	
1	0	0	79 tpcyc*[1]	*[3]
		1	175 tpcyc*[1]	
	1	0	159 tpcyc*[1]	
		1	351 tpcyc*[1]	

Notes: 1. tpcyc indicates the frequency of the peripheral clock (Pϕ).
2. This bit is reserved for 1-Mbyte version.
3. These settings are valid only in 640-Kbyte version.

17.7 Usage Notes

17.7.1 Note on Setting for Multi-Master Operation

In multi-master operation, when the transfer rate setting for this module (ICCR1.CKS[3:0]) makes this LSI slower than the other masters, pulse cycles with an unexpected length will infrequently be output on SCL.

Be sure to specify a transfer rate that is at least 1/1.8 of the fastest transfer rate among the other masters.

17.7.2 Note on Master Receive Mode

Reading ICDRR around the falling edge of the 8th clock might fail to fetch the receive data.

In addition, when RCVD is set to 1 around the falling edge of the 8th clock and the receive buffer full, a stop condition may not be issued.

Use either 1 or 2 below as a measure against the situations above.

1. In master receive mode, read ICDRR before the rising edge of the 8th clock.
2. In master receive mode, set the RCVD bit to 1 so that transfer proceeds in byte units.

chanhvinh.com

17.7.3 Note on Setting ACKBT in Master Receive Mode

In master receive mode operation, set ACKBT before the falling edge of the 8th SCL cycle of the last data being continuously transferred. Not doing so can lead to an overrun for the slave transmission device.

17.7.4 Note on the States of Bits MST and TRN when Arbitration is Lost

When sequential bit-manipulation instructions are used to set the MST and TRS bits to select master transmission in multi-master operation, a conflicting situation where AL in ICSR = 1 but the mode is master transmit mode (MST = 1 and TRS = 1) may arise; this depends on the timing of the loss of arbitration when the bit manipulation instruction for TRS is executed.

This can be avoided in either of the following ways.

- In multi-master operation, use the MOV instruction to set the MST and TRS bits.
- When arbitration is lost, check whether the MST and TRS bits are 0. If the MST and TRS bits have been set to a value other than 0, clear the bits to 0.

17.7.5 Note on I²C-bus Interface Master Receive Mode

After a master receive operation is completed, confirm the falling edge of the ninth clock cycle of the SCL signal and generate a stop condition or regenerate a start condition.

17.7.6 Note on IICRST and BBSY bits

When 1 is written to IICRST in ICCR2, this LSI release SCL and SDA pins. Then, if the SDA level changes from low to high under the condition of SCL = high, BBSY in ICCR2 is cleared to 0 assuming that the stop condition has been issued.

chanhvinh.com

Section 18 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.

18.1 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
  Channel 0 supports 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 value as the 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.
- Selects the oversampling clock input from among the following pins:
  AUDIO_CLK (1 to 25 MHz)
  AUDIO_X1, AUDIO_X2 (when connecting a crystal resonator: 10 to 25 MHz, when used to input external clock: 1 to 25 MHz)
- Includes 8-stage FIFO buffers in transmitter and receiver

chanhvinh.com

Figure 18.1 shows a schematic diagram of the four channels in this module.

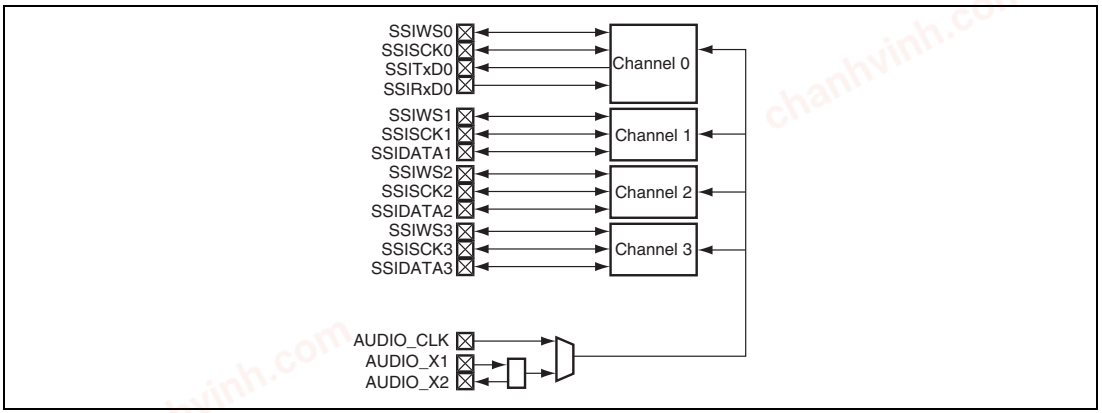


Figure 18.1 Schematic Diagram of Serial Sound Interface

chanhvinh.com

Figure 18.2 shows a block diagram of this module.

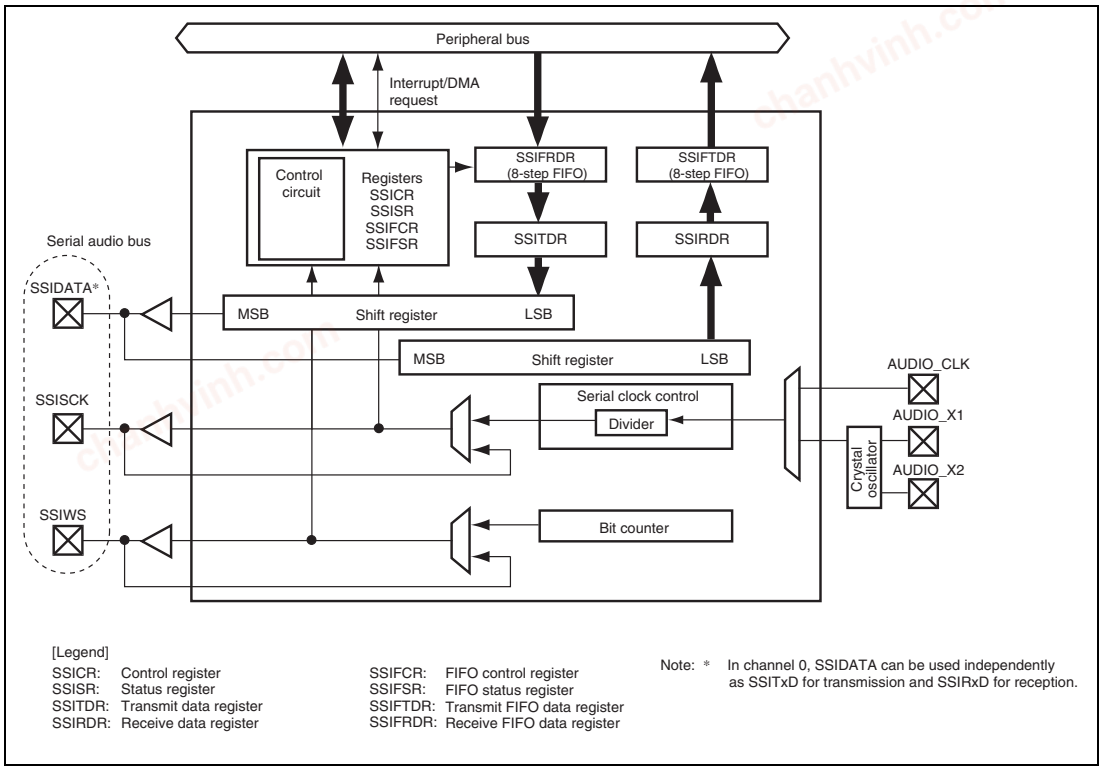


Figure 18.2 Block Diagram of Serial Sound Interface

chanhvinh.com

18.2 Input/Output Pins

Table 18.1 shows the pin assignments relating to this module.

Table 18.1 Pin Assignments

Channel	Pin Name	I/O	Description
0	SSISCK0	I/O	Serial bit clock
	SSIWS0	I/O	Word selection
	SSITxD0	Output	Serial data output
	SSIRxD0	Input	Serial data input
1 to 3	SSISCK1 to SSISCK3	I/O	Serial bit clock
	SSIWS1 to SSIWS3	I/O	Word selection
	SSIDATA1 to SSIDATA3	I/O	Serial data input/output
Common	AUDIO_CLK	Input	External clock for audio (input oversampling clock)
	AUDIO_X1	Input	Crystal resonator/external clock for audio (input oversampling clock)
	AUDIO_X2	Output	

chanhvinh.com

18.3 Register Description

Table 18.2 lists the register configuration. Note that explanation in the text does not refer to the channels.

Table 18.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
0	Control register 0	SSICR_0	R/W	H'00000000	H'FFFF0000	8, 16, 32
	Status register 0	SSISR_0	R/W*[1]	H'02000013	H'FFFF0004	8, 16, 32
	FIFO control register 0	SSIFCR_0	R/W	H'00000000	H'FFFF0010	8, 16, 32
	FIFO status register 0	SSIFSR_0	R/(W)*[2]	H'00010000	H'FFFF0014	8, 16, 32
	Transmit FIFO data register 0	SSIFTDR_0	W	Undefined	H'FFFF0018	32
	Receive FIFO data register 0	SSIFRDR_0	R	Undefined	H'FFFF001C	32
1	Control register 1	SSICR_1	R/W	H'00000000	H'FFFF0800	8, 16, 32
	Status register 1	SSISR_1	R/W*[1]	H'02000013	H'FFFF0804	8, 16, 32
	FIFO control register 1	SSIFCR_1	R/W	H'00000000	H'FFFF0810	8, 16, 32
	FIFO status register 1	SSIFSR_1	R/(W)*[2]	H'00010000	H'FFFF0814	8, 16, 32
	Transmit FIFO data register 1	SSIFTDR_1	W	Undefined	H'FFFF0818	32
	Receive FIFO data register 1	SSIFRDR_1	R	Undefined	H'FFFF081C	32

chanhvinh.com

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
2	Control register 2	SSICR_2	R/W	H'00000000	H'FFFF1000	8, 16, 32
	Status register 2	SSISR_2	R/W*1	H'02000013	H'FFFF1004	8, 16, 32
	FIFO control register 2	SSIFCR_2	R/W	H'00000000	H'FFFF1010	8, 16, 32
	FIFO status register 2	SSIFSR_2	R/(W)*2	H'00010000	H'FFFF1014	8, 16, 32
	Transmit FIFO data register 2	SSIFTDR_2	W	Undefined	H'FFFF1018	32
	Receive FIFO data register 2	SSIFRDR_2	R	Undefined	H'FFFF101C	32
3	Control register 3	SSICR_3	R/W	H'00000000	H'FFFF1800	8, 16, 32
	Status register 3	SSISR_3	R/W*1	H'02000013	H'FFFF1804	8, 16, 32
	FIFO control register 3	SSIFCR_3	R/W	H'00000000	H'FFFF1810	8, 16, 32
	FIFO status register 3	SSIFSR_3	R/(W)*2	H'00010000	H'FFFF1814	8, 16, 32
	Transmit FIFO data register 3	SSIFTDR_3	W	Undefined	H'FFFF1818	32
	Receive FIFO data register 3	SSIFRDR_3	R	Undefined	H'FFFF181C	32

Notes: 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 18.3.2, Status Register (SSISR).

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 18.3.6, FIFO Status Register (SSIFSR).

chanhvinh.com

18.3.1 Control Register (SSICR)

SSICR is a readable/writable 32-bit 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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 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
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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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) and (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) and (1,1). Other settings are prohibited.

chanhvinh.com

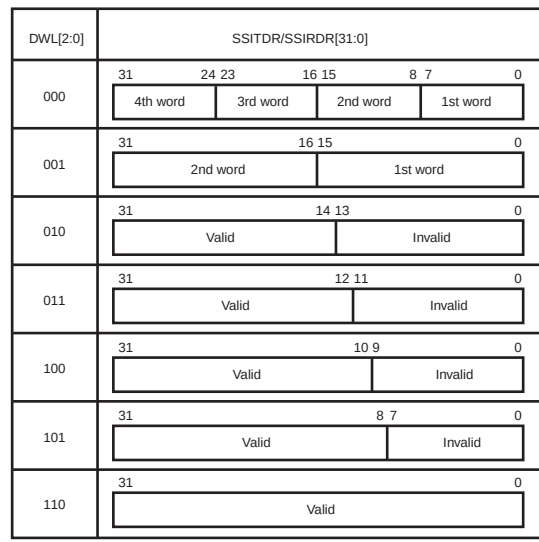
Bit	Bit Name	Initial Value	R/W	Description
13	SCKP	0	R/W	Serial Bit Clock Polarity 0: SSIWS and SSIDATA change at the SSISCK falling edge (sampled at the SCK rising edge). 1: SSIWS and SSIDATA change at the SSISCK rising edge (sampled at the SCK falling edge).

	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 0: SSIWS is low for 1st channel, high for 2nd channel. 1: SSIWS is high for 1st channel, low for 2nd channel.
11	SPDP	0	R/W	Serial Padding Polarity 0: Padding bits are low. 1: Padding bits are high.
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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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



chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
9	PDTA	0	R/W	• 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

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 0: This module is not muted. 1: This module is muted. Note: When this module is muted, the value of outputting serial data is re-written to 0 but data transmission 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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
0	REN	0	R/W	Receive Enable 0: Disables the receive operation. 1: Enables the receive operation.

18.3.2 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 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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 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 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 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 interrupt status flag indicates whether this module is in idle state. This bit is set regardless of the value of the IIEN bit to allow polling. The interrupt can be masked by clearing IIEN, but cannot be cleared by writing to this bit. If IIRQ = 1 and IIEN = 1, an 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 which channel 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.
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.
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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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.
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 are 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

18.3.3 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:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

18.3.4 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

18.3.5 FIFO Control Register (SSIFCR)

SSIFCR is a readable/writable 32-bit register that specifies the data trigger numbers and selects between transmission and reception for the FIFO data register, and enables or disables FIFO data reset 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 These bits specify the number of transmit data bytes in the FIFO (transmit trigger number) at which the TDE flag in the FIFO status register (SSIFSR) is set while the FIFO is operating for transmission. The TDE flag is set to 1 when the number of transmit data bytes in the transmit FIFO data register (SSIFTDR) has become equal to or less than the set trigger number shown below. 00: 7 (1)* 01: 6 (2)* 10: 4 (4)* 11: 2 (6)* Note: * The values in parenthesis are the number of empty stages in SSIFTDR at which the TDE flag is set.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
5, 4	RTRG[1:0]	00	R/W	Receive Data Trigger Number These bits specify the number of received data bytes in the FIFO (receive trigger number) at which the RDF flag in the FIFO status register (SSIFSR) is set while the FIFO is operating for reception. The RDF flag is set to 1 when the number of received data bytes in the receive FIFO data register (SSIFRDR) has become equal to or greater than the set trigger number shown below. 00: 1 01: 2 10: 4 11: 6
3	TIE	0	R/W	Transmit Interrupt Enable 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) is 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; and thus 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.

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.
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

18.3.6 FIFO Status Register (SSIFSR)

SSIFSR consists of status flags indicating the operating status of the FIFO data register.

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 the FIFO data register has become less than the transmit trigger number specified by TTRG[1:0] in the FIFO control register (SSIFCR), and thus writing of data transmission to SSIFTDR has been enabled.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
16	TDE	1	R/(W)*	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. * [Setting conditions] • Power-on reset • Number of transmission data bytes to be 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 This bit is 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 the FIFO data register 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 empty 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

18.3.7 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 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 during reception.

18.3.8 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 by the CPU but cannot be written to. Note that when SSIFRDR is read when it stores no received data, undefined values will be read and a receive underflow occurs.

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

18.4 Operation Description

18.4.1 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 eight major modes shown in table 18.3.

Table 18.3 Bus Format for SSIF Module

	Non-Compression Slave Receiver	Non-Compression Slave Transmitter	Non-Compression Slave Transceiver	Non-Compression Master Receiver	Non-Compression Master Transmitter	Non-Compression Master Transceiver
TEN	0	1	1	0	1	1
REN	1	0	1	1	0	1
SCKD	0	0	0	1	1	1
SWSD	0	0	0	1	1	1
MUEN	Control Bits					
IIEN						
TOIEN						
TUIEN						
ROIEN						
RUIEN						
DEL	Configuration Bits					
PDTA						
SDTA						
SPDP						
SWSP						
SCKP						
SWL[2:0]						
DWL[2:0]						
CHNL[1:0]						

chanhvinh.com

18.4.2 Non-Compressed Modes

The non-compressed modes support all serial audio streams split into channels. It supports 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 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.

chanhvinh.com

(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) Operating 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 the formats described below are I²S compatible, MSB-first left-aligned, and MSB-first right-aligned.

- I²S Compatible Format

Figures 18.3 and 18.4 demonstrate the supported I²S compatible format both with and without padding. Padding occurs when the data word length is smaller than the system word length.

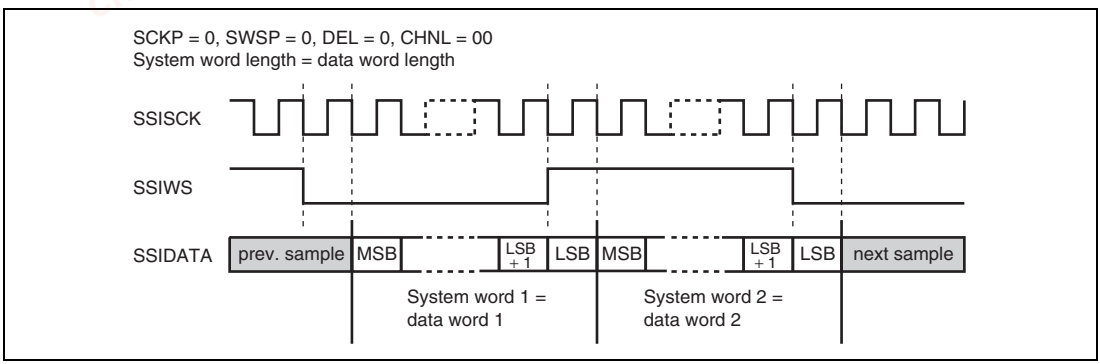


Figure 18.3 I²S Compatible Format (without Padding)

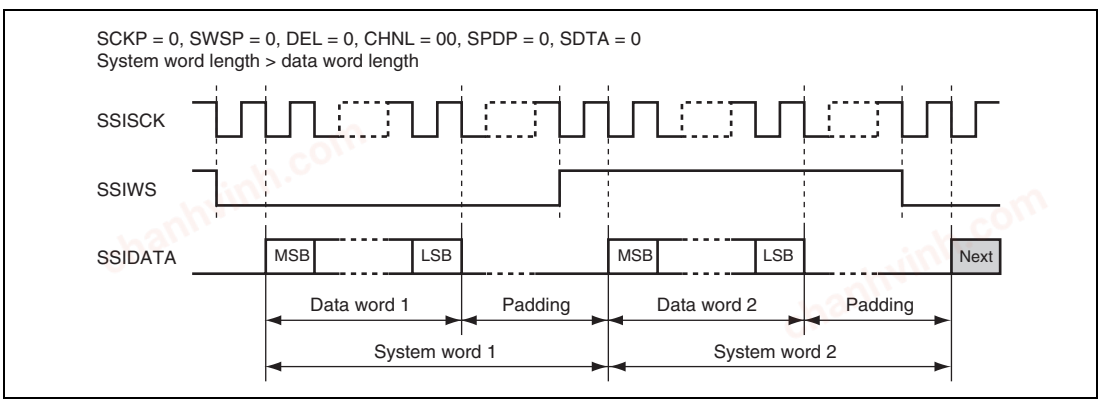


Figure 18.4 I²S Compatible Format (with Padding)

chanhvinh.com

Figure 18.5 shows MSB-first left-aligned format, and figure 18.6 shows MSB-first right-aligned format. Padding is assumed in both cases, but may not be present in a final implementation if the system word length equals the data word length.

- MSB-First Left-Aligned Format

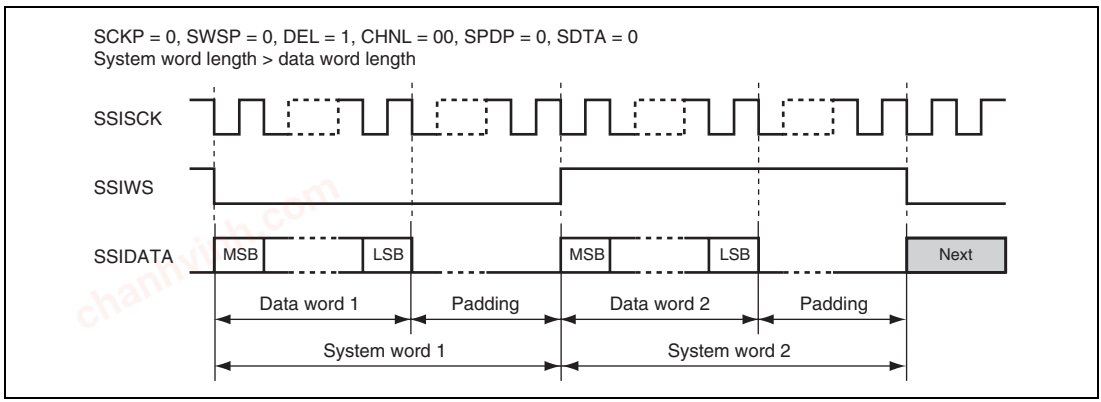


Figure 18.5 MSB-First Left-Aligned Format
(Transmitted and Received in the order of Serial Data and Padding Bits)

- MSB-First Right-Aligned Format

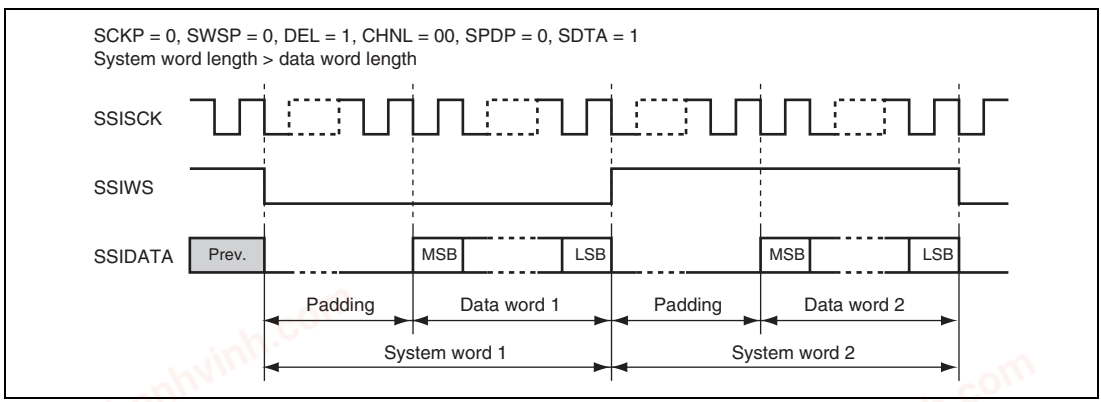


Figure 18.6 MSB-First Right-Aligned Format
(Transmitted and Received in the Order of Padding Bits and Serial Data)

chanhvinh.com

(8) Multi-channel Formats

Some devices extend the definition of the specification by I^2S bus 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 18.4 shows the number of padding bits for each of the valid setting. If setting is not valid, "—" is indicated instead of a number.

Table 18.4 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

chanhvinh.com

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
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 in the order received from the SSIRDR register.

Figures 18.7 to 18.9 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, the second example is left-aligned and the third is right-aligned. The other conditions in these examples have been selected arbitrarily.

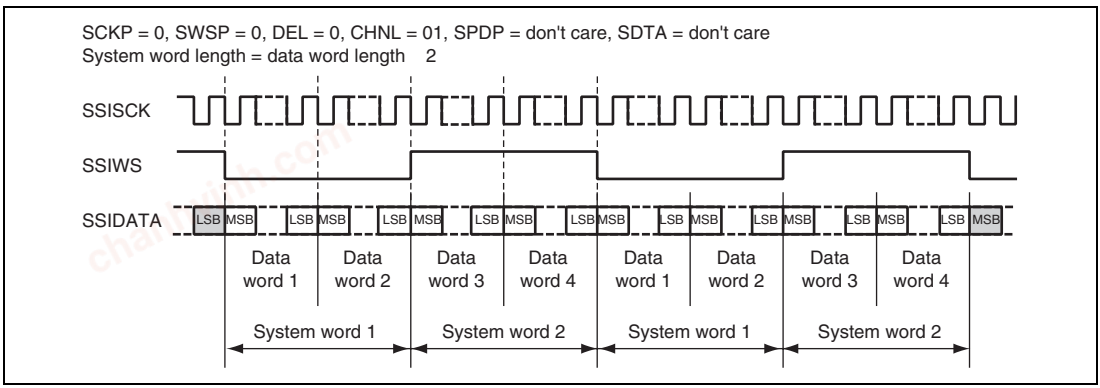


Figure 18.7 Multi-Channel Format (4 Channels Without Padding)

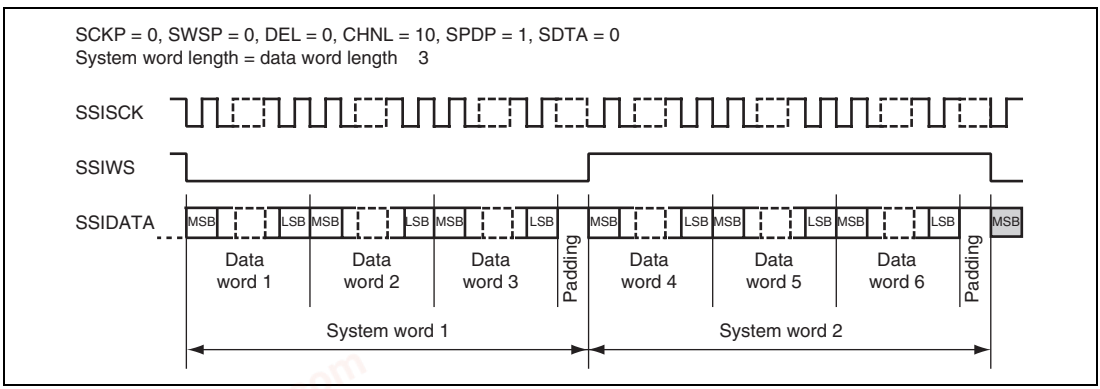


Figure 18.8 Multi-Channel Format (6 Channels with High Padding)

chanhvinh.com

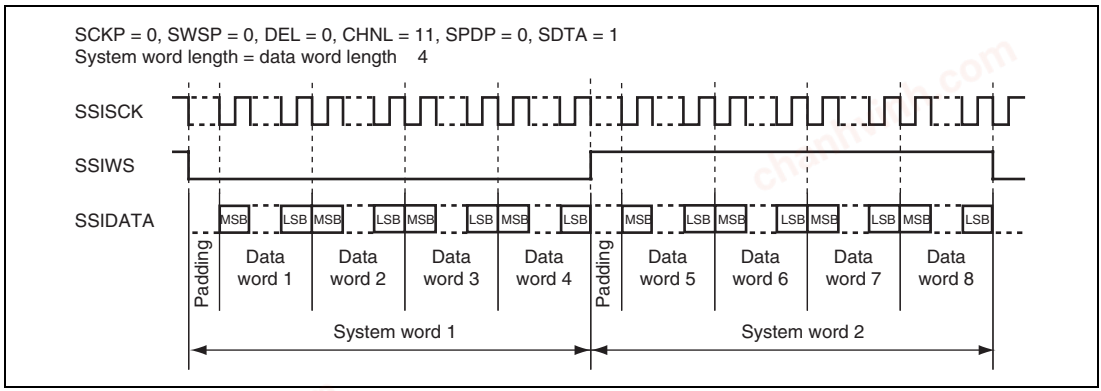


Figure 18.9 Multi-Channel Format (8 Channels; Transmitting and Receiving in the Order of Serial Data and Padding Bits; with Padding)

(9) Bit Setting Configuration Format

Several more configuration bits in non-compressed mode are shown below. These bits are not mutually exclusive, but some combinations may not be useful for any other device.

These configuration bits are described below with reference to figure 18.10.

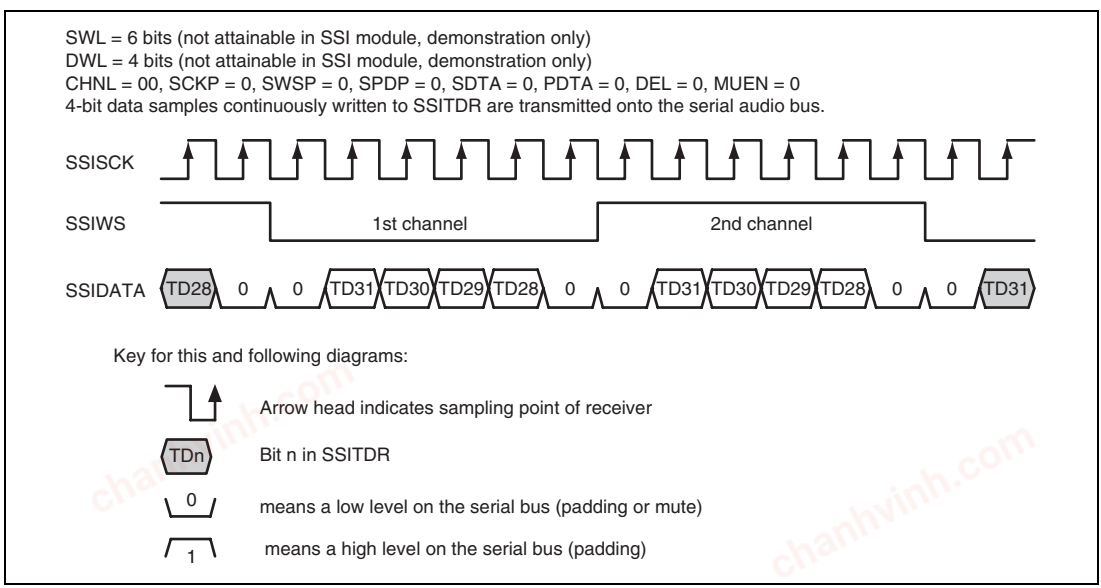


Figure 18.10 Basic Sample Format (Transmit Mode with Example System/Data Word Length)

chanhvinh.com

Figure 18.10 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 only for clarification of the other configuration bits.

- Inverted Clock

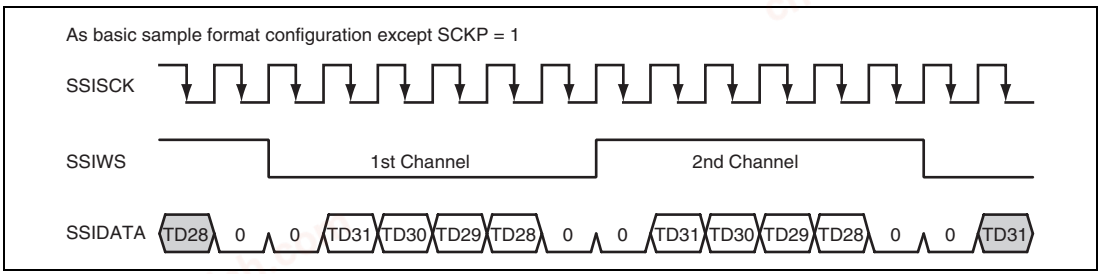


Figure 18.11 Inverted Clock

- Inverted Word Select

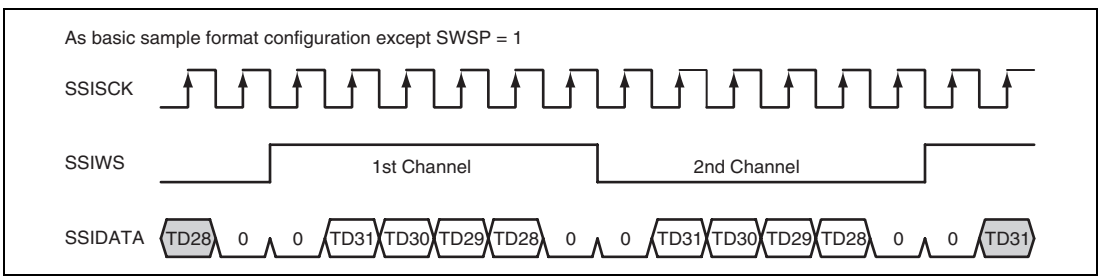


Figure 18.12 Inverted Word Select

- Inverted Padding Polarity

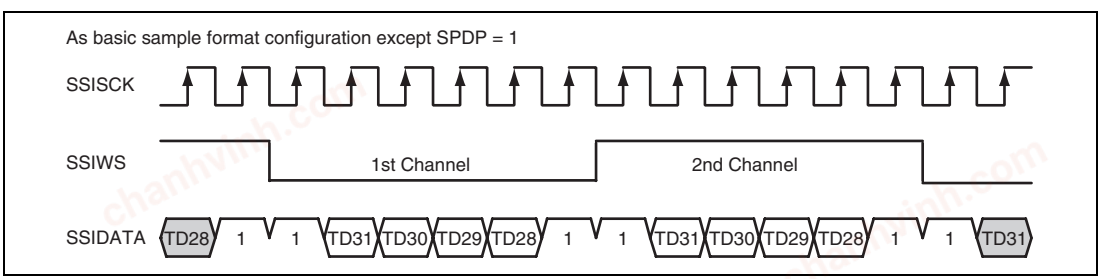


Figure 18.13 Inverted Padding Polarity

chanhvinh.com

- Transmitting and Receiving in the Order of Padding Bits and Serial Data; with Delay

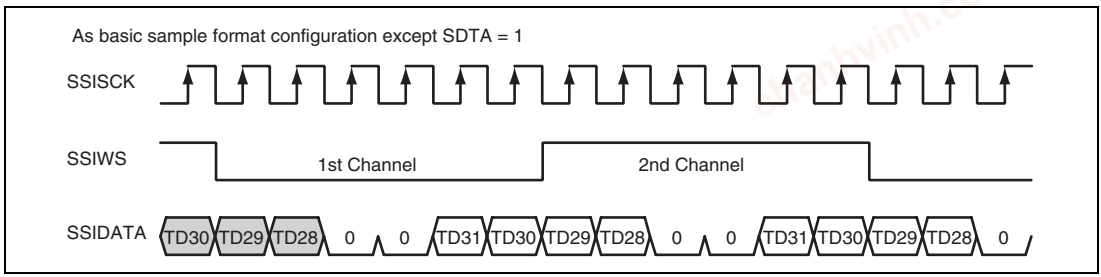


Figure 18.14 Transmitting and Receiving in the Order of Padding Bits and Serial Data; with Delay

- Transmitting and Receiving in the Order of Padding Bits and Serial Data; without Delay

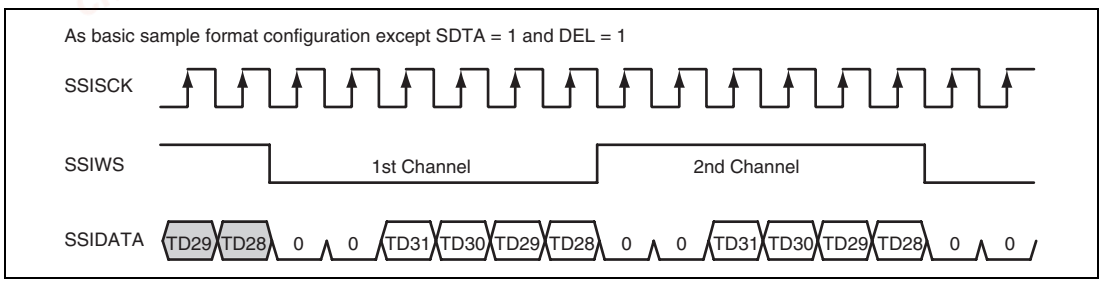


Figure 18.15 Transmitting and Receiving in the Order of Padding Bits and Serial Data; without Delay

- Transmitting and Receiving in the Order of Serial Data and Padding Bits; without Delay

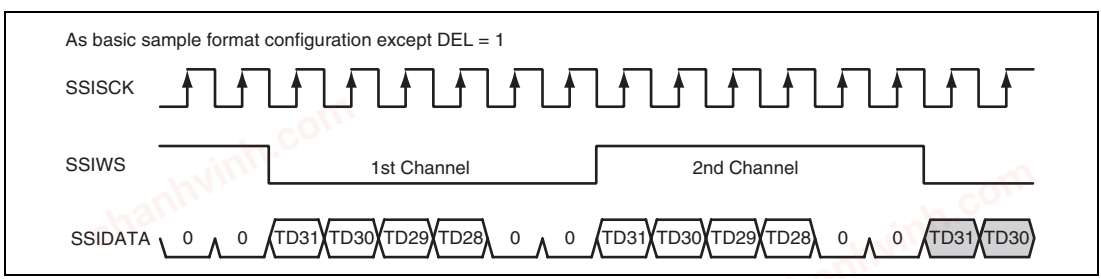


Figure 18.16 Transmitting and Receiving in the Order of Serial Data and Padding Bits; without Delay

chanhvinh.com

- Parallel Right-Aligned with Delay

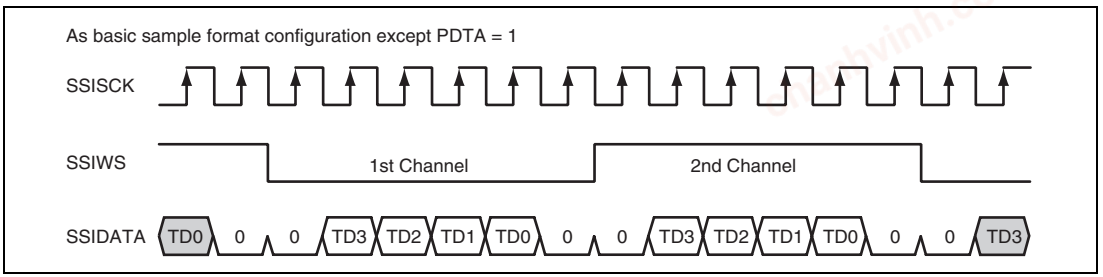


Figure 18.17 Parallel Right-Aligned with Delay

- Mute Enabled

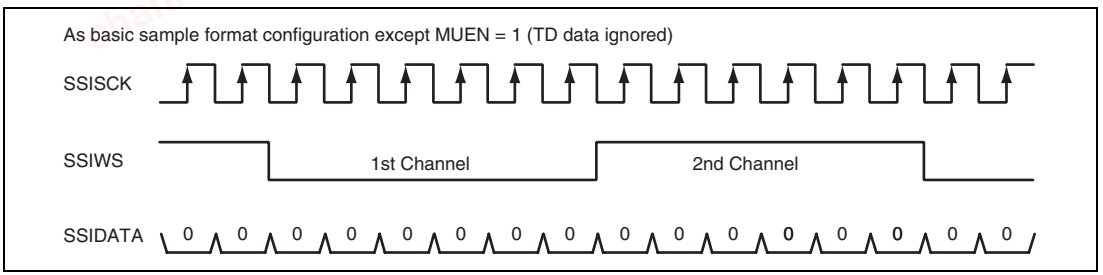


Figure 18.18 Mute Enabled

chanhvinh.com

18.4.3 Operation Modes

There are three modes of operation: configuration, enabled and disabled. Figure 18.19 shows how the module enters each of these modes.

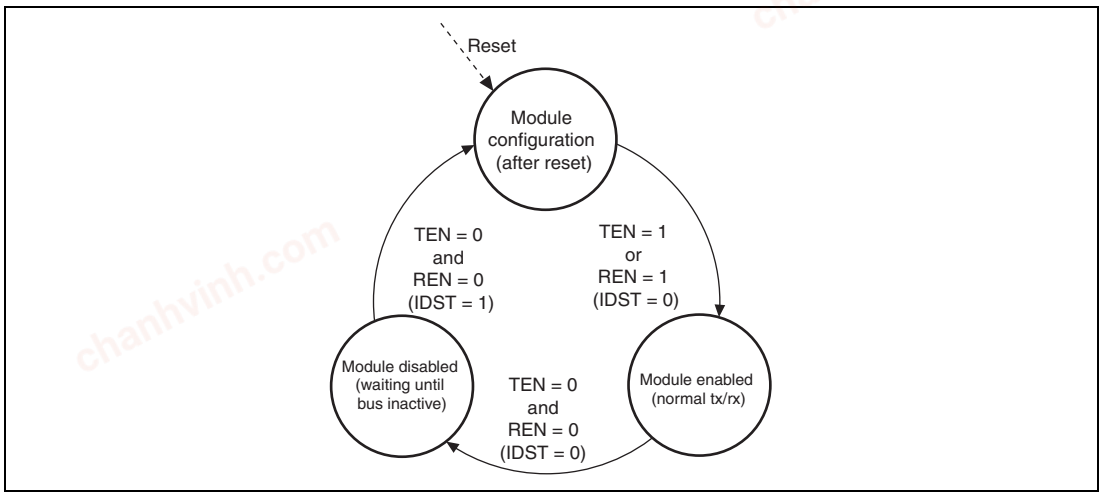


Figure 18.19 Operation Modes

(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 18.4.4, Transmit Operation and section 18.4.5, Receive Operation, below.

chanhvinh.com

18.4.4 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.

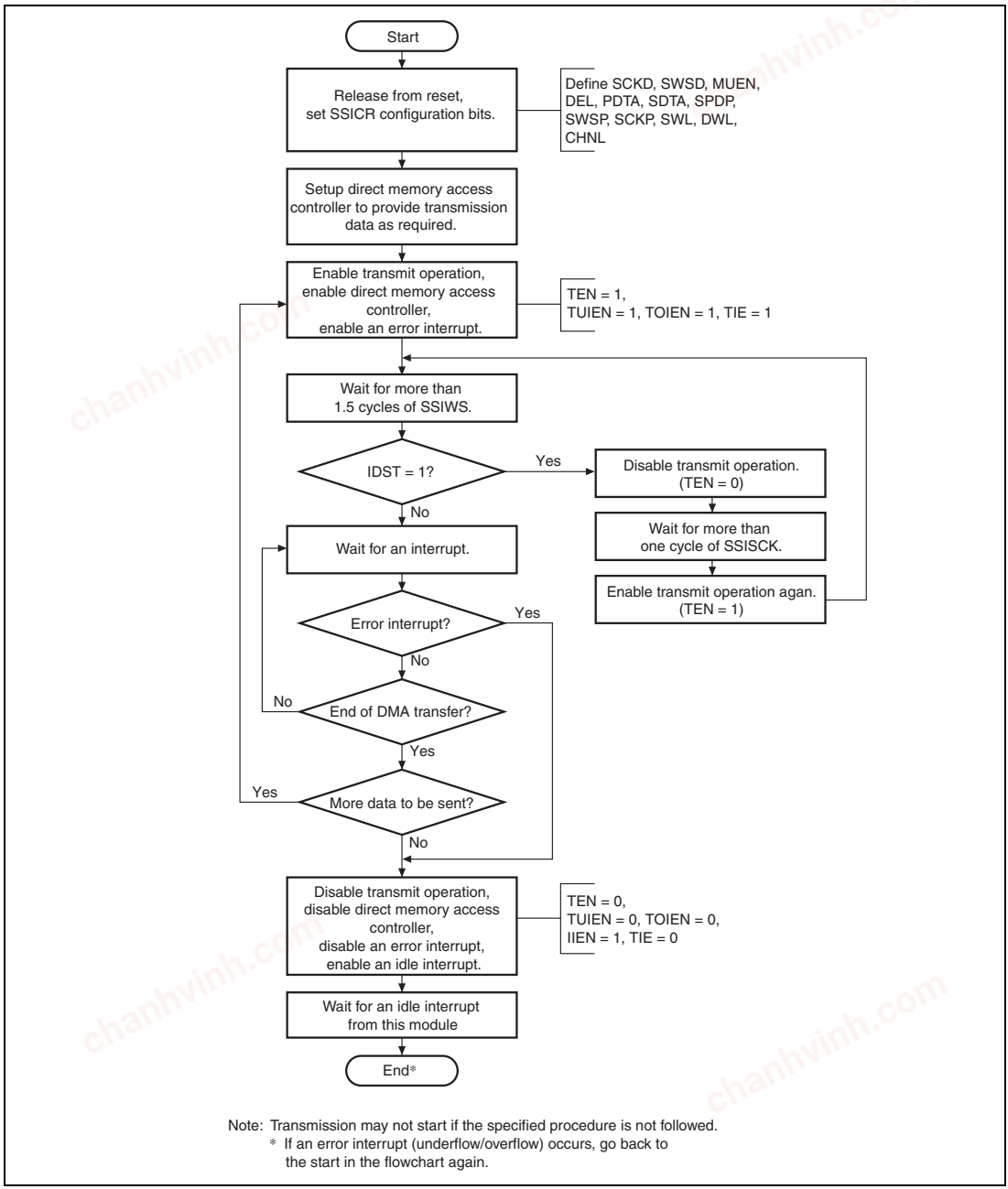
When disabling this module, the clock* must be kept supplied to this module until the IIRQ bit indicates that the module is in the idle state.

Figure 18.20 shows the transmit operation in DMA control mode, and figure 18.21 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

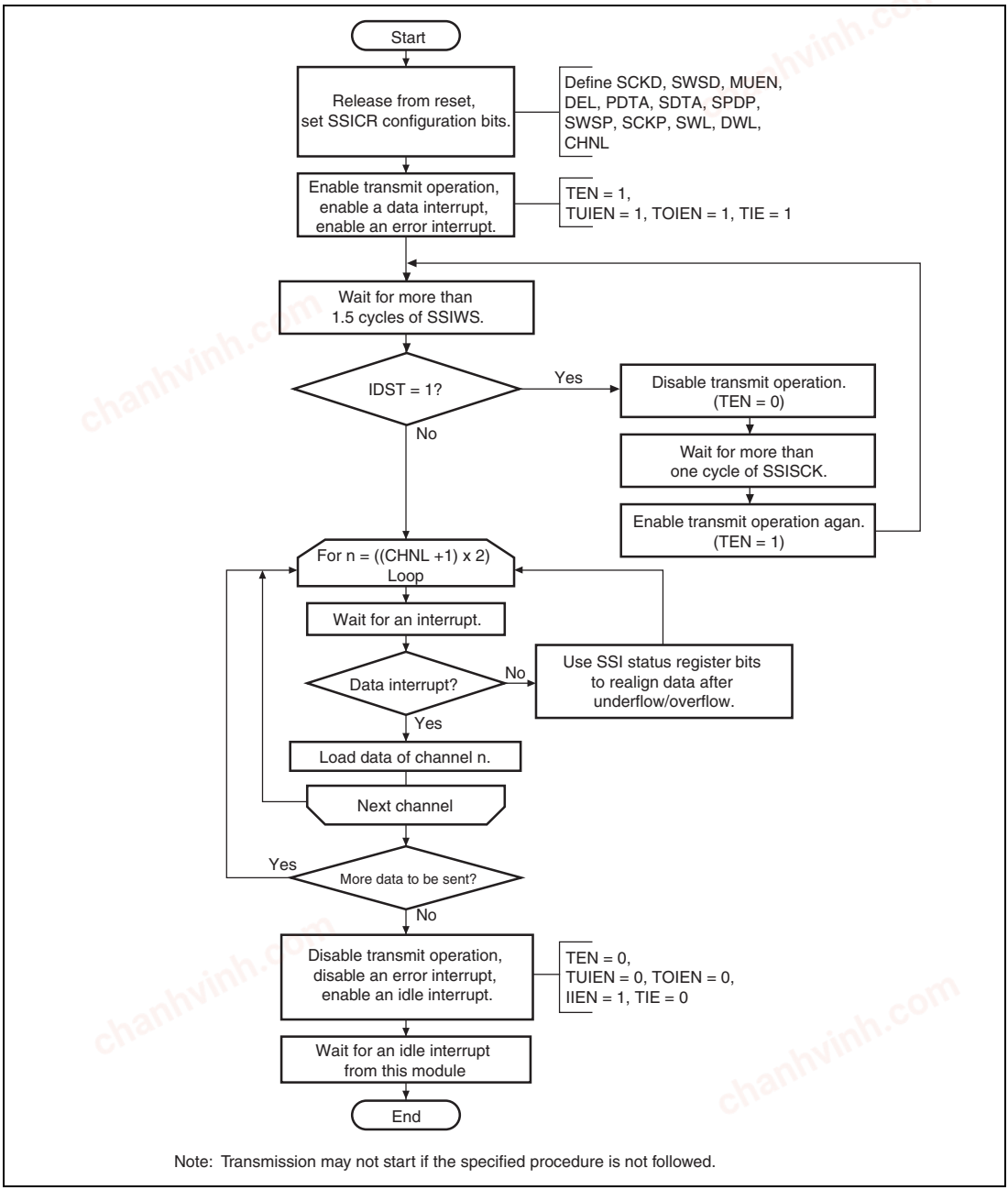


Note: Transmission may not start if the specified procedure is not followed.
* If an error interrupt (underflow/overflow) occurs, go back to the start in the flowchart again.

Figure 18.20 Transmission Using Direct Memory Access Controller

chanhvinh.com

(2) Transmission Using Interrupt-Driven Data Flow Control



Note: Transmission may not start if the specified procedure is not followed.

Figure 18.21 Transmission Using Interrupt-Driven Data Flow Control

chanhvinh.com

18.4.5 Receive Operation

Like transmission, reception can be controlled either by DMA transfer or interrupt.

Figures 18.22 and 18.23 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.

chanhvinh.com

(1) Reception Using Direct Memory Access Controller

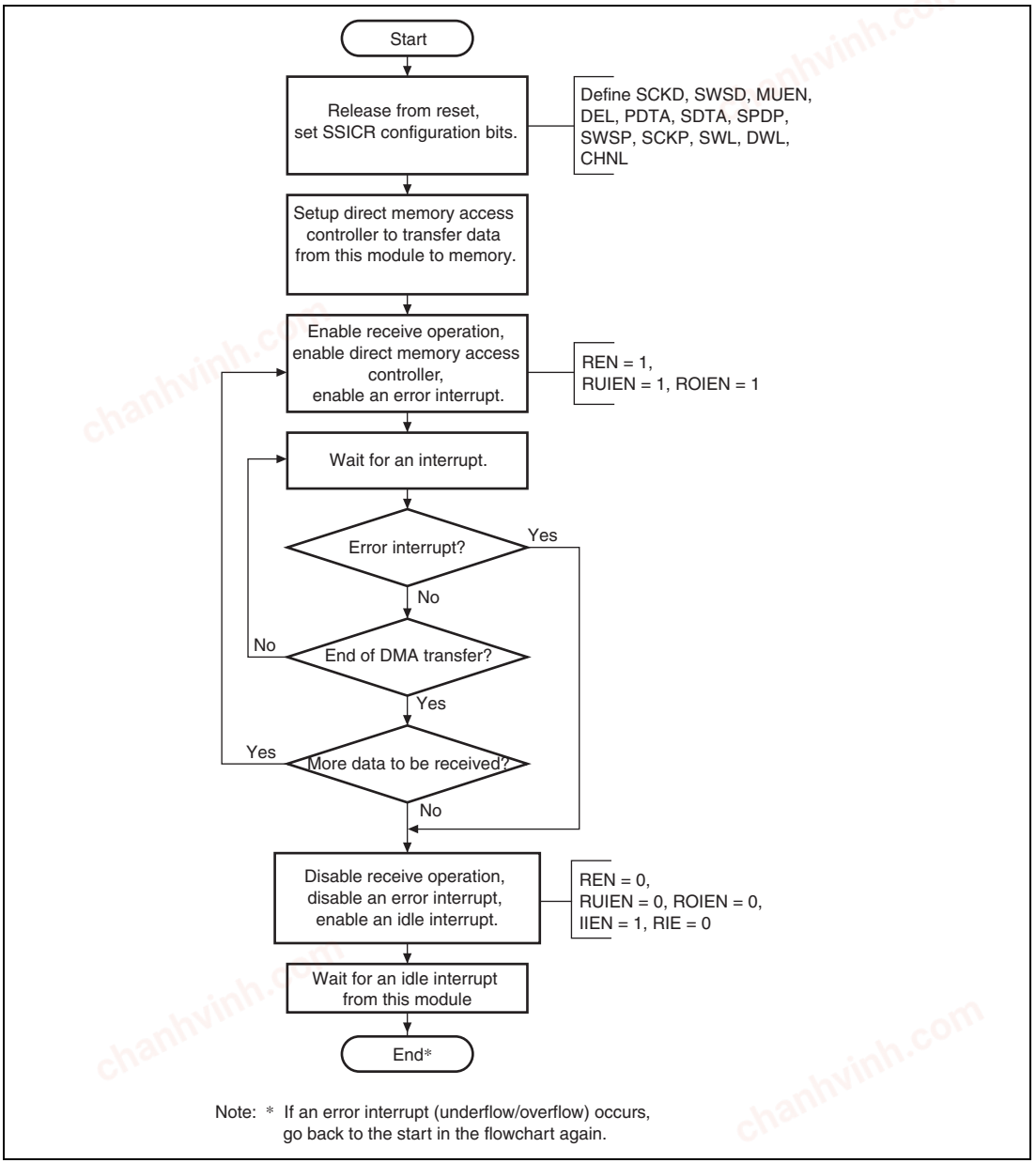


Figure 18.22 Reception Using Direct Memory Access Controller

chanhvinh.com

(2) Reception Using Interrupt-Driven Data Flow Control

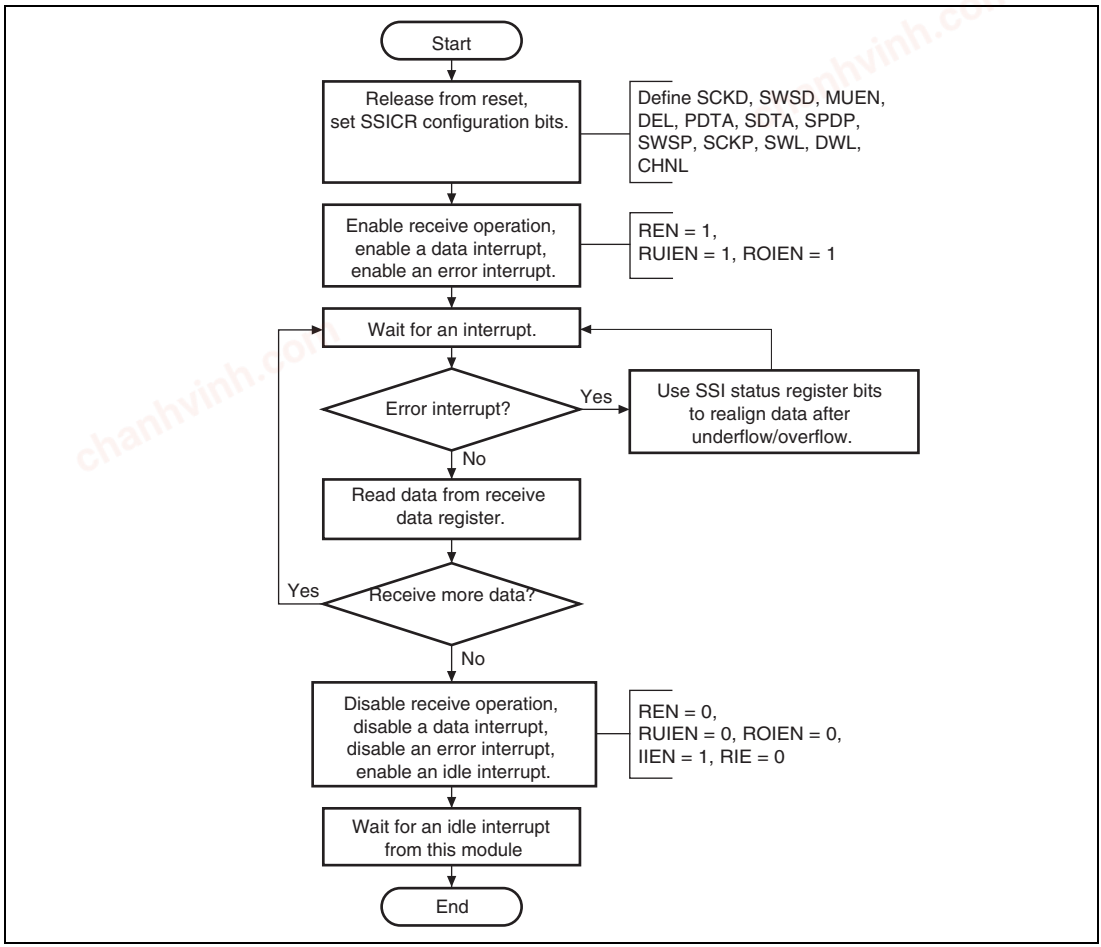


Figure 18.23 Reception Using Interrupt-Driven Data Flow Control

chanhvinh.com

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 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 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.

18.4.6 Serial Bit Clock Control

This function is used to control and select which clock is used for the serial bus interface.

If the serial 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 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 the module pin, SSISCK, is the same as the bit clock.

chanhvinh.com

18.5 Usage Notes

18.5.1 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 occurrence is confirmed with four types of error interrupts (transmit underflow, transmit overflow, receive underflow, and receive overflow) or the corresponding error status flag (the bits TUIRQ, TOIRQ, RUIRQ, and ROIRQ in SSISR), write 0 to the TEN or REN bit in SSICR to disable DMA transfer requests in this module, thus stopping the operation. (In this case, the direct memory access controller setting should also be stopped.) After this, write 0 to the error status flag bit to clear the error status, set the direct memory access controller again and restart the transfer.

18.5.2 Note on Changing Mode from Master Transceiver to Master Receiver

If a transmit underflow occurs in master transceiver mode 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 SSITDR to suppress transmit underflow.

chanhvinh.com

chanhvinh.com
chanhvinh.com
chanhvinh.com
chanhvinh.com

chanhvinh.com

Section 19 Serial I/O with FIFO

This LSI includes a clock-synchronized serial I/O module with FIFO.

19.1 Features

- Serial transfer
  - 16-stage 32-bit FIFOs (independent transmission and reception)
  - Supports 8-bit monaural/16-bit monaural/16-bit stereo audio input and output
  - MSB first for data transmission
  - Supports a maximum of 48-kHz sampling rate
  - Synchronization by frame synchronization pulse
  - Connectable to linear, audio, or A-Law or μ-Law CODEC chip
  - Supports both master and slave modes
- Serial clock
  - AUDIO_CLK or AUDIO_X1 can be selected as the clock source.
- Interrupts: One type
- DMA transfer: Two types
  - Transmit FIFO transfer requests and receive FIFO transfer requests

chanhvinh.com

Figure 19.1 shows a block diagram.

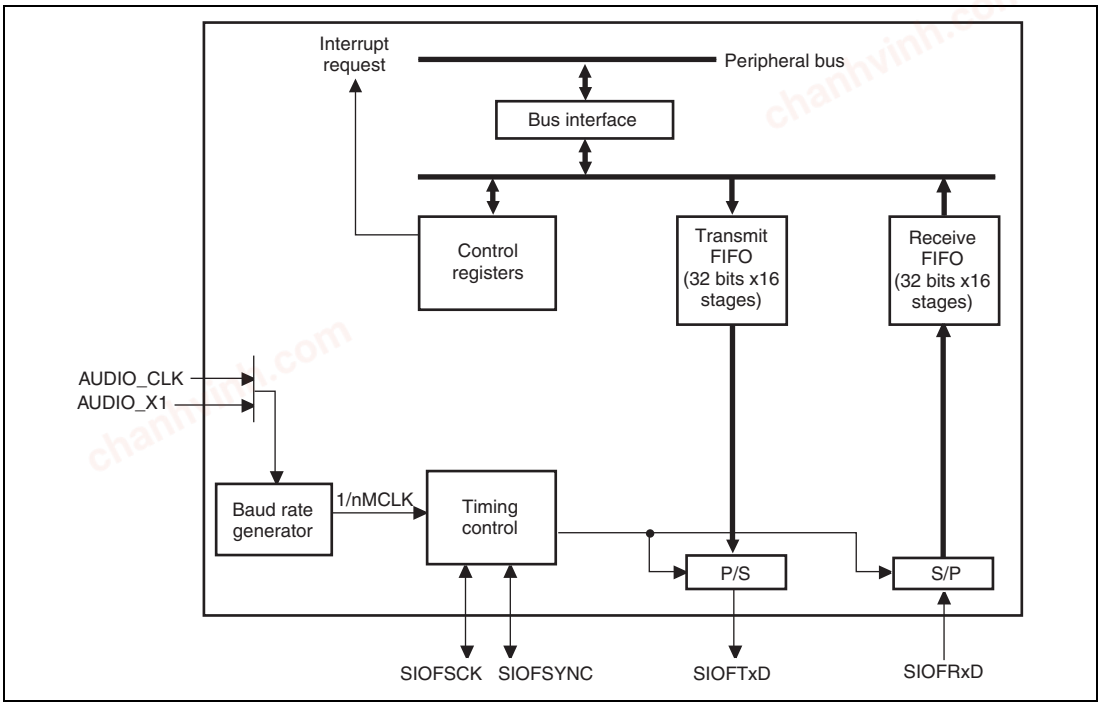


Figure 19.1 Block Diagram

chanhvinh.com

19.2 Input/Output Pins

Table 19.1 shows the pin configuration.

Table 19.1 Pin Configuration

Pin Name	I/O	Function
AUDIO_CLK	Input	External clock for audio
AUDIO_X1	Input	Crystal resonator/external clock for audio
AUDIO_X2	Output	
SIOFSCK	I/O	Serial clock (common to transmission/reception)
SIOFSYNC	I/O	Frame synchronous signal (common to transmission/reception)
SIOFTxD	Output	Transmit data
SIOFRxD	Input	Receive data

chanhvinh.com

19.3 Register Descriptions

Table 19.2 shows the register configuration.

Table 19.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Mode register	SIMDR	R/W	H'8000	H'FFFF4800	16
Clock select register	SISCR	R/W	H'8000	H'FFFF4802	16
Transmit data assign register	SITDAR	R/W	H'0000	H'FFFF4804	16
Receive data assign register	SIRDAR	R/W	H'0000	H'FFFF4806	16
Control register	SICTR	R/W	H'0000	H'FFFF480C	16
FIFO control register	SIFCTR	R/W*	H'1000	H'FFFF4810	16
Status register	SISTR	R/W*	H'0000	H'FFFF4814	16
Interrupt enable register	SIIER	R/W	H'0000	H'FFFF4816	16
Transmit data register	SITDR	W	Undefined	H'FFFF4820	8, 16, 32
Receive data register	SIRDR	R	Undefined	H'FFFF4824	8, 16, 32

Note: * This register has readable/writable bits and read-only bits. For details, see descriptions for each register.

chanhvinh.com

19.3.1 Mode Register (SIMDR)

SIMDR sets the operating mode for this module.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TRMD1	TRMD0	SYNCAT	REDG	FL3	FL2	FL1	FL0	TXDIZ	-	SYNCAC	SYNCDL	-	-	-	-
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	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	TRMD1	1	R/W	Transfer Mode 1, 0
14	TRMD0	0	R/W	Select transfer mode. 00: Slave mode 01: Setting prohibited 10: Master mode 11: Setting prohibited
13	SYNCAT	0	R/W	SIOFSYNC Pin Valid Timing Indicates the position where the SIOFSYNC signal is output. This bit is valid in master mode. 0: At the start-bit data of frame 1: At the last-bit data of slot Note: If this bit is set to 1, make sure that valid data is transmitted/received or transmitted.
12	REDG	0	R/W	Receive Data Sampling Edge This bit is valid in master mode. 0: The SIOFRxD signal is sampled at the falling edge of SIOFSCK (The SIOFTxD signal is transmitted at the rising edge of SIOFSCK.) 1: The SIOFRxD signal is sampled at the rising edge of SIOFSCK (The SIOFTxD signal is transmitted at the falling edge of SIOFSCK.)

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
11	FL3	0	R/W	Frame Length 3 to 0
10	FL2	0	R/W	00xx: Data length is 8 bits and frame length is 8 bits.
9	FL1	0	R/W	0100: Data length is 8 bits and frame length is 16 bits.
8	FL0	0	R/W	0101: Data length is 8 bits and frame length is 32 bits. 0110: Data length is 8 bits and frame length is 64 bits. 0111: Data length is 8 bits and frame length is 128 bits. 10xx: Data length is 16 bits and frame length is 16 bits. 1100: Data length is 16 bits and frame length is 32 bits. 1101: Data length is 16 bits and frame length is 64 bits. 1110: Data length is 16 bits and frame length is 128 bits. 1111: Data length is 16 bits and frame length is 256 bits. Note: When data length is specified as 8 bits, control data cannot be transmitted or received. x: Don't care
7	TXDIZ	0	R/W	SIOFTxD Pin Output when Transmission is Invalid* 0: High output (1 output) when invalid 1: High-impedance state when invalid Note: Invalid means when disabled, and when a slot that is not assigned as transmit data or control data is being output.
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5	SYNCAC	0	R/W	SIOFSYNC Pin Polarity This bit is valid in master mode. 0: Active-high 1: Active-low
4	SYNCDL	0	R/W	Data Pin Bit Delay for SIOFSYNC Pin Only 1-bit delay is valid in slave mode. 0: No bit delay 1: 1-bit delay

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

19.3.2 Control Register (SICTR)

SICTR sets the operating state for this module.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCKE	FSE	-	-	-	-	TXE	RXE	-	-	-	-	-	-	TXRST	RXRST
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/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	SCKE	0	R/W	Serial Clock Output Enable This bit is valid in master mode. 0: Disables the SIOFSCK output (outputs 0) 1: Enables the SIOFSCK output • If this bit is set to 1, this module initializes the baud rate generator and initiates the operation. At the same time, the clock generated by the baud rate generator is output to the SIOFSCK pin.
14	FSE	0	R/W	Frame Synchronous Signal Output Enable This bit is valid in master mode. 0: Disables the SIOFSYNC output (outputs 0) 1: Enables the SIOFSYNC output • If this bit is set to 1, this module initializes the frame counter and initiates the operation.
13 to 10	—	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
9	TXE	0	R/W	Transmit Enable 0: Disables data transmission from the SIOFTxD pin 1: Enables data transmission from the SIOFTxD pin • This bit setting becomes valid at the start of the next frame (at the rising edge of the SIOFSYNC signal). • When the 1 setting for this bit becomes valid, this module issues a transmit transfer request according to the setting of the TFWM bit in SIFCTR. When transmit data is stored in the transmit FIFO, transmission of data from the SIOFTxD pin begins. • This bit is initialized upon a transmit reset.
8	RXE	0	R/W	Receive Enable 0: Disables data reception from SIOFRxD 1: Enables data reception from SIOFRxD • This bit setting becomes valid at the start of the next frame (at the rising edge of the SIOFSYNC signal). • When the 1 setting for this bit becomes valid, this module begins the reception of data from the SIOFRxD pin. When receive data is stored in the receive FIFO, a reception transfer request is issued according to the setting of the RFWM bit in SIFCTR. • This bit is initialized upon receive reset.
7 to 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	TXRST	0	R/W	Transmit Reset 0: Does not reset transmit operation 1: Resets transmit operation • This bit setting becomes valid immediately. This bit should be cleared to 0 before setting the register to be initialized. • When the 1 setting for this bit becomes valid, this module immediately sets the SIOFTxD pin output to 1, and initializes the following registers and data: — SITDR — Valid data in transmit FIFO — The TFEMP and TDREQ bits in SISTR — The TXE bit Note: Set this bit to 1 for more than one transfer clock period.
0	RXRST	0	R/W	Receive Reset 0: Does not reset receive operation 1: Resets receive operation • This bit setting becomes valid immediately. This bit should be cleared to 0 before setting the register to be initialized. • When the 1 setting for this bit becomes valid, this module immediately disables reception from the SIOFRxD pin, and initializes the following registers and data: — SIRDR — Valid data in receive FIFO — The RFFUL and RDREQ bits in SISTR — The RXE bit Note: Set this bit to 1 for more than one transfer clock period.

chanhvinh.com

19.3.3 Transmit Data Register (SITDR)

SITDR specifies transmit data. The data set in SITDR will be stored in the transmit FIFO.

SITDR is initialized by a transmit reset caused by the TXRST bit in SICTR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SITDL[15:0]															
Initial Value:	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
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
	SITDR[15:0]															
Initial Value:	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
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 16	SITDL [15:0]	Undefined	W	Left-Channel Transmit Data Specify data to be transmitted from the SIOFTxD pin as left-channel data. The position of the left-channel data in the transmit frame is specified by the TDLA bit in SITDAR. • These bits are valid only when the TDLE bit in SITDAR is set to 1.
15 to 0	SITDR [15:0]	Undefined	W	Right-Channel Transmit Data Specify data to be transmitted from the SIOFTxD pin as right-channel data. The position of the right-channel data in the transmit frame is specified by the TDRA bit in SITDAR. • These bits are valid only when the TDRE bit is set to 1 and the TLREP bit is cleared to 0 in SITDAR.

chanhvinh.com

19.3.4 Receive Data Register (SIRDR)

SIRDR reads receive data of this module. SIRDR stores data in the receive FIFO.

SIRDR is initialized by a receive reset caused by the RXRST bit in SICTR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SIRDL[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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SIRDR[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 16	SIRDL [15:0]	Undefined	R	Left-Channel Receive Data Store data received from the SIOFRxD pin as left-channel data. The position of the left-channel data in the receive frame is specified by the RDLA bit in SIRDAR. • These bits are valid only when the RDLE bit in SIRDAR is set to 1.
15 to 0	SIRDR [15:0]	Undefined	R	Right-Channel Receive Data Store data received from the SIOFRxD pin as right-channel data. The position of the right-channel data in the receive frame is specified by the RDRA bit in SIRDAR. • These bits are valid only when the RDRE bit in SIRDAR is set to 1.

chanhvinh.com

19.3.5 Status Register (SISTR)

SISTR shows the state of this module. Each bit in this register becomes an interrupt source for this module when the corresponding bit in SIIER is set to 1.

SISTR is initialized in module stop mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	TFEMP	TDREQ	-	-	RFFUL	RDREQ	-	-	-	FSERR	TFOVF	TFUDF	RFUDF	RFOVF
Initial Value:	0	0	0	0	0	0	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
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	—	0	R	Reserved The read value is undefined. The write value should always be 0.
13	TFEMP	0	R	Transmit FIFO Empty 0: Indicates that transmit FIFO is not empty 1: Indicates that transmit FIFO is empty • This bit is valid when the TXE bit in SICTR is 1. • If SITDR is written, this module clears this bit. Note: When this bit is set to 1, a transmit FIFO underflow may have occurred. Do not use this bit at the timing of writing to the transmit data register.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
12	TDREQ	0	R	Transmit Data Transfer Request 0: Indicates that the size of empty space in the transmit FIFO is less than the size specified by the TFWM bit in SIFCTR. 1: Indicates that the size of empty space in the transmit FIFO is equal to or greater than the size specified by the TFWM bit in SIFCTR. A transmit data transfer request is issued when the empty space in the transmit FIFO exceeds the size specified by the TFWM bit in SIFCTR. When transmit data is transferred through the direct memory access controller, this bit is always cleared by an access of the direct memory access controller. If the condition for setting this bit is satisfied after the access of the direct memory access controller, this module again sets this bit to 1. • This bit is valid when the TXE bit in SICTR is 1. • If the size of empty space in the transmit FIFO is less than the size specified by the TFWM bit in SIFCTR, this module clears this bit.
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	RFFUL	0	R	Receive FIFO Full 0: Receive FIFO not full 1: Receive FIFO full • This bit is valid when the RXE bit in SICTR is 1. • If SIRDR is read, this module clears this bit.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
8	RDREQ	0	R	Receive Data Transfer Request 0: Indicates that the size of valid space in the receive FIFO is less than the size specified by the RFWM bit in SIFCTR. 1: Indicates that the size of valid space in the receive FIFO is equal to or greater than the size specified by the RFWM bit in SIFCTR. A receive data transfer request is issued when the valid space in the receive FIFO exceeds the size specified by the RFWM bit in SIFCTR. When receive data is transferred through the direct memory access controller, this bit is always cleared by an access of the direct memory access controller. If the condition for setting this bit is satisfied after the access of the direct memory access controller, this module again sets this bit to 1. • This bit is valid when the RXE bit in SICTR is 1. • If the size of valid space in the receive FIFO is less than the size specified by the RFWM bit in SIFCTR, this module clears this bit.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	FSERR	0	R/W	Frame Synchronization Error 0: Indicates that no frame synchronization error occurs 1: Indicates that a frame synchronization error occurs A frame synchronization error occurs when the next frame synchronization timing appears before the previous data transfer has been completed. If a frame synchronization error occurs, this module performs transmission or reception for slots that can be transferred. • This bit is valid when the TXE or RXE bit in SICTR is 1. • When this bit is set to 1, it is cleared to 0 by this module. Writing 0 to this bit is invalid.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
3	TFOVF	0	R/W	Transmit FIFO Overflow 0: No transmit FIFO overflow 1: Transmit FIFO overflow A transmit FIFO overflow means that there has been an attempt to write to SITDR when the transmit FIFO is full. When an overflow of the transmit FIFO occurs, the write which caused the overflow is invalid. • This bit is valid when the TXE bit in SICTR is 1. • When this bit is set to 1, it is cleared to 0 by this module. Writing 0 to this bit is invalid.
2	TFUDF	0	R/W	Transmit FIFO Underflow 0: No transmit FIFO underflow 1: Transmit FIFO underflow A transmit FIFO underflow means that loading for transmission has occurred when the transmit FIFO is empty. When a transmit FIFO underflow occurs, this module repeatedly sends the previous transmit data. • This bit is valid when the TXE bit in SICTR is 1. • When this bit is set to 1, it is cleared to 0 by this module. Writing 0 to this bit is invalid.
1	RFUDF	0	R/W	Receive FIFO Underflow 0: No receive FIFO underflow 1: Receive FIFO underflow A receive FIFO underflow means that reading of SIRDR has occurred when the receive FIFO is empty. When a receive FIFO underflow occurs, the value of data read from SIRDR is not guaranteed. • This bit is valid when the RXE bit in SICTR is 1. • When this bit is set to 1, it is cleared to 0 by this module. Writing 0 to this bit is invalid.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
0	RFOVF	0	R/W	Receive FIFO Overflow 0: No receive FIFO overflow 1: Receive FIFO overflow A receive FIFO overflow means that writing has occurred due to reception operation when the receive FIFO is full. When an overflow of the receive FIFO occurs, the receive data which caused the overflow is lost. • When this bit is set to 1, it is cleared to 0 by this module. Writing 0 to this bit is invalid.

chanhvinh.com

19.3.6 Interrupt Enable Register (SIIER)

SIIER enables the issue of interrupts from this module. When a bit in this register is set to 1 and the corresponding bit in SISTR is set to 1, this module issues an interrupt.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TDMAE	-	TFEMPE	TDREQE	RDMAE	-	RFFULE	RDREQE	-	-	-	FSERRE	TFOVFE	TFUDFE	RFUDFE	RFOVFE
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R/W	R/W	R/W	R	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
15	TDMAE	0	R/W	Transmit FIFO DMA Transfer Request Enable Uses a transmit FIFO transfer request as an interrupt or a DMA transfer request. 0: Used as an interrupt to the CPU 1: Used as a DMA transfer request to the direct memory access controller
14	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
13	TFEMPE	0	R/W	Transmit FIFO Empty Enable 0: Disables interrupts due to transmit FIFO empty 1: Enables interrupts due to transmit FIFO empty
12	TDREQE	0	R/W	Transmit FIFO Transfer Request Enable 0: Disables interrupts/DMA transfer requests due to transmit FIFO transfer requests 1: Enables interrupts/DMA transfer requests due to transmit FIFO transfer requests
11	RDMAE	0	R/W	Receive FIFO DMA Transfer Request Enable Uses a receive FIFO transfer request as an interrupt or a DMA transfer request. 0: Used as a CPU interrupt 1: Used as a DMA transfer request to the direct memory access controller
10	—	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
9	RFFULE	0	R/W	Receive FIFO Full Enable 0: Disables interrupts due to receive FIFO full 1: Enables interrupts due to receive FIFO full
8	RDREQE	0	R/W	Receive FIFO Transfer Request Enable 0: Disables interrupts/DMA transfer requests due to receive FIFO transfer requests 1: Enables interrupts/DMA transfer requests due to receive FIFO transfer requests
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	FSERRE	0	R/W	Frame Synchronization Error Enable 0: Disables interrupts due to frame synchronization error 1: Enables interrupts due to frame synchronization error
3	TFOVFE	0	R/W	Transmit FIFO Overflow Enable 0: Disables interrupts due to transmit FIFO overflow 1: Enables interrupts due to transmit FIFO overflow
2	TFUDFE	0	R/W	Transmit FIFO Underflow Enable 0: Disables interrupts due to transmit FIFO underflow 1: Enables interrupts due to transmit FIFO underflow
1	RFUDFE	0	R/W	Receive FIFO Underflow Enable 0: Disables interrupts due to receive FIFO underflow 1: Enables interrupts due to receive FIFO underflow
0	RFOVFE	0	R/W	Receive FIFO Overflow Enable 0: Disables interrupts due to receive FIFO overflow 1: Enables interrupts due to receive FIFO overflow

chanhvinh.com

19.3.7 FIFO Control Register (SIFCTR)

SIFCTR indicates the area available for the transmit/receive FIFO transfer.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TFWM2	TFWM1	TFWM0	TFUA4	TFUA3	TFUA2	TFUA1	TFUA0	RFWM2	RFWM1	RFWM0	RFUA4	RFUA3	RFUA2	RFUA1	RFUA0
Initial Value:	0	0	0	1	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/W	R/W	R/W	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	TFWM2	0	R/W	Transmit FIFO Watermark
14	TFWM1	0	R/W	000: Issue a transfer request when 16 stages of the transmit FIFO are empty.
13	TFWM0	0	R/W	001: Setting prohibited 010: Setting prohibited 011: Setting prohibited 100: Issue a transfer request when 12 or more stages of the transmit FIFO are empty. 101: Issue a transfer request when 8 or more stages of the transmit FIFO are empty. 110: Issue a transfer request when 4 or more stages of the transmit FIFO are empty. 111: Issue a transfer request when 1 or more stages of transmit FIFO are empty. • A transfer request to the transmit FIFO is issued by the TDREQE bit in SISTR. • The transmit FIFO is always used as 16 stages of the FIFO regardless of these bit settings.
12	TFUA4	1	R	Transmit FIFO Usable Area
11	TFUA3	0	R	Indicate the number of stages of FIFO that can be transferred as B'00000 (full) to B'10000 (empty).
10	TFUA2	0	R	
9	TFUA1	0	R	
8	TFUA0	0	R	

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
7	RFWM2	0	R/W	Receive FIFO Watermark
6	RFWM1	0	R/W	000: Issue a transfer request when 1 stage or more of the receive FIFO are valid.
5	RFWM0	0	R/W	001: Setting prohibited 010: Setting prohibited 011: Setting prohibited 100: Issue a transfer request when 4 or more stages of the receive FIFO are valid. 101: Issue a transfer request when 8 or more stages of the receive FIFO are valid. 110: Issue a transfer request when 12 or more stages of the receive FIFO are valid. 111: Issue a transfer request when 16 stages of the receive FIFO are valid. • A transfer request to the receive FIFO is issued by the RDREQE bit in SISTR. • The receive FIFO is always used as 16 stages of the FIFO regardless of these bit settings.
4	RFUA4	0	R	Receive FIFO Usable Area
3	RFUA3	0	R	Indicate the number of stages of FIFO that can be transferred as B'00000 (empty) to B'10000 (full).
2	RFUA2	0	R	
1	RFUA1	0	R	
0	RFUA0	0	R	

chanhvinh.com

19.3.8 Clock Select Register (SISCR)

SISCR sets the serial clock generation conditions for the master clock. SISCR can be specified when the TRMD1 and TRMD0 bits in SIMDR are specified as B'10.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MSSEL	-	-	BRPS4	BRPS3	BRPS2	BRPS1	BRPS0	-	-	-	-	-	BRDV2	BRDV1	BRDV0
Initial Value:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R/W	R/W	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
15	MSSEL	1	R/W	Master Clock Source Selection 0: Uses AUDIO_X1 as the master clock 1: Uses AUDIO_CLK as the master clock The master clock is the clock input to the baud rate generator.
14, 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	BRPS4	0	R/W	Prescalar Setting Set the master clock division ratio according to the count value of the prescalar of the baud rate generator. The range of settings is from B'00000 (× 1/1) to B'11111 (× 1/32).
11	BRPS3	0	R/W	
10	BRPS2	0	R/W	
9	BRPS1	0	R/W	
8	BRPS0	0	R/W	
7 to 3	—	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
2	BRDV2	0	R/W	Baud rate generator's Division Ratio Setting
1	BRDV1	0	R/W	Set the frequency division ratio for the output stage of the baud rate generator.
0	BRDV0	0	R/W	000: Prescalar output × 1/2 001: Prescalar output × 1/4 010: Prescalar output × 1/8 011: Prescalar output × 1/16 100: Prescalar output × 1/32 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited The final frequency division ratio of the baud rate generator is determined by BRPS × BRDV (maximum 1/1024).

19.3.9 Transmit Data Assign Register (SITDAR)

SITDAR specifies the position of the transmit data in a frame (slot number).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TDLE	-	-	-	TDLA3	TDLA2	TDLA1	TDLA0	TDRE	TLREP	-	-	TDRA3	TDRA2	TDRA1	TDRA0
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/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	TDLE	0	R/W	Transmit Left-Channel Data Enable 0: Disables left-channel data transmission 1: Enables left-channel data transmission
14 to 12	—	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
11	TDLA3	0	R/W	Transmit Left-Channel Data Assigns 3 to 0
10	TDLA2	0	R/W	Specify the position of left-channel data in a transmit frame as B'0000 (0) to B'1110 (14).
9	TDLA1	0	R/W	1111: Setting prohibited
8	TDLA0	0	R/W	• Transmit data for the left channel is specified in the SITDL bit in SITDR.
7	TDRE	0	R/W	Transmit Right-Channel Data Enable 0: Disables right-channel data transmission 1: Enables right-channel data transmission
6	TLREP	0	R/W	Transmit Left-Channel Repeat 0: Transmits data specified in the SITDR bit in SITDR as right-channel data 1: Repeatedly transmits data specified in the SITDL bit in SITDR as right-channel data • This bit setting is valid when the TDRE bit is set to 1. • When this bit is set to 1, the SITDR settings are ignored.
5, 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	TDRA3	0	R/W	Transmit Right-Channel Data Assigns 3 to 0
2	TDRA2	0	R/W	Specify the position of right-channel data in a transmit frame as B'0000 (0) to B'1110 (14).
1	TDRA1	0	R/W	1111: Setting prohibited
0	TDRA0	0	R/W	• Transmit data for the right channel is specified in the SITDR bit in SITDR.

chanhvinh.com

19.3.10 Receive Data Assign Register (SIRDAR)

SIRDAR specifies the position of the receive data in a frame (slot number).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDLE	-	-	-	RDLA3	RDLA2	RDLA1	RDLA0	RDRE	-	-	-	RDRA3	RDRA2	RDRA1	RDRA0
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/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	RDLE	0	R/W	Receive Left-Channel Data Enable 0: Disables left-channel data reception 1: Enables left-channel data reception
14 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	RDLA3	0	R/W	Receive Left-Channel Data Assigns 3 to 0
10	RDLA2	0	R/W	Specify the position of left-channel data in a receive frame as B'0000 (0) to B'1110 (14).
9	RDLA1	0	R/W	1111: Setting prohibited
8	RDLA0	0	R/W	• Receive data for the left channel is stored in the SIRDL bit in SIRDR.
7	RDRE	0	R/W	Receive Right-Channel Data Enable 0: Disables right-channel data reception 1: Enables right-channel data reception
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	RDRA3	0	R/W	Receive Right-Channel Data Assigns 3 to 0
2	RDRA2	0	R/W	Specify the position of right-channel data in a receive frame as B'0000 (0) to B'1110 (14).
1	RDRA1	0	R/W	1111: Setting prohibited
0	RDRA0	0	R/W	• Receive data for the right channel is stored in the SIRDR bit in SIRDR.

chanhvinh.com

19.4 Operation

19.4.1 Serial Clocks

(1) Master/Slave Modes

The following two modes are available as a clock mode for this module.

- Slave mode: SIOFSCK, SIOFSYNC input
- Master mode: SIOFSCK, SIOFSYNC output

(2) Baud Rate Generator: In master mode, the baud rate generator (BRG) is used to generate the serial clock. The division ratio is from 1/2 to 1/1024.

Figure 19.2 shows connections for supply of the serial clock.

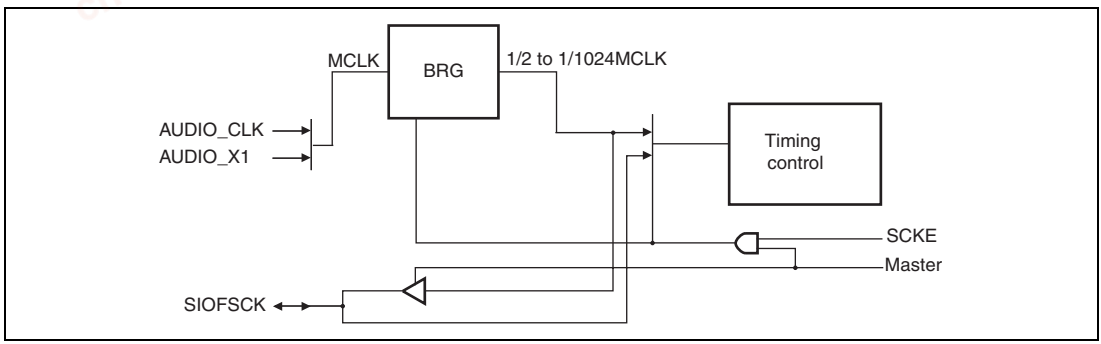


Figure 19.2 Serial Clock Supply

Table 19.3 shows an example of serial clock frequency.

Table 19.3 Serial Clock Frequency

	Sampling Rate		
Frame Length	**8 kHz**	**44.1 kHz**	**48 kHz**
32 bits	256 kHz	1.4112 MHz	1.536 MHz
64 bits	512 kHz	2.8224 MHz	3.072 MHz
128 bits	1.024 MHz	5.6448 MHz	6.144 MHz
256 bits	2.048 MHz	11.289 MHz	12.289 MHz

chanhvinh.com

19.4.2 Serial Timing

(1) SIOFSYNC

The SIOFSYNC is a frame synchronous signal.

Figure 19.3 shows the SIOFSYNC synchronization timing.

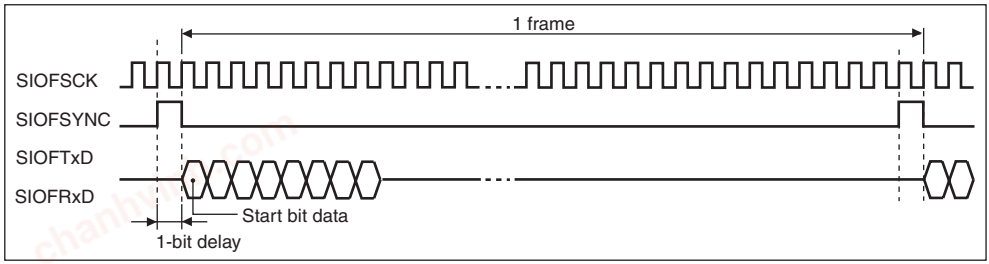


Figure 19.3 Serial Data Synchronization Timing

chanhvinh.com

(2) Transmit/Receive Timing

The SIOFTxD transmit timing and SIOFRxD receive timing relative to the SIOFSCK can be set as the sampling timing in the following ways. The transmit/receive timing is set using the REDG bit in SIMDR.

- Falling-edge sampling
- Rising-edge sampling (possible only in master mode)

Figure 19.4 shows the transmit/receive timing.

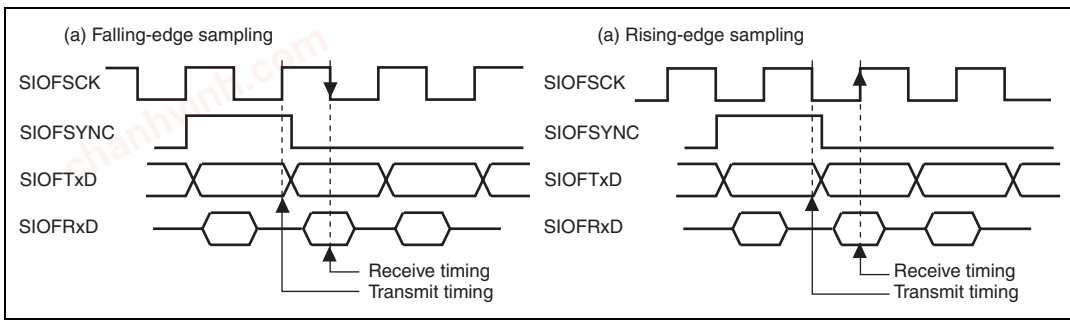


Figure 19.4 Transmit/Receive Timing

19.4.3 Transfer Data Format

This module performs the following transfer.

- Transmit/receive data: Transfer of 8-bit monaural/16-bit monaural/16-bit stereo data

(1) Transfer Mode

This module supports the following two transfer modes as listed in table 19.4. The transfer mode can be specified by the TRMD1 and TRMD0 bits in SIMDR.

Table 19.4 Serial Transfer Modes

Transfer Mode	SIOFSYNC	Bit Delay
Slave mode	Synchronous pulse	SYNCDL bit
Master mode		

chanhvinh.com

(2) Frame Length

The length of the frame to be transferred by this module is specified with the FL3 to FL0 bits in SIMDR. Table 19.5 shows the relationship between the FL3 to FL0 bit settings and frame length.

Table 19.5 Frame Length

FL3 to FL0	Slot Length	Number of Bits in a Frame	Transfer Data
00xx	8	8	8-bit monaural data
0100	8	16	8-bit monaural data
0101	8	32	8-bit monaural data
0110	8	64	8-bit monaural data
0111	8	128	8-bit monaural data
10xx	16	16	16-bit monaural data
1100	16	32	16-bit monaural/stereo data
1101	16	64	16-bit monaural/stereo data
1110	16	128	16-bit monaural/stereo data
1111	16	256	16-bit monaural/stereo data

Note: x: Don't care.

(3) Slot Position

This module can specify the position of transmit data and receive data in a frame by slot numbers. The slot number of each data is specified by the following registers.

- Transmit data: SITDAR
- Receive data: SIRDAR

19.4.4 Register Allocation of Transfer Data

Writing and reading of transmit/receive data is performed for the following registers.

- Transmit data writing: SITDR (8-, 16-, or 32-bit access)
- Receive data reading: SIRDR (8-, 16-, or 32-bit access)

Figure 19.5 shows the transmit/receive data and the SITDR and SIRDR bit alignment.

chanhvinh.com

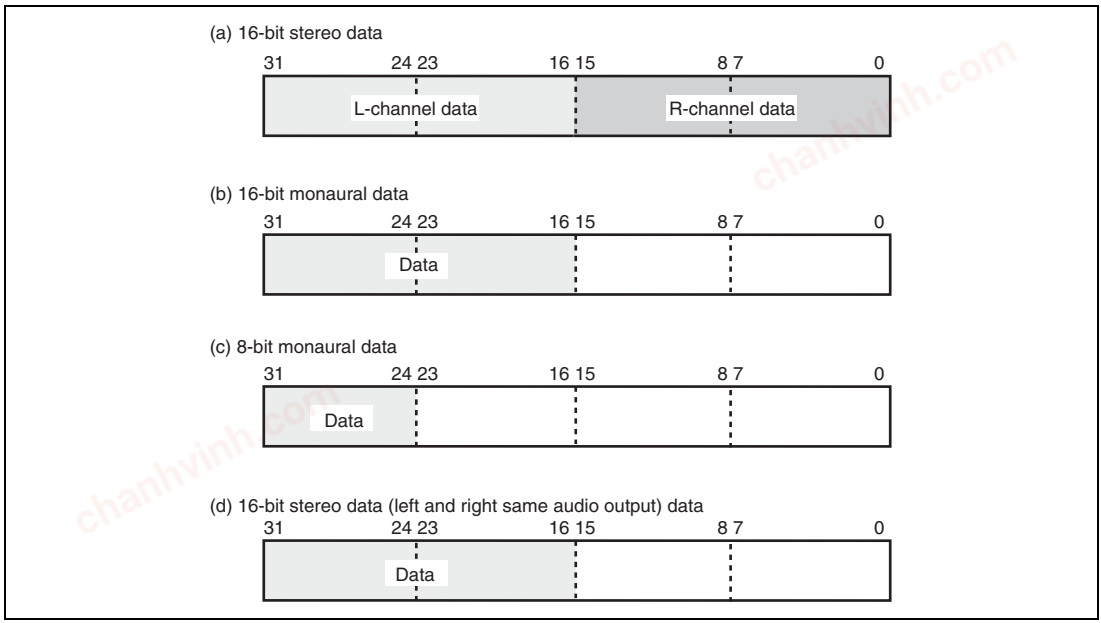


Figure 19.5 Transmit/Receive Data Bit Alignment

Note: In the figure, only the shaded areas are transmitted or received as valid data. Data in unshaded areas is not transmitted or received.

Monaural or stereo can be specified for transmit data by the TDLE bit and TDRE bit in SITDAR. Monaural or stereo can be specified for receive data by the RDLE bit and RDRE bit in SIRDAR. To achieve left and right same audio output while stereo is specified for transmit data, specify the TLREP bit in SITDAR. Tables 19.6 and 19.7 show the audio mode specifications for transmit data and that for receive data, respectively.

Table 19.6 Audio Mode Specification for Transmit Data

	Bit		
Mode	**TDLE**	**TDRE**	**TLREP**
Monaural	1	0	x
Stereo	1	1	0
Left and right same audio output	1	1	1

Note: x: Don't care

chanhvinh.com

Table 19.7 Audio Mode Specification for Receive Data

	Bit	
Mode	**RDLE**	**RDRE**
Monaural	1	0
Stereo	1	1

Note: Left and right same audio mode is not supported in receive data.
To execute monaural transmission or reception, use the left channel.

19.4.5 FIFO

(1) Overview

The transmit and receive FIFOs of this module have the following features.

- 16-stage 32-bit FIFOs for transmission and reception
- One FIFO buffer stage is used regardless of the access size. (One-stage 32-bit FIFO access cannot be divided into multiple accesses.)

(2) Transfer Request

The following FIFO transfer requests can be issued to the CPU or direct memory access controller.

- Transmit request: TDREQ (transmit FIFO transfer request)
- Receive request: RDREQ (receive FIFO transfer request)

The conditions to issue the transmit/receive FIFO transfer requests can be specified individually. The transmit request condition is specified with the TFWM2 to TFWM0 bits in SIFCTR, and the receive FIFO transfer request is specified with the RFWM2 to RFWM0 bits in SIFCTR. Tables 19.8 and 19.9 summarize the conditions specified by SIFCTR.

chanhvinh.com

Table 19.8 Conditions to Issue Transmit Request

TFWM2 to TFWM0	Number of Requested Stages	Transmit Request Issued	Used Areas
000	1	There are sixteen stages of empty area.	Smallest
100	4	There are twelve or more stages of empty area.	↕
101	8	There are eight or more stages of empty area.	
110	12	There are four or more stages of empty area.	
111	16	There is one or more stage of empty area.	Largest

Table 19.9 Conditions to Issue Receive Request

RFWM2 to RFWM0	Number of Requested Stages	Receive Request Issued	Used Areas
000	1	There is one or more stage of valid data.	Smallest
100	4	There are four stages of valid data or more.	↕
101	8	There are eight stages of valid data or more.	
110	12	There are twelve stages of valid data or more.	
111	16	There are sixteen stages of valid data.	Largest

The number of stages of the FIFO is sixteen. Accordingly, an overflow error or underflow error occurs if data area or empty area exceeds sixteen FIFO stages. The transfer request is canceled when the above condition is not satisfied even if the FIFO is not empty or full.

(3) Number of FIFOs

The usage state of the transmit FIFO and receive FIFO are indicated by the TFUA and FRUA bits in the FIFO control register as below:

- Transmit FIFO: The number of empty FIFO stages is indicated by the TFUA4 to TFUA0 bits in SIFCTR.
- Receive FIFO: The number of valid data stages is indicated by the RFUA4 to RFUA0 bits in SIFCTR.

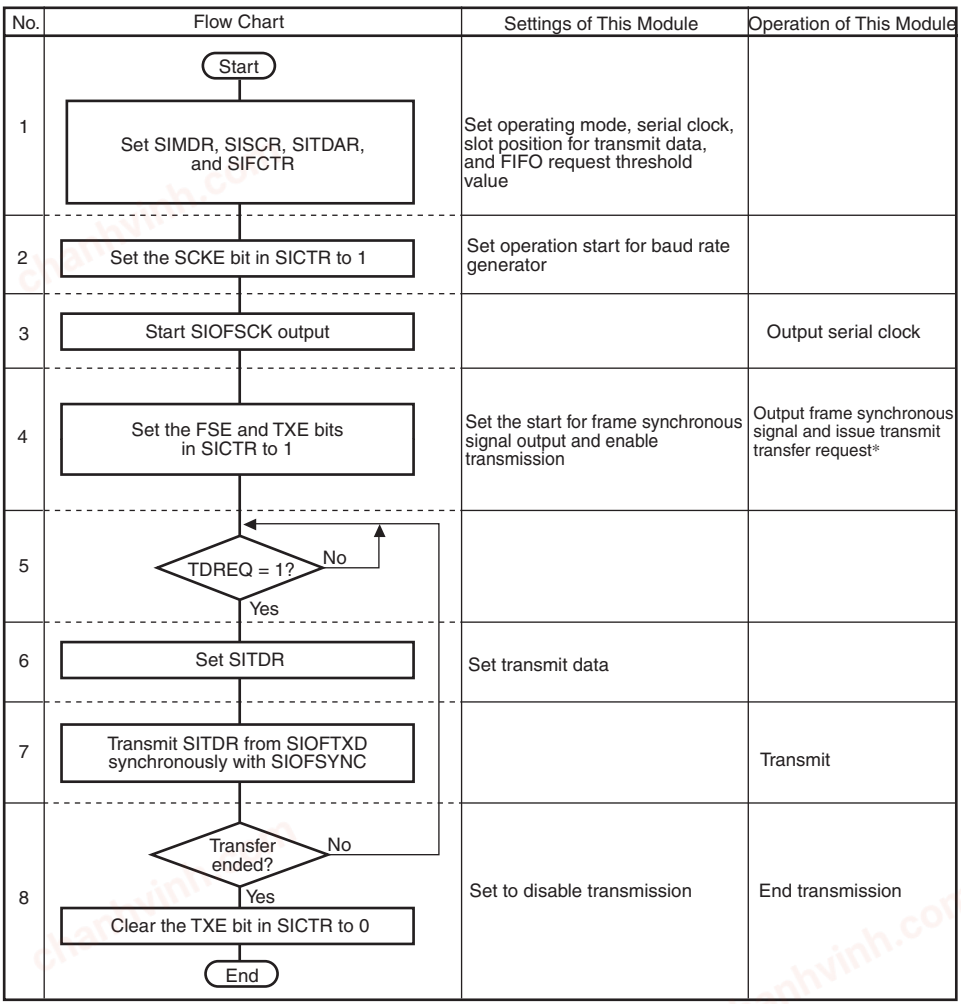
The above register contents indicate the possible data numbers that can be transferred by the CPU or direct memory access controller.

chanhvinh.com

19.4.6 Transmit and Receive Procedures

(1) Transmission in Master Mode

Figure 19.6 shows an example of transmission settings and operation when this module is used as a master.



Note: * To avoid occurrence of a transmit data underflow, the TXE bit should be set to 1 after setting the no. 6 transmit data,

Figure 19.6 Example of Transmit Operation in Master Mode

chanhvinh.com

(2) Reception in Master Mode

Figure 19.7 shows an example of reception settings and operation when this module is used as a master.

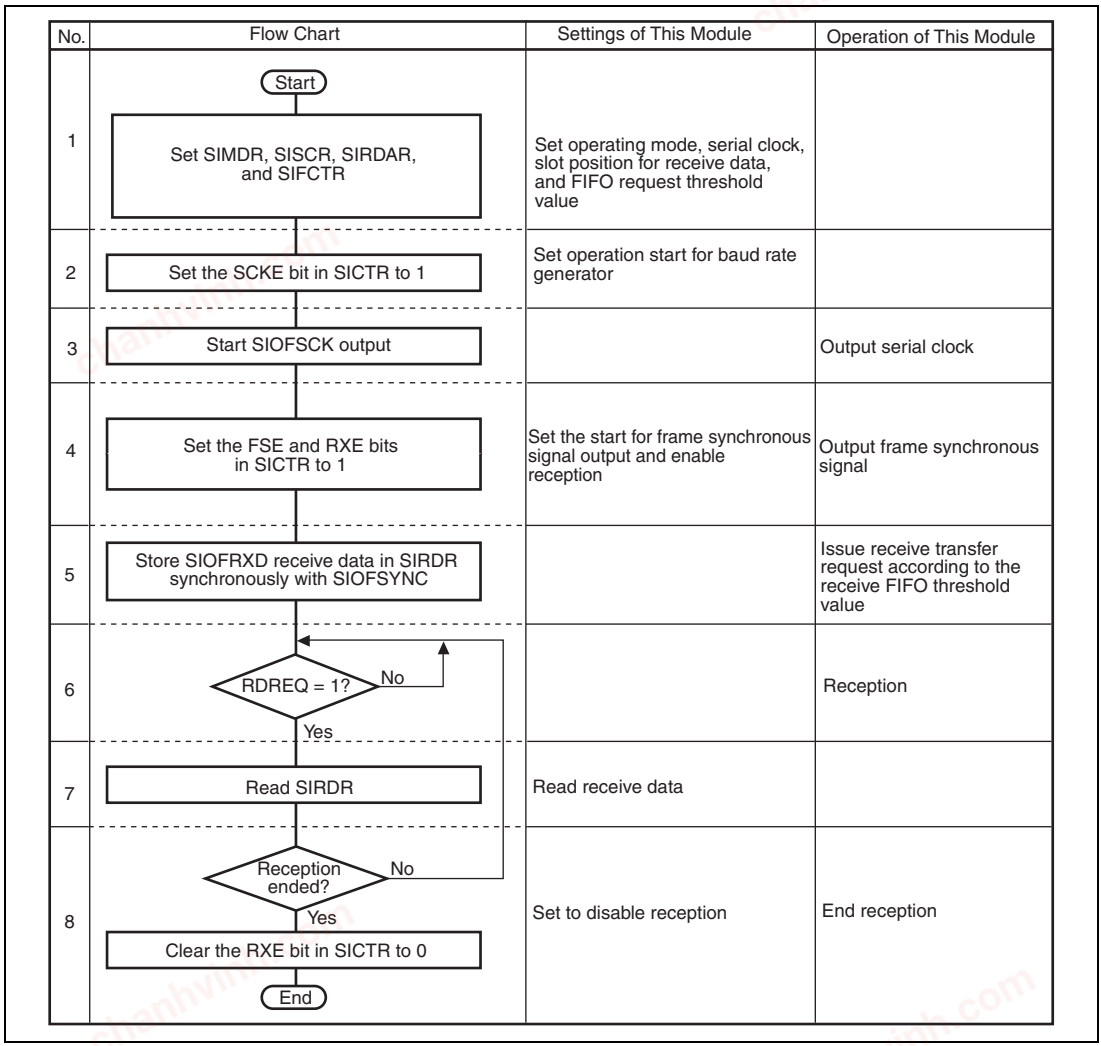


Figure 19.7 Example of Receive Operation in Master Mode

chanhvinh.com

(3) Transmission in Slave Mode

Figure 19.8 shows an example of transmission settings and operation for when this module is used as a slave.

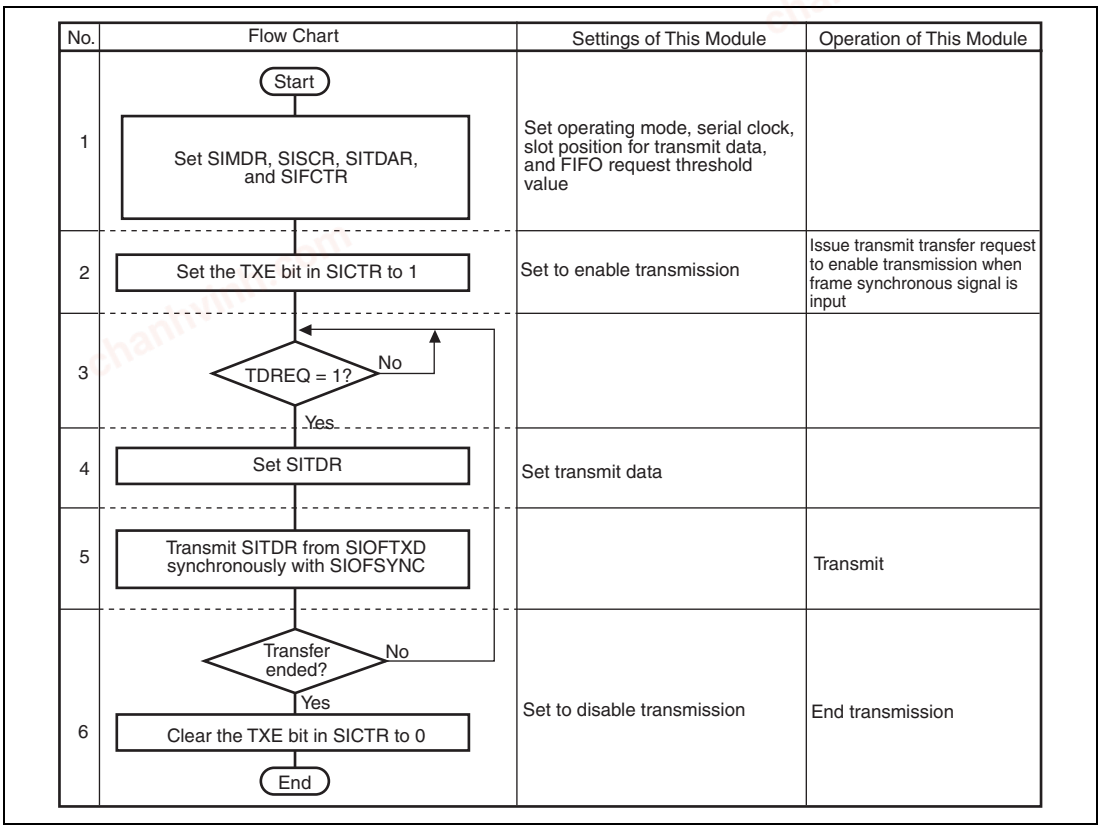


Figure 19.8 Example of Transmit Operation in Slave Mode

chanhvinh.com

(4) Reception in Slave Mode

Figure 19.9 shows an example of reception settings and operation when this module is used as a slave.

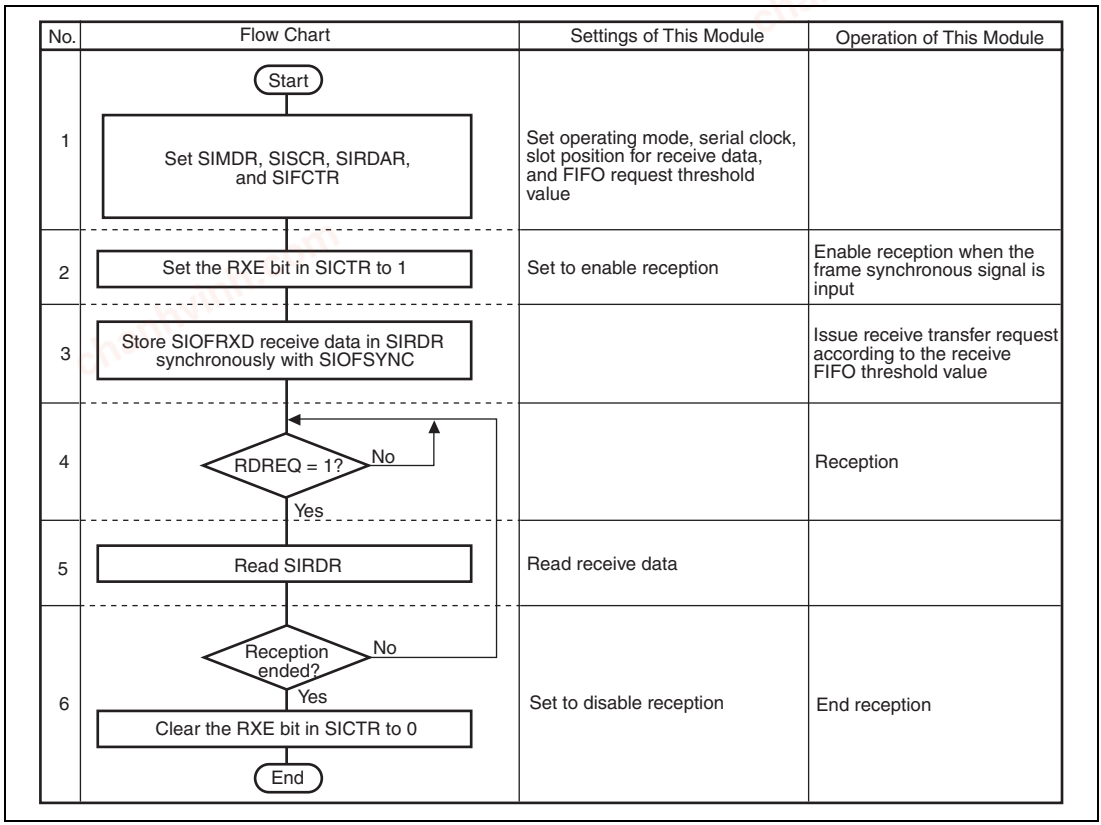


Figure 19.9 Example of Receive Operation in Slave Mode

chanhvinh.com

(5) Transmit/Receive Reset

This module can separately reset the transmit and receive units by setting the following bits to 1.

- Transmit reset: TXRST bit in SICTR
- Receive reset: RXRST bit in SICTR

Table 19.10 shows the details of initialization upon the transmit or receive reset.

Table 19.10 Transmit and Receive Reset

Type	Objects Initialized
Transmit reset	SITDR
	Valid data in transmit FIFO
	The TFEMP and TDREQ bits in SISTR
	The TXE bit in SICTR
Receive reset	SIRDR
	Valid data in receive FIFO
	The RFFUL and RDREQ bits in SISTR
	The RXE bit in SICTR

chanhvinh.com

19.4.7 Interrupts

This module has one type of interrupt.

(1) Interrupt Requests

Interrupts can be issued by several requests. Each source is shown as an status in SISTR. Table 19.11 lists the interrupt requests.

Table 19.11 Interrupt Requests

No.	Classification	Bit Name	Function Name	Description
1	Transmission	TDREQ	Transmit FIFO transfer request	The transmit FIFO stores data of specified size or more.
2		TFEMP	Transmit FIFO empty	The transmit FIFO is empty.
3	Reception	RDREQ	Receive FIFO transfer request	The receive FIFO stores data of specified size or more.
4		RFFUL	Receive FIFO full	The receive FIFO is full.
5	Error	TFUDF	Transmit FIFO underflow	Serial data transmit timing has arrived while the transmit FIFO is empty.
6		TFOVF	Transmit FIFO overflow	Write to the transmit FIFO is performed while the transmit FIFO is full.
7		RFOVF	Receive FIFO overflow	Serial data is received while the receive FIFO is full.
8		RFUDF	Receive FIFO underflow	The receive FIFO is read while the receive FIFO is empty.
9		FSERR	FS error	A synchronous signal is input before the specified bit number has been passed (in slave mode).

Whether the interrupt is issued or not by the request is determined by the SIIER settings. If an interrupt request is generated when the corresponding bit in SIIER is set to 1, this module issues the interrupt.

chanhvinh.com

(2) Regarding Transmit and Receive Classification

The transmit request and receive request are signals indicating the state; after being set, if the state of the transmit/receive FIFO change, they are automatically cleared by this module.

When the DMA transfer is used, the signal is cleared to 0 by the direct memory access controller. If the setting condition is still satisfied after the access using the direct memory access controller, it is set to 1 again.

(3) Processing when Errors Occur

On occurrence of each of the errors indicated as a status in SISTR, this module performs the following operations.

- Transmit FIFO underflow (TFUDF)
  The immediately preceding transmit data is again transmitted.
- Transmit FIFO overflow (TFOVF)
  The contents of the transmit FIFO are protected, and the write operation causing the overflow is ignored.
- Receive FIFO overflow (RFOVF)
  Data causing the overflow is discarded and lost.
- Receive FIFO underflow (RFUDF)
  The read value is undefined.
- FS error (FSERR)
  The internal counter is reset according to the sync signal in which an error occurs.

chanhvinh.com

19.4.8 Transmit and Receive Timing

Examples of serial transmission and reception with this module are shown in figures 19.10 to 19.15.

(1) 8-bit Monaural Data (1)

Falling edge sampling, slot No.0 used for transmit and receive data, an frame length = 8 bits

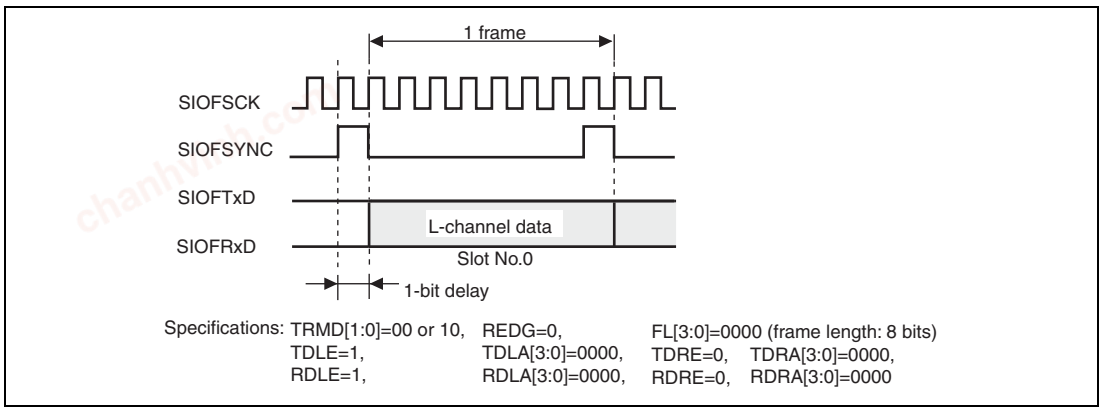


Figure 19.10 Transmit and Receive Timing (8-Bit Monaural Data (1))

(2) 8-bit Monaural Data (2)

Falling edge sampling, slot No.0 used for transmit and receive data, and frame length = 16 bits

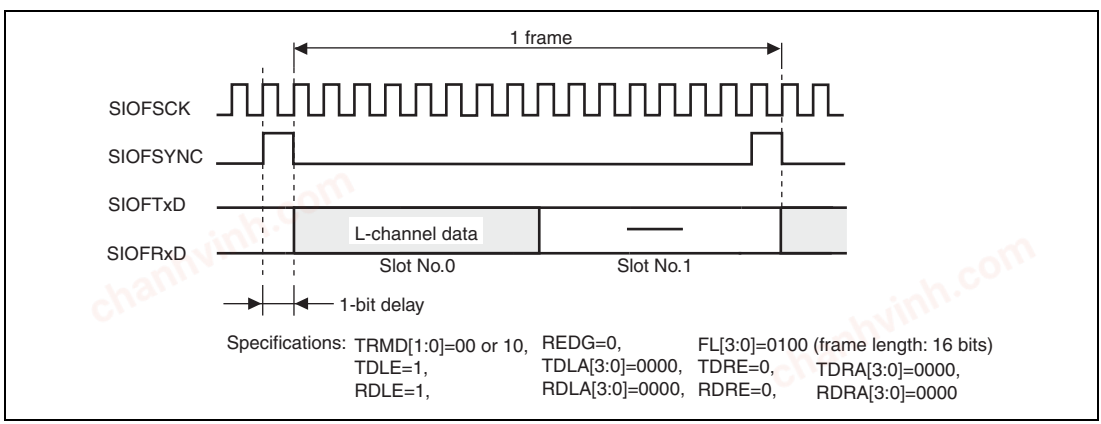


Figure 19.11 Transmit and Receive Timing (8-Bit Monaural Data (2))

chanhvinh.com

(3) 16-bit Monaural Data

Falling edge sampling, slot No.0 used for transmit and receive data, and frame length = 64 bits

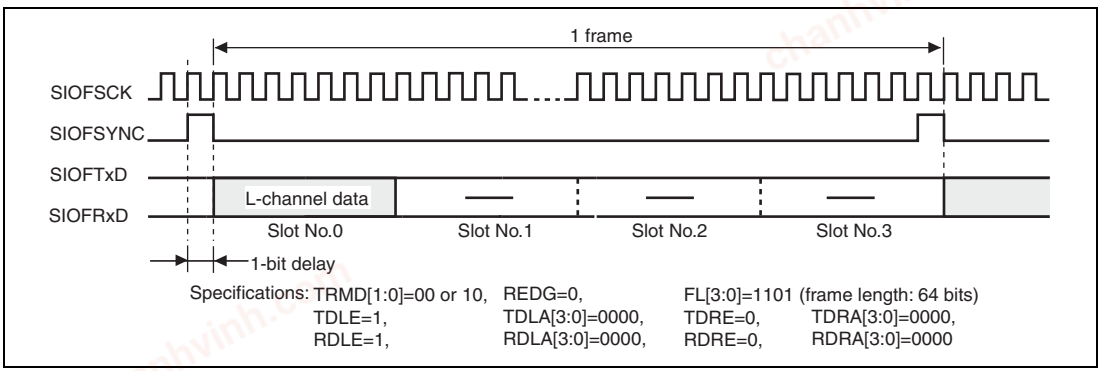


Figure 19.12 Transmit and Receive Timing (16-Bit Monaural Data)

(4) 16-bit Stereo Data (1)

Falling edge sampling, slot No.0 used for left channel data, slot No.1 used for right channel data, and frame length = 128 bits

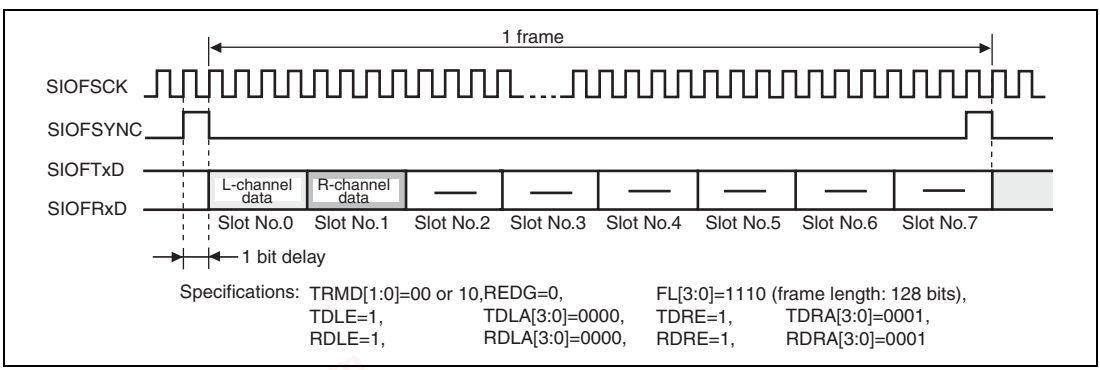


Figure 19.13 Transmit and Receive Timing (16-Bit Stereo Data (1))

chanhvinh.com

(5) 16-bit Stereo Data (2)

Falling edge sampling, slot No.0 used for left channel data, slot No.2 used for right channel data, and frame length = 128 bits

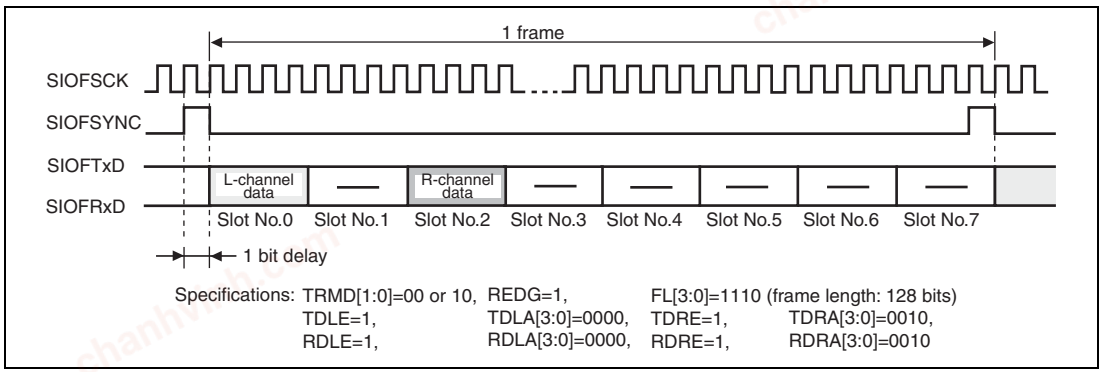


Figure 19.14 Transmit and Receive Timing (16-Bit Stereo Data (2))

(6) Synchronization-Pulse Output Mode at End of Each Slot (SYNCAT Bit = 1)

Falling edge sampling, slot No.0 used for left channel data, slot No.1 used for right-channel data, and frame length = 128 bits

In this mode, valid data must be set to slot No. 0. In addition, make sure that valid data is transmitted/received or transmitted.

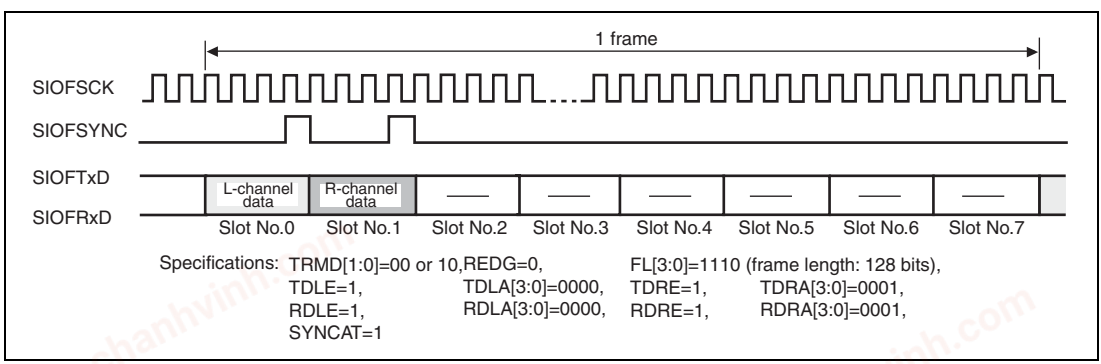


Figure 19.15 Transmit and Receive Timing (16-Bit Stereo Data)

chanhvinh.com

chanhvinh.com
chanhvinh.com
chanhvinh.com
chanhvinh.com

chanhvinh.com

Section 20 Controller Area Network

20.1 Summary

20.1.1 Overview

This document primarily describes the programming interface for the controller area network (Renesas CAN Time Trigger Level 1) module. It serves to facilitate the hardware/software interface so that engineers involved in this module implementation can ensure the design is successful.

Deep standby mode can be canceled by change on CRxn (PJ3, PJ1) pin. For details, refer to section 33, Power-Down Modes.

20.1.2 Scope

The CAN Data Link Controller function is not described in this document. It is the responsibility of the reader to investigate the CAN Specification Document (see references). The interfaces from the CAN Controller are described, in so far as they pertain to the connection with the User Interface.

The programming model is described in some detail. It is not the intention of this document to describe the implementation of the programming interface, but to simply present the interface to the underlying CAN functionality.

The document places no constraints upon the implementation of this module in terms of process, packaging or power supply criteria. These issues are resolved where appropriate in implementation specifications.

20.1.3 Audience

In particular this document provides the design reference for software authors who are responsible for creating a CAN application using this module.

In the creation of this module user interface LSI engineers must use this document to understand the hardware requirements.

20.1.4 References

1. CAN Specification Version 2.0 part A, Robert Bosch GmbH, 1991
2. CAN Specification Version 2.0 part B, Robert Bosch GmbH, 1991

chanhvinh.com

3. Implementation Guide for the CAN Protocol, CAN Specification 2.0 Addendum, CAN In Automation, Erlangen, Germany, 1997
4. Road vehicles - Controller area network (CAN): Part 1: Data link layer and physical signalling (ISO-11898-1, 2003)
5. Road vehicles - Controller area network (CAN): Part 4: Time triggered communication (ISO-11898-4, 2004)

20.1.5 Features

- Supports CAN specification 2.0B
- Bit timing compliant with ISO-11898-1
- 32 Mailbox version
- Clock frequency: Up to 36 MHz
- 31 programmable Mailboxes for transmit / receive + 1 receive-only mailbox
- Sleep mode for low power consumption and automatic recovery from sleep mode by detecting CAN bus activity
- Programmable receive filter mask (standard and extended identifier) supported by all Mailboxes
- Programmable CAN data rate up to 1MBit/s
- Transmit message queuing with internal priority sorting mechanism against the problem of priority inversion for real-time applications
- Data buffer access without SW handshake requirement in reception
- Flexible micro-controller interface
- Flexible interrupt structure
- 16-bit free running timer with flexible clock sources and pre-scaler, 3 Timer Compare Match Registers
- 6-bit Basic Cycle Counter for Time Trigger Transmission
- Timer Compare Match Registers with interrupt generation
- Timer counter clear / set capability
- Registers for Time-Trigger: Local_Time, Cycle_time, Ref_Mark, Tx_Enable Window, Ref_Trigger_Offset
- Flexible TimeStamp at SOF for both transmission and reception supported
- Time-Trigger Transmission, Periodic Transmission supported (on top of Event Trigger Transmission)
- Basic Cycle value can be embedded into a CAN frame and transmitted

chanhvinh.com

20.2 Architecture

This module device offers a flexible and sophisticated way to organise and control CAN frames, providing the compliance to CAN2.0B Active and ISO-11898-1. The module is formed from 5 different functional entities. These are the Micro Processor Interface (MPI), Mailbox, Mailbox Control, Timer, and CAN Interface. The figure below shows the block diagram of the Module. The bus interface timing is designed according to the peripheral bus I/F required for each product.

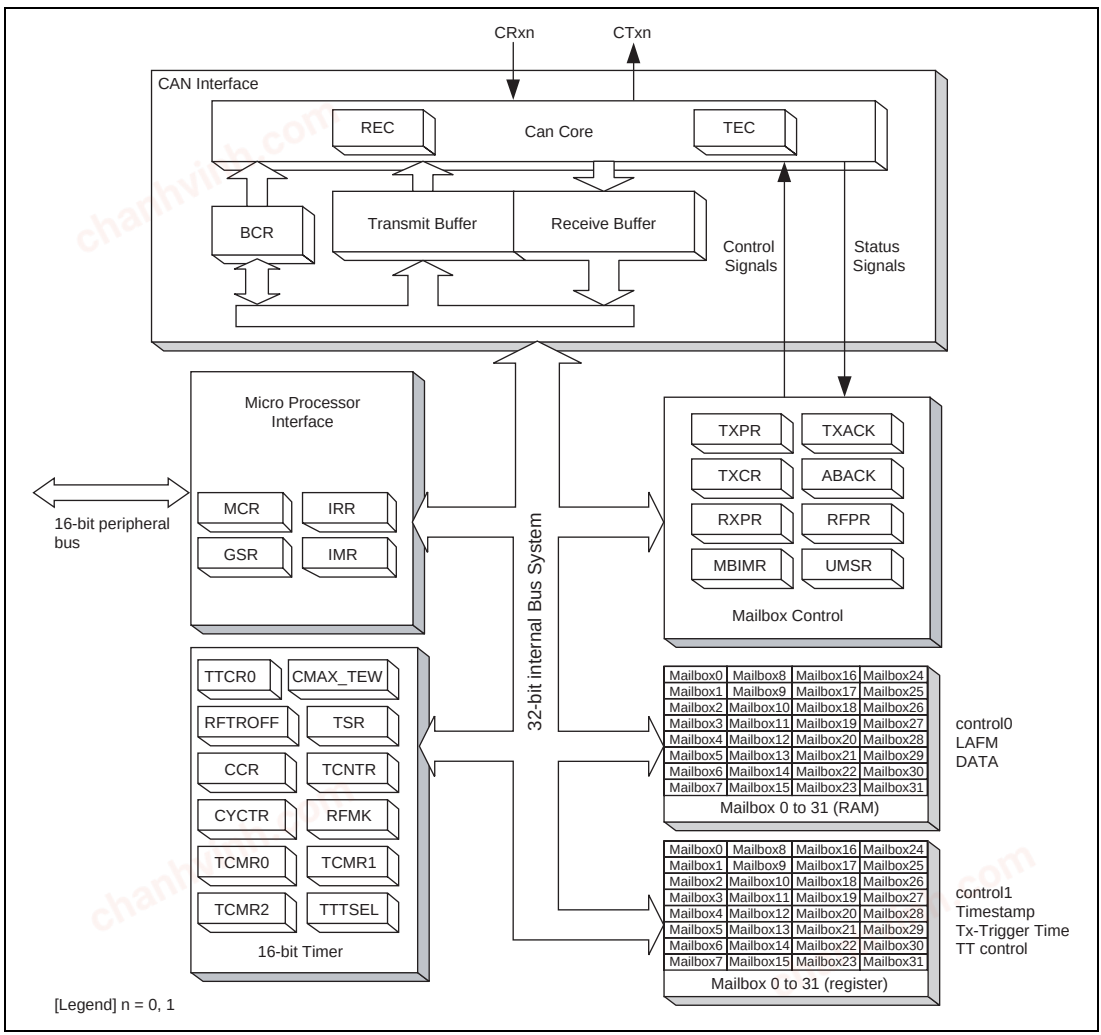


Figure 20.1 This Module Architecture

chanhvinh.com

Important: LongWord (32-bit) accesses are converted into two consecutive word (16-bit) accesses by the bus interface.

- Micro Processor Interface (MPI)

  The MPI allows communication between the Renesas CPU and this module's registers/mailboxes to control the memory interface. It also contains the Wakeup Control logic that detects the CAN bus activities and notifies the MPI and the other parts of this module so that this module can automatically exit the Sleep mode.

  It contains registers such as MCR, IRR, GSR and IMR.

- Mailbox

  The Mailboxes consists of RAM configured as message buffers and registers. There are 32 Mailboxes, and each mailbox has the following information.

  <RAM>

  — CAN message control (identifier, rtr, ide,etc)

  — CAN message data (for CAN Data frames)

  — Local Acceptance Filter Mask for reception

  <Registers>

  — CAN message control (dlc)

  — Time Stamp for message reception/transmission

  — 3-bit wide Mailbox Configuration, Disable Automatic Re-Transmission bit, Auto-Transmission for Remote Request bit, New Message Control bit

  — Tx-Trigger Time

- Mailbox Control

  The Mailbox Control handles the following functions.

  — For received messages, compare the IDs and generate appropriate RAM addresses/data to store messages from the CAN Interface into the Mailbox and set/clear appropriate registers accordingly.

  — To transmit event-triggered messages, run the internal arbitration to pick the correct priority message, and load the message from the Mailbox into the Tx-buffer of the CAN Interface and set/clear appropriate registers accordingly. In the case of time-triggered transmission, compare match of Tx-Trigger time invoke loading the messages.

  — Arbitrates Mailbox accesses between the CPU and the Mailbox Control.

  — Contains registers such as TXPR, TXCR, TXACK, ABACK, RXPR, RFPR, UMSR and MBIMR.

chanhvinh.com

- Timer

  The Timer function is the functional entity, which provides this module with support for transmitting messages at a specific time frame and recording the result.

  The Timer is a 16-bit free running up counter which can be controlled by the CPU. It provides one 16-bit Compare Match Register to compare with Local Time and two 16-bit ones to compare with Cycle Time. The Compare Match Registers can generate interrupt signals and clear the Counter.

  The clock period of this Timer offers a wide selection derived from the system clock or can be programmed to be incremented with one nominal bit timing of CAN Bus.

  Contains registers such as TCNTR, TTCR0, CMAX_TEW, RETROFF, TSR, CCR, CYCTR, RFMK, TCMR0, TCMR1, TCMR2 and TTTSEL.

- CAN Interface

  This block conforms to the requirements for a CAN Bus Data Link Controller which is specified in Ref. [2, 4]. It fulfils all the functions of a standard DLC as specified by the OSI 7 Layer Reference model. This functional entity also provides the registers and the logic which are specific to a given CAN bus, which includes the Receive Error Counter, Transmit Error Counter, the Bit Configuration Registers and various useful Test Modes. This block also contains functional entities to hold the data received and the data to be transmitted for the CAN Data Link Controller.

chanhvinh.com

20.3 Programming Model - Overview

The purpose of this programming interface is to allow convenient, effective access to the CAN bus for efficient message transfer. Please bear in mind that the user manual reports all settings allowed by this module IP. Different use of this module is not allowed.

20.3.1 Memory Map

The diagram of the memory map is shown below.

chanhvinh.com

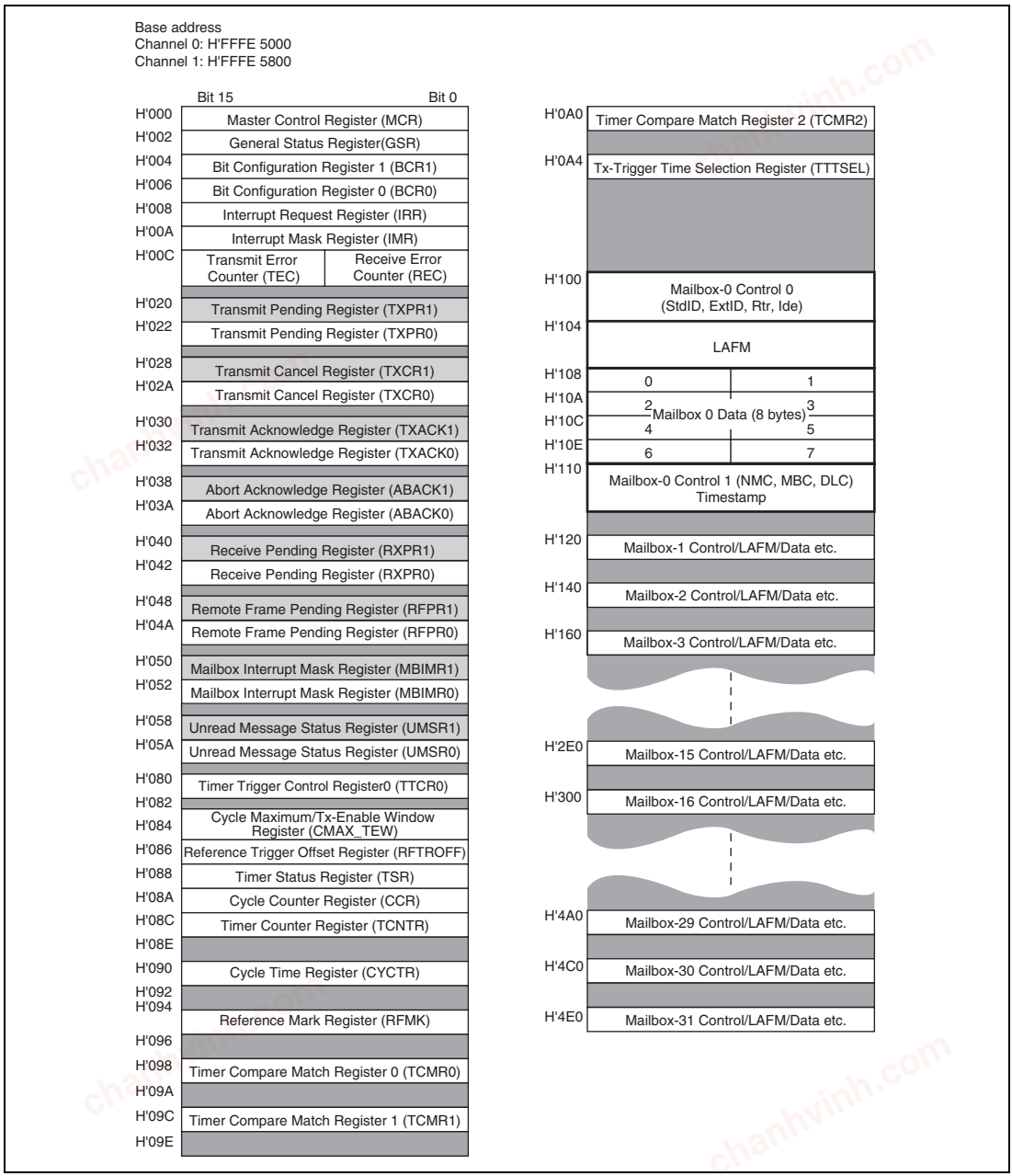


Figure 20.2 Memory Map

The locations not used (between H'000 and H'4F3) are reserved and cannot be accessed.

chanhvinh.com

20.3.2 Mailbox Structure

Mailboxes play a role as message buffers to transmit/receive CAN frames. Each Mailbox is comprised of 3 identical storage fields that are 1): Message Control, 2): Local Acceptance Filter Mask, 3): Message Data. In addition some Mailboxes contain the following extra Fields: 4): Time Stamp, 5): Time Trigger configuration and 6): Time Trigger Control. The following table shows the address map for the control, LAFM, data, timestamp, Transmission Trigger Time and Time Trigger Control addresses for each mailbox.

	Address						
	Control0	LAFM	Data	Control1	Time Stamp	Trigger Time	TT control
Mailbox	4 bytes	4 bytes	8 bytes	2 bytes	2 bytes	2 bytes	2 bytes
0 (Receive Only)	100 – 103	104– 107	108 – 10F	110 – 111	112 – 113	No	No
1	120 – 123	124 – 127	128 – 12F	130 – 131	132 – 133	No	No
2	140 – 143	144 – 147	148 – 14F	150 – 151	152 – 153	No	No
3	160 – 163	164 – 167	168 – 16F	170 – 171	172 – 173	No	No
4	180 – 183	184 – 187	188 – 18F	190 – 191	192 – 193	No	No
5	1A0 – 1A3	1A4 – 1A7	1A8 – 1AF	1B0 – 1B1	1B2 – 1B3	No	No
6	1C0 – 1C3	1C4 – 1C7	1C8 – 1CF	1D0 – 1D1	1D2 – 1D3	No	No
7	1E0 – 1E3	1E4 – 1E7	1E8 – 1EF	1F0 – 1F1	1F2 – 1F3	No	No
8	200 – 203	204 – 207	208 – 20F	210 – 211	212 – 213	No	No
9	220 – 223	224 – 227	228 – 22F	230 – 231	232 – 233	No	No
10	240 – 243	244 – 247	248 – 24F	250 – 251	252 – 253	No	No
11	260 – 263	264 – 267	268 – 26F	270 – 271	272 – 273	No	No
12	280 – 283	284 – 287	288 – 28F	290 – 291	292 – 293	No	No
13	2A0 – 2A3	2A4 – 2A7	2A8 – 2AF	2B0 – 2B1	2B2 – 2B3	No	No
14	2C0 – 2C3	2C4 – 2C7	2C8 – 2CF	2D0 – 2D1	2D2 – 2D3	No	No
15	2E0 – 2E3	2E4 – 2E7	2E8 – 2EF	2F0 – 2F1	2F2 – 2F3	No	No
16	300 – 303	304 – 307	308 – 30F	310 – 311	No	No	No
17	320 – 323	324 – 327	328 – 32F	330 – 331	No	No	No
18	340 – 343	344 – 347	348 – 34F	350 – 351	No	No	No

chanhvinh.com

	Address						
	Control0	LAFM	Data	Control1	Time Stamp	Trigger Time	TT control
Mailbox	4 bytes	4 bytes	8 bytes	2 bytes	2 bytes	2 bytes	2 bytes
19	360 – 363	364 – 367	368 – 36F	370 – 371	No	No	No
20	380 – 383	384 – 387	388 – 38F	390 – 391	No	No	No
21	3A0 – 3A3	3A4 – 3A7	3A8 – 3AF	3B0 – 3B1	No	No	No
22	3C0 – 3C3	3C4 – 3C7	3C8 – 3CF	3D0 – 3D1	No	No	No
23	3E0 – 3E3	3E4 – 3E7	3E8 – 3EF	3F0 – 3F1	No	No	No
24	400 – 403	404 – 407	408 – 40F	410 – 411	No	414 – 415	416 – 417
25	420 – 423	424 – 427	428 – 42F	430 – 431	No	434 – 435	436 – 437
26	440 – 443	444 – 447	448 – 44F	450 – 451	No	454 – 455	456 – 457
27	460 – 463	464 – 467	468 – 46F	470 – 471	No	474 – 475	476 – 477
28	480 – 483	484 – 487	488 – 48F	490 – 491	No	494 – 495	496 – 497
29	4A0 – 4A3	4A4 – 4A7	4A8 – 4AF	4B0 – 4B1	No	4B4 – 4B5	4B6 – 4B7
30	4C0 – 4C3	4C4 – 4C7	4C8 – 4CF	4D0 – 4D1	4D2 – 4D3 (Local Time)	4D4 – 4D5	No
31	4E0 – 4E3	4E4 – 4E7	4E8 – 4EF	4F0 – 4F1	4F2 – 4F3 (Local Time)	No	No

Mailbox-0 is a receive-only box, and all the other Mailboxes can operate as both receive and transmit boxes, dependant upon the MBC (Mailbox Configuration) bits in the Message Control. The following diagram shows the structure of a Mailbox in detail.

chanhvinh.com

Table 20.1 Roles of Mailboxes

	Event Trigger		Time Trigger		Remark	
	Tx	Rx	Tx	Rx	TimeStamp	Tx-Trigger Time
MB31	Settable	Settable	—	Time reference reception	Available	—
MB30	Settable	Settable	Time reference transmission in time master mode	Reception in time slave mode	Available	Available
MB29 - 24	Settable	Settable	Settable	Settable	—	Available
MB23 - 16	Settable	Settable	— (ET)	Settable	—	—
MB15 - 1	Settable	Settable	— (ET)	Settable	Available	—
MB0	—	Settable	—	Settable	Available	—

(ET) shows that it works during merged arbitrating window, after completion of time-triggered transmission.

MB0 (reception MB with timestamp)

Byte: 8-bit access, Word: 16-bit access, LW (LongWord) : 32-bit access

Address	Data Bus 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Access Size	Field Name
H'100 + N*32	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		Word/LW	Control 0
H'102 + N*32	EXTID[15:0]																Word	
H'104 + N*32	IDE LAFM	0	0	STDID_LAFM[10:0]											EXTID_LAFM[17:16]		Word/LW	LAFM
H'106 + N*32	EXTID_LAFM[15:0]																Word	
H'108 + N*32	MSG_DATA_0 (first Rx/Tx Byte)								MSG_DATA_1								Byte/Word/LW	Data
H'10A + N*32	MSG_DATA_2								MSG_DATA_3								Byte/Word	
H'10C + N*32	MSG_DATA_4								MSG_DATA_5								Byte/Word/LW	
H'10E + N*32	MSG_DATA_6								MSG_DATA_7								Byte/Word	
H'110 + N*32	0	0	NMC	0	0	MBC[2:0]			0	0	0	0	DLC[3:0]				Byte/Word	Control 1
H'112 + N*32	TimeStamp[15:0] (CYCTR[15:0] or CCR[5:0]/CYCTR[15:6] at SOF)																Word	TimeStamp

MBC[1] is fixed to "1"

MB15 to 1 (MB with timestamp)

Address	Data Bus 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Access Size	Field Name
H'100 + N*32	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		Word/LW	Control 0
H'102 + N*32	EXTID[15:0]																Word	
H'104 + N*32	IDE LAFM	0	0	STDID_LAFM[10:0]											EXTID_LAFM[17:16]		Word/LW	LAFM
H'106 + N*32	EXTID_LAFM[15:0]																Word	
H'108 + N*32	MSG_DATA_0 (first Rx/Tx Byte)								MSG_DATA_1								Byte/Word/LW	Data
H'10A + N*32	MSG_DATA_2								MSG_DATA_3								Byte/Word	
H'10C + N*32	MSG_DATA_4								MSG_DATA_5								Byte/Word/LW	
H'10E + N*32	MSG_DATA_6								MSG_DATA_7								Byte/Word	
H'110 + N*32	0	0	NMC	ATX	DART	MBC[2:0]			0	0	0	0	DLC[3:0]				Byte/Word	Control 1
H'112 + N*32	TimeStamp[15:0] (CYCTR[15:0] or CCR[5:0]/CYCTR[15:6] at SOF)																Word	TimeStamp

Figure 20.3 Mailbox-N Structure

chanhvinh.com

MB23 to 16 (MB without timestamp)

Address	Data Bus 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Access Size	Field Name
H'100 + N*32	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		Word/LW	Control 0
H'102 + N*32	EXTID[15:0]																Word	
H'104 + N*32	IDE LAFM	0	0	STDID_LAFM[10:0]											EXTID LAFM[17:16]		Word/LW	LAFM
H'106 + N*32	EXTID_LAFM[15:0]																Word	
H'108 + N*32	MSG_DATA_0 (first Rx/Tx Byte)								MSG_DATA_1								Byte/Word/LW	Data
H'10A + N*32	MSG_DATA_2								MSG_DATA_3								Byte/Word	
H'10C + N*32	MSG_DATA_4								MSG_DATA_5								Byte/Word/LW	
H'10E + N*32	MSG_DATA_6								MSG_DATA_7								Byte/Word	
H'110 + N*32	0	0	NMC	ATX	DART	MBC[2:0]			0	0	0	0	DLC[3:0]				Byte/Word	Control 1

MB29 to 24 (Time-Triggered Transmission in Time Trigger mode)

Address	Data Bus 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Access Size	Field Name
H'100 + N*32	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		Word/LW	Control 0
H'102 + N*32	EXTID[15:0]																Word	
H'104 + N*32	IDE LAFM	0	0	STDID_LAFM[10:0]											EXTID LAFM[17:16]		Word/LW	LAFM
H'106 + N*32	EXTID_LAFM[15:0]																Word	
H'108 + N*32	MSG_DATA_0 (first Rx/Tx Byte)								MSG_DATA_1								Byte/Word/LW	Data
H'10A + N*32	MSG_DATA_2								MSG_DATA_3								Byte/Word	
H'10C + N*32	MSG_DATA_4								MSG_DATA_5								Byte/Word/LW	
H'10E + N*32	MSG_DATA_6								MSG_DATA_7								Byte/Word	
H'110 + N*32	0	0	NMC	ATX	DART	MBC[2:0]			0	0	0	0	DLC[3:0]				Byte/Word	Control 1
H'112 + N*32	reserved																-	-
H'114 + N*32	Tx-Triggered Time (TTT)																Word	Trigger Time
H'116 + N*32	TTW[1:0]		offset						0	0	0	0	0	Rep_Factor			Word	TT control

Figure 20.3 Mailbox-N Structure (continued)

chanhvinh.com

MB30 (Time Reference Transmitssion in Time Trigger mode)

Address	Data Bus																Access Size	Field Name
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'100 + N*32	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		Word/LW	Control 0
H'102 + N*32	EXTID[15:0]																Word	
H'104 + N*32	IDE_LAFM	0	0	STDID_LAFM[10:0]											EXTID_LAFM[17:16]		Word/LW	LAFM
H'106 + N*32	EXTID_LAFM[15:0]																Word	
H'108 + N*32	MSG_DATA_0 (first Rx/Tx Byte)								MSG_DATA_1								Byte/Word/LW	Data
H'10A + N*32	MSG_DATA_2								MSG_DATA_3								Byte/Word	
H'10C + N*32	MSG_DATA_4								MSG_DATA_5								Byte/Word/LW	
H'10E + N*32	MSG_DATA_6								MSG_DATA_7								Byte/Word	
H'110 + N*32	0	0	NMC	ATX	DART	MBC[2:0]			0	0	0	0	DLC[3:0]				Byte/Word	Control 1
H'112 + N*32	TimeStamp[15:0] (TCNTR at SOF)																Word	TimeStamp
H'114 + N*32	Tx-Triggered Time (TTT) as Time Reference																Word	Trigger Time

MB31 (Time Reference Reception in Time Trigger mode)

Address	Data Bus																Access Size	Field Name
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'100 + N*32	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		Word/LW	Control 0
H'102 + N*32	EXTID[15:0]																Word	
H'104 + N*32	IDE_LAFM	0	0	STDID_LAFM[10:0]											EXTID_LAFM[17:16]		Word/LW	LAFM
H'106 + N*32	EXTID_LAFM[15:0]																Word	
H'108 + N*32	MSG_DATA_0 (first Rx/Tx Byte)								MSG_DATA_1								Byte/Word/LW	Data
H'10A + N*32	MSG_DATA_2								MSG_DATA_3								Byte/Word	
H'10C + N*32	MSG_DATA_4								MSG_DATA_5								Byte/Word/LW	
H'10E + N*32	MSG_DATA_6								MSG_DATA_7								Byte/Word	
H'110 + N*32	0	0	NMC	ATX	DART	MBC[2:0]			0	0	0	0	DLC[3:0]				Byte/Word	Control 1
H'112 + N*32	TimeStamp[15:0] (TCNTR at SOF)																Word	TimeStamp

Figure 20.3 Mailbox-N Structure (continued)

Notes: 1. All bits shadowed in grey are reserved and must be written LOW. The value returned by a read may not always be '0' and should not be relied upon.

2. ATX and DART are not supported by Mailbox-0, and the MBC setting of Mailbox-0 is limited.

3. ID Reorder (MCR15) can change the order of STDID, RTR, IDE and EXTID of both message control and LAFM.

chanhvinh.com

(1) Message Control Field

STDID[10:0]: These bits set the identifier (standard identifier) of data frames and remote frames.

EXTID[17:0]: These bits set the identifier (extended identifier) of data frames and remote frames.

RTR (Remote Transmission Request bit): Used to distinguish between data frames and remote frames. This bit is overwritten by received CAN Frames depending on Data Frames or Remote Frames.

Important: Please note that, when ATX bit is set with the setting MBC = 001(bin), the RTR bit will never be set. When a Remote Frame is received, the CPU can be notified by the corresponding RFPR set or IRR[2] (Remote Frame Receive Interrupt), however, as this module needs to transmit the current message as a Data Frame, the RTR bit remains unchanged.

Important: In order to support automatic answer to remote frame when MBC = 001 (bin) is used and ATX = 1 the RTR flag must be programmed to zero to allow data frame to be transmitted.

Note: when a Mailbox is configured to send a remote frame request the DLC used for transmission is the one stored into the Mailbox.

RTR	Description
0	Data frame
1	Remote frame

IDE (Identifier Extension bit): Used to distinguish between the standard format and extended format of CAN data frames and remote frames.

IDE	Description
0	Standard format
1	Extended format

chanhvinh.com

- Mailbox-0

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

Note: MBC[1] of MB0 is always "1".

- Mailbox-31 to 1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	NMC	ATX	DART	MBC[2:0]			0	0	0	0	DLC[3:0]			
Initial value:	0	0	0	0	0	1	1	1	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/W	R/W	R/W	R/W

NMC (New Message Control): When this bit is set to '0', the Mailbox of which the RXPR or RFPR bit is already set does not store the new message but maintains the old one and sets the UMSR correspondent bit. When this bit is set to '1', the Mailbox of which the RXPR or RFPR bit is already set overwrites with the new message and sets the UMSR correspondent bit.

Important: Please note that if a remote frame is overwritten with a data frame or vice versa could be that both RXPR and RFPR flags (together with UMSR) are set for the same Mailbox. In this case the RTR bit within the Mailbox Control Field should be relied upon.

Important: Please note that when the Time Triggered mode is used NMC needs to be set to ‘1’ for Mailbox 31 to allow synchronization with all incoming reference messages even when RXPR[31] is not cleared.

NMC	Description
0	Overrun mode (Initial value)
1	Overwrite mode

chanhvinh.com

ATX (Automatic Transmission of Data Frame): When this bit is set to ‘1’ and a Remote Frame is received into the Mailbox DLC is stored. Then, a Data Frame is transmitted from the same Mailbox using the current contents of the message data and updated DLC by setting the corresponding TXPR automatically. The scheduling of transmission is still governed by ID priority or Mailbox priority as configured with the Message Transmission Priority control bit (MCR.2). In order to use this function, MBC[2:0] needs to be programmed to be ‘001’ (Bin). When a transmission is performed by this function, the DLC (Data Length Code) to be used is the one that has been received. Application needs to guarantee that the DLC of the remote frame correspond to the DLC of the data frame requested.

Important: When ATX is used and MBC = 001 (Bin) the filter for the IDE bit cannot be used since ID of remote frame has to be exactly the same as that of data frame as the reply message.

Important: Please note that, when this function is used, the RTR bit will never be set despite receiving a Remote Frame. When a Remote Frame is received, the CPU will be notified by the corresponding RFPR set, however, as this module needs to transmit the current message as a Data Frame, the RTR bit remains unchanged.

Important: Please note that in case of overrun condition (UMSR flag set when the Mailbox has its NMC = 0) the message received is discarded. In case a remote frame is causing overrun into a Mailbox configured with ATX = 1, the transmission of the corresponding data frame may be triggered only if the related PFPR flag is cleared by the CPU when the UMSR flag is set. In such case PFPR flag would get set again.

ATX	Description
0	Automatic Transmission of Data Frame disabled (Initial value)
1	Automatic Transmission of Data Frame enabled

DART (Disable Automatic Re-Transmission): When this bit is set, it disables the automatic re-transmission of a message in the event of an error on the CAN bus or an arbitration lost on the CAN bus. In effect, when this function is used, the corresponding TXCR bit is automatically set at the start of transmission. When this bit is set to '0', this module tries to transmit the message as many times as required until it is successfully transmitted or it is cancelled by the TXCR.

DART	Description
0	Re-transmission enabled (Initial value)
1	Re-Transmission disabled

chanhvinh.com

MBC[2:0] (Mailbox Configuration): These bits configure the nature of each Mailbox as follows. When MBC = 111 (Bin), the Mailbox is inactive, i.e., it does not receive or transmit a message regardless of TXPR or other settings. The MBC = '110', '101' and '100' settings are prohibited. When the MBC is set to any other value, the LAFM field becomes available. Please don't set TXPR when MBC is set as reception as there is no hardware protection, and TXPR will remain set. MBC[1] of Mailbox-0 is fixed to "1" by hardware. This is to ensure that MB0 cannot be configured to transmit Messages.

MBC[2]	MBC[1]	MBC[0]	Data Frame Transmit	Remote Frame Transmit	Data Frame Receive	Remote Frame Receive	Remarks
0	0	0	Yes	Yes	No	No	• Not allowed for Mailbox-0 • Time-Triggered transmission can be used
0	0	1	Yes	Yes	No	Yes	• Can be used with ATX* • Not allowed for Mailbox-0 • LAFM can be used
0	1	0	No	No	Yes	Yes	• Allowed for Mailbox-0 • LAFM can be used
0	1	1	No	No	Yes	No	• Allowed for Mailbox-0 • LAFM can be used
1	0	0	Setting prohibited				
1	0	1	Setting prohibited				
1	1	0	Setting prohibited				
1	1	1	Mailbox inactive (Initial value)				

Notes: * In order to support automatic retransmission, RTR shall be "0" when MBC = 001(bin) and ATX = 1.
When ATX = 1 is used the filter for IDE must not be used.

chanhvinh.com

DLC[3:0] (Data Length Code): These bits encode the number of data bytes from 0,1, 2, … 8 that will be transmitted in a data frame. Please note that when a remote frame request is transmitted the DLC value to be used must be the same as the DLC of the data frame that is requested.

DLC[3]	DLC[2]	DLC[1]	DLC[0]	Description
0	0	0	0	Data Length = 0 bytes (Initial value)
0	0	0	1	Data Length = 1 byte
0	0	1	0	Data Length = 2 bytes
0	0	1	1	Data Length = 3 bytes
0	1	0	0	Data Length = 4 bytes
0	1	0	1	Data Length = 5 bytes
0	1	1	0	Data Length = 6 bytes
0	1	1	1	Data Length = 7 bytes
1	x	x	x	Data Length = 8 bytes

(2) Local Acceptance Filter Mask (LAFM)

This area is used as Local Acceptance Filter Mask (LAFM) for receive boxes**.**

LAFM: When MBC is set to 001, 010, 011(Bin), this field is used as LAFM Field. It allows a Mailbox to accept more than one identifier. The LAFM is comprised of two 16-bit read/write areas as follows.

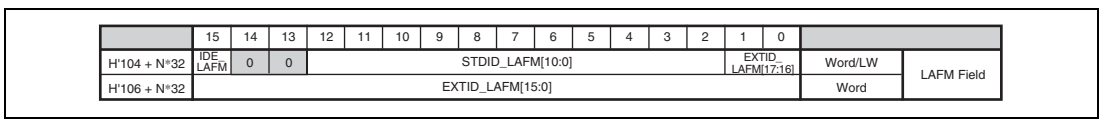


Figure 20.4 Acceptance filter

If a bit is set in the LAFM, then the corresponding bit of a received CAN identifier is ignored when this module searches a Mailbox with the matching CAN identifier. If the bit is cleared, then the corresponding bit of a received CAN identifier must match to the STDID/IDE/EXTID set in the mailbox to be stored. The structure of the LAFM is same as the message control in a Mailbox. If this function is not required, it must be filled with '0'.

Important: This module starts to find a matching identifier from Mailbox-31 down to Mailbox-0. As soon as this module finds one matching, it stops the search. The message will be stored or not depending on the NMC and RXPR/RFPR flags. This means that, even using LAFM, a received message can only be stored into 1 Mailbox.

chanhvinh.com

Important: When a message is received and a matching Mailbox is found, the whole message is stored into the Mailbox. This means that, if the LAFM is used, the STDID, RTR, IDE and EXTID may differ to the ones originally set as they are updated with the STDID, RTR, IDE and EXTID of the received message.

STD_LAFM[10:0] — Filter mask bits for the CAN base identifier [10:0] bits.

STD_LAFM[10:0]	Description
0	Corresponding STD_ID bit is cared
1	Corresponding STD_ID bit is "don't cared"

EXT_LAFM[17:0] — Filter mask bits for the CAN Extended identifier [17:0] bits.

EXT_LAFM[17:0]	Description
0	Corresponding EXT_ID bit is cared
1	Corresponding EXT_ID bit is "don't cared"

IDE_LAFM — Filter mask bit for the CAN IDE bit.

IDE_LAFM	Description
0	Corresponding IDE bit is cared
1	Corresponding IDE bit is "don't cared"

(3) Message Data Fields

Storage for the CAN message data that is transmitted or received. MSG_DATA[0] corresponds to the first data byte that is transmitted or received. The bit order on the CAN bus is bit 7 through to bit 0.

When CMAX!= 3'b111/MBC[30] = 3'b000 and TXPR[30] is set, Mailbox-30 is configured as transmission of time reference. Its DLC must be greater than 0 and its RTR must be zero (as specified for TTCAN Level 1) so that the Cycle_count (CCR register) is embedded in the first byte of the data field instead of MSG_DATA_0[5:0] when this Mailbox starts transmission. This function shall be used when this module is enabled to work in TTCAN mode to perform a Potential Time Master role to send the Time reference message. MSG_DATA_0[7:6] is still transmitted as stored in the Mailbox. User can set MSG_DATA_0[7] when a Next_is_Gap needs to be transmitted.

chanhvinh.com

Please note that the CCR value is only embedded on the frame transmitted but not stored back into Mailbox 30.

When CMAX!= 3'b111, MBC[31] = 3'b011 and TXPR[31] is cleared, Mailbox-31 is configured as reception of time reference. When a valid reference message is received (DLC > 0) this module performs internal synchronisation (modifying its RFMK and basic cycle CCR).

MB30 - 31

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'108 + N*32	Next_is_Gap/Cycle_Counter (first Rx/Tx Byte)								MSG_DATA_1								Byte/Word/LW	Data
H'10A + N*32	MSG_DATA_2								MSG_DATA_3								Byte/Word	
H'10C + N*32	MSG_DATA_4								MSG_DATA_5								Byte/Word/LW	
H'10E + N*32	MSG_DATA_6								MSG_DATA_7								Byte/Word	

Figure 20.5 Message Data Field

(4) Timestamp

Storage for the Timestamp recorded on messages for transmit/receive. The Timestamp will be a useful function to monitor if messages are received/transmitted within expected schedule.

- Timestamp

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TS15	TS14	TS13	TS12	TS11	TS10	TS9	TS8	TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Message Receive: For received messages of Mailbox-15 to 0, Timestamp always captures the CYCTR (Cycle Time Register) value or Cycle_Counter CCR[5:0] + CYCTR[15:6] value, depending on the programmed value in the bit 14 of TTCR0 (Timer Trigger Control Register 0) at SOF.

For messages received into Mailboxes 30 and 31, Timestamp captures the TCNTR (Timer Counter Register) value at SOF.

Message Transmit: For transmitted messages of Mailbox-15 to 1, Timestamp always captures the CYCTR (Cycle Time Register) value or Cycle_Counter CCR[5:0] + CYCTR[15:6] value, depending on the programmed value in the bit 14 of TTCR0 (Timer Trigger Control Register 0), at SOF.

For messages transmitted from Mailboxes30 and 31, Timestamp captures the TCNTR (Timer Counter Register) value at SOF.

chanhvinh.com

Important: Please note that the TimeStamp is stored in a temporary register. Only after a successful transmission or reception the value is then copied into the related Mailbox field. The TimeStamp may also be updated if the CPU clears RXPR[N]/RFPR[N] at the same time that UMSR[N] is set in overrun, however it can be read properly before clearing RXPR[N]/RFPR[N].

(5) Tx-Trigger Time (TTT) and Time Trigger control

For Mailbox-29 to 24, when MBC is set to 000 (Bin) in time trigger mode (CMAX!= 3'b111), Tx-Trigger Time works as Time_Mark to determine the boundary between time windows. The TTT and TT control are comprised of two 16-bit read/write areas as follows. Mailbox-30 doesn't have TT control and works as Time_Ref.

Mailbox 30 to 24 can be used for reception if not used for transmission in TT mode. However they cannot join the event trigger transmission queue when the TT mode is used.

- Tx-Trigger Time

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TTT15	TTT14	TTT13	TTT12	TTT11	TTT10	TTT9	TTT8	TTT7	TTT6	TTT5	TTT4	TTT3	TTT2	TTT1	TTT0
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

- Time Trigger control

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TTW[1:0]		Offset[5:0]						0	0	0	0	0	rep_factor[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/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

The following figure shows the differences between all Mailboxes supporting Time Triggered mode.

MB29 to 24

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'114 + N*32	Tx-Trigger Time (Cycle Time)																Word	Trigger Time
H'116 + N*32	TTW[1:0]		Offset[5:0]						0	0	0	0	0	rep_factor[2:0]			Word	TT control

MB30

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'114 + N*32	Tx-Trigger Time (Cycle Time)																Word	Trigger Time

Figure 20.6 Tx-Trigger control field

chanhvinh.com

- **TTW[1:0] (Time Trigger Window):** These bits show the attribute of time windows. Please note that once a merged arbitrating window is opened by TTW = 2'b10, the window must be closed by TTW = 2'b11. Several messages with TTW = 2'b10 may be used within the start and the end of a merged arbitrating window.

TTW[1]	TTW[0]	Description
0	0	Exclusive window (initial value)
0	1	Arbitrating window
1	0	Start of merged arbitrating window
1	1	End of merged arbitrating window

The first 16-bit area specifies the time that triggers the transmission of the message in cycle time. The second 16-bit area specifies the basic cycle in the system matrix where the transmission must start (Offset) and the frequency for periodic transmission. When the internal TTT register matches to the CYCTR value, and the internal Offset matches to CCR value transmission is attempted from the corresponding Mailbox. In order to enable this function, the CMAX (Cycle Maximum Register) must be set to a value different from 3'b111, the Timer (TCNTR) must be running (TTCR0 bit15 = 1), the corresponding MBC must be set to 3'b000 and the corresponding TXPR bit must be set. Once TXPR is set by S/W, this module does not clear the corresponding TXPR bit (among Mailbox-30 to 24) to carry on performing the periodic transmission. In order to stop the periodic transmission, TXPR must be cleared by TXCR. Please note that in this case it is possible that both TXACK and ABACK are set for the same Mailbox if TXACK is not cleared right after completion of transmission. Please refer to figure 20.7.

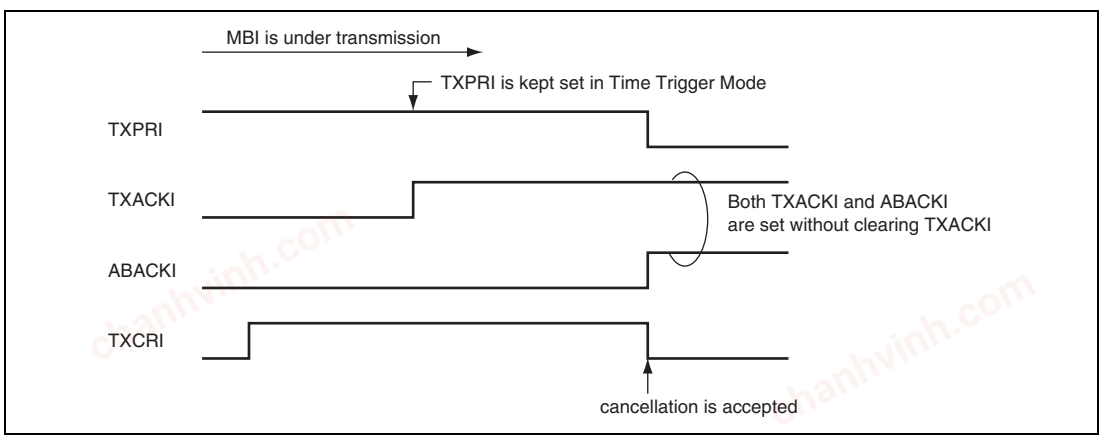


Figure 20.7 TXACK and ABACK in Time Trigger Transmission

chanhvinh.com

Please note that for Mailbox 30 TTW is fixed to '01', Offset to '00' and rep_factor to ‘0’.The following tables report the combinations for the rep_factor and the offset.

Rep_factor	Description
3'b000	Every basic cycle (initial value)
3'b001	Every two basic cycle
3'b010	Every four basic cycle
3'b011	Every eight basic cycle
3'b100	Every sixteen basic cycle
3'b101	Every thirty two basic cycle
3'b110	Every sixty four basic cycle (once in system matrix)
3'b111	Reserved

The Offset Field determines the first cycle in which a Time Triggered Mailbox may start transmitting its Message.

Offset	Description
6'b000000	Initial Offset = 1st Basic Cycle (initial value)
6'b000001	Initial Offset = 2nd Basic Cycles
6'b000010	Initial Offset = 3rd Basic Cycles
6'b000011	Initial Offset = 4th Basic Cycles
6'b000100	Initial Offset = 5th Basic Cycles
…	
…	
6'b111110	Initial Offset = 63rd Basic Cycles
6'b111111	Initial Offset = 64th Basic Cycles

The following relation must be maintained:

$$Cycle_Count_Maximum + 1 >= Repeat_Factor > Offset$$

$$Cycle_Count_Maximum = 2^{CMAX} - 1$$

$$Repeat_Factor = 2^{rep_factor}$$

chanhvinh.com

CMAX, Repeat_Factor, and Offset are register values

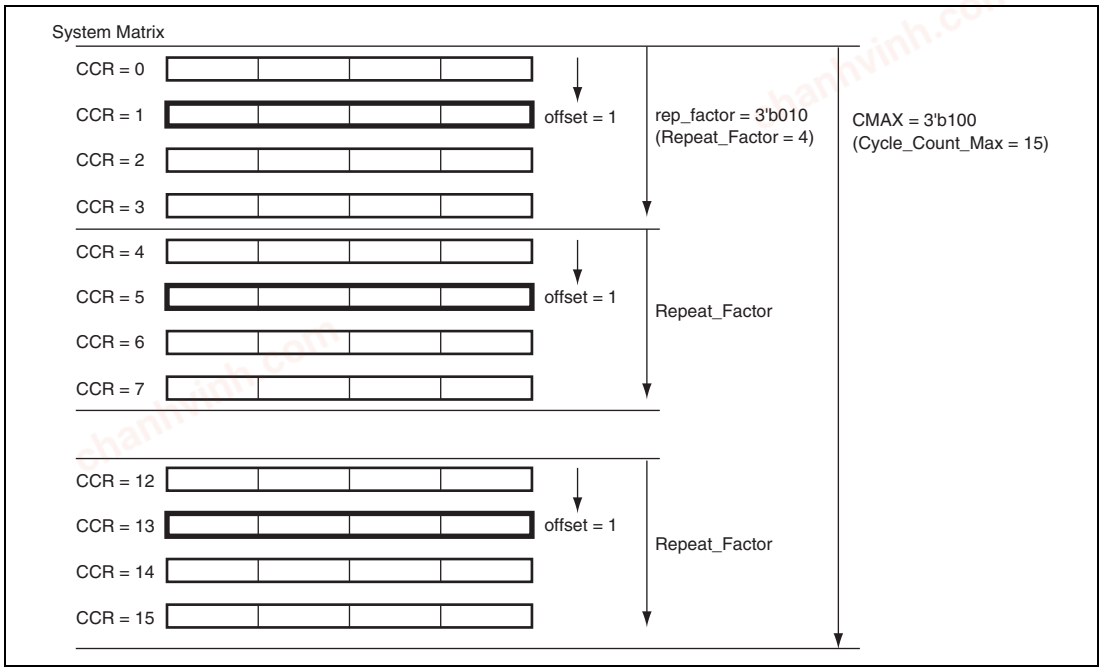


Figure 20.8 System Matrix

Tx-Trigger Times must be set in ascending order such that the difference between them satisfies the following condition.

TTT(mailbox i) –1 TTT(mailbox i-1) > TEW + Maximum frame length + 9

chanhvinh.com

20.3.3 Control Registers

The following sections describe control registers. The address is mapped as follow.

Important: These registers can only be accessed in Word size (16-bit).

Register Name	Address	Abbreviation	Access Size (bits)
Master Control Register	000	MCR	16
General Status Register	002	GSR	16
Bit Configuration Register 1	004	BCR1	16
Bit Configuration Register 0	006	BCR0	16
Interrupt Request Register	008	IRR	16
Interrupt Mask Register	00A	IMR	16
Error Counter Register	00C	TEC/REC	16

Figure 20.9 Control Registers

(1) Master Control Register (MCR)

The Master Control Register (MCR) is a 16-bit read/write register that controls this module.

- MCR (Address = H'000)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MCR15	MCR14	-	-	-	TST[2:0]			MCR7	MCR6	MCR5	-	-	MCR2	MCR1	MCR0
Initial value:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W

Bit 15 — ID Reorder (MCR15): This bit changes the order of STDID, RTR, IDE and EXTID of both message control and LAFM.

Bit15: MCR15	Description
0	This module is the same as HCAN2
1	This module is not the same as HCAN2 (Initial value)

chanhvinh.com

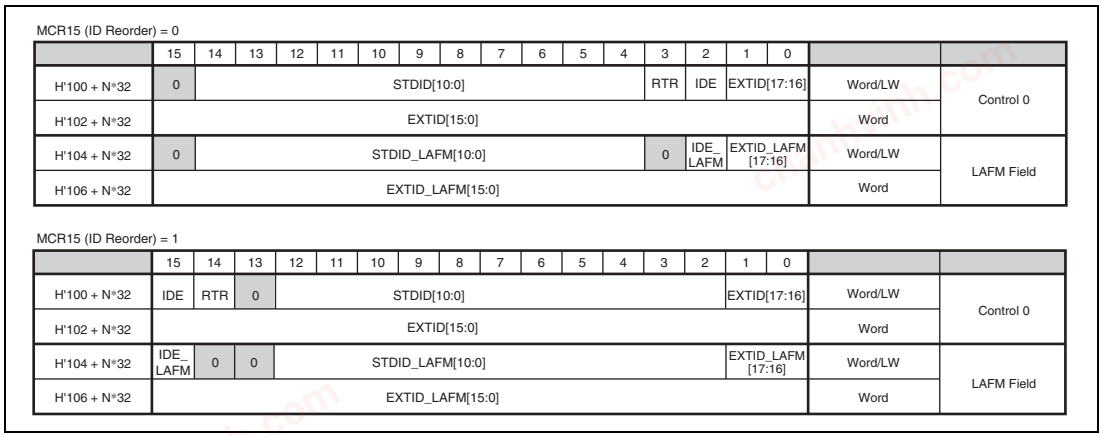


Figure 20.10 ID Reorder

This bit can be modified only in reset mode.

Bit 14 — Auto Halt Bus Off (MCR14): If both this bit and MCR6 are set, MCR1 is automatically set as soon as this module enters BusOff.

Bit14: MCR14	Description
0	This module remains in BusOff for normal recovery sequence (128 x 11 Recessive Bits) (Initial value)
1	This module moves directly into Halt Mode after it enters BusOff if MCR6 is set.

This bit can be modified only in reset mode.

Bit 13 — Reserved. The written value should always be ‘0’ and the returned value is '0'.

Bit 12 — Reserved. The written value should always be ‘0’ and the returned value is '0'.

Bit 11 — Reserved. The written value should always be ‘0’ and the returned value is '0'.

Bit 10 - 8 — Test Mode (TST[2:0]): This bit enables/disables the test modes. Please note that before activating the Test Mode it is requested to move this module into Halt mode or Reset mode. This is to avoid that the transition to Test Mode could affect a transmission/reception in progress. For details, please refer to section 20.4.1, Test Mode Settings.

Please note that the test modes are allowed only for diagnosis and tests and not when this module is used in normal operation.

chanhvinh.com

Bit10: TST2	Bit9: TST1	Bit8: TST0	Description
0	0	0	Normal Mode (initial value)
0	0	1	Listen-Only Mode (Receive-Only Mode)
0	1	0	Self Test Mode 1 (External)
0	1	1	Self Test Mode 2 (Internal)
1	0	0	Write Error Counter
1	0	1	Error Passive Mode
1	1	0	Setting prohibited
1	1	1	Setting prohibited

Bit 7 — Auto-wake Mode (MCR7): MCR7 enables or disables the Auto-wake mode. If this bit is set, this module automatically cancels the sleep mode (MCR5) by detecting CAN bus activity (dominant bit). If MCR7 is cleared this module does not automatically cancel the sleep mode.

This module cannot store the message that wakes it up.

Note: This bit can be modified only Reset or Halt mode.

Bit7: MCR7	Description
0	Auto-wake by CAN bus activity disabled (Initial value)
1	Auto-wake by CAN bus activity enabled

Bit 6 — Halt during Bus Off (MCR6): MCR6 enables or disables entering Halt mode immediately when MCR1 is set during Bus Off. This bit can be modified only in Reset or Halt mode. Please note that when Halt is entered in Bus Off the CAN engine is also recovering immediately to Error Active mode.

Bit6: MCR6	Description
0	If MCR[1] is set, this module will not enter Halt mode during Bus Off but wait up to end of recovery sequence (Initial value)
1	Enter Halt mode immediately during Bus Off if MCR[1] or MCR[14] are asserted.

chanhvinh.com

Bit 5 — Sleep Mode (MCR5): Enables or disables Sleep mode transition. If this bit is set, while this module is in halt mode, the transition to sleep mode is enabled. Setting MCR5 is allowed after entering Halt mode. The two Error Counters (REC, TEC) will remain the same during Sleep mode. This mode will be exited in two ways:

1. by writing a '0' to this bit position,
2. or, if MCR[7] is enabled, after detecting a dominant bit on the CAN bus.

If Auto wake up mode is disabled, this module will ignore all CAN bus activities until the sleep mode is terminated. When leaving this mode this module will synchronise to the CAN bus (by checking for 11 recessive bits) before joining CAN Bus activity. This means that, when the No.2 method is used, this module will miss the first message to receive. CAN transceivers stand-by mode will also be unable to cope with the first message when exiting stand by mode, and the S/W needs to be designed in this manner.

In sleep mode only the following registers can be accessed: MCR, GSR, IRR and IMR.

Important: This module is required to be in Halt mode before requesting to enter in Sleep mode. That allows the CPU to clear all pending interrupts before entering sleep mode. Once all interrupts are cleared this module must leave the Halt mode and enter Sleep mode simultaneously (by writing MCR[5] = 1 and MCR[1] = 0 at the same time).

Bit 5: MCR5	Description
0	This module sleep mode released (Initial value)
1	Transition to this module sleep mode enabled

Bit 4 — Reserved. The written value should always be '0' and the returned value is '0'.

Bit 3 — Reserved. The written value should always be '0' and the returned value is '0'.

Bit 2 — Message Transmission Priority (MCR2): MCR2 selects the order of transmission for pending transmit data. If this bit is set, pending transmit data are sent in order of the bit position in the Transmission Pending Register (TXPR). The order of transmission starts from Mailbox-31 as the highest priority, and then down to Mailbox-1 (if those mailboxes are configured for transmission). Please note that this feature cannot be used for time trigger transmission of the Mailboxes 24 to 30.

If MCR2 is cleared, all messages for transmission are queued with respect to their priority (by running internal arbitration). The highest priority message has the Arbitration Field (STDID + IDE bit + EXTID (if IDE = 1) + RTR bit) with the lowest digital value and is transmitted first. The internal arbitration includes the RTR bit and the IDE bit (internal arbitration works in the same

chanhvinh.com

way as the arbitration on the CAN Bus between two CAN nodes starting transmission at the same time).

This bit can be modified only in Reset or Halt mode.

Bit 2: MCR2	Description
0	Transmission order determined by message identifier priority (Initial value)
1	Transmission order determined by mailbox number priority (Mailbox-31 → Mailbox-1)

Bit 1—Halt Request (MCR1): Setting the MCR1 bit causes the CAN controller to complete its current operation and then enter Halt mode (where it is cut off from the CAN bus). This module remains in Halt Mode until the MCR1 is cleared. During the Halt mode, the CAN Interface does not join the CAN bus activity and does not store messages or transmit messages. All the user registers (including Mailbox contents and TEC/REC) remain unchanged with the exception of IRR0 and GSR4 which are used to notify the halt status itself. If the CAN bus is in idle or intermission state regardless of MCR6, this module will enter Halt Mode within one Bit Time. If MCR6 is set, a halt request during Bus Off will be also processed within one Bit Time. Otherwise the full Bus Off recovery sequence will be performed beforehand. Entering the Halt Mode can be notified by IRR0 and GSR4.

If both MCR14 and MCR6 are set, MCR1 is automatically set as soon as this module enters BusOff.

In the Halt mode, this module configuration can be modified with the exception of the Bit Timing setting, as it does not join the bus activity. MCR[1] has to be cleared by writing a '0' in order to re-join the CAN bus. After this bit has been cleared, this module waits until it detects 11 recessive bits, and then joins the CAN bus.

Notes: 1. After issuing a Halt request the CPU is not allowed to set TXPR or TXCR or clear MCR1 until the transition to Halt mode is completed (notified by IRR0 and GSR4). After MCR1 is set this can be cleared only after entering Halt mode or through a reset operation (SW or HW).
2. Transition into or recovery from HALT mode, is only possible if the BCR1 and BCR0 registers are configured to a proper Baud Rate.

Bit 1: MCR1	Description
0	Clear Halt request (Initial value)
1	Halt mode transition request

chanhvinh.com

Bit 0 — Reset Request (MCR0): Controls resetting of this module. When this bit is changed from '0' to '1' this module controller enters its reset routine, re-initialising the internal logic, which then sets GSR3 and IRR0 to notify the reset mode. During a re-initialisation, all user registers are initialised.

This module can be re-configured while this bit is set. This bit has to be cleared by writing a '0' to join the CAN bus. After this bit is cleared, this module waits until it detects 11 recessive bits, and then joins the CAN bus. The Baud Rate needs to be set up to a proper value in order to sample the value on the CAN Bus.

After Power On Reset, this bit and GSR3 are always set. This means that a reset request has been made and this module needs to be configured.

The Reset Request is equivalent to a Power On Reset but controlled by Software.

Bit 0: MCR0	Description
0	Clear Reset Request
1	CAN Interface reset mode transition request (Initial value)

(2) General Status Register (GSR)

The General Status Register (GSR) is a 16-bit read-only register that indicates the status of this module.

- GSR (Address = H'002)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bits 15 to 6: Reserved. The written value should always be '0' and the returned value is '0'.

Bit 5 — Error Passive Status Bit (GSR5): Indicates whether the CAN Interface is in Error Passive or not. This bit will be set high as soon as this module enters the Error Passive state and is cleared when the module enters again the Error Active state (this means the GSR5 will stay high during Error Passive and during Bus Off). Consequently to find out the correct state both GSR5 and GSR0 must be considered.

chanhvinh.com

Bit 5: GSR5	Description
0	This module is not in Error Passive or in Bus Off status (Initial value) [Reset condition] This module is in Error Active state
1	This module is in Error Passive (if GSR0 = 0) or Bus Off (if GSR0 = 1) [Setting condition] When TEC • 128 or REC • 128 or if Error Passive Test Mode is selected

Bit 4 — Halt/Sleep Status Bit (GSR4): Indicates whether the CAN engine is in the halt/sleep state or not. Please note that the clearing time of this flag is not the same as the setting time of IRR12.

Please note that this flag reflects the status of the CAN engine and not of the full this module IP. This module exits sleep mode and can be accessed once MCR5 is cleared. The CAN engine exits sleep mode only after two additional transmission clocks on the CAN Bus.

Bit 4: GSR4	Description
0	This module is not in the Halt state or Sleep state (Initial value)
1	Halt mode (if MCR1 = 1) or Sleep mode (if MCR5 = 1) [Setting condition] If MCR1 is set and the CAN bus is either in intermission or idle or MCR5 is set and this module is in the halt mode or this module is moving to Bus Off when MCR14 and MCR6 are both set

Bit 3 — Reset Status Bit (GSR3): Indicates whether this module is in the reset state or not.

Bit 3: GSR3	Description
0	This module is not in the reset state
1	Reset state (Initial value) [Setting condition] After an internal reset of this module (due to SW or HW reset)

chanhvinh.com

Bit 2 — Message Transmission in progress Flag (GSR2): Flag that indicates to the CPU if this module is in Bus Off or transmitting a message or an error/overload flag due to error detected during transmission. The timing to set TXACK is different from the time to clear GSR2. TXACK is set at the 7th bit of End Of Frame. GSR2 is set at the 3rd bit of intermission if there are no more messages ready to be transmitted. It is also set by arbitration lost, bus idle, reception, reset or halt transition.

Bit 2: GSR2	Description
0	This module is in Bus Off or a transmission is in progress
1	[Setting condition] Not in Bus Off and no transmission in progress (Initial value)

Bit 1—Transmit/Receive Warning Flag (GSR1): Flag that indicates an error warning.

Bit 1: GSR1	Description
0	[Reset condition] When (TEC < 96 and REC < 96) or Bus Off (Initial value)
1	[Setting condition] When 96 ≤ TEC • 256 or 96 ≤ REC • 256

Note: REC is incremented during Bus Off to count the recurrences of 11 recessive bits as requested by the Bus Off recovery sequence. However the flag GSR1 is not set in Bus Off.

Bit 0—Bus Off Flag (GSR0): Flag that indicates that this module is in the bus off state.

Bit 0: GSR0	Description
0	[Reset condition] Recovery from bus off state or after a HW or SW reset (Initial value)
1	[Setting condition] When TEC • 256 (bus off state)

Note: Only the lower 8 bits of TEC are accessible from the user interface. The 9th bit is equivalent to GSR0.

chanhvinh.com

(3) Bit Configuration Register (BCR0, BCR1)

The bit configuration registers (BCR0 and BCR1) are 2 X 16-bit read/write register that are used to set CAN bit timing parameters and the baud rate pre-scaler for the CAN Interface.

The Time quanta is defined as:

$$Timequanta = \frac{2 * BRP}{f_{clk}}$$

Where: BRP (Baud Rate Pre-scaler) is the value stored in BCR0 incremented by 1 and fclk is the used peripheral clock frequency.

- BCR1 (Address = H'004)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG1[3:0]				-	TSG2[2:0]			-	-	SJW[1:0]		-	-	-	BSP
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/W	R/W	R	R	R	R/W

Bits 15 to 12 — Time Segment 1 (TSG1[3:0] = BCR1[15:12]): These bits are used to set the segment TSEG1 (= PRSEG + PHSEG1) to compensate for edges on the CAN Bus with a positive phase error. A value from 4 to 16 time quanta can be set.

Bit 15: TSG1[3]	Bit 14: TSG1[2]	Bit 13: TSG1[1]	Bit 12: TSG1[0]	Description
0	0	0	0	Setting prohibited (Initial value)
0	0	0	1	Setting prohibited
0	0	1	0	Setting prohibited
0	0	1	1	PRSEG + PHSEG1 = 4 time quanta
0	1	0	0	PRSEG + PHSEG1 = 5 time quanta
:	:	:	:	:
:	:	:	:	:
1	1	1	1	PRSEG + PHSEG1 = 16 time quanta

Bit 11: Reserved. The written value should always be '0' and the returned value is '0'.

chanhvinh.com

Bits 10 to 8 — Time Segment 2 (TSG2[2:0] = BCR1[10:8]): These bits are used to set the segment TSEG2 (= PHSEG2) to compensate for edges on the CAN Bus with a negative phase error. A value from 2 to 8 time quanta can be set as shown below.

Bit 10: TSG2[2]	Bit 9: TSG2[1]	Bit 8: TSG2[0]	Description
0	0	0	Setting prohibited (Initial value)
0	0	1	PHSEG2 = 2 time quanta (conditionally prohibited)
0	1	0	PHSEG2 = 3 time quanta
0	1	1	PHSEG2 = 4 time quanta
1	0	0	PHSEG2 = 5 time quanta
1	0	1	PHSEG2 = 6 time quanta
1	1	0	PHSEG2 = 7 time quanta
1	1	1	PHSEG2 = 8 time quanta

Bits 7 and 6: Reserved. The written value should always be '0' and the returned value is '0'.

Bits 5 and 4 - ReSynchronisation Jump Width (SJW[1:0] = BCR0[5:4]): These bits set the synchronisation jump width.

Bit 5: SJW[1]	Bit 4: SJW[0]	Description
0	0	Synchronisation Jump width = 1 time quantum (Initial value)
0	1	Synchronisation Jump width = 2 time quanta
1	0	Synchronisation Jump width = 3 time quanta
1	1	Synchronisation Jump width = 4 time quanta

Bits 3 to 1: Reserved. The written value should always be '0' and the returned value is '0'.

Bit 0 — Bit Sample Point (BSP = BCR1[0]): Sets the point at which data is sampled.

Bit 0 : BSP	Description
0	Bit sampling at one point (end of time segment 1) (Initial value)
1	Bit sampling at three points (rising edge of the last three clock cycles of PHSEG1)

chanhvinh.com

- BCR0 (Address = H'006)

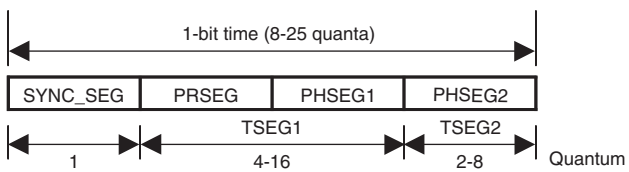
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	BRP[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 8 to 15: Reserved. The written value should always be '0' and the returned value is '0'.

Bits 7 to 0—Baud Rate Pre-scale (BRP[7:0] = BCR0 [7:0]): These bits are used to define the peripheral clock periods contained in a Time Quantum.

Bit 7: BRP[7]	Bit 6: BRP[6]	Bit 5: BRP[5]	Bit 4: BRP[4]	Bit 3: BRP[3]	Bit 2: BRP[2]	Bit 1: BRP[1]	Bit 0: BRP[0]	Description
0	0	0	0	0	0	0	0	2 X peripheral clock (Initial value)
0	0	0	0	0	0	0	1	4 X peripheral clock
0	0	0	0	0	0	1	0	6 X peripheral clock
: :	: :	: :	: :	: :	: :	: :	: :	2*(register value + 1) X peripheral clock
1	1	1	1	1	1	1	1	512 X peripheral clock

- Requirements of Bit Configuration Register



SYNC_SEG: Segment for establishing synchronisation of nodes on the CAN bus. (Normal bit edge transitions occur in this segment.)

PRSEG: Segment for compensating for physical delay between networks.

PHSEG1: Buffer segment for correcting phase drift (positive). (This segment is extended when synchronisation (resynchronisation) is established.)

PHSEG2: Buffer segment for correcting phase drift (negative). (This segment is shortened when synchronisation (resynchronisation) is established)

TSEG1: TSG1 + 1

chanhvinh.com

TSEG2: TSG2 + 1

The Bit Rate Calculation is:

$$\text{Bit Rate} = \frac{f_{clk}}{2 \times (BRP + 1) \times (TSEG1 + TSEG2 + 1)}$$

Where BRP is given by the register value and TSEG1 and TSEG2 are derived values from TSG1 and TSG2 register values. The '+1' in the above formula is for the Sync-Seg which duration is 1 time quanta.

f_{CLK} = Peripheral Clock

BCR Setting Constraints

TSEG1min > TSEG2 ≥ SJWmax (SJW = 1 to 4)

8 ≤ TSEG1 + TSEG2 + 1 ≤ 25 time quanta (TSEG1 + TSEG2 + 1 = 7 is not allowed)

TSEG2 ≥ 2

These constraints allow the setting range shown in the table below for TSEG1 and TSEG2 in the Bit Configuration Register. The number in the table shows possible setting of SJW. "No" shows that there is no allowed combination of TSEG1 and TSEG2.

chanhvinh.com

		001	010	011	100	101	110	111	TSG2
		2	3	4	5	6	7	8	TSEG2
TSG1	TSEG1								
0011	4	No	1-3	No	No	No	No	No	
0100	5	1-2	1-3	1-4	No	No	No	No	
0101	6	1-2	1-3	1-4	1-4	No	No	No	
0110	7	1-2	1-3	1-4	1-4	1-4	No	No	
0111	8	1-2	1-3	1-4	1-4	1-4	1-4	No	
1000	9	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1001	10	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1010	11	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1011	12	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1100	13	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1101	14	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1110	15	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1111	16	1-2	1-3	1-4	1-4	1-4	1-4	1-4	

Example 1: To have a Bit rate of 500Kbps with a frequency of fclk = 24MHz it is possible to set: BRP = 1, TSEG1 = 6, TSEG2 = 5.

Then the configuration to write is BCR1 = H'5400 and BCR0 = H'0001.

Example 2: To have a Bit rate of 500Kbps with a frequency of fclk = 36MHz it is possible to set: BRP = 1, TSEG1 = 10, TSEG2 = 7.

Then the configuration to write is BCR1 = H'9600 and BCR0 = H'0001.

chanhvinh.com

(4) Interrupt Request Register (IRR)

The interrupt request register (IRR) is a 16-bit read/write-clearable register containing status flags for the various interrupt sources.

- IRR (Address = H'008)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRR15	IRR14	IRR13	IRR12	IRR11	IRR10	IRR9	IRR8	IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0
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	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W

Bit 15 — Timer Compare Match Interrupt 1 (IRR15): Indicates that a Compare-Match condition occurred to the Timer Compare Match Register 1 (TCMR1). When the value set in the TCMR1 matches to Cycle Time (TCMR1 = CYCTR), this bit is set.

Bit 15: IRR15	Description
0	Timer Compare Match has not occurred to the TCMR1 (Initial value) [Clearing condition] Writing 1
1	Timer Compare Match has occurred to the TCMR1 [Setting condition] TCMR1 matches to Cycle Time (TCMR1 = CYCTR)

Bit 14 — Timer Compare Match Interrupt 0 (IRR14): Indicates that a Compare-Match condition occurred to the Timer Compare Match Register 0 (TCMR0). When the value set in the TCMR0 matches to Local Time (TCMR0 = TCNTR), this bit is set.

Bit 14: IRR14	Description
0	Timer Compare Match has not occurred to the TCMR0 (Initial value) [Clearing condition] Writing 1
1	Timer Compare Match has occurred to the TCMR0 [Setting condition] TCMR0 matches to the Timer value (TCMR0 = TCNTR)

Bit 13 - Timer Overrun Interrupt/Next_is_Gap Reception Interrupt/Message Error Interrupt (IRR13): This interrupt assumes a different meaning depending on this module mode. It indicates that:

— The Timer (TCNTR) has overrun when this module is working in event-trigger mode (including test modes)

chanhvinh.com

— Time reference message with Next_is_Gap set has been received when working in time-trigger mode. Please note that when a Next_is_Gap is received the application is responsible to stop all transmission at the end of the current basic cycle (including test modes)

— Message error has occurred when in test mode. Note: If a Message Overload condition occurs when in Test Mode, then this bit will not be set.

Bit 13: IRR13	Description
0	Timer (TCNTR) has not overrun in event-trigger mode (including test modes) (Initial value) Time reference message with Next_is_Gap has not been received in time-trigger mode (including test modes) Message error has not occurred in test mode [Clearing condition] Writing 1
1	[Setting condition] Timer (TCNTR) has overrun and changed from H'FFFF to H'0000 in event-trigger mode (including test modes) Time reference message with Next_is_Gap has been received in time-trigger mode (including test modes) Message error has occurred in test mode

Bit 12 – Bus activity while in sleep mode (IRR12): IRR12 indicates that a CAN bus activity is present. While this module is in sleep mode and a dominant bit is detected on the CAN bus, this bit is set. This interrupt is cleared by writing a '1' to this bit position. Writing a '0' has no effect. If auto wakeup is not used and this interrupt is not requested it needs to be disabled by the related interrupt mask register. If auto wake up is not used and this interrupt is requested it should be cleared only after recovering from sleep mode. This is to avoid that a new falling edge of the reception line causes the interrupt to get set again.

Please note that the setting time of this interrupt is different from the clearing time of GSR4.

Bit 12: IRR12	Description
0	Bus idle state (Initial value) [Clearing condition] Writing 1
1	CAN bus activity detected in this module sleep mode [Setting condition] Dominant bit level detection on the Rx line while in sleep mode

chanhvinh.com

Bit 11 — Timer Compare Match Interrupt 2 (IRR11): Indicates that a Compare-Match condition occurred to the Timer Compare Match Register 2 (TCMR2). When the value set in the TCMR2 matches to Cycle Time (TCMR2 = CYCTR), this bit is set.

Bit 11: IRR11	Description
0	Timer Compare Match has not occurred to the TCMR2 (initial value) [Clearing condition] Writing 1
1	Timer Compare Match has occurred to the TCMR2 [Setting condition] TCMR2 matches to Cycle Time (TCMR2 = CYCTR)

Bit 10 — Start of new system matrix Interrupt (IRR10): Indicates that a new system matrix is starting.

When CCR = 0, this bit is set at the successful completion of reception/transmission of time reference message. Please note that when CMAX = 0 this interrupt is set at every basic cycle.

Bit 10: IRR10	Description
0	A new system matrix is not starting (initial value) [Clearing condition] Writing 1
1	Cycle counter reached zero. [Setting condition] Reception/transmission of time reference message is successfully completed when CMAX!= 3'b111 and CCR = 0

Bit 9 – Message Overrun/Overwrite Interrupt Flag (IRR9): Flag indicating that a message has been received but the existing message in the matching Mailbox has not been read as the corresponding RXPR or RFPR is already set to '1' and not yet cleared by the CPU. The received message is either abandoned (overrun) or overwritten dependant upon the NMC (New Message Control) bit. This bit is cleared when all bit in UMSR (Unread Message Status Register) are cleared (by writing '1') or by setting MBIMR (MailBox interrupt Mast Register) for all UMSR flag set. It is also cleared by writing a '1' to all the correspondent bit position in MBIMR. Writing to this bit position has no effect.

chanhvinh.com

Bit 9: IRR9	Description
0	No pending notification of message overrun/overwrite [Clearing condition] Clearing of all bit in UMSR/setting MBIMR for all UMSR set (initial value)
1	A receive message has been discarded due to overrun condition or a message has been overwritten [Setting condition] Message is received while the corresponding RXPR and/or RFPR = 1 and MBIMR = 0

Bit 8 - Mailbox Empty Interrupt Flag (IRR8): This bit is set when one of the messages set for transmission has been successfully sent (corresponding TXACK flag is set) or has been successfully aborted (corresponding ABACK flag is set). In Event Triggered mode the related TXPR is also cleared and this mailbox is now ready to accept a new message data for the next transmission. In Time Trigger mode TXPR for the Mailboxes from 30 to 24 is not cleared after a successful transmission in order to keep transmitting at each programmed basic cycle. In effect, this bit is set by an OR'ed signal of the TXACK and ABACK bits not masked by the corresponding MBIMR flag. Therefore, this bit is automatically cleared when all the TXACK and ABACK bits are cleared. It is also cleared by writing a '1' to all the correspondent bit position in MBIMR. Writing to this bit position has no effect.

Bit 8: IRR8	Description
0	Messages set for transmission or transmission cancellation request NOT progressed. (Initial value) [Clearing Condition] All the TXACK and ABACK bits are cleared/setting MBIMR for all TXACK and ABACK set
1	Message has been transmitted or aborted, and new message can be stored (in TT mode Mailbox 24 to 30 can be programmed with a new message only in case of abortion) [Setting condition] When a TXACK or ABACK bit is set (if related MBIMR = 0).

chanhvinh.com

Bit 7 - Overload Frame (IRR7): Flag indicating that this module has detected a condition that should initiate the transmission of an overload frame. Note that in the condition of transmission being prevented, such as listen only mode, an Overload Frame will NOT be transmitted, but IRR7 will still be set. IRR7 remains asserted until reset by writing a '1' to this bit position - writing a '0' has no effect.

Bit 7: IRR7	Description
0	[Clearing condition] Writing 1 (Initial value)
1	[Setting conditions] Overload condition detected

Bit 6 - Bus Off Interrupt Flag (IRR6): This bit is set when this module enters the Bus-off state or when this module leaves Bus-off and returns to Error-Active. The cause therefore is the existing condition TEC ≥ 256 at the node or the end of the Bus-off recovery sequence (128X11 consecutive recessive bits) or the transition from Bus Off to Halt (automatic or manual). This bit remains set even if this module node leaves the bus-off condition, and needs to be explicitly cleared by S/W. The S/W is expected to read the GSR0 to judge whether this module is in the bus-off or error active status. It is cleared by writing a '1' to this bit position even if the node is still bus-off. Writing a '0' has no effect.

Bit 6: IRR6	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Enter Bus off state caused by transmit error or Error Active state returning from Bus-off [Setting condition] When TEC becomes ≥ 256 or End of Bus-off after 128X11 consecutive recessive bits or transition from Bus Off to Halt

Bit 5 - Error Passive Interrupt Flag (IRR5): Interrupt flag indicating the error passive state caused by the transmit or receive error counter or by Error Passive forced by test mode. This bit is reset by writing a '1' to this bit position, writing a '0' has no effect. If this bit is cleared the node may still be error passive. Please note that the SW needs to check GSR0 and GSR5 to judge whether this module is in Error Passive or Bus Off status.

Bit 5: IRR5	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Error passive state caused by transmit/receive error [Setting condition] When TEC ≥ 128 or REC ≥ 128 or Error Passive test mode is used

chanhvinh.com

Bit 4 - Receive Error Counter Warning Interrupt Flag (IRR4): This bit becomes set if the receive error counter (REC) reaches a value greater than 95 when this module is not in the Bus Off status. The interrupt is reset by writing a '1' to this bit position, writing '0' has no effect.

Bit 4: IRR4	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Error warning state caused by receive error [Setting condition] When REC ≥ 96 and this module is not in Bus Off

Bit 3 - Transmit Error Counter Warning Interrupt Flag (IRR3): This bit becomes set if the transmit error counter (TEC) reaches a value greater than 95. The interrupt is reset by writing a '1' to this bit position, writing '0' has no effect.

Bit 3: IRR3	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Error warning state caused by transmit error [Setting condition] When TEC ≥ 96

Bit 2 - Remote Frame Receive Interrupt Flag (IRR2): Flag indicating that a remote frame has been received in a mailbox. This bit is set if at least one receive mailbox, with related MBIMR not set, contains a remote frame transmission request. This bit is automatically cleared when all bits in the Remote Frame Receive Pending Register (RFPR), are cleared. It is also cleared by writing a '1' to all the correspondent bit position in MBIMR. Writing to this bit has no effect.

Bit 2: IRR2	Description
0	[Clearing condition] Clearing of all bits in RFPR (Initial value)
1	At least one remote request is pending [Setting condition] When remote frame is received and the corresponding MBIMR = 0

Bit 1 – Data Frame Received Interrupt Flag (IRR1): IRR1 indicates that there are pending Data Frames received. If this bit is set at least one receive mailbox contains a pending message. This bit is cleared when all bits in the Data Frame Receive Pending Register (RXPR) are cleared, i.e. there is no pending message in any receiving mailbox. It is in effect a logical OR of the RXPR flags from each configured receive mailbox with related MBIMR not set. It is also cleared by writing a '1' to all the correspondent bit position in MBIMR. Writing to this bit has no effect.

chanhvinh.com

Bit 1: IRR1	Description
0	[Clearing condition] Clearing of all bits in RXPR (Initial value)
1	Data frame received and stored in Mailbox [Setting condition] When data is received and the corresponding MBIMR = 0

Bit 0 – Reset/Halt/Sleep Interrupt Flag (IRR0): This flag can get set for three different reasons. It can indicate that:

1. Reset mode has been entered after a SW (MCR0) or HW reset
2. Halt mode has been entered after a Halt request (MCR1)
3. Sleep mode has been entered after a sleep request (MCR5) has been made while in Halt mode.

The GSR may be read after this bit is set to determine which state this module is in.

Important: When a Sleep mode request needs to be made, the Halt mode must be used beforehand. Please refer to the MCR5 description and Figure 20.15 Halt Mode/Sleep Mode.

IRR0 is set by the transition from "0" to "1" of GSR3 or GSR4 or by transition from Halt mode to Sleep mode. So, IRR0 is not set if this module enters Halt mode again right after exiting from Halt mode, without GSR4 being cleared. Similarly, IRR0 is not set by direct transition from Sleep mode to Halt Request. At the transition from Halt/Sleep mode to Transition/Reception, clearing GSR4 needs (one-bit time - TSEG2) to (one-bit time * 2 - TSEG2).

In the case of Reset mode, IRR0 is set, however, the interrupt to the CPU is not asserted since IMR0 is automatically set by initialisation.

Bit 0: IRR0	Description
0	[Clearing condition] Writing 1
1	Transition to S/W reset mode or transition to halt mode or transition to sleep mode (Initial value) [Setting condition] When reset/halt/sleep transition is completed after a reset (MCR0 or HW) or Halt mode (MCR1) or Sleep mode (MCR5) is requested

chanhvinh.com

(5) Interrupt Mask Register (IMR)

The interrupt mask register is a 16 bit register that protects all corresponding interrupts in the Interrupt Request Register (IRR) from generating an output signal on the IRQ. An interrupt request is masked if the corresponding bit position is set to '1'. This register can be read or written at any time. The IMR directly controls the generation of IRQ, but does not prevent the setting of the corresponding bit in the IRR.

- IMR (Address = H'00A)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IMR15	IMR14	IMR13	IMR12	IMR11	IMR10	IMR9	IMR8	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0
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 15 to 0: Maskable interrupt sources corresponding to IRR[15:0] respectively. When a bit is set, the interrupt signal is not generated, although setting the corresponding IRR bit is still performed.

Bit[15:0]: IMRn	Description
0	Corresponding IRR is not masked (IRQ is generated for interrupt conditions)
1	Corresponding interrupt of IRR is masked (Initial value)

(6) Transmit Error Counter (TEC) and Receive Error Counter (REC)

The Transmit Error Counter (TEC) and Receive Error Counter (REC) is a 16-bit read/(write) register that functions as a counter indicating the number of transmit/receive message errors on the CAN Interface. The count value is stipulated in the CAN protocol specification Refs. [1], [2], [3] and [4]. When not in (Write Error Counter) test mode this register is read only, and can only be modified by the CAN Interface. This register can be cleared by a Reset request (MCR0) or entering to bus off.

In Write Error Counter test mode (i.e. TST[2:0] = 3'b100), it is possible to write to this register. The same value can only be written to TEC/REC, and the value written into TEC is set to TEC and REC. When writing to this register, this module needs to be put into Halt Mode. This feature is only intended for test purposes.

chanhvinh.com

- TEC/REC (Address = H'00C)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0
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: * It is only possible to write the value in test mode when TST[2:0] in MCR is 3'b100.
REC is incremented during Bus Off to count the recurrences of 11 recessive bits as requested by the Bus Off recovery sequence.

20.3.4 Mailbox Registers

The following sections describe Mailbox registers that control/flag individual Mailboxes. The address is mapped as follows.

Important: LongWord access is carried out as two consecutive Word accesses.

chanhvinh.com

32-Mailboxes version

Description	Address	Name	Access Size (bits)
Transmit Pending 1	020	TXPR1	LW
Transmit Pending 0	022	TXPR0	—
	024		
	026		
Transmit Cancel 1	028	TXCR1	Word/LW
Transmit Cancel 0	02A	TXCR0	Word
	02C		
	02E		
Transmit Acknowledge 1	030	TXACK1	Word/LW
Transmit Acknowledge 0	032	TXACK0	Word
	034		
	036		
Abort Acknowledge 1	038	ABACK1	Word/LW
Abort Acknowledge 0	03A	ABACK0	Word
	03C		
	03E		
Data Frame Receive Pending 1	040	RXPR1	Word/LW
Data Frame Receive Pending 0	042	RXPR0	Word
	044		
	046		
Remote Frame Receive Pending 1	048	RFPR1	Word/LW
Remote Frame Receive Pending 0	04A	RFPR0	Word
	04C		
	04E		
Mailbox Interrupt Mask Register 1	050	MBIMR1	Word/LW
Mailbox Interrupt Mask Register 0	052	MBIMR0	Word
	054		
	056		
Unread message Status Register 1	058	UMSR1	Word/LW
Unread message Status Register 0	05A	UMSR0	Word
	05C		
	05E		

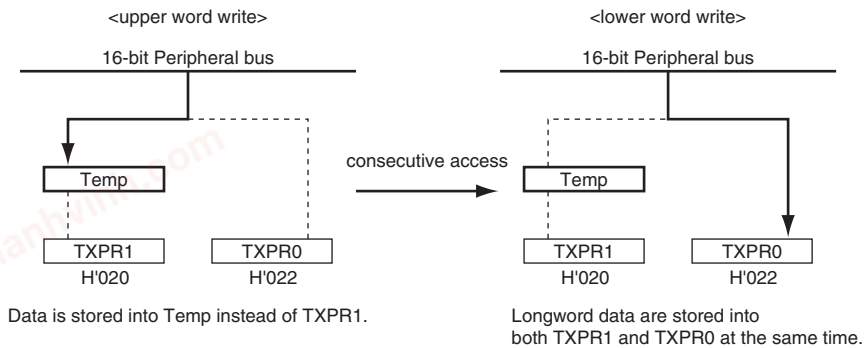
Figure 20.11 Mailbox Registers

chanhvinh.com

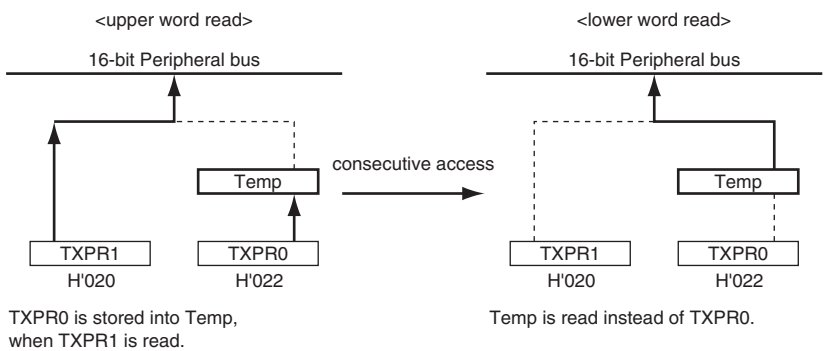
(1) Transmit Pending Register (TXPR1, TXPR0)

The concatenation of TXPR1 and TXPR0 is a 32-bit register that contains any transmit pending flags for the CAN module. In the case of 16-bit bus interface, Long Word access is carried out as two consecutive word accesses.

<Longword Write Operation>



<Longword Read Operation>



The TXPR1 controls Mailbox-31 to Mailbox-16, and the TXPR0 controls Mailbox-15 to Mailbox-1. The CPU may set the TXPR bits to affect any message being considered for transmission by writing a '1' to the corresponding bit location. Writing a '0' has no effect, and TXPR cannot be cleared by writing a '0' and must be cleared by setting the corresponding TXCR bits. TXPR may be read by the CPU to determine which, if any, transmissions are pending or in progress. In effect there is a transmit pending bit for all Mailboxes except for the Mailbox-0. Writing a '1' to a bit location when the mailbox is not configured to transmit is not allowed.

chanhvinh.com

In Event Triggered Mode this module will clear a transmit pending flag after successful transmission of its corresponding message or when a transmission abort is requested successfully from the TXCR. In Time Trigger Mode, TXPR for the Mailboxes from 30 to 24 is NOT cleared after a successful transmission, in order to keep transmitting at each programmed basic cycle. The TXPR flag is not cleared if the message is not transmitted due to the CAN node losing the arbitration process or due to errors on the CAN bus, and this module automatically tries to transmit it again unless its DART bit (Disable Automatic Re-Transmission) is set in the Message-Control of the corresponding Mailbox. In such case (DART set), the transmission is cleared and notified through Mailbox Empty Interrupt Flag (IRR8) and the correspondent bit within the Abort Acknowledgement Register (ABACK).

If the status of the TXPR changes, this module shall ensure that in the identifier priority scheme (MCR2 = 0), the highest priority message is always presented for transmission in an intelligent way even under circumstances such as bus arbitration losses or errors on the CAN bus. Please refer to the Application Note for details.

When this module changes the state of any TXPR bit position to a '0', an empty slot interrupt (IRR8) may be generated. This indicates that either a successful or an aborted mailbox transmission has just been made. If a message transmission is successful it is signalled in the TXACK register, and if a message transmission abortion is successful it is signalled in the ABACK register. By checking these registers, the contents of the Message of the corresponding Mailbox may be modified to prepare for the next transmission.

- TXPR1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXPR1[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*

Note: * It is possible only to write a '1' for a Mailbox configured as transmitter.

Bit 15 to 0 — Requests the corresponding Mailbox to transmit a CAN Frame. The bit 15 to 0 corresponds to Mailbox-31 to 16 respectively. When multiple bits are set, the order of the transmissions is governed by the MCR2 – CAN-ID or Mailbox number.

chanhvinh.com

Bit[15:0]: TXPR1	Description
0	Transmit message idle state in corresponding mailbox (Initial value) [Clearing Condition] Completion of message transmission (for Event Triggered Messages) or message transmission abortion (automatically cleared)
1	Transmission request made for corresponding mailbox

- TXPR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXPR0[15:1]															-
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

Note: * It is possible only to write a '1' for a Mailbox configured as transmitter.

Bit 15 to 1 — Indicates that the corresponding Mailbox is requested to transmit a CAN Frame. The bit 15 to 1 corresponds to Mailbox-15 to 1 respectively. When multiple bits are set, the order of the transmissions is governed by the MCR2 – CAN-ID or Mailbox number.

Bit[15:1]: TXPR0	Description
0	Transmit message idle state in corresponding mailbox (Initial value) [Clearing Condition] Completion of message transmission (for Event Triggered Messages) or message transmission abortion (automatically cleared)
1	Transmission request made for corresponding mailbox

Bit 0— Reserved: This bit is always '0' as this is a receive-only Mailbox. Writing a '1' to this bit position has no effect. The returned value is '0'.

chanhvinh.com

(2) Transmit Cancel Register (TXCR1, TXCR0)

The TXCR1 and TXCR0 are 16-bit read/conditionally-write registers. The TXCR1 controls Mailbox-31 to Mailbox-16, and the TXCR0 controls Mailbox-15 to Mailbox-1.This register is used by the CPU to request the pending transmission requests in the TXPR to be cancelled. To clear the corresponding bit in the TXPR the CPU must write a '1' to the bit position in the TXCR. Writing a '0' has no effect.

When an abort has succeeded the CAN controller clears the corresponding TXPR + TXCR bits, and sets the corresponding ABACK bit. However, once a Mailbox has started a transmission, it cannot be cancelled by this bit. In such a case, if the transmission finishes in success, the CAN controller clears the corresponding TXPR + TXCR bit, and sets the corresponding TXACK bit, however, if the transmission fails due to a bus arbitration loss or an error on the bus, the CAN controller clears the corresponding TXPR + TXCR bit, and sets the corresponding ABACK bit. If an attempt is made by the CPU to clear a mailbox transmission that is not transmit-pending it has no effect. In this case the CPU will be not able at all to set the TXCR flag.

- TXCR1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXCR1[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*

Note: * Only writing a ‘1’ to a Mailbox that is requested for transmission and is configured as transmit.

Bit 15 to 0 — Requests the corresponding Mailbox, that is in the queue for transmission, to cancel its transmission. The bit 15 to 0 corresponds to Mailbox-31 to 16 (and TXPR1[15:0]) respectively.

Bit[15:0]:TXCR1	Description
0	Transmit message cancellation idle state in corresponding mailbox (Initial value) [Clearing Condition] Completion of transmit message cancellation (automatically cleared)
1	Transmission cancellation request made for corresponding mailbox

chanhvinh.com

- TXCR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXCR0[15:1]															-
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

Note: * Only writing a '1' to a Mailbox that is requested for transmission and is configured as transmit.

Bit 15 to 1 — Requests the corresponding Mailbox, that is in the queue for transmission, to cancel its transmission. The bit 15 to 1 corresponds to Mailbox-15 to 1 (and TXPR0[15:1]) respectively.

Bit[15:1]: TXCR0	Description
0	Transmit message cancellation idle state in corresponding mailbox (Initial value) [Clearing Condition] Completion of transmit message cancellation (automatically cleared)
1	Transmission cancellation request made for corresponding mailbox

Bit 0 — This bit is always '0' as this is a receive-only mailbox. Writing a '1' to this bit position has no effect and always read back as a ‘0’.

(3) Transmit Acknowledge Register (TXACK1, TXACK0)

The TXACK1 and TXACK0 are 16-bit read/conditionally-write registers. These registers are used to signal to the CPU that a mailbox transmission has been successfully made. When a transmission has succeeded this module sets the corresponding bit in the TXACK register. The CPU may clear a TXACK bit by writing a '1' to the corresponding bit location. Writing a '0' has no effect.

- TXACK1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXACK1[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*

Note: * Only when writing a '1' to clear.

chanhvinh.com

Bit 15 to 0 — Notifies that the requested transmission of the corresponding Mailbox has been finished successfully. The bit 15 to 0 corresponds to Mailbox-31 to 16 respectively.

Bit[15:0]:TXACK1	Description
0	[Clearing Condition] Writing '1' (Initial value)
1	Corresponding Mailbox has successfully transmitted message (Data or Remote Frame) [Setting Condition] Completion of message transmission for corresponding mailbox

- TXACK0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXACK0[15:1]															-
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*	-

Note: * Only when writing a '1' to clear.

Bit 15 to 1 — Notifies that the requested transmission of the corresponding Mailbox has been finished successfully. The bit 15 to 1 corresponds to Mailbox-15 to 1 respectively.

Bit[15:1]:TXACK0	Description
0	[Clearing Condition] Writing '1' (Initial value)
1	Corresponding Mailbox has successfully transmitted message (Data or Remote Frame) [Setting Condition] Completion of message transmission for corresponding mailbox

Bit 0 — This bit is always '0' as this is a receive-only mailbox. Writing a '1' to this bit position has no effect and always read back as a '0'.

chanhvinh.com

(4) Abort Acknowledge Register (ABACK1, ABACK0)

The ABACK1 and ABACK0 are 16-bit read/conditionally-write registers. These registers are used to signal to the CPU that a mailbox transmission has been aborted as per its request. When an abort has succeeded this module sets the corresponding bit in the ABACK register. The CPU may clear the Abort Acknowledge bit by writing a '1' to the corresponding bit location. Writing a '0' has no effect. An ABACK bit position is set by this module to acknowledge that a TXPR bit has been cleared by the corresponding TXCR bit.

- ABACK1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ABACK1[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*

Note: * Only when writing a '1' to clear.

Bit 15 to 0 — Notifies that the requested transmission cancellation of the corresponding Mailbox has been performed successfully. The bit 15 to 0 corresponds to Mailbox-31 to 16 respectively.

Bit[15:0]:ABACK1	Description
0	[Clearing Condition] Writing '1' (Initial value)
1	Corresponding Mailbox has cancelled transmission of message (Data or Remote Frame) [Setting Condition] Completion of transmission cancellation for corresponding mailbox

- ABACK0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ABACK0[15:1]															-
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

Note: * Only when writing a '1' to clear.

Bit 15 to 1 — Notifies that the requested transmission cancellation of the corresponding Mailbox has been performed successfully. The bit 15 to 1 corresponds to Mailbox-15 to 1 respectively.

chanhvinh.com

Bit[15:1]:ABACK0	Description
0	[Clearing Condition] Writing '1' (Initial value)
1	Corresponding Mailbox has cancelled transmission of message (Data or Remote Frame) [Setting Condition] Completion of transmission cancellation for corresponding mailbox

Bit 0 — This bit is always '0' as this is a receive-only mailbox. Writing a '1' to this bit position has no effect and always read back as a '0'.

(5) Data Frame Receive Pending Register (RXPR1, RXPR0)

The RXPR1 and RXPR0 are 16-bit read/conditionally-write registers. The RXPR is a register that contains the received Data Frames pending flags associated with the configured Receive Mailboxes. When a CAN Data Frame is successfully stored in a receive mailbox the corresponding bit is set in the RXPR. The bit may be cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect. However, the bit may only be set if the mailbox is configured by its MBC (Mailbox Configuration) to receive Data Frames. When a RXPR bit is set, it also sets IRR1 (Data Frame Received Interrupt Flag) if its MBIMR (Mailbox Interrupt Mask Register) is not set, and the interrupt signal is generated if IMR1 is not set. Please note that these bits are only set by receiving Data Frames and not by receiving Remote frames.

- RXPR1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXPR1[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*

Note : * Only when writing a '1' to clear.

Bit 15 to 0 — Configurable receive mailbox locations corresponding to each mailbox position from 31 to 16 respectively.

Bit[15:0]: RXPR1	Description
0	[Clearing Condition] Writing '1' (Initial value)
1	Corresponding Mailbox received a CAN Data Frame [Setting Condition] Completion of Data Frame receive on corresponding mailbox

chanhvinh.com

- RXPR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXPR0[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*

Note: * Only when writing a '1' to clear.

Bit 15 to 0 — Configurable receive mailbox locations corresponding to each mailbox position from 15 to 0 respectively.

Bit[15:0]: RXPR0	Description
0	[Clearing Condition] Writing '1' (Initial value)
1	Corresponding Mailbox received a CAN Data Frame [Setting Condition] Completion of Data Frame receive on corresponding mailbox

(6) Remote Frame Receive Pending Register (RFPR1, RFPR0)

The RFPR1 and RFPR0 are 16-bit read/conditionally-write registers. The RFPR is a register that contains the received Remote Frame pending flags associated with the configured Receive Mailboxes. When a CAN Remote Frame is successfully stored in a receive mailbox the corresponding bit is set in the RFPR. The bit may be cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect. In effect there is a bit position for all mailboxes. However, the bit may only be set if the mailbox is configured by its MBC (Mailbox Configuration) to receive Remote Frames. When a RFPR bit is set, it also sets IRR2 (Remote Frame Receive Interrupt Flag) if its MBIMR (Mailbox Interrupt Mask Register) is not set, and the interrupt signal is generated if IMR2 is not set. Please note that these bits are only set by receiving Remote Frames and not by receiving Data frames.

- RFPR1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFPR1[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*

Note: * Only when writing a '1' to clear.

chanhvinh.com

Bit 15 to 0 — Remote Request pending flags for mailboxes 31 to 16 respectively.

Bit[15:0]: RFPR1	Description
0	[Clearing Condition] Writing '1' (Initial value)
1	Corresponding Mailbox received Remote Frame [Setting Condition] Completion of remote frame receive in corresponding mailbox

- RFPR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFPR0[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*

Note: * Only when writing a '1' to clear.

Bit 15 to 0 — Remote Request pending flags for mailboxes 15 to 0 respectively.

Bit[15:0]: RFPR0	Description
0	[Clearing Condition] Writing '1' (Initial value)
1	Corresponding Mailbox received Remote Frame [Setting Condition] Completion of remote frame receive in corresponding mailbox

(7) Mailbox Interrupt Mask Register (MBIMR)

The MBIMR1 and MBIMR0 are 16-bit read/write registers. The MBIMR only prevents the setting of IRR related to the Mailbox activities, that are IRR[1] – Data Frame Received Interrupt, IRR[2] – Remote Frame Receive Interrupt, IRR[8] – Mailbox Empty Interrupt, and IRR[9] – Message OverRun/OverWrite Interrupt. If a mailbox is configured as receive, a mask at the corresponding bit position prevents the generation of a receive interrupt (IRR[1] and IRR[2] and IRR[9]) but does not prevent the setting of the corresponding bit in the RXPR or RFPR or UMSR. Similarly when a mailbox has been configured for transmission, a mask prevents the generation of an Interrupt signal and setting of an Mailbox Empty Interrupt due to successful transmission or abortion of transmission (IRR[8]), however, it does not prevent this module from clearing the corresponding TXPR/TXCR bit + setting the TXACK bit for successful transmission, and it does not prevent this module from clearing the corresponding TXPR/TXCR bit + setting the ABACK bit for abortion of the transmission.

chanhvinh.com

A mask is set by writing a '1' to the corresponding bit position for the mailbox activity to be masked. At reset all mailbox interrupts are masked.

- MBIMR1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIMR1[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

Bit 15 to 0 — Enable or disable interrupt requests from individual Mailbox-31 to Mailbox-16 respectively.

Bit[15:0]: MBIMR1	Description
0	Interrupt Request from IRR1/IRR2/IRR8/IRR9 enabled
1	Interrupt Request from IRR1/IRR2/IRR8/IRR9 disabled (initial value)

- MBIMR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIMR0[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

Bit 15 to 0 — Enable or disable interrupt requests from individual Mailbox-15 to Mailbox-0 respectively.

Bit[15:0]: MBIMR0	Description
0	Interrupt Request from IRR1/IRR2/IRR8/IRR9 enabled
1	Interrupt Request from IRR1/IRR2/IRR8/IRR9 disabled (initial value)

chanhvinh.com

(8) Unread Message Status Register (UMSR)

This register is a 32-bit read/conditionally write register and it records the mailboxes whose contents have not been accessed by the CPU prior to a new message being received. If the CPU has not cleared the corresponding bit in the RXPR or RFPR when a new message for that mailbox is received, the corresponding UMSR bit is set to '1'. This bit may be cleared by writing a '1' to the corresponding bit location in the UMSR. Writing a '0' has no effect.

If a mailbox is configured as transmit box, the corresponding UMSR will not be set.

- UMSR1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UMSR1[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*

Note: * Only when writing a '1' to clear.

Bit 15 to 0 — Indicate that an unread received message has been overwritten or overrun condition has occurred for Mailboxes 31 to 16.

Bit[15:0]: UMSR1	Description
0	[Clearing Condition] Writing '1' (initial value)
1	Unread received message is overwritten by a new message or overrun condition [Setting Condition] When a new message is received before RXPR or RFPR is cleared

- UMSR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UMSR0[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*

Note: * Only when writing a '1' to clear.

Bit 15 to 0 — Indicate that an unread received message has been overwritten or overrun condition has occurred for Mailboxes 15 to 0.

chanhvinh.com

Bit[15:0]: UMSR0	Description
0	[Clearing Condition] Writing '1' (initial value)
1	Unread received message is overwritten by a new message or overrun condition [Setting Condition] When a new message is received before RXPR or RFPR is cleared

20.3.5 Timer Registers

The Timer is 16 bits and supports several source clocks. A pre-scale counter can be used to reduce the speed of the clock. It also supports three Compare Match Registers (TCMR2, TCMR1, TCMR0). The address map is as follows.

Important: These registers can only be accessed in Word size (16-bit).

Description	Address	Name	Access Size (bits)
Timer Trigger Control Register 0	080	TTCR0	Word (16)
Cycle Maximum/Tx-Enable Window Register	084	CMAX_TEW	Word (16)
Reference Trigger Offset Register	086	RFTROFF	Word (16)
Timer Status Register	088	TSR	Word (16)
Cycle Counter Register	08A	CCR	Word (16)
Timer Counter Register	08C	TCNTR	Word (16)
Cycle Time Register	090	CYCTR	Word (16)
Reference Mark Register	094	RFMK	Word (16)
Timer Compare Match Register 0	098	TCMR0	Word (16)
Timer Compare Match Register 1	09C	TCMR1	Word (16)
Timer Compare Match Register 2	0A0	TCMR2	Word (16)
Tx-Trigger Time Selection Register	0A4	TTTSEL	Word (16)

Figure 20.12 Timer Registers

chanhvinh.com

(1) Time Trigger Control Register0 (TTCR0)

The Time Trigger Control Register0 is a 16-bit read/write register and provides functions to control the operation of the Timer. When operating in Time Trigger Mode, please refer to section 20.4.3 (1), Time Triggered Transmission.

- TTCR0 (Address = H'080)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCR15	TCR14	TCR13	TCR12	TCR11	TCR10	-	-	-	TCR6	TPSC5	TPSC4	TPSC3	TPSC2	TPSC1	TPSC0
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	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 15 — Enable Timer: When this bit is set, the timer TCNTR is running. When this bit is cleared, TCNTR and CCR are cleared.

Bit15: TTCR0 15	Description
0	Timer and CCR are cleared and disabled (initial value)
1	Timer is running

Bit 14 — TimeStamp value: Specifies if the Timestamp for transmission and reception in Mailboxes 15 to 1 must contain the Cycle Time (CYCTR) or the concatenation of CCR[5:0] + CYCTR[15:6]. This feature is very useful for time triggered transmission to monitor Rx_Trigger.

This register does not affect the TimeStamp for Mailboxes 30 and 31.

Bit14: TTCR0 14	Description
0	CYCTR[15:0] is used for the TimeStamp in Mailboxes 15 to 1 (initial value)
1	CCR[5:0] + CYCTR[15:6] is used for the TimeStamp in Mailboxes 15 to 1

Bit 13 — Cancellation by TCMR2: The messages in the transmission queue are cancelled by setting TXCR, when both this bit and bit12 are set and compare match occurs when this module is not in the Halt status, causing the setting of all TXCR bits with the corresponding TXPR bits set.

Bit13: TTCR0 13	Description
0	Cancellation by TCMR2 compare match is disabled (initial value)
1	Cancellation by TCMR2 compare match is enabled

chanhvinh.com

Bit 12 — TCMR2 compare match enable: When this bit is set, IRR11 is set by TCMR2 compare match.

Bit12 TTCR0 12	Description
0	IRR11 isn't set by TCMR2 compare match (initial value)
1	IRR11 is set by TCMR2 compare match

Bit 11 — TCMR1 compare match enable: When this bit is set, IRR15 is set by TCMR1 compare match.

Bit11 TTCR0 11	Description
0	IRR15 isn't set by TCMR1 compare match (initial value)
1	IRR15 is set by TCMR1 compare match

Bit 10 — TCMR0 compare match enable: When this bit is set, IRR14 is set by TCMR0 compare match.

Bit10 TTCR0 10	Description
0	IRR14 isn't set by TCMR0 compare match (initial value)
1	IRR14 is set by TCMR0 compare match

Bits 9 to 7: Reserved. The written value should always be '0' and the returned value is '0'.

Bit 6 — Timer Clear-Set Control by TCMR0: Specifies if the Timer is to be cleared and set to H'0000 when the TCMR0 matches to the TCNTR. Please note that the TCMR0 is also capable to generate an interrupt signal to the CPU via IRR14.

Note: If this module is working in TTCAN mode (CMAX isn't 3'b111), TTCR0 bit6 has to be '0' to avoid clearing Local Time.

Bit6: TTCR0 6	Description
0	Timer is not cleared by the TCMR0 (initial value)
1	Timer is cleared by the TCMR0

chanhvinh.com

Bit5 to 0 — Timer Prescaler (TPSC[5:0]): This control field allows the timer source clock (4*[this module system clock]) to be divided before it is used for the timer. This function is available only in event-trigger mode. In time trigger mode (CMAX is not 3'b111), one nominal Bit Timing (= one bit length of CAN bus) is automatically chosen as source clock of TCNTR.

The following relationship exists between source clock period and the timer period.

Bit[5:0]: TPSC[5:0]	Description
0 0 0 0 0 0	1 X Source Clock (initial value)
0 0 0 0 0 1	2 X Source Clock
0 0 0 0 1 0	3 X Source Clock
0 0 0 0 1 1	4 X Source Clock
0 0 0 1 0 0	5 X Source Clock
. . . . .	. . . . .
. . . . .	. . . . .
1 1 1 1 1 1	64 X Source Clock

(2) Cycle Maximum/Tx-Enable Window Register (CMAX_TEW)

This register is a 16-bit read/write register. CMAX specifies the maximum value for the cycle counter (CCR) for TT Transmissions to set the number of basic cycles in the matrix system. When the Cycle Counter reaches the maximum value (CCR = CMAX), after a full basic cycle, it is cleared to zero and an interrupt is generated on IRR.10.

TEW specifies the width of Tx-Enable window.

- CMAX_TEW (Address = H'084)

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

Bits 15 to 11: Reserved. The written value should always be '0' and the returned value is '0'.

chanhvinh.com

Bit 10 to 8 — Cycle Count Maximum (CMAX): Indicates the maximum number of CCR. The number of basic cycles available in the matrix cycle for Timer Triggered transmission is (Cycle Count Maximum + 1).

Unless CMAX = 3'b111, this module is in time-trigger mode and time trigger function is available. If CMAX = 3'b111, this module is in event-trigger mode.

Bit[10:8]: CMAX[2:0]	Description
0 0 0	Cycle Count Maximum = 0
0 0 1	Cycle Count Maximum = 1
0 1 0	Cycle Count Maximum = 3
0 1 1	Cycle Count Maximum = 7
1 0 0	Cycle Count Maximum = 15
1 0 1	Cycle Count Maximum = 31
1 1 0	Cycle Count Maximum = 63
1 1 1	CCR is cleared and this module is in event-trigger mode. (initial value)

Important: Please set CMAX = 3'b111 when event-trigger mode is used.

Bits 7 to 4: Reserved. The written value should always be '0' and the returned value is ‘0’.

Bit 3 to 0 — Tx-Enable Window (TEW): Indicates the width of Tx-Enable Window. TEW = H'00 shows the width is one nominal Bit Timing. All values from 0 to 15 are allowed to be set.

Bit[3:0]: TEW[3:0]	Description
0 0 0 0	The width of Tx-Enable Window = 1 (initial value)
0 0 0 1	The width of Tx-Enable Window = 2
0 0 1 0	The width of Tx-Enable Window = 3
0 0 1 1	The width of Tx-Enable Window = 4
. . . .	. . . . .
. . . .	. . . . .
1 1 1 1	The width of Tx-Enable Window = 16

Note: The CAN core always needs a time between 1 to 2 bit timing to initiate transmission. The above values are not considering this accuracy.

chanhvinh.com

(3) Reference Trigger Offset Register (RFTROFF)

This is a 8-bit read/write register that affects Tx-Trigger Time (TTT) of Mailbox-30. The TTT of Mailbox-30 is compared with CYCTR after RFTROFF extended with sign is added to the TTT. However, the value of TTT is not modified. The offset value doesn't affect others except Mailbox-30.

- RFTROFF (Address = H'086)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFTROFF[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	R	R	R	R	R	R	R

Bit 15 to 8 — Indicate the value of Reference Trigger Offset.

Bits 7 to 0: Reserved. The written value should always be '0' and the returned value is '0'.

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Description
0	0	0	0	0	0	0	0	Ref_trigger_offset = +0 (initial value)
0	0	0	0	0	0	0	1	Ref_trigger_offset = +1
0	0	0	0	0	0	1	0	Ref_trigger_offset = +2
.	.	.	.	.	.	.	.	
0	1	1	1	1	1	1	1	Ref_trigger_offset = +127
.	.	.	.	.	.	.	.	
1	1	1	1	1	1	1	1	Ref_trigger_offset = –1
1	1	1	1	1	1	1	0	Ref_trigger_offset = –2
.	.	.	.	.	.	.	.	
1	0	0	0	0	0	0	1	Ref_trigger_offset = –127
1	0	0	0	0	0	0	0	Prohibited

chanhvinh.com

(4) Timer Status Register (TSR)

This register is a 16-bit read-only register, and allows the CPU to monitor the Timer Compare Match status and the Timer Overrun Status.

- TSR (Address = H'088)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	TSR4	TSR3	TSR2	TSR1	TSR0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bits 15 to 5: Reserved. The written value should always be '0' and the returned value is '0'.

Bit 4 to 0 — Timer Status (TSR[4:0]): This read-only field allows the CPU to monitor the status of the Cycle Counter, the Timer and the Compare Match registers. Writing to this field has no effect.

Bit 4 — Start of New System Matrix (TSR4): Indicates that a new system matrix is starting. When CCR = 0, this bit is set at the successful completion of reception/transmission of time reference message.

Bit4: TSR4	Description
0	A new system matrix is not starting (initial value) [Clearing condition] Writing '1' to IRR10 (Cycle Counter Overflow Interrupt)
1	Cycle counter reached zero [Setting condition] When the Cycle Counter value changes from the maximum value (CMAX) to H'0. Reception/transmission of time reference message is successfully completed when CMAX!= 3'b111 and CCR = 0

chanhvinh.com

Bit 3 — Timer Compare Match Flag 2 (TSR3): Indicates that a Compare-Match condition occurred to the Timer Compare Match Register 2 (TCMR2). When the value set in the TCMR2 matches to Cycle Time Register (TCMR2 = CYCTR), this bit is set if TTCR0 bit12 = 1. Please note that this bit is read-only and is cleared when IRR11 (Timer Compare Match Interrupt 2) is cleared.

Bit3: TSR3	Description
0	Timer Compare Match has not occurred to the TCMR2 (Initial value) [Clearing condition] Writing '1' to IRR11 (Timer Compare Match Interrupt 1)
1	Timer Compare Match has occurred to the TCMR2 [Setting condition] TCMR2 matches to Cycle Time (TCMR2 = CYCTR), if TTCR0 bit12 = 1.

Bit 2 — Timer Compare Match Flag 1 (TSR2): Indicates that a Compare-Match condition occurred to the Timer Compare Match Register 1 (TCMR1). When the value set in the TCMR1 matches to Cycle Time Register (TCMR1 = CYCTR), this bit is set if TTCR0 bit11 = 1. Please note that this bit is read-only and is cleared when IRR15 (Timer Compare Match Interrupt 1) is cleared.

Bit2: TSR2	Description
0	Timer Compare Match has not occurred to the TCMR1 (Initial value) [Clearing condition] Writing '1' to IRR15 (Timer Compare Match Interrupt 1)
1	Timer Compare Match has occurred to the TCMR1 [Setting condition] TCMR1 matches to Cycle Time (TCMR1 = CYCTR), if TTCR0 bit11 = 1.

Bit 1 — Timer Compare Match Flag 0 (TSR1): Indicates that a Compare-Match condition occurred to the Compare Match Register 0 (TCMR0). When the value set in the TCMR0 matches to the Timer value (TCMR0 = TCNTR), this bit is set if TTCR0 bit10 = 1. Please note that this bit is read-only and is cleared when IRR14 (Timer Compare Match Interrupt 0) is cleared.

Bit1: TSR1	Description
0	Compare Match has not occurred to the TCMR0 (Initial value) [Clearing condition] Writing '1' to IRR14 (Timer Compare Match Interrupt 0)
1	Compare Match has occurred to the TCMR0 [Setting condition] TCMR0 matches to the Timer value (TCMR0 = TCNTR)

chanhvinh.com

Bit 0 — Timer Overrun/Next_is_Gap Reception/Message Error (TSR0): This flag is assigned to three different functions. It indicates that the Timer has overrun when working in event-trigger mode, time reference message with Next_is_Gap set has been received in time-trigger mode, and error detected on the CAN bus has occurred in test mode, respectively. Test mode has higher priority with respect to the other settings.

Bit0: TSR0	Description
0	Timer (TCNTR) has not overrun in event-trigger mode (Initial value) Time reference message with Next_is_Gap has not been received in time-trigger mode message error has not occurred in test mode. [Clearing condition] Writing '1' to IRR13
1	[Setting condition] Timer (TCNTR) has overrun and changed from H'FFFF to H'0000 in event-trigger mode.time reference message with Next_is_Gap has been received in time-trigger mode message error has occurred in test mode

(5) Cycle Counter Register (CCR)

This register is a 6-bit read/write register. Its purpose is to store the number of the basic cycle for Time -Triggered Transmissions. Its value is updated in different fashions depending if this module is programmed to work as a potential time master or as a time slave. If this module is working as (potential) time master, CCR is:

— Incremented by one every time the cycle time (CYCTR) matches to Tx-Trigger Time of Mailbox-30 or

— Overwritten with the value contained in MSG_DATA_0[5:0] of Mailbox 31 when a valid reference message is received.

If this module is working as a time slave, CCR is only overwritten with the value of MSG_DATA_0[5:0] of Mailbox 31 when a valid reference message is received.

If CMAX = 3'111, CCR is always H'0000.

- CCR (Address = H'08A)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	CCR[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

chanhvinh.com

Bits 15 to 6: Reserved. The written value should always be '0' and the returned value is '0'.

Bit 5 to 0 — Cycle Counter Register (CCR): Indicates the number of the current Base Cycle of the matrix cycle for Timer Triggered transmission.

(6) Timer Counter Register (TCNTR)

This is a 16-bit read/write register that allows the CPU to monitor and modify the value of the Free Running Timer Counter. When the Timer meets TCMR0 (Timer Compare Match Register 0) + TTCR0 [6] is set to '1', the TCNTR is cleared to H'0000 and starts running again. In Time-Trigger mode, this timer can be used as Local Time and TTCR0[6] has to be cleared to work as a free running timer.

Notes: 1. It is possible to write into this register only when it is enabled by the bit 15 in TTCR0. If TTCR0 bit15 = 0, TCNTR is always H'0000.
2. There could be a delay of a few clock cycles between the enabling of the timer and the moment where TCNTR starts incrementing. This is caused by the internal logic used for the pre-scaler.

- TCNTR (Address = H'08C)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCNTR[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*

Note: * The register can be written only when enabled in TTCR0[15]. Write operation is not allowed in Time Trigger mode (i.e. CMAX is not 3'b111).

Bit 15 to 0 — Indicate the value of the Free Running Timer.

(7) Cycle Time register (CYCTR)

This register is a 16-bit read-only register. This register shows Cycle Time = Local Time (TCNTR) - Reference_Mark (RFMK). In ET mode this register is the exact copy of TCNTR as RFMK is always fixed to zero.

- CYCTR (Address = H'090)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CYCTR[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

chanhvinh.com

(8) Reference Mark Register (RFMK)

This register is a 16-bit read-only register. The purpose of this register is to capture Local Time (TCNTR) at SOF of the reference message when the message is received or transmitted successfully. In ET mode this register is not used and it is always cleared to zero.

- RFMK (Address = H'094)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFMK[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 15 to 0 — Reference Mark Register (RFMK): Indicates the value of TCNTR at SOF of time reference message.

(9) Timer Compare Match Registers (TCMR0, TCMR1, TCMR2)

These three registers are 16-bit read/write registers and are capable of generating interrupt signals, clearing-setting the Timer value (only supported by TCMR0) or clear the transmission messages in the queue (only supported by TCMR2). TCMR0 is compared with TCNTR, however, TCMR1 and TCMR2 are compared with CYCTR.

The value used for the compare can be configured independently for each register. In order to set flags, TTCR0 bit 12-10 needs to be set.

In Time-Trigger mode, TTCR0 bit6 has to be cleared by software to prevent TCNTR from being cleared.

TMCR0 is for Init_Watch_Trigger, and TCMR2 is for Watch_Trigger.

Interrupt:

The interrupts are flagged by the Bit11, Bit15 and 14 in the IRR accordingly when a Compare Match occurs, and setting these bits can be enabled by Bit12, Bit11, Bit10 in TTCR0. The generation of interrupt signals itself can be prevented by the Bit11, Bit15 and Bit14 in the IMR. When a Compare Match occurs and the IRR11 (or IRR15 or IRR14) is set, the Bit3 or Bit2 or Bit1 in the TSR (Timer Status Register) is also set. Clearing the IRR bit also clears the corresponding bit of TSR.

chanhvinh.com

Timer Clear-Set:

The Timer value can only be cleared when a Compare Match occurs if it is enabled by the Bit6 in the TTCR0. TCMR1 and TCMR2 do not have this function.

Cancellation of the messages in the transmission queue:

The messages in the transmission queue can only be cleared by the TCMR2 through setting TXCR when a Compare Match occurs while this module is not in the halt status. TCMR1 and TCMR0 do not have this function.

- TCMR0 (Address = H'098)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCMR0[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

Bit 15 to 0 — Timer Compare Match Register (TCMR0): Indicates the value of TCNTR when compare match occurs.

- TCMR1 (Address = H'09C)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCMR1[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

Bit 15 to 0 — Timer Compare Match Register (TCMR1): Indicates the value of CYCTR when compare match occurs.

- TCMR2 (Address = H'0A0)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCMR2[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

Bit 15 to 0 — Timer Compare Match Register (TCMR2): Indicates the value of CYCTR when compare match occurs.

chanhvinh.com

(10) Tx-Trigger Time Selection Register (TTTSEL)

This register is a 16-bit read/write register and specifies the Tx-Trigger Time waiting for compare match with Cycle Time. Only one bit is allowed to be set. Please don't set more bits than one, or clear all bits.

This register may only be modified during configuration mode. The modification algorithm is shown in figure 20.13.

Please note that this register is only indented for test and diagnosis. When not in test mode, this register must not be written to and the returned value is not guaranteed.

- TTTSEL (Address = H'0A4)

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

Note: Only one bit is allowed to be set.

Bit 15: Reserved. The written value should always be '0' and the returned value is '0'.

Bit 14 to 8 — Specifies the Tx-Trigger Time waiting for compare match with CYCTR The bit 14 to 8 corresponds to Mailbox-30 to 24, respectively.

Bits 7 to 0: Reserved. The written value should always be '0' and the returned value is '0'.

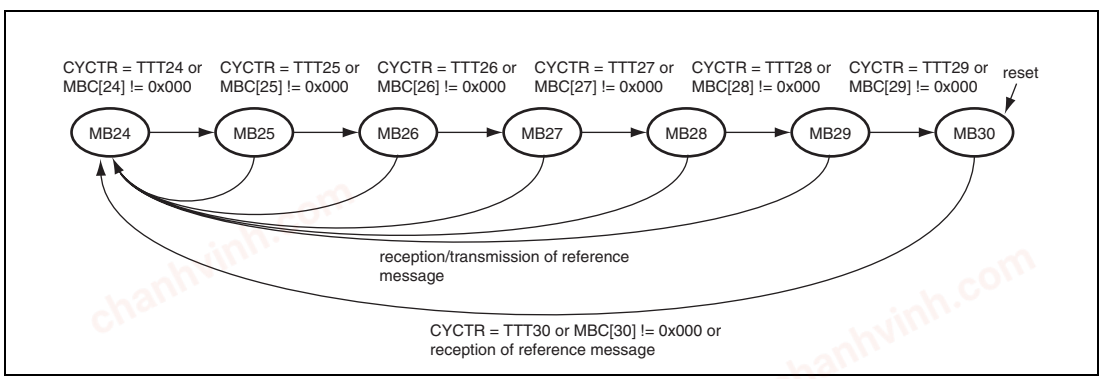


Figure 20.13 TTTSEL modification algorithm

chanhvinh.com

20.4 Application Note

20.4.1 Test Mode Settings

This module has various test modes. The register TST[2:0] (MCR[10:8]) is used to select this module test mode. The default (initialised) settings allow this module to operate in Normal mode. The following table is examples for test modes.

Test Mode can be selected only while in configuration mode. The user must then exit the configuration mode (ensuring BCR0/BCR1 is set) in order to run the selected test mode.

Bit10: TST2	Bit9: TST1	Bit8: TST0	Description
0	0	0	Normal Mode (initial value)
0	0	1	Listen-Only Mode (Receive-Only Mode)
0	1	0	Self Test Mode 1 (External)
0	1	1	Self Test Mode 2 (Internal)
1	0	0	Write Error Counter
1	0	1	Error Passive Mode
1	1	0	Setting prohibited
1	1	1	Setting prohibited

Normal Mode: This module operates in the normal mode.

Listen-Only Mode: ISO-11898 requires this mode for baud rate detection. The Error Counters are cleared and disabled so that the TEC/REC does not increase the values, and the CTxn (n = 0, 1) Output is disabled so that this module does not generate error frames or acknowledgment bits. IRR13 is set when a message error occurs.

Self Test Mode 1: This module generates its own Acknowledge bit, and can store its own messages into a reception mailbox (if required). The CRxn/CTxn (n = 0, 1) pins must be connected to the CAN bus.

chanhvinh.com

Self Test Mode 2: This module generates its own Acknowledge bit, and can store its own messages into a reception mailbox (if required). The CRxn/CTxn (n = 0, 1) pins do not need to be connected to the CAN bus or any external devices, as the internal CTxn (n = 0, 1) is looped back to the internal CRxn (n = 0, 1). CTxn (n = 0, 1) pin outputs only recessive bits and CRxn (n = 0, 1) pin is disabled.

Write Error Counter: TEC/REC can be written in this mode. This module can be forced to become an Error Passive mode by writing a value greater than 127 into the Error Counters. The value written into TEC is used to write into REC, so only the same value can be set to these registers. Similarly, this module can be forced to become an Error Warning by writing a value greater than 95 into them.

This module needs to be in Halt Mode when writing into TEC/REC (MCR1 must be "1" when writing to the Error Counter). Furthermore this test mode needs to be exited prior to leaving Halt mode.

Error Passive Mode: This module can be forced to enter Error Passive mode.
Note: The REC will not be modified by implementing this Mode. However, once running in Error Passive Mode, the REC will increase normally should errors be received. In this Mode, this module will enter BusOff if TEC reaches 256 (Dec). However when this mode is used this module will not be able to become Error Active. Consequently, at the end of the Bus Off recovery sequence, this module will move to Error Passive and not to Error Active.

When message error occurs, IRR13 is set in all test modes.

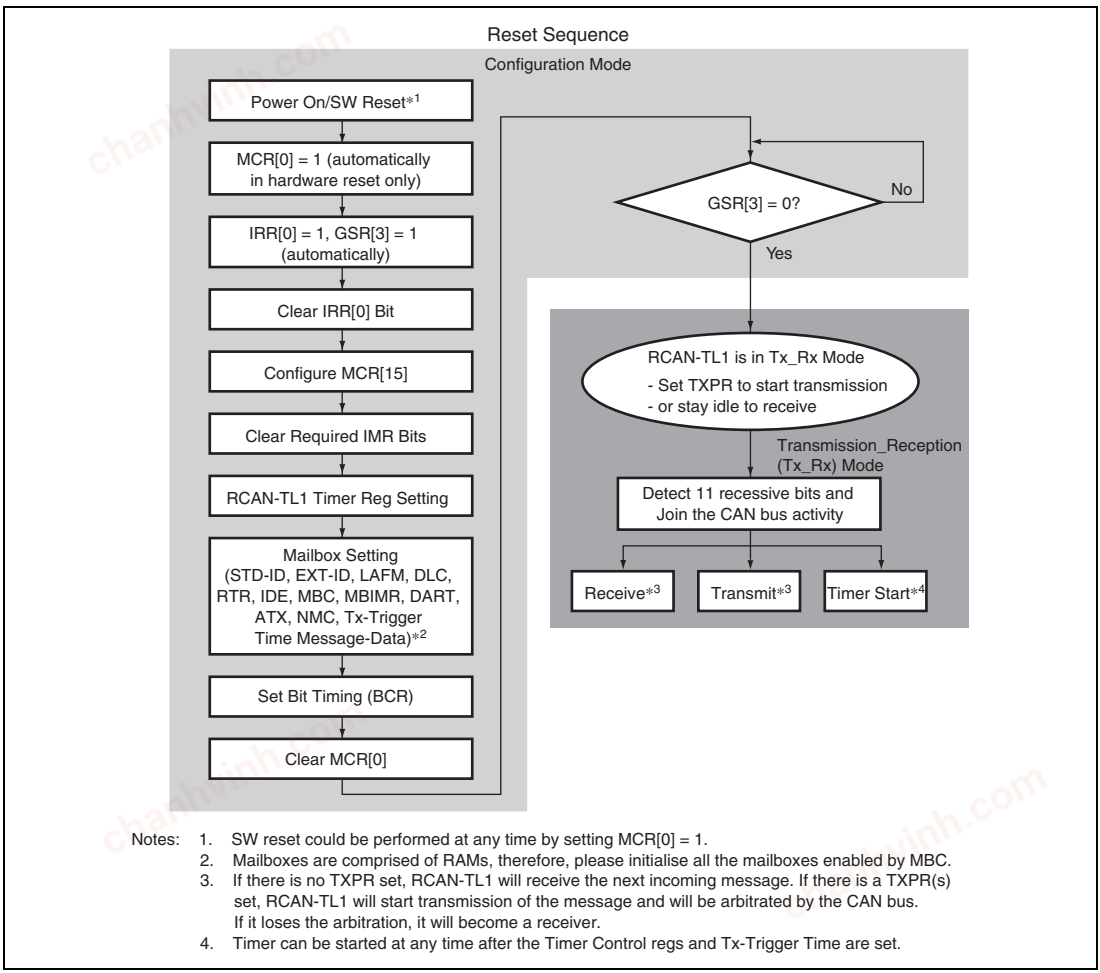
chanhvinh.com

20.4.2 Configuration of This Module

This module is considered in configuration mode or after a H/W (Power On Reset)/S/W (MCR[0]) reset or when in Halt mode. In both conditions this module cannot join the CAN Bus activity and configuration changes have no impact on the traffic on the CAN Bus.

- After a Reset request

The following sequence must be implemented to configure this module after (S/W or H/W) reset. After reset, all the registers are initialised, therefore, this module needs to be configured before joining the CAN bus activity. Please read the notes carefully.



Notes:
1. SW reset could be performed at any time by setting MCR[0] = 1.
2. Mailboxes are comprised of RAMs, therefore, please initialise all the mailboxes enabled by MBC.
3. If there is no TXPR set, RCAN-TL1 will receive the next incoming message. If there is a TXPR(s) set, RCAN-TL1 will start transmission of the message and will be arbitrated by the CAN bus. If it loses the arbitration, it will become a receiver.
4. Timer can be started at any time after the Timer Control regs and Tx-Trigger Time are set.

Figure 20.14 Reset Sequence

chanhvinh.com

- Halt mode

When this module is in Halt mode, it cannot take part to the CAN bus activity. Consequently the user can modify all the requested registers without influencing existing traffic on the CAN Bus. It is important for this that the user waits for this module to be in halt mode before to modify the requested registers - note that the transition to Halt Mode is not always immediate (transition will occurs when the CAN Bus is idle or in intermission). After this module transit to Halt Mode, GSR4 is set.

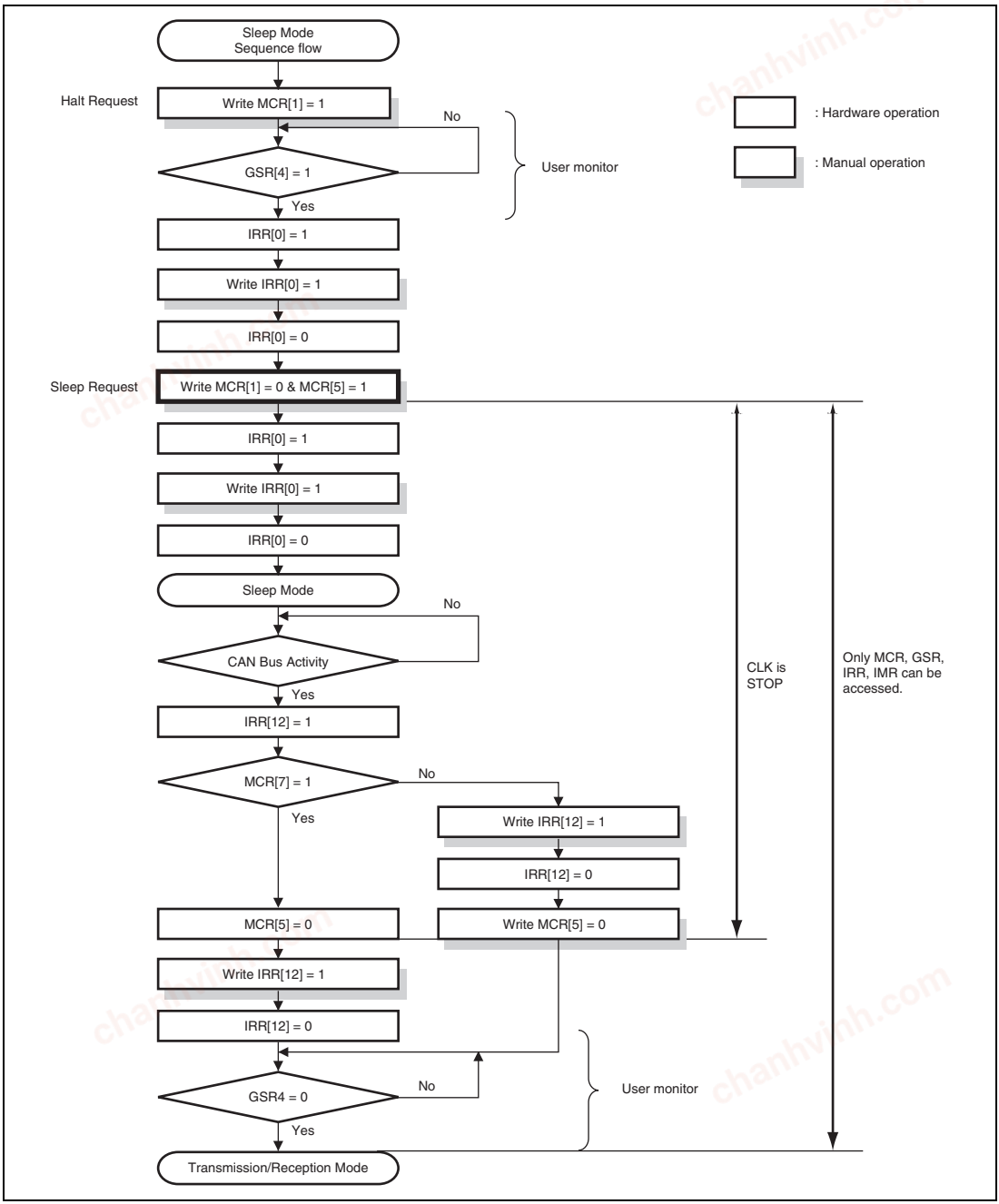
Once the configuration is completed the Halt request needs to be released. This module will join CAN Bus activity after the detection of 11 recessive bits on the CAN Bus.

- Sleep mode

When this module is in sleep mode the clock for the main blocks of the IP is stopped in order to reduce power consumption. Only the following user registers are clocked and can be accessed: MCR, GSR, IRR and IMR. Interrupt related to transmission (TXACK and ABACK) and reception (RXPR and RFPR) cannot be cleared when in sleep mode (as TXACK, ABACK, RXPR and RFPR are not accessible) and must to be cleared beforehand.

chanhvinh.com

The following diagram shows the flow to follow to move this module into sleep mode.



chanhvinh.com

Figure 20.15 shows allowed state transitions.

— Please don't set MCR5 (Sleep Mode) without entering Halt Mode.

— After MCR1 is set, please don't clear it before GSR4 is set and this module enters Halt Mode.

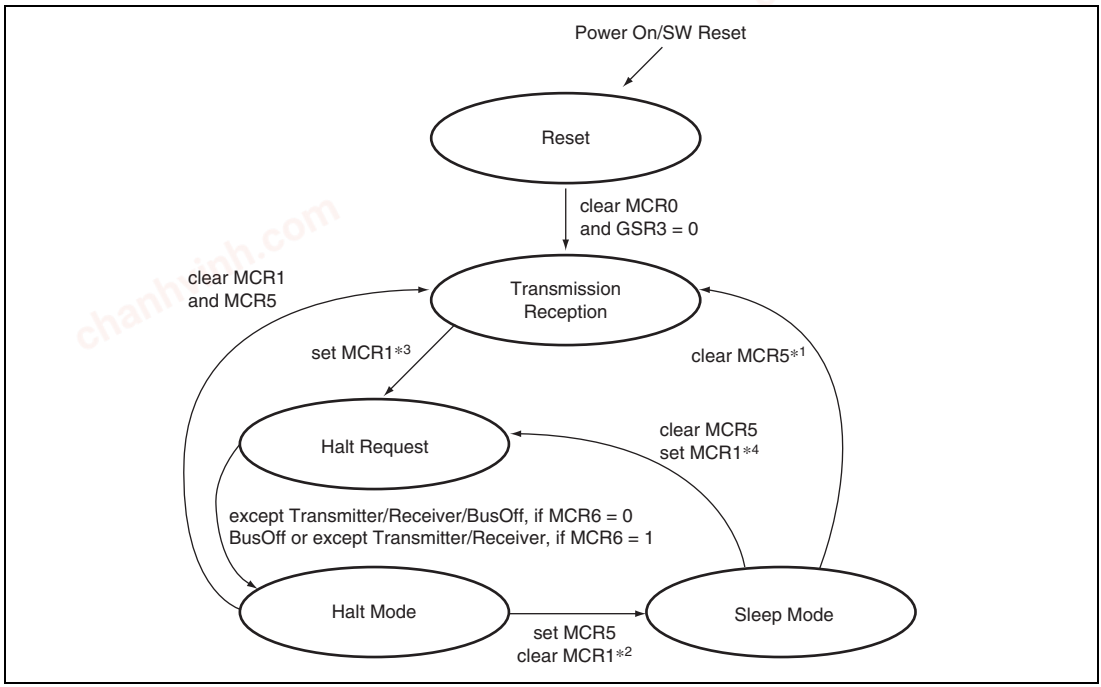


Figure 20.15 Halt Mode/Sleep Mode

Notes: 1. MCR5 can be cleared by automatically by detecting a dominant bit on the CAN Bus if MCR7 is set or by writing '0'.

2. MCR1 is cleared in SW. Clearing MCR1 and setting MCR5 have to be carried out by the same instruction.

3. MCR1 must not be cleared in SW, before GSR4 is set. MCR1 can be set automatically in HW when this module moves to Bus Off and MCR14 and MCR6 are both set.

4. When MCR5 is cleared and MCR1 is set at the same time, this module moves to Halt Request. Right after that, it moves to Halt Mode with no reception/transmission.

The following table shows conditions to access registers.

chanhvinh.com

	Registers									
Status Mode	MCR GSR	IRR IMR	BCR	MBIMR timer TT_register	Flag_ register	Mailbox (ctrl0, LAFM)		Mailbox (data)	Mailbox (ctrl1)		Mailbox Trigger Time TT control
Reset	yes	yes	yes	yes	yes	yes		yes	yes		yes
Transmission Reception Halt Request	yes	yes	no*[1]	yes	yes	no*[1]	yes*[2]	yes*[2]	no*[1]	yes*[2]	yes*[2]
Halt	yes	yes	no*[1]	yes	yes	yes		yes	yes		yes
Sleep	yes	yes	no	no	no	no		no	no		no

Notes: 1. No hardware protection.
2. When TXPR is not set.

20.4.3 Message Transmission Sequence

- Message Transmission Request

The following sequence is an example to transmit a CAN frame onto the bus. As described in the previous register section, please note that IRR8 is set when one of the TXACK or ABACK bits is set, meaning one of the Mailboxes has completed its transmission or transmission abortion and is now ready to be updated for the next transmission, whereas, the GSR2 means that there is currently no transmission request made (No TXPR flags set).

chanhvinh.com

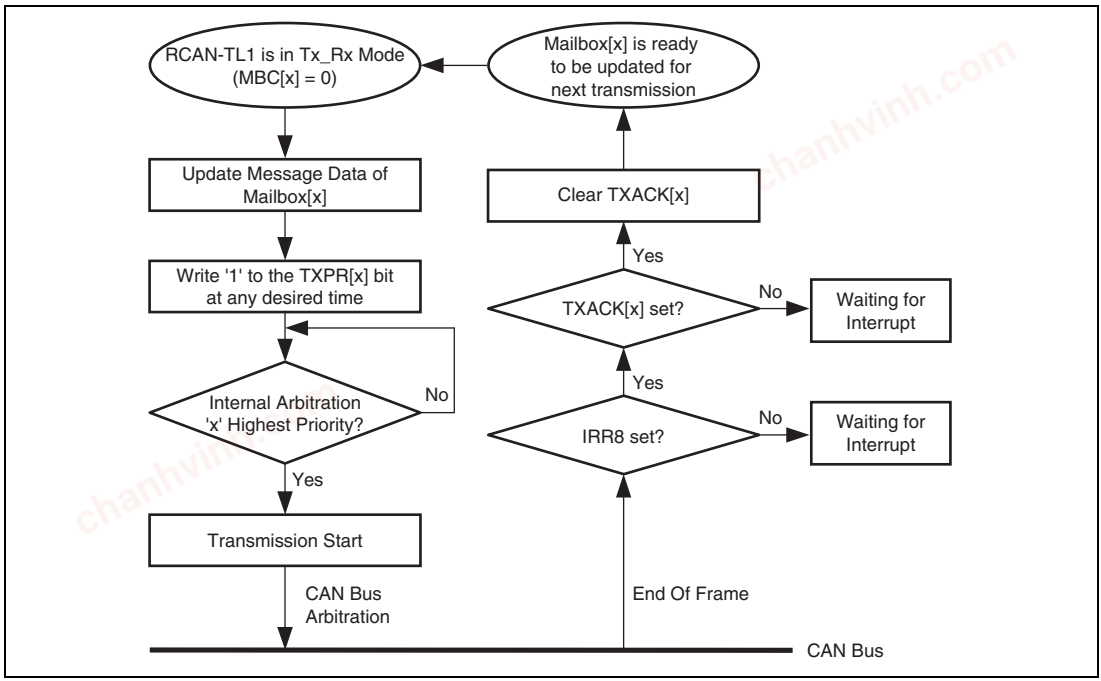


Figure 20.16 Transmission request

- Internal Arbitration for transmission

The following diagram explains how this module manages to schedule transmission-requested messages in the correct order based on the CAN identifier. 'Internal arbitration' picks up the highest priority message amongst transmit-requested messages.

chanhvinh.com

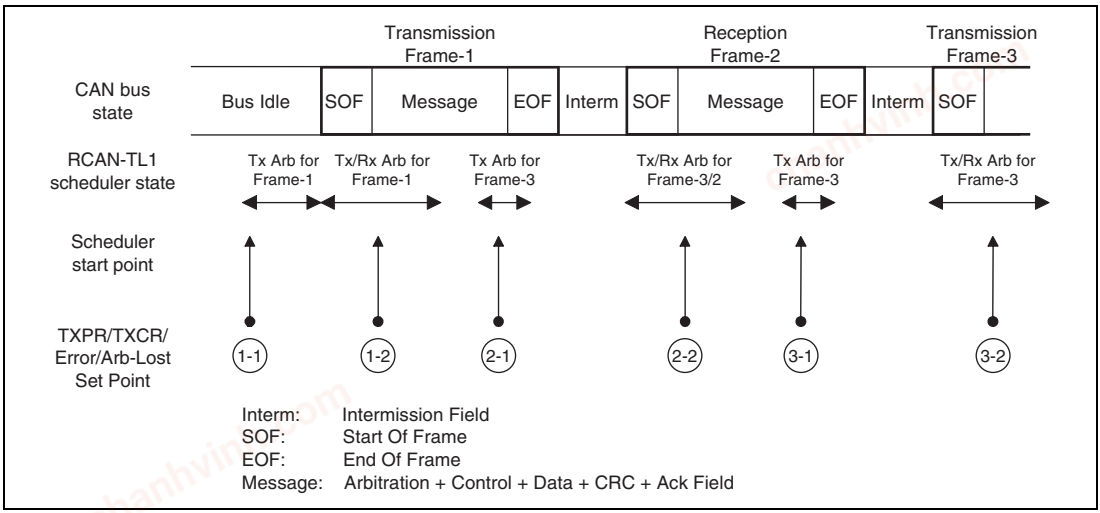


Figure 20.17 Internal Arbitration for transmission

This module has two state machines. One is for transmission, and the other is for reception.

1-1: When a TXPR bit(s) is set while the CAN bus is idle, the internal arbitration starts running immediately and the transmission is started.

1-2: Operations for both transmission and reception starts at SOF. Since there is no reception frame, this module becomes transmitter.

2-1: At crc delimiter, internal arbitration to search next message transmitted starts.

2-2: Operations for both transmission and reception starts at SOF. Because of a reception frame with higher priority, this module becomes receiver. Therefore, Reception is carried out instead of transmitting Frame-3.

3-1: At crc delimiter, internal arbitration to search next message transmitted starts.

3-2: Operations for both transmission and reception starts at SOF. Since a transmission frame has higher priority than reception one, this module becomes transmitter.

Internal arbitration for the next transmission is also performed at the beginning of each error delimiter in case of an error is detected on the CAN Bus. It is also performed at the beginning of error delimiters following overload frame.

As the arbitration for transmission is performed at CRC delimiter, in case a remote frame request is received into a Mailbox with ATX = 1 the answer can join the arbitration for transmission only at the following Bus Idle, CRC delimiter or Error Delimiter.

Depending on the status of the CAN bus, following the assertion of the TXCR, the corresponding Message abortion can be handled with a delay of maximum 1 CAN Frame.

chanhvinh.com

(1) Time Triggered Transmission

This module offers a H/W support to perform communication in Time Trigger mode in line with the emerging ISO-11898-4 Level 1 Specification.

This section reports the basic procedures to use this mode.

- Setting Time Trigger Mode

In order to set up the time trigger mode the following settings need to be used.

— CMAX in CMAX_TEW must be programmed to a value different from 3'b111.
— Bit 15 in TTCR0 has to be set, to start TCNTR.
— Bit 6 in TTCR0 has to be cleared to prevent TCNTR from being cleared after a match.
— DART in Mailboxes used for time-triggered transmission cannot be used, since for Time Triggered Mailboxes, TXPR is not cleared to support periodic transmission.

- Roles of Registers

The user registers of this module can be used to handle the main functions requested by the TTCAN standard.

TCNTR	Local Time
RFMK	Ref_Mark
CYCTR	Cycle Time = TCNTR - RFMK
RFTROFF	Ref_Trigger_Offset for Mailbox-30
Mailbox-31	Mailbox dedicated to the reception of time reference message
Mailbox-30	Mailbox dedicated to the transmission of time reference message when working as a potential time master
Mailbox-29 to 24	Mailboxes supporting time-triggered transmission
Mailbox-23 to 16	Mailboxes supporting reception without timestamp (may also be implemented as Mailboxes supporting Event Triggered transmission)
Mailbox-15 to 0	Mailboxes supporting reception with timestamp timestamp (may also be implemented as Mailboxes supporting Event Triggered transmission)
Tx-Trigger Time	Time_Mark to specify when a message should be transmitted

chanhvinh.com

CMAX	Specifies the maximum number of basic cycles when working as potential time master
TEW	Specify the width of Tx_Enable
TCMR0	Init_Watch_Trigger (compare match with Local Time)
TCMR1	Compare match with Cycle Time to monitor users-specified events
TCMR2	Watch_Trigger (compare match with Cycle Time). This can be programmed to abort all pending transmissions
TTW	Specifies the attribute of a time window used for transmission
TTTSEL	Specifies the next Mailbox waiting for transmission

- Time Master/Time Slave

This module can be programmed to work as a potential time master of the network or as a time slave. The following table shows the settings and the operation automatically performed by this module in each mode.

chanhvinh.com

mode	requested setting	function
Time Slave	TXPR[30] = 0 & MBC[30]!= 3'b000 & CMAX!= 3'b111 & MBC[31] = 3'b011	TCNTR is sampled at each SOF detected on the CAN Bus and stored into an internal register. When a valid Time Reference Message is received into Mailbox-31 the value of TCNTR (stored at the SOF) is copied into Ref_Mark. CCR embedded in the received Reference Message is copied to CCR. If Next_is_Gap = 1, IRR13 is set.
(Potential) Time Master	TXPR[30] = 1 & MBC[30] = 3'b000 & DLC[30] > 0 & CMAX!= 3'b111 & MBC[31] = 3'b011	Two cases are covered: (1) When a valid Time Reference message is received into Mailbox-31 the value of TCNTR stored into an internal register at the SOF is copied into Ref_Mark. CCR embedded in the received Reference Message is copied to CCR. If Next_is_Gap = 1, IRR13 is set. (2) When a Time Reference message is transmitted from Mailbox-30 the value of TCNTR stored into an internal register at the SOF is copied into Ref_Mark. CCR is incremented when TTT of Mailbox-30 matches with CYCTR . CCR is embedded into the first data byte of the time reference message { Data0[7:6], CCR[5:0] } .

- Setting Tx-Trigger Time

The Tx-Trigger Time(TTT) must be set in ascending order shown below, and the difference between them has to satisfy the following expressions. TEW in the following expressions is the register value.

TTT (Mailbox-24) < TTT (Mailbox-25) < TTT (Mailbox-26) < TTT (Mailbox-27) < TTT (Mailbox-28) < TTT (Mailbox-29) < TTT (Mailbox-30)

and

TTT (Mailbox-i) – TTT (Mailbox- i-1) > TEW + the maximum frame length + 9

TTT (Mailbox-24) to TTT (Mailbox-29) correspond to Time_Marks, and TTT (Mailbox-30) corresponds to Time_Ref showing the length of a basic cycle, respectively when working as potential time master.

chanhvinh.com

The above limitation is not applied to mailboxes which are not set as time-triggered transmission.

Important: Because of limitation on setting Tx-Trigger Time, only one Mailbox can be assigned to one time window.

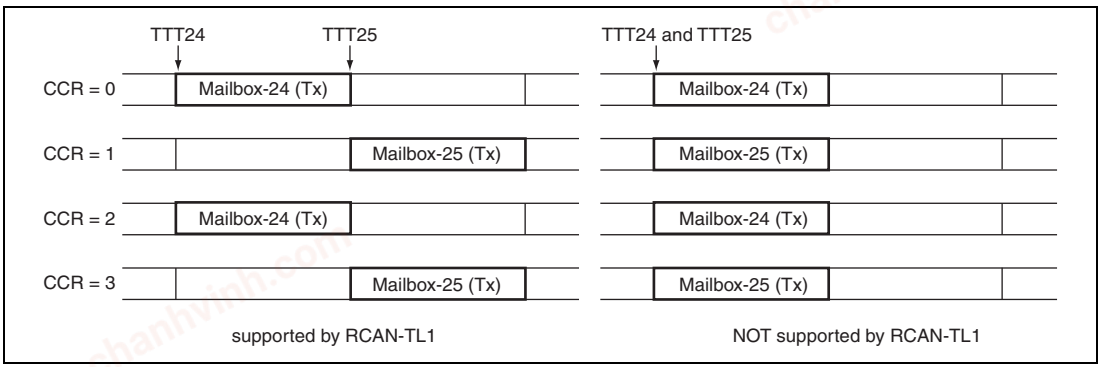


Figure 20.18 Limitation on Tx-Trigger Time

The value of TCMR2 as Watch_Trigger has to be larger than TTT(Mailbox-30), which shows the length of a basic cycle.

chanhvinh.com

Figures 20.19 and 20.20 show examples of configurations for (Potential) Time Master and Time Slave. "L" in diagrams shows the length in time of the time reference messages.

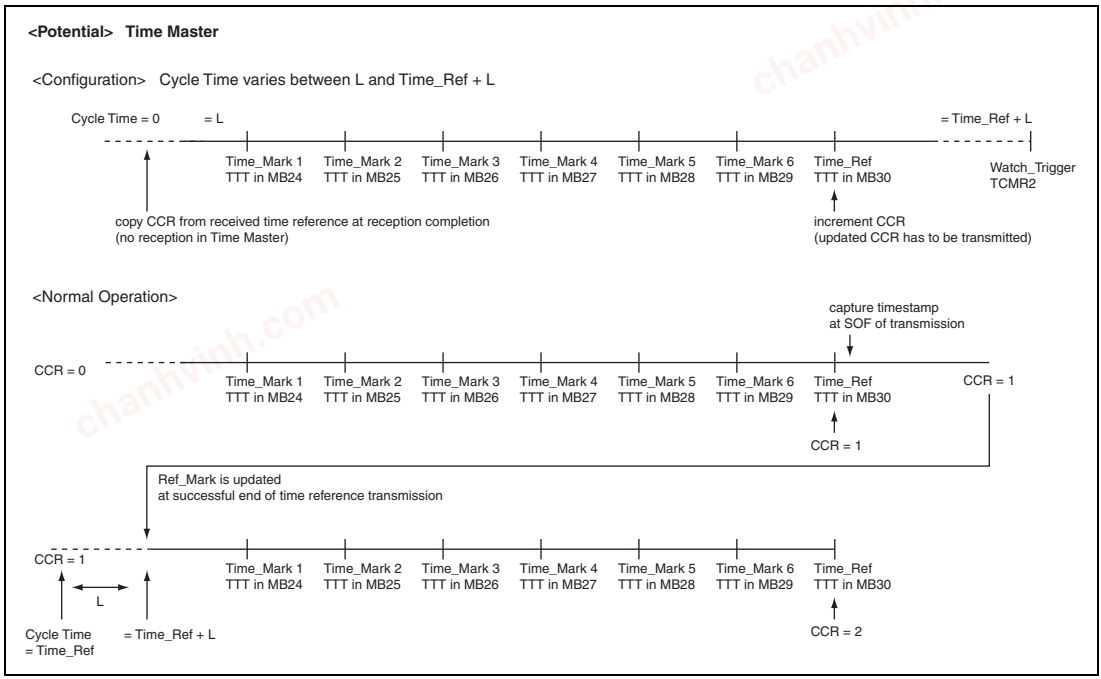


Figure 20.19 (Potential) Time Master

chanhvinh.com

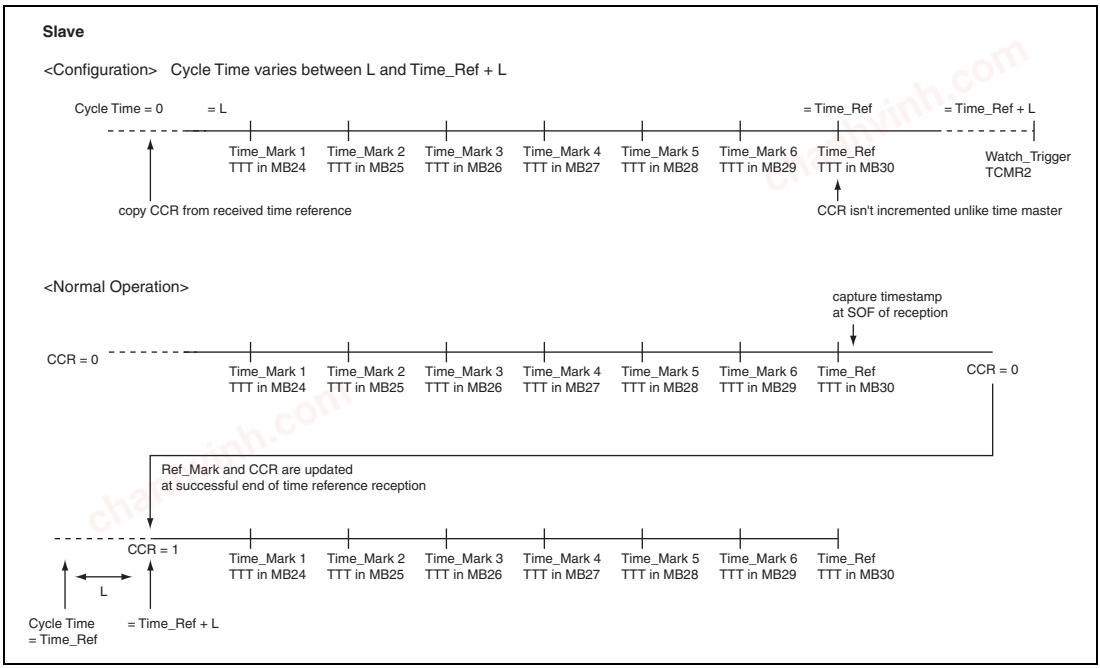


Figure 20.20 Time Slave

- Function to be implemented by software

Some of the TTCAN functions need to be implemented in software. The main details are reported hereafter. Please refer to ISO-11898-4 for more details.

— Change from Init_Watch_Trigger to Watch_Trigger

This module offers the two registers TCMR0 and TCMR2 as H/W support for Init_Watch_Trigger and Watch_Trigger respectively. The SW is requested to enable TCMR0 and disable TCMR2 up to the first reference message is detected on the CAN Bus and then disable TCMR0 and enable TCMR2.- Schedule Synchronization state machine.

Only reception of Next_is_Gap interrupt is supported. The application needs to take care of stopping all transmission at the end of the current basic cycle by setting the related TXCR flags.Master-Slave Mode control.

Only automatic cycle time synchronization and CCR increment is supported.

— Message status count

Software has to count scheduling errors for periodic messages in exclusive windows.

chanhvinh.com

- Message Transmission Request for Time Triggered communication

When the Time Triggered mode is used communications must fulfils the ISO11898-4 requirements.

The following procedure should be used.

— Send this module to reset or halt mode
— Set TCMR0 to the Init_Watch_Trigger (0xFFFF)
— Enable TCMR0 compare match setting bit 10 of TTCR0
— Set TCMR2 to the specified Watch_Trigger value
— Keep TCMR2 compare match disabled by keeping cleared the bit 12 of TTCR0
— Set CMAX to the requested value (different from 111 bin)
— Set TEW to the requested value
— Configure the necessary Mailboxes for Time Trigger transmission and reception
— Set LAFM for the 3 LSBs of Mailbox 31
— Configure MCR, BCR1 and BCR0 to the requested values
— If working as a potential time master:
  - Set RFTROFF to the requested Init_Ref_Offset value
  - Set TXPR for Mailbox 30
  - Write H'4000 into TTTSEL
— Enable the TCNTR timer through the bit 15 of TTCR0
— Move to Transmission_Reception mode
— Wait for the reception or transmission of a valid reference message or for TCMR0 match
— If the local time reaches the value of TCMR0 the Init_Watch_Trigger is reached and the application needs to set TXCR for Mailbox 30 and start again
— If the reference message is transmitted (TXACK[30] is set) set RFTROFF to zero
— If a valid reference message is received (RXPR[31] is set) then:
  - If 3 LSBs of ID of Mailbox 31 have high priority than the 3 LSBs of Mailbox 30 (if working as potential time master) keep RFTROFF to Init_Ref_Offset
  - If 3 LSBs of ID of Mailbox 31 have lower priority than the 3 LSBs of Mailbox 30 (if working as potential time master) decrement by 1 the value in RFTROFF
— Disable TCMR0 compare match by clearing bit 10 of TTCR0
— Enable TCMR2 compare match by setting bit 12 of TTCR0
— Only after two reference messages have been detected on the CAN Bus (transmitted or received) can the application set TXPR for the other Time Triggered Mailboxes.

chanhvinh.com

If, at any time, a reference message cannot be detected on the CAN Bus, and the cycle time CYCTR reaches TCMR2, this module automatically aborts all pending transmissions (including the Reference Message).

The following is the sequence to request further transmission in Time Triggered mode.

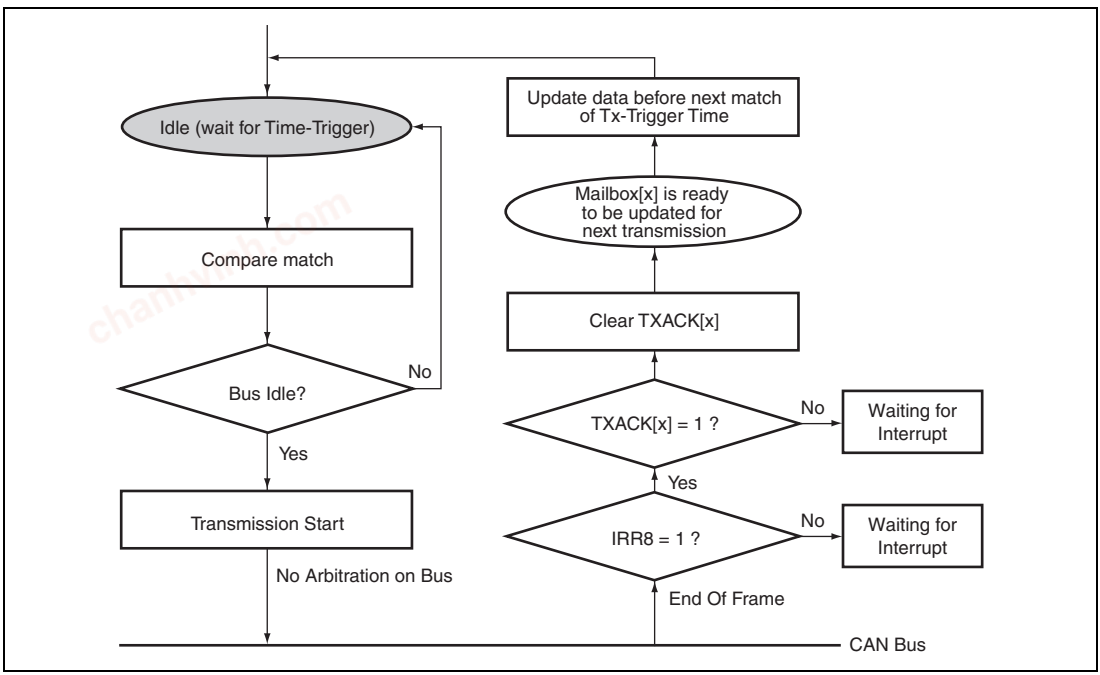


Figure 20.21 Message transmission request

S/W has to ensure that a message is updated before a Tx trigger for transmission occurs.

When the CYCTR reaches to TTT (Tx-Trigger Time) of a Mailbox and CCR matches with the programmed cycle for transmission, this module immediately transfers the message into the Tx buffer. At this point, this module will attempt a transmission within the specified Time Enable Window. If this module misses this time slot, it will suspend the transmission request up to the next Tx Trigger, keeping the corresponding TXPR bit set to '1' if the transmission is periodic (Mailbox-24 to 30). There are three factors that may cause this module to miss the time slot –

1. The CAN bus currently used
2. An error on the CAN bus during the time triggered message transmission
3. Arbitration loss during the time triggered message transmission

chanhvinh.com

In case of Merged Arbitrating Window the slot for transmission goes from the Tx_Trig of the Mailbox opening the Window (TTW = 10 bin) to the end to the TEW of the Mailbox closing the Window (TTW = 11 bin).The TXPR can be modified at any time. This module ensures the transmission of Time Triggered messages is always scheduled correctly. However, in order to guarantee the correct schedule, there are some important rules that are :

— TTT (Tx Trigger Time) can be modified during configuration mode.
— TTT cannot be set outside the range of Time_Ref, which specifies the length of basic cycle. This could cause a scheduling problem.
— TXPR is not automatically cleared for periodic transmission. If a periodic transmission needs to be cancelled, the corresponding TXCR bit needs to be set by the application.

- Example of Time Triggered System

The following diagram shows a simple example of how time trigger system works using this module in time slave mode.

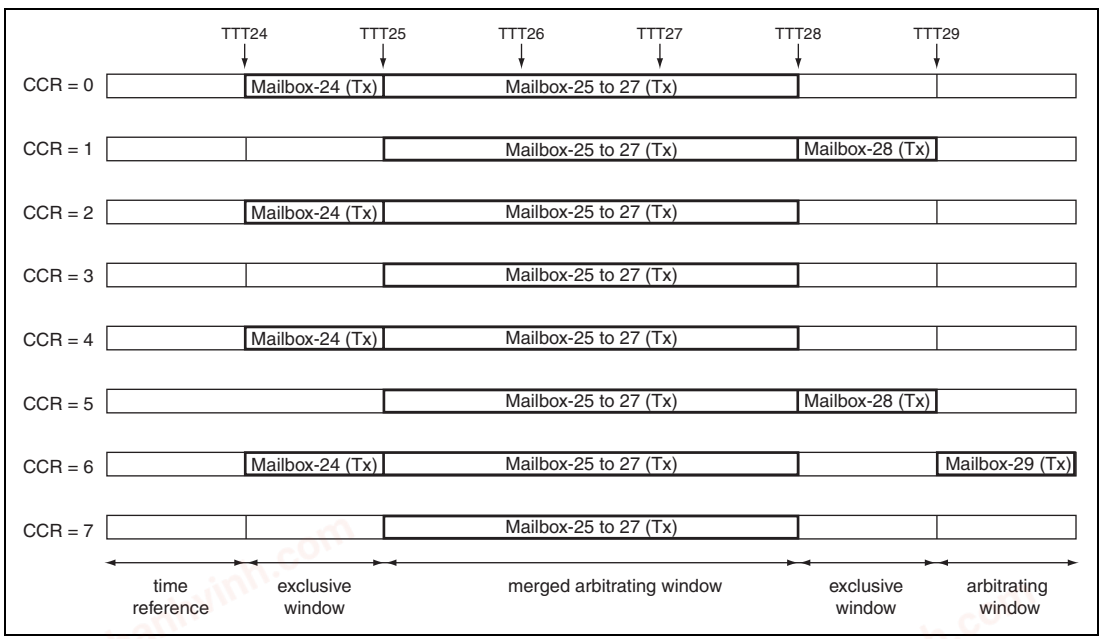


Figure 20.22 Example of Time trigger system as Time Slave

chanhvinh.com

The following settings were used in the above example:

	rep_factor (register)	Offset	TTW[1:0]	MBC[2:0]
Mailbox-24	3'b001	6'b000000	2'b00	3'b000
Mailbox-25	3'b000	6'b000000	2'b10	3'b000
Mailbox-26	3'b000	6'b000000	2'b10	3'b000
Mailbox-27	3'b000	6'b000000	2'b11	3'b000
Mailbox-28	3'b010	6'b000001	2'b00	3'b000
Mailbox-29	3'b011	6'b000110	2'b01	3'b000
Mailbox-30	—	—	—	3'b111
Mailbox-31	—	—	—	3'b011

CMAX = 3'b011, TXPR[30] = 0

During merged arbitrating window, request by time-triggered transmission is served in the way of FCFS (First Come First Served). For example, if Mailbox-25 cannot be transmitted between Tx-Trigger Time 25 (TTT25) and TTT26, Mailbox-25 has higher priority than Mailbox-26 between TTT26 and 28.

MBC needs to be set into 3'b111, in order to disable time-triggered transmission. If this module is Time Master, MBC[30] has to be 3'b000 and time reference window is automatically recognized as arbitrating window.

- Timer Operation

Figure 20.23 shows the timing diagram of the timer. By setting Tx-Trigger Time = n, time trigger transmission starts between CYCTR = n + 2 and
CYCTR = n + 3.

chanhvinh.com

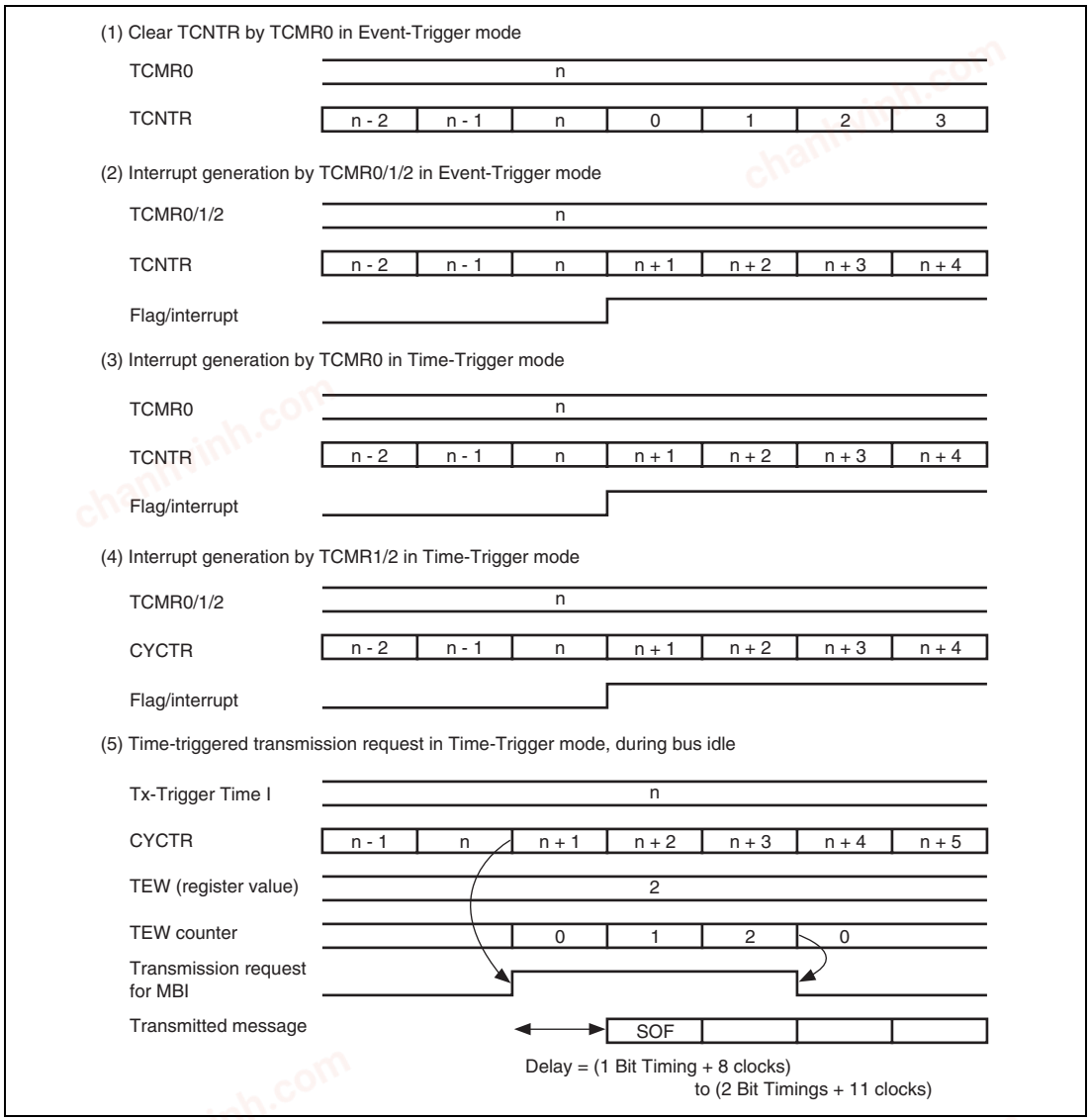


Figure 20.23 Timing Diagram of Timer

chanhvinh.com

During merged arbitrating window, event-trigger transmission is served after completion of time-triggered transmission. For example, If transmission of Mailbox-25 is completed and CYCTR doesn't reach TTT26, event-trigger transmission starts based on message transmission priority specified by MCR2. TXPR of time-triggered transmission is not cleared after transmission completion, however, that of event-triggered transmission is cleared.

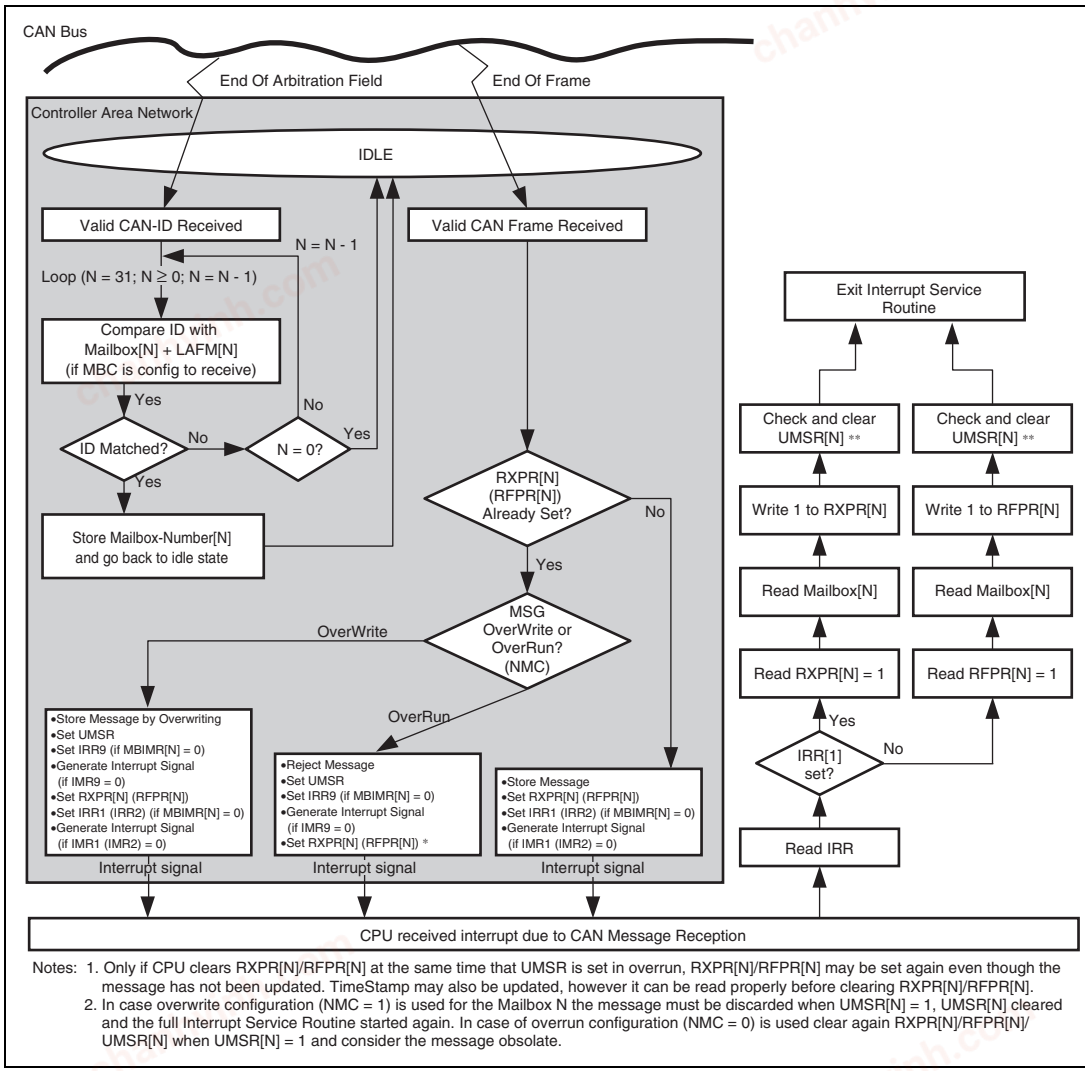
Note: that in the case that the TXPR is not set for the Mailbox which is assigned to close the Merged Arbitrating Window (MAW), then the MAW will still be closed (at the end of the TEW following the TTT of the assigned Mailbox.

Please refer to Table Roles of Mailboxes in section 20.3.2, Mailbox Structure.

chanhvinh.com

20.4.4 Message Receive Sequence

The diagram below shows the message receive sequence.



Notes: 1. Only if CPU clears RXPR[N]/RFPR[N] at the same time that UMSR is set in overrun, RXPR[N]/RFPR[N] may be set again even though the message has not been updated. TimeStamp may also be updated, however it can be read properly before clearing RXPR[N]/RFPR[N].
2. In case overwrite configuration (NMC = 1) is used for the Mailbox N the message must be discarded when UMSR[N] = 1, UMSR[N] cleared and the full Interrupt Service Routine started again. In case of overrun configuration (NMC = 0) is used clear again RXPR[N]/RFPR[N]/ UMSR[N] when UMSR[N] = 1 and consider the message obsolete.

Figure 20.24 Message receive sequence

chanhvinh.com

When this module recognises the end of the Arbitration field while receiving a message, it starts comparing the received identifier to the identifiers set in the Mailboxes, starting from Mailbox-31 down to Mailbox-0. It first checks the MBC if it is configured as a receive box, and reads LAFM, and reads the CAN-ID of Mailbox-31 (if configured as receive) to finally compare them to the received ID. If it does not match, the same check takes place at Mailbox-30 (if configured as receive). Once this module finds a matching identifier, it stores the number of Mailbox-[N] into an internal buffer, stops the search, and goes back to idle state, waiting for the EndOfFrame (EOF) to come. When the 6th bit of EOF is notified by the CAN Interface logic, the received message is written or abandoned, depending on the NMC bit. No modification of configuration during communication is allowed. Entering Halt Mode is one of ways to modify configuration. If it is written into the corresponding Mailbox, including the CAN-ID, i.e., there is a possibility that the CAN-ID is overwritten by a different CAN-ID of the received message due to the LAFM used. This also implies that, if the identifier of a received message matches to ID + LAFM of 2 or more Mailboxes, the higher numbered Mailbox will always store the relevant messages and the lower numbered Mailbox will never receive messages. Therefore, the settings of the identifiers and LAFMs need to be carefully selected.

With regards to the reception of data and remote frames described in the above flow diagram the clearing of the UMSR flag after the reading of IRR is to detect situations where a message is overwritten by a new incoming message stored in the same mailbox (if its NMC = 1) while the interrupt service routine is running. If during the final check of UMSR a overwrite condition is detected the message needs to be discarded and read again.

In case UMSR is set and the Mailbox is configured for overrun (NMC = 0) the message is still valid, however it is obsolete as it is not reflecting the latest message monitored on the CAN Bus.

Please access the full Mailbox content before clearing the related RXPR/RFPR flag.

Please note that in the case a received remote frame is overwritten by a data frame, both the remote frame receive interrupt (IRR2) and data frame received interrupt (IRR1) and also the Receive Flags (RXPR and RFPR) are set. In an analogous way, the overwriting of a data frame by a remote frame, leads to setting both IRR2 and IRR1.

When a message is received and stored into a Mailbox all the fields of the data not received are stored as zero. The same applies when a standard frame is received. The extended identifier part (EXTID[17:0]) is written as zero.

chanhvinh.com

20.4.5 Reconfiguration of Mailbox

When re-configuration of Mailboxes is required, the following procedures should be taken.

- Change configuration of transmit box

  Two cases are possible.

  — Change of ID, RTR, IDE, LAFM, Data, DLC, NMC, ATX, DART

  This change is possible only when MBC = 3'b000. Confirm that the corresponding TXPR is not set. The configuration (except MBC bit) can be changed at any time.

  — Change from transmit to receive configuration (MBC)

  Confirm that the corresponding TXPR is not set. The configuration can be changed only in Halt or reset state. Please note that it might take longer for this module to transit to halt state if it is receiving or transmitting a message (as the transition to the halt state is delayed until the end of the reception/transmission), and also this module will not be able to receive/transmit messages during the Halt state.

  In case this module is in the Bus Off state the transition to halt state depends on the configuration of the bit 6 of MCR and also bit and 14 of MCR.

- Change configuration (ID, RTR, IDE, LAFM, Data, DLC, NMC, ATX, DART, MBC) of receiver box or Change receiver box to transmitter box

  The configuration can be changed only in Halt Mode.

  This module will not lose a message if the message is currently on the CAN bus and this module is a receiver. This module will be moving into Halt Mode after completing the current reception. Please note that it might take longer if this module is receiving or transmitting a message (as the transition to the halt state is delayed until the end of the reception/transmission), and also this module will not be able to receive/transmit messages during the Halt Mode.

  In case this module is in the Bus Off state the transition to halt mode depends on the configuration of the bit 6 and 14 of MCR.

chanhvinh.com

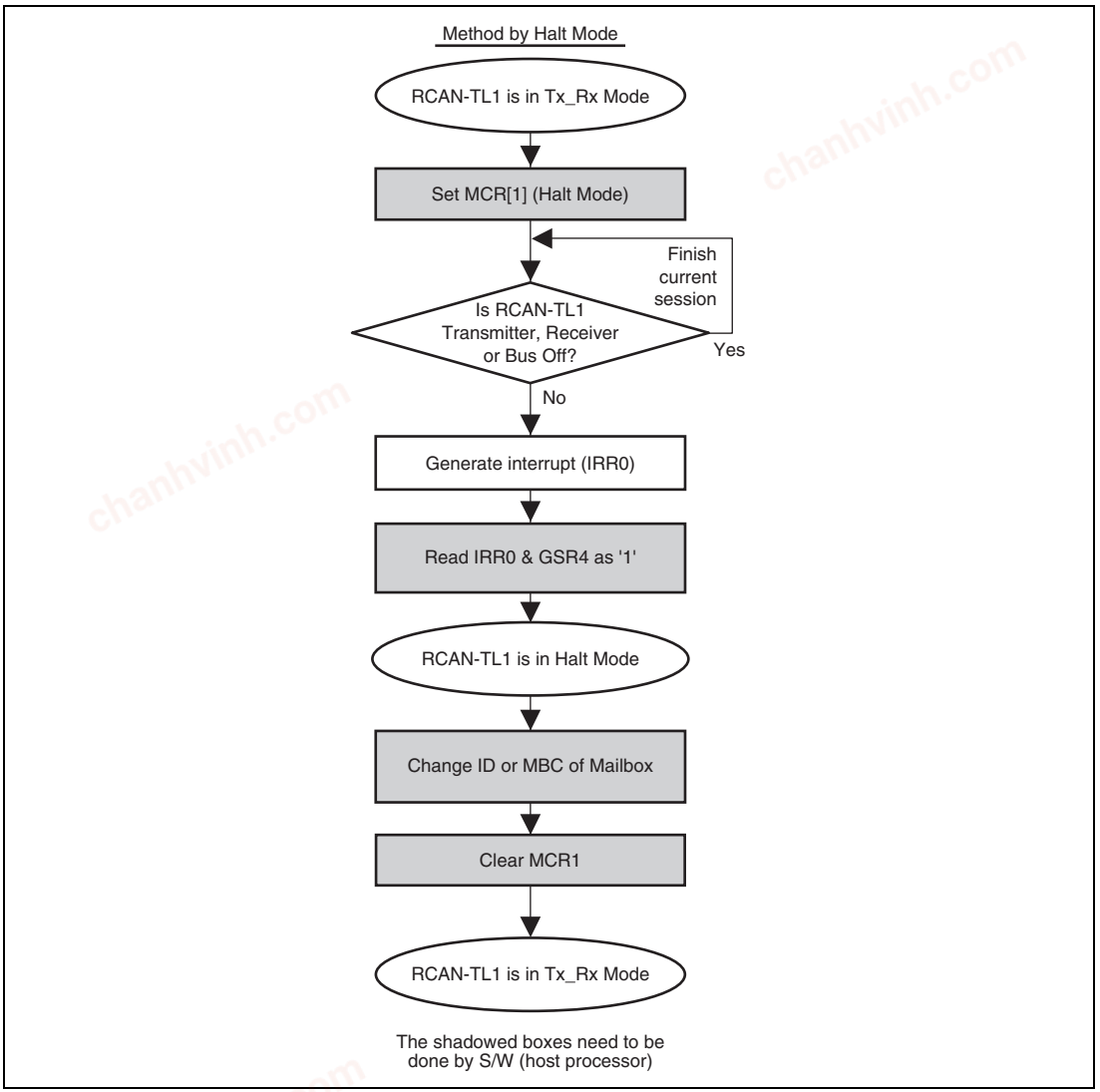


Figure 20.25 Change ID of receive box or Change receive box to transmit box

chanhvinh.com

20.5 Interrupt Sources

Table 20.2 lists this module interrupt sources. These sources can be masked. Masking is implemented using the mailbox interrupt mask registers (MBIMR) and interrupt mask register (IMR). For details on the interrupt vector of each interrupt source, see section 7, Interrupt Controller.

Table 20.2 Interrupt Sources

Interrupt	Description	Interrupt Flag	DMAC Activation
ERSn*[1]	Error Passive Mode (TEC ≥ 128 or REC ≥ 128)	IRR5	Not possible
	Bus Off (TEC ≥ 256)/Bus Off recovery	IRR6	
	Error warning (TEC ≥ 96)	IRR3	
	Error warning (REC ≥ 96)	IRR4	
OVRn*[1]	Reset/halt/CAN sleep transition	IRR0	
	Overload frame transmission	IRR7	
	Unread message overwrite (overrun)	IRR9	
	Start of new system matrix	IRR10	
	TCMR2 compare match	IRR11	
	Bus activity while in sleep mode	IRR12	
	Timer overrun/Next_is_Gap reception/message error	IRR13	
	TCMR0 compare match	IRR14	
	TCMR1 compare match	IRR15	
RM0n*[1]*[2], RM1n*[1]*[2]	Data frame reception	IRR1*[3]	Possible*[4]
	Remote frame reception	IRR2*[3]	
SLEn*[1]	Message transmission/transmission disabled (slot empty)	IRR8	Not possible

Notes: 1. n = 0, 1

2. RM0 is an interrupt generated by the remote request pending flag for mailbox 0 (RFPR0[0]) or the data frame receive flag for mailbox 0 (RXPR0[0]). RM1 is an interrupt generated by the remote request pending flag for mailbox n (RFPR0[n]) or the data frame receive flag for mailbox n (RXPR0[n]) (n = 1 to 31).
3. IRR1 is a data frame received interrupt flag for mailboxes 0 to 31, and IRR2 is a remote frame request interrupt flag for mailboxes 0 to 31.
4. The direct memory access controller is activated only by an RM0n interrupt.

chanhvinh.com

20.6 DMAC Interface

The DMAC can be activated by the reception of a message in mailbox 0. When DMAC transfer ends after DMAC activation has been set, flags of RXPR0 and RFPR0 are cleared automatically. An interrupt request due to a receive interrupt from this module cannot be sent to the CPU in this case. Figure 20.26 shows a DMAC transfer flowchart.

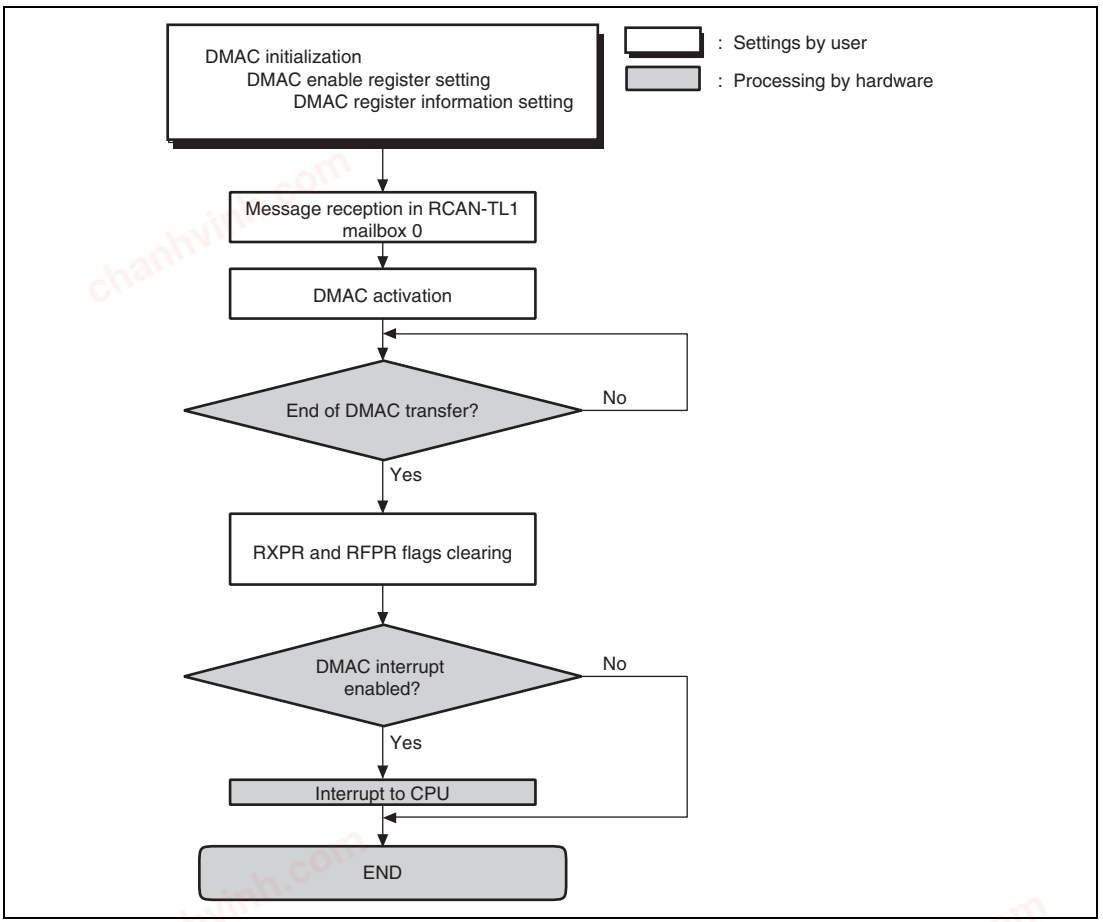


Figure 20.26 DMAC Transfer Flowchart

chanhvinh.com

20.7 CAN Bus Interface

A bus transceiver IC is necessary to connect this LSI to a CAN bus. A Renesas HA13721 transceiver IC and its compatible products are recommended. As the CRx and CTx pins use 3 V, an external level shifter is necessary. Figure 20.27 shows a sample connection diagram.

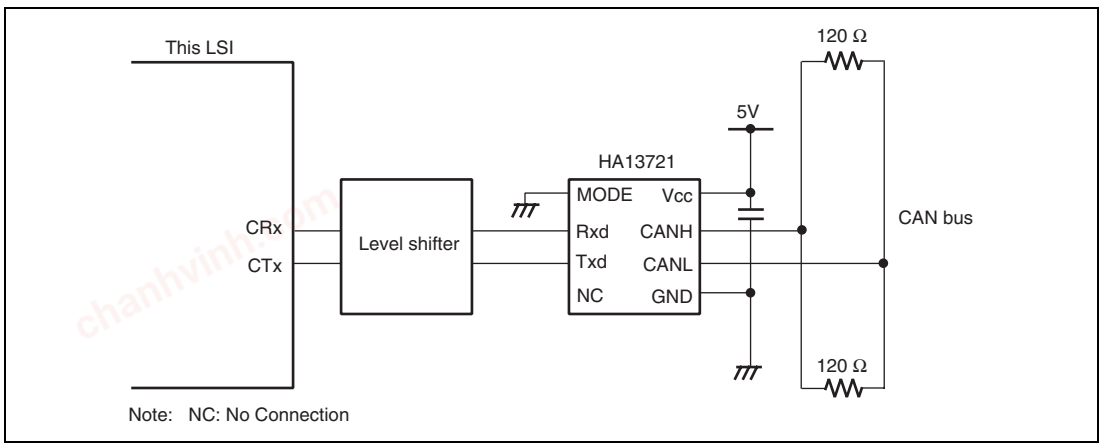


Figure 20.27 High-Speed CAN Interface Using HA13721

chanhvinh.com

20.8 Setting I/O Ports

The I/O ports for this module must be specified before or during the configuration mode. For details on the settings of I/O ports, see section 32, General Purpose I/O Ports. Two methods are available using two channels of this module in this LSI.

- Using this module as a 2-channel module (channels 0 and 1)
  Each channel has 32 Mailboxes.
- Using this module as a 1-channel module (channels 0 and 1 functioning as a single channel)

When the second method is used, see section 20.9.1, Notes on Port Setting for Multiple Channels Used as Single Channel.

Figures 20.28 and 20.29 show connection examples for individual port settings.

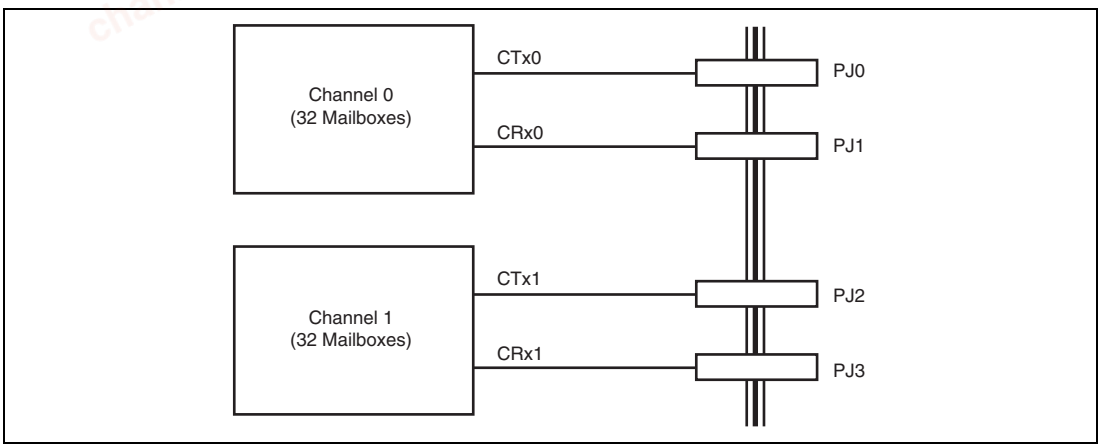


Figure 20.28 Connection Example when Using This Module as 2-Channel Module (32 Mailboxes × 2 Channels)

chanhvinh.com

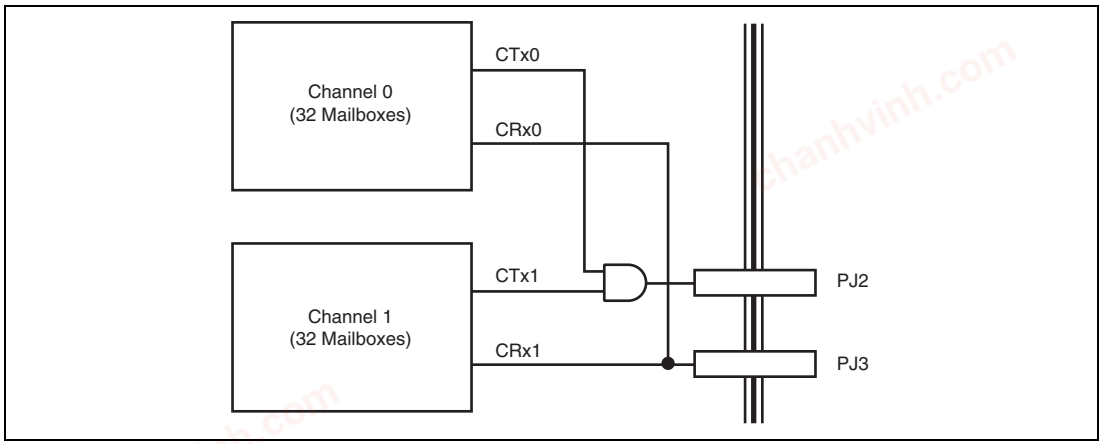


Figure 20.29 Connection Example when Using This Module as 1-Channel Module (64 Mailboxes × 1 Channel)

chanhvinh.com

20.9 Usage Notes

20.9.1 Notes on Port Setting for Multiple Channels Used as Single Channel

This module in this LSI has two channels and some of these channels can be used as a single channel. When using multiple channels as a single channel, keep the following in mind.

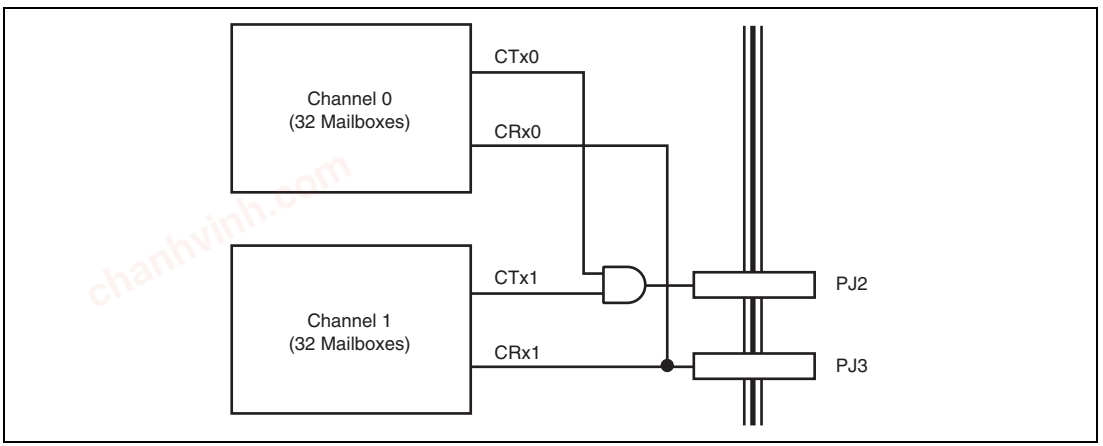


Figure 20.30 Connection Example when Using This Module as 1-Channel Module (64 Mailboxes × 1 Channel)

1. No ACK error is detected even when any other nodes are not connected to the CAN bus. This occurs when channel 1 transmits an ACK in the ACK field in response to a message channel 0 has transmitted.

   Channel 1 receives a message which channel 0 has transmitted on the CAN bus and then transmits an ACK in the ACK field. After that, channel 0 receives the ACK.

   To avoid this, make channel 1 which is not currently used for transmission the listen-only mode (TST[2:0] = B'001) or the reset state (MCR0 = 1). With this setting, only a channel which transmits a message transmits an ACK.
2. Internal arbitration for channels 0 and 1 is independently controlled to determine the order of transmission.

   Although the internal arbitration is performed on 31 Mailboxes at a time, it is not performed on 64 Mailboxes at a time even though multiple channels function as a single channel.
3. Do not set the same transmission message ID in both channels 0 and 1.

   Two messages may be transmitted from the two channels after arbitration on the CAN bus.

chanhvinh.com

Section 21 IEBus™ Controller

This LSI has an on-chip one-channel IEBus controller. The Inter Equipment Bus™ (IEBus™)* is a small-scale digital data transfer system for inter-equipment data transfer.

This LSI does not have an on-chip IEBus driver/receiver, so it is necessary to mount a dedicated driver/receiver externally. In addition, as the IERxD and IETxD pins need 3V to operate, a dedicated external level shifter is necessary.

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

21.1 Features

- IEBus protocol control (layer 2) supported
  - Half-duplex asynchronous communications
  - Multi-master system
  - Broadcast communications function
  - Selectable mode (three types) with different transfer speeds
- On-chip buffers for data transmission and reception
  - Transmission and reception buffers: 128 bytes each
  - Up to 128 bytes of consecutive transmit/reception (maximum number of transfer bytes in mode 2)
- Operating frequency
  - 12 MHz, 12.58 MHz (This module uses 1/2 divided clocks of Pϕ, or AUDIO_X1*, AUDIO_X2*.)
  - 18 MHz, 18.87 MHz (This module uses 1/3 divided clocks of Pϕ, or AUDIO_X1*, AUDIO_X2*.)
  - 24 MHz (This module uses 1/4 divided clocks of Pϕ, or AUDIO_X1*, AUDIO_X2*.)
  - 25.16 MHz (This module uses 1/4 divided clocks of Pϕ.)
  - 30 MHz, 31.45 MHz (This module uses 1/5 divided clocks of Pϕ.)
  - 36 MHz (This module uses 1/6 divided clocks of Pϕ.)

Note: * Available as this module clock input only when not used as the clock input for serial sound interface, serial I/O with FIFO, or Renesas SPDIF interface.

- Module standby mode can be set.

chanhvinh.com

21.1.1 IEBus Communications Protocol

An overview of the IEBus is provided below.

- Communications method: Half-duplex asynchronous communications
- Multi-master system
  All units connected to the IEBus can transfer data to other units.
- Broadcast communications function (one-to-many communications)
  - Group broadcast communications: Broadcast communications to group unit
  - General broadcast communications: Broadcast communications to all units
- Mode is selectable (three modes with different transfer speeds)

Table 21.1 Mode Types

Mode	IEBϕ*1 = 12, 18, 24*2, 30, 36 MHz	IEBϕ*1 = 12.58, 18.87*2, 25.16, 31.45 MHz	Maximum Number of Transfer Bytes (byte/frame)
0	About 3.9 kbps	About 4.1 kbps	16
1	About 17 kbps	About 18 kbps	32
2	About 26 kbps	About 27 kbps	128

Notes: 1. Peripheral clock (Pϕ), or clocks for AUDIO_X1 and AUDIO_X2
2. Oscillation frequency when this LSI is used

- Access control: CSMA/CD (Carrier Sense Multiple Access with Collision Detection)
  Priority of bus mastership is as follows.
  - Broadcast communications (one-to-many communications) have priority over normal communications (one-to-one communications).
  - A smaller master address has priority.
- Communications scale
  - Number of units: Up to 50
  - Cable length: Up to 150 m (when using a twisted-pair cable)

Note: The communications scale of the actual system depends on the characteristics of the externally mounted IEBus driver/receiver and the cable used.

chanhvinh.com

(1) Determination of Bus Mastership (Arbitration)

A unit connected to the IEBus performs an operation to get the bus to control other units. This operation is called arbitration. In arbitration, when multiple units start transferring simultaneously, the bus mastership is given to one unit among them.

Only one unit can obtain bus mastership through arbitration, so the following priority for bus mastership is determined.

(a) Priority according to communications type

Broadcast communications (one-to-many communications) has priority over normal communications (one-to-one communications).

(b) Priority according to master address

The unit with the smallest master address has priority among units of the same communications type.

Example: The master address is configured with 12 bits. A unit with H'000 has the highest priority, while a unit with H'FFF has the lowest priority.

Note: When a unit loses in arbitration, the unit can automatically enter retransfer mode (0 to 7 retransfer times can be selected by the RN bit in IEMCR).

chanhvinh.com

(2) Communications Mode

The IEBus has three communications modes with different transfer speeds. Table 21.2 shows the transfer speed in each communications mode and the maximum number of transfer bytes in one communications frame.

Table 21.2 Transfer Speed and Maximum Number of Transfer Bytes in Each Communications Mode

Communications Mode	Maximum Number of Transfer Bytes (bytes/frame)	Effective Transfer Speed*1 (kbps)	
		IEBϕ*2 = 12, 18, 24*3, 30, 36 MHz	IEBϕ*2 = 12.58, 18.87*3, 25.16, 31.45 MHz
0	16	About 3.9	About 4.1
1	32	About 17	About 18
2	128	About 26	About 27

Notes: Each unit connected to the IEBus should select a communications mode prior to performing communications. Note that correct communications is not guaranteed if the master and slave units do not adopt the same communications mode.

In the case of communications between a unit with IEBϕ = 6 MHz and a unit with IEBϕ = 6.29 MHz, correct communications are not possible even if the same communications mode is adopted. Communications must be done with the same oscillation frequency.

1. Effective transfer speed when the maximum number of transfer bytes is transmitted.
2. Peripheral clock (Pϕ), or clocks for AUDIO_X1 and AUDIO_X2
3. Oscillation frequency when this LSI is used

(3) Communications Address

In the IEBus, a specific 12-bit communications address is allocated to each individual unit. A communications address is configured as follows.

- Upper four bits: group number (number identifying a group to which the unit belongs)
- Lower eight bits: unit number (number identifying individual units in a group)

chanhvinh.com

(4) Broadcast Communications

In normal transfer, a single master unit communicates with a single slave unit, so one-to-one transfer or reception takes place. In broadcast communications, a single master unit communicates with multiple slave units. Since there are multiple slave units, no acknowledgements are returned from the slave units during communications.

A broadcast bit decides whether broadcast or normal communications is done. (For details of the broadcast bit, see section 21.1.2 (1) (b), Broadcast Bit.

There are two types of broadcast communications.

(a) Group broadcast communications

Broadcast communications is aimed at units with the same group number, meaning that those units have the same upper four bits of the communications address.

(b) General broadcast communications

Broadcast communications is aimed at all units regardless of group number.

Group broadcast and general broadcast communications are identified by a slave address. (For details on the slave address, see section 21.1.2 (3), Slave Address Field.)

chanhvinh.com

21.1.2 Communications Protocol

Figure 21.1 shows an IEBus transfer signal format.

Communications data is transferred as a series of signals referred to as a communications frame. The number of data, which can be transmitted in a single communications frame and the transfer speed, differs according to the communications mode.

(When IEBϕ = 12, 18, 24, 30, or 36 MHz)

Field name	Header		Master address field		Slave address field			Control field			Message length field			Data field								
Number of bits	1	1	12	1	12	1	1	4	1	1	8	1	1	8	1	1		8	1	1
	Start bit	Broad-cast bit	Master address	P	Slave address	P	A	Control bits	P	A	Message length bits	P	A	Data bits	P	A		Data bits	P	A
Transfer time																					
Mode 0	Approximately 7330 µs													Approximately 1590 × N µs							
Mode 1	Approximately 2090 µs													Approximately 410 × N µs							
Mode 2	Approximately 1590 µs													Approximately 300 × N µs							

P: Parity bit (1 bit)
A: Acknowledge bit (1 bit)
When A = 0: ACK
When A = 1: NAK
N: Number of bytes

Note: The value of acknowledge bit is ignored in broadcast communications.

Figure 21.1 Transfer Signal Format

(1) Header

A header is comprised of a start bit and a broadcast bit.

(a) Start Bit

The start bit is a signal to inform other units of the start of data transfer. A unit attempting to start data transfer outputs a low-level signal (the start bit) for a specified period and then outputs the broadcast bit.

If another unit is already outputting a start bit when a unit attempts to output a start bit, the unit waits for completion of the start bit from the other unit without outputting its own start bit, and then outputs the broadcast bit synchronized with the completion timing.

Other units enter the receive state after detecting the start bit.

chanhvinh.com

(b) Broadcast Bit

The broadcast bit is a bit to identify the type of communications: broadcast or normal.

When this bit is cleared to 0, it indicates broadcast communications. When it is set to 1, it indicates normal communications. Broadcast communications includes group broadcast and general broadcast, which are identified by a value of the slave address. (For details of the slave address, see section 21.1.2 (3), Slave Address Field.)

Since multiple slave units are communications destination units, in the case of broadcast communications, the acknowledge bit is not returned from each field described in (2) and below.

When more than one unit starts to transfer a communications frame with the same timing, broadcast communications has priority over normal communications, and arbitration occurs.

(2) Master Address Field

The master address field is a field for transmitting the unit address (master address) to other units. The master address field is comprised of master address bits and a parity bit.

The master address consists of 12 bits and the MSB is output first.

When more than one unit start to transfer broadcast bits having the same value with the same timing, arbitration is decided by the master address field.

In the master address field, self-output data and data on the bus are compared for every one-bit transfer. If the self-output master address and data on the bus are different, the unit that loses arbitration will stop its transfer and enter the receive state.

Since the IEBus is configured with wired AND, the unit having the smallest master address of the units in arbitration (arbitration master) wins in arbitration.

Finally, only a single unit remains in the transfer state as a master unit after outputting a 12-bit master address.

Next, this master unit outputs a parity bit*, defines the master address for other units, and then enters the slave address field output state.

Note: * Since even parity is used, when the number of one bit in the master address is odd, the parity bit is 1.

chanhvinh.com

(3) Slave Address Field

The slave address field is a field to transmit an address (the slave address) of a unit (the slave unit) to be transmitted. The slave address field is comprised of slave address bits, a parity bit, and an acknowledge bit.

The slave address consists of 12 bits and the MSB is output first. The parity bit is output after the 12-bit slave address is transmitted to avoid receiving the slave address accidentally. The master unit then detects the acknowledgement from the slave unit to confirm that the slave unit exists on the bus. When the acknowledgement is detected, the master unit enters the control field output state. However, the master unit enters the control field output state without detecting the acknowledgement in broadcast communications.

The slave unit returns an acknowledgement when the slave addresses match and the parities of the master and slave addresses are correct. When the parity of either the master or slave address is incorrect, the slave unit decides that the master or slave address was not correctly received and does not return the acknowledgement. In this case, the master unit enters the waiting (monitor) state and communications ends.

In the case of broadcast communications, the slave address is used to identify the type of broadcast communications (group or general) as follows:

- When the slave address is H'FFF: General broadcast communications
- When the slave address is other than H'FFF: Group broadcast communications

Note: The group number is the upper 4-bit value of the slave address in group broadcast communications.

(4) Control Field

The control field is a field for transmitting the type and direction of the following data field. The control field is comprised of control bits, a parity bit, and an acknowledge bit.

The control bits consist of four bits and the MSB is output first.

The parity bit is output following the control bits. When the parity is correct, and the slave unit can implement the function required from the master unit, the slave unit returns an acknowledgement and enters the message length field output state. However, if the slave unit cannot implement the requirements from the master unit even though the parity is correct, or if the parity is not correct, the slave unit does not return an acknowledgement and returns to the waiting (monitor) state.

chanhvinh.com

The master unit enters the subsequent message length field output state after confirming the acknowledgement.

When the acknowledgement is not confirmed, the master unit enters the waiting (monitor) state, and communications ends. However, in the case of broadcast communications, the master unit enters the following message length field output state without confirming the acknowledgement. For details of the contents of the control bit, see table 21.4.

(5) Message Length Field

The message length field is a field for specifying the number of transfer bytes. The message length field is comprised of message length bits, a parity bit, and an acknowledge bit.

The message length has eight bits and the MSB is output first. Table 21.3 shows the number of transfer bytes.

Table 21.3 Contents of Message Length bits

Message Length bits (Hexadecimal)	Number of Transfer Bytes
H'01	1 byte
H'02	2 bytes
:	:
H'FF	255 bytes
H'00	256 bytes

Note: If a number greater than the maximum number of transfer bytes in one frame is specified, communications are done in multiple frames depending on the communications mode. In this case, the message length bits indicate the number of remaining communications data after the first transfer. In this LSI, the message length bits must be smaller than the maximum number of transfer bytes in one frame. Set these within the ranges shown below.
Mode 0: 1 to 16 bytes
Mode 1: 1 to 32 bytes
Mode 2: 1 to 128 bytes

This field operation differs depending on the value of bit 3 in the control field: master transmission (the bit 3 of the control bits is 1) or master reception (the bit 3 of the control bits is 0).

chanhvinh.com

(a) Master Transmission

The master unit outputs the message length bits and the parity bit. When the parity is even, the slave unit returns an acknowledgement and enters the following data field. Note that the slave unit does not return an acknowledgement in broadcast communications.

When the parity is odd, the slave unit decides that the message length field is not correctly received, does not return an acknowledgement, and returns to the waiting (monitor) state. In this case, the master unit also returns to the waiting state and communications end.

(b) Master Reception

The slave unit outputs the message length bits and parity bit. When even parity is confirmed, the master unit returns an acknowledgement.

When the parity is not correct, the master unit decides that the message length bits are not correctly received, does not return an acknowledgement, and returns to the waiting state. In this case, the slave unit also returns to the waiting state and communications end.

(6) Data Field

The data field is a field for data transmission/reception to and from the slave unit. The master unit transmits/receives data to and from the slave unit using the data field. The data field is comprised of data bits, a parity bit, and an acknowledge bit.

The data bits consist of eight bits and the MSB is output first.

The parity and acknowledge bits are output following the data bits from the master unit and slave unit, respectively.

Broadcast communications are performed only for the transmission of the master unit. In this case, the acknowledge bit is ignored. Operations in master transmission and master reception are described below.

chanhvinh.com

(a) Master Transmission

The master unit transmits the data bits and parity bit to the slave unit to write data from the master unit to the slave unit. The slave unit receives the data bits and parity bit, and returns an acknowledgement if the parity bit is even and the receive buffer is empty. If the parity bit is odd or the receive buffer is not empty, the slave unit does not accept the corresponding data and does not return an acknowledgement.

When the slave unit does not return an acknowledgement, the master unit retransmits the data. This operation is repeated until either an acknowledgement from the slave unit is detected or the maximum number of data transfer bytes is reached.

When the parity is even and the acknowledgement is output from the slave unit, the master unit transmits the subsequent data if data remains and the maximum number of transfer bytes is not exceeded.

In the case of broadcast communications, the slave unit does not return the acknowledgement, and the master unit transfers data byte by byte.

(b) Master Reception

The master unit outputs synchronous signals corresponding to all data bits to be read from the slave unit.

The slave unit outputs the data bits and parity bit on the bus in accordance with the synchronous signals from the master unit.

The master unit reads the parity bit output from the slave unit, and checks the parity. If the parity is not even, or the receive buffer is not empty, the master unit rejects acceptance of the data, and does not return the acknowledgement. The master unit reads the same data repeatedly if the number of data does not exceed the maximum number of transfer bytes in one frame. If the parity is even and the receive buffer is empty, the master unit accepts data and returns an acknowledgement. The master unit reads in the subsequent data if the number of data does not exceed the maximum number of transfer bytes in one frame.

chanhvinh.com

(7) Parity bit

The parity bit is used to confirm that transfer data occurs with no errors.

The parity bit is added to respective data of the master address, slave address, control, message length, and data bits.

Even parity is used. When the number of bits having the value 1 is odd, the parity bit is 1. When the number of bits having the value 1 is even, the parity bit is 0.

(8) Acknowledge bit

In normal communications (single unit to single unit communications), the acknowledge bit is added in the following positions to confirm that data is correctly accepted.

- At the end of the slave address field
- At the end of the control field
- At the end of the message length field
- At the end of the data field

The acknowledge bit is defined below.

- 0: indicates that the transfer data is acknowledged. (ACK)
- 1: indicates that the transfer data is not acknowledged. (NAK)

Note that the acknowledge bit is ignored in the case of broadcast communications.

(a) Acknowledge bit at the End of the Slave Address Field

The acknowledge bit at the end of the slave address field becomes NAK in the following cases and transfer is stopped.

- When the parity of the master address or slave address bits is incorrect
- When a timing error (an error in bit format) occurs
- When there is no slave unit

chanhvinh.com

(b) Acknowledge bit at the End of the Control Field

The acknowledge bit at the end of the control field becomes NAK in the following cases and transfer is stopped.

- When the parity of the control bits is incorrect
- When the bit 3 of the control bits is 1 (data write) although the slave receive buffer* is not empty
- When the control bits are set to data read (H'3, H'7) although the slave transmit buffer* is empty
- When another unit which locked the slave unit requests H'3, H'6, H'7, H'A, H'B, H'E, or H'F in the control bits although the slave unit has been locked
- When the control bits are the locked address read (H'4, H'5) although the unit is not locked
- When a timing error occurs
- When the control bits are undefined

Note: See section 21.1.3 (1), Slave Status Read (Control Bits: H'0, H'6).

(c) Acknowledge Bit at the End of the Message Length Field

The acknowledge bit at the end of the message length field becomes NAK in the following cases and transfer is stopped.

- When the parity of the message length bits is incorrect
- When a timing error occurs

(d) Acknowledge Bit at the End of the Data Field

The acknowledge bit at the end of the data field becomes NAK in the following cases and transfer is stopped.

- When the parity of the data bits is incorrect*
- When a timing error occurs after the previous transfer of the acknowledge bit
- When the receive buffer becomes full and cannot accept further data*

Note: * In this case, the data field is transferred repeatedly until the number of data reaches the maximum number of transfer bytes if the number of data does not exceed the maximum number of transfer bytes in one frame.

chanhvinh.com

21.1.3 Transfer Data (Data Field Contents)

The data field contents are specified by the control bits.

Table 21.4 Control Bit Contents

Setting Value	Bit 3*[1]	Bit 2	Bit 1	Bit 0	Function*[2]
H'0	0	0	0	0	Reads slave status (SSR)
H'1	0	0	0	1	Undefined.
H'2	0	0	1	0	Undefined.
H'3	0	0	1	1	Reads data and locks
H'4	0	1	0	0	Reads locked address (lower 8 bits)
H'5	0	1	0	1	Reads locked address (upper 4 bits)
H'6	0	1	1	0	Reads slave status (SSR) and unlocks
H'7	0	1	1	1	Reads data
H'8	1	0	0	0	Undefined.
H'9	1	0	0	1	Undefined.
H'A	1	0	1	0	Writes command and locks
H'B	1	0	1	1	Writes data and locks
H'C	1	1	0	0	Undefined.
H'D	1	1	0	1	Undefined.
H'E	1	1	1	0	Writes command
H'F	1	1	1	1	Writes data

Notes: 1. Depending on the value of bit 3 (MSB), the transfer directions of the message length bits in the following message length field and data in the data field vary.
When bit 3 is 1: Data is transferred from the master unit to the slave unit.
When bit 3 is 0: Data is transferred from the slave unit to the master unit.
2. H'3, H'6, H'A, and H'B are control bits to specify lock setting and cancellation.
When the undefined values of H'1, H'2, H'8, H'9, H'C, and H'D are transmitted, the acknowledge signal is not returned.

When the control bits received from another unit which locked are not included in table 21.5, the slave unit which has been locked by the master unit does not accept the control bits and does not return the acknowledge bit.

chanhvinh.com

Table 21.5 Control Field for Locked Slave Unit

Setting Value	Bit 3	Bit 2	Bit 1	Bit 0	Function
H'0	0	0	0	0	Reads slave status
H'4	0	1	0	0	Reads locked address (upper 8 bits)
H'5	0	1	0	1	Reads locked address (lower 4 bits)

(1) Slave Status Read (Control Bits: H'0, H'6)

The master unit can decide the reason the slave unit does not return the acknowledgement (ACK) by reading the slave status (H'0, H'6). The slave status indicates the result of the last communications that the slave unit performed. All slave units can provide slave status information. Figure 21.2 shows the bit configuration of the slave status.

MSB LSB

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bit	Value	Description	
Bit 7, bit 6	00	Mode 0	Indicates the highest mode supported by a unit. *1
	01	Mode 1	
	10	Mode 2	
	11	For future use	
Bit 5	0	Fixed 0	
Bit 4*2	0	Slave transmission halted	
	1	Slave transmission enabled	
Bit 3	0	Fixed 0	
Bit 2	0	Unit is unlocked	
	1	Unit is locked	
Bit 1*3	0	Slave receive buffer is empty	
	1	Slave receive buffer is not empty	
Bit 0*4	0	Slave transmit buffer is empty	
	1	Slave transmit buffer is not empty	

Notes:
1. Since this LSI can support up to mode 2, bits 6 and 7 are fixed to 10.
2. The value of bit 4 can be selected by the STE bit in the IEBus master unit address register 1 (IEAR1).
3. The slave receive buffer is a buffer which is accessed during data write (control bits: H'A, H'B, H'E, H'F).
   In this LSI, the slave receive buffer corresponds to the IEBus receive buffer register (IERB001 to IERB128); and bit 1 is the value of the RXBSY bit in the IEBus receive status register (IERSR).
4. The slave transmit buffer is a buffer which is accessed during data read (control bits: H'3, H'7).
   In this LSI, the slave transmit buffer corresponds to the IEBus transmit buffer register (IETB001 to IETB128) and bit 0 is the value of the SRQ bit in the IEBus general flag registers (IEFLG).

Figure 21.2 Bit Configuration of Slave Status (SSR)

chanhvinh.com

(2) Data Command Transfer (Control Bits: Read (H'3, H'7), Write (H'A, H'B, H'E, H'F))

In the case of data read (H'3, H'7), data in the data buffer of the slave unit is read in the master unit. In the case of data write (H'B or H'F) or command write (H'A or H'E), data received in the slave unit is processed in accordance with the operation specification of the slave unit.

Notes: 1. The user can select data and commands freely in accordance with the system.
2. H'3, H'A, or H'B may lock depending on the communications condition and status.

(3) Locked Address Read (Control Bits: H'4, H'5)

In the case of the locked address read (H'4 or H'5), the address (12 bits) of the master unit, which issues the lock instruction, is configured in bytes as shown in figure 21.3.

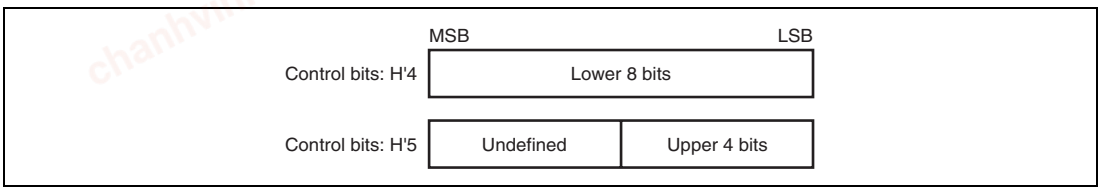


Figure 21.3 Locked Address Configuration

(4) Locking/Unlocking (Control Bits: Setting (H'3, H'A, H'B), Cancellation: (H'6))

The lock function is used for message transfer over multiple communications frames. A locked unit receives data only from the unit which locked it.

Locking and unlocking are described below.

(a) Locking

When an acknowledge bit of 0 in the message length field is transmitted/received with the control bits (H'3, H'A, H'B) indicating the lock operation, and then the communications frame is completed before completion of data transmission/reception for the number of bytes specified by the message length bits, the slave unit is locked by the master unit. In this case, the bit (bit 2) relevant to locking in the byte data indicating the slave status is set to 1.

Lock is set only when the number of data exceeds the maximum number of transfer bytes in one frame. Lock is not set by other error terminations.

chanhvinh.com

(b) Unlocking

When the control bits indicate the lock (H'3, H'A, or H'B) or unlock (H'6) operation and the byte data for the number of bytes specified by the message length bits are transmitted/received in a single communications frame, the slave unit is unlocked by the master unit. In this case, the bit (bit 2) relevant to locking in the byte indicating the slave status is cleared to 0.

Note that locking and unlocking are not done in broadcast communications.

Note: * There are three ways to cause a locked unit to unlock itself.
- Perform a power-on reset
- Put the unit in deep standby mode
- Issue an unlock command through the IEBus command register (IECMR)

Note that the LCK flag in IEFLG can be used to check whether the unit is locked or unlocked.

21.1.4 Bit Format

Figure 21.4 shows the bit format (conceptual diagram) configuring the IEBus communications frame.

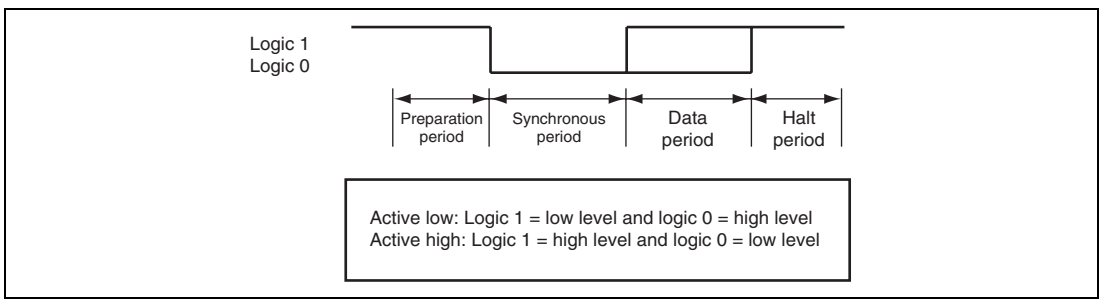


Figure 21.4 IEBus Bit Format (Conceptual Diagram)

Each period of the bit format for use of active high signals is described below.

- Preparation period: first logic 1 period (high level)
- Synchronous period: subsequent logic 0 period (low level)
- Data period: period indicating bit value (logic 1: high level, logic 0: low level)
- Halt period: last logic 1 period (high level)

chanhvinh.com

For use of active low signals, levels are reversed from the active high signals.

The synchronous and data periods have approximately the same length.

The IEBus is synchronized bit by bit. The specifications for the time of all bits and the periods allocated to the bits differ depending on the type of transfer bits and the unit (master or slave unit).

21.1.5 Configuration

Figure 21.5 shows the entire block configuration and table 21.6 lists the functions of each block.

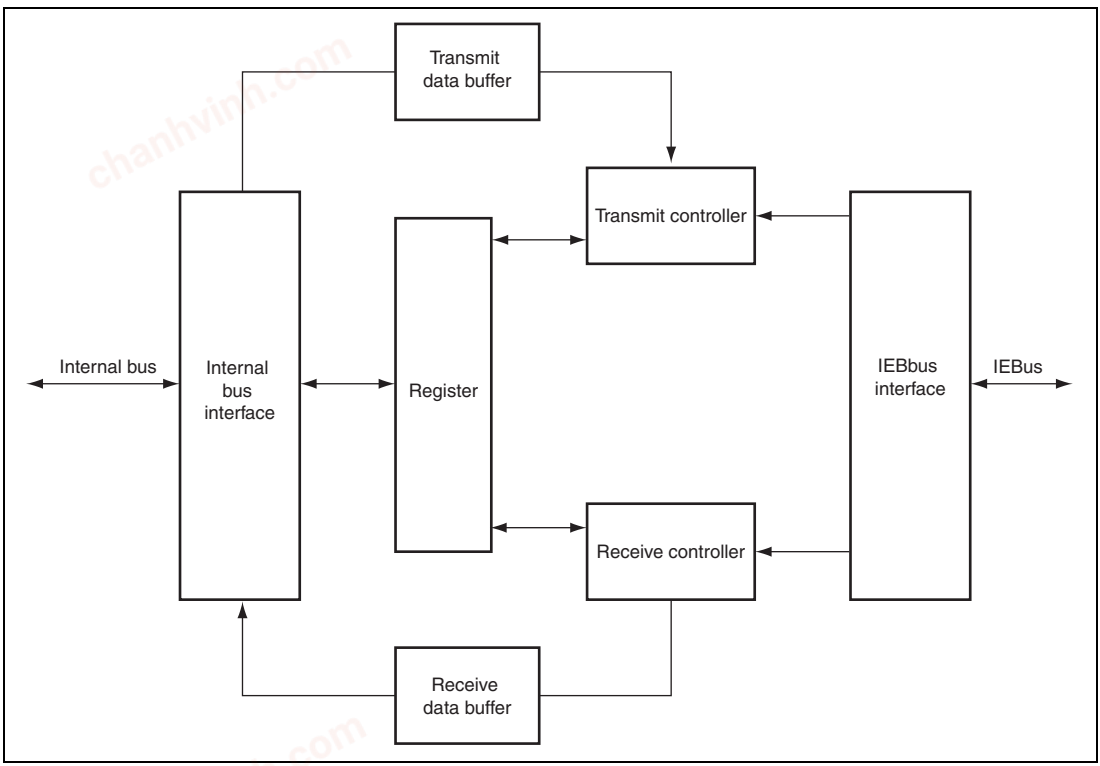


Figure 21.5 Block Diagram

chanhvinh.com

Table 21.6 Functions of Each Block

Block	Function
Internal bus interface	Internal bus interface
	• Data width: 8 bits • Register access
IEBus interface	Interface conforms to IEBus specifications
	• Outputs data from transmit controller to IEBus in IEBus specification bit format • Picks out frame data in IEBus specification bit format to transfer to receive controller
Register	Control register
	• Register to control this module • Readable/writable from internal bus
Transmit controller	Transmits data in transmit buffer to IEBus
	• Generates transmit frame combining header information in register and data in transmit buffer to transmits • Detects transmit error
Receive controller	Stores data from IEBus in receive buffer
	• Stores header information and data in received frame in register and receive buffer, respectively • Detects receive error
Transmit data buffer	Buffer for data transmission
	• Buffer that stores data to be transmitted to IEBus • Buffer size: 128 bytes
Receive data buffer	Buffer for data reception
	• Buffer that stores data received from IEBus • Buffer size: 128 bytes

21.2 Input/Output Pins

Table 21.7 Pin Configuration

Name	Abbreviation	I/O	Function
IEBus receive data pin	IERxD	Input	Receive data input pin
IEBus transmit data pin	IETxD	Output	Transmit data output pin

chanhvinh.com

21.3 Register Descriptions

Table 21.8 shows the register configuration.

Table 21.8 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
IEBus control register	IECTR	R/W	H'00	H'FFFE F000	8
IEBus command register	IECMR	W	H'00	H'FFFE F001	8
IEBus master control register	IEMCR	R/W	H'00	H'FFFE F002	8
IEBus master unit address register 1	IEAR1	R/W	H'00	H'FFFE F003	8
IEBus master unit address register 2	IEAR2	R/W	H'00	H'FFFE F004	8
IEBus slave address setting register 1	IESA1	R/W	H'00	H'FFFE F005	8
IEBus slave address setting register 2	IESA2	R/W	H'00	H'FFFE F006	8
IEBus transmit message length register	IETBFL	R/W	H'00	H'FFFE F007	8
IEBus reception master address register 1	IEMA1	R	H'00	H'FFFE F009	8
IEBus reception master address register 2	IEMA2	R	H'00	H'FFFE F00A	8
IEBus receive control field register	IERCTL	R	H'00	H'FFFE F00B	8
IEBus receive message length register	IERBFL	R	H'00	H'FFFE F00C	8
IEBus lock address register 1	IELA1	R	H'00	H'FFFE F00E	8
IEBus lock address register 2	IELA2	R	H'00	H'FFFE F00F	8
IEBus general flag register	IEFLG	R	H'00	H'FFFE F010	8
IEBus transmit status register	IETSR	R/(W)*	H'00	H'FFFE F011	8
IEBus transmit interrupt enable register	IEIET	R/W	H'00	H'FFFE F012	8
IEBus receive status register	IERSR	R/(W)*	H'00	H'FFFE F014	8

chanhvinh.com

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
IEBus receive interrupt enable register	IEIER	R/W	H'00	H'FFFE F015	8
IEBus clock select register	IECKSR	R/W	H'01	H'FFFE F018	8
IEBus transmit data buffer registers 001 to 128	IETB001 to IETB128	W	Undefined	H'FFFE F100 to H'FFFE F17F	8
IEBus receive data buffer registers 001 to 128	IERB001 to IERB128	R	Undefined	H'FFFE F200 to H'FFFE F27F	8

Note: * Only 1 can be written to clear the flag.

chanhvinh.com

21.3.1 IEBus Control Register (IECTR)

IECTR is used to control the operation of this module.

Bit:	7	6	5	4	3	2	1	0
	-	IOL	DEE	-	RE	-	-	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R	R/W	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	IOL	0	R/W	Input/Output Level Selects input/output pin level (polarity) for the IERxD and IETxD pins. 0: Pin input/output is set to active low. (Logic 1 is low level and logic 0 is high level.) 1: Pin input/output is set to active high. (Logic 1 is high level and logic 0 is low level.)
5	DEE	0	R/W	Broadcast Receive Error Interrupt Enable If this bit is set to 1, a reception error interrupt occurs when the receive buffer is not in the receive enabled state during broadcast reception (when the RE bit is not set to 1 or the RXBSY flag is set.). At this time, the master address is stored in IEBus reception master address register 1 and 2. While this bit is 0, a reception error interrupt does not occur when the receive buffer is not in the receive enabled state, and the reception stops and enters the wait state. The master address is not saved. 0: A broadcast receive error is not generated up to the control field. 1: A broadcast receive error is generated up to the control field.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
4	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
3	RE	0	R/W	Receive Enable Enables/disables reception. This bit must be set at the initial setting before frame reception. 0: Reception is disabled. 1: Reception is enabled.
2 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

21.3.2 IEBus Command Register (IECMR)

IECMR issues commands to control communications. Since this register is a write-only register, the read value is undefined.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	CMD		
Initial value:	0	0	0	0	0	0	0	0
R/W:	-	-	-	-	-	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	—	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
2 to 0	CMD	000	W	Command These bits issue a command to control communications. When the CMX flag in IEFLG is set after the command issuance, the command is indicated to be in execution. When the CMX flag becomes 0, the operation state is entered. 000: No operation. Operation is not affected. 001: Unlock (required from other units)*1 010: Requires communications as the master 011: Stops master communications*2 100: Undefined bits*4 101: Requires data transfer from the slave 110: Stops data transfer from the slave*3 111: Undefined bits*4

Notes:
1. Do not execute this command in slave communications.
2. This command is valid during master communications (MRQ = 1). In other states, this command issuance is ignored. If this command is issued in master communications, the communications controller immediately enters the wait state. At this time, the issued master transmission request ends (MRQ = 0).
3. This command is valid during slave communications (SRQ = 1). In other states, this command issuance is ignored. Once this command is issued in slave transmission, the SRQ flag is 0 before slave transmission. Therefore, a transmit request from the master is not responded to. If a transmit request is issued during slave transmission, the transmission stops and the wait state is entered (SRQ = 0).
4. Undefined bits. Issuing this command does not affect operation.

chanhvinh.com

21.3.3 IEBus Master Control Register (IEMCR)

IEMCR sets the communication conditions for master communications.

Bit:	7	6	5	4	3	2	1	0
	SS	RN			CTL*1			
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	SS	0	R/W	Broadcast/Normal Communications Select Selects broadcast or normal communications for master communications. 0: Broadcast communications for master communications 1: Normal communications for master communications
6 to 4	RN	000	R/W	Retransmission Counts Set the number of times retransmission is done when arbitration is lost in master communications. If arbitration is lost, the TXEAL flag in IETSR is set and transmission ends. 000: 0 001: 1 010: 2 011: 3 100: 4 101: 5 110: 6 111: 7

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	CTL*[1]	0000	R/W	Control Set the control bits in the control field for master transmission. 0000: Reads slave status 0001: Undefined*[3] 0010: Undefined*[3] 0011: Reads data and locks*[2] 0100: Reads locked address (lower 8 bits) 0101: Reads locked address (upper 4 bits) 0110: Reads slave status and unlocks*[2] 0111: Reads data 1000: Undefined*[3] 1001: Undefined*[3] 1010: Writes command and locks*[2] 1011: Writes data and locks*[2] 1100: Undefined*[3] 1101: Undefined*[3] 1110: Writes command 1111: Writes data

Notes: 1. CTL3 decides the data transfer direction of the message length bits in the message length field and data bits in the data field:
CTL3 = 1: Transfer is from master unit to slave unit
CTL3 = 0: Transfer is from slave unit to master unit
2. Control bits to lock and unlock
3. Setting prohibited.

chanhvinh.com

21.3.4 IEBus Master Unit Address Register 1 (IEAR1)

IEAR1 sets the lower four bits of the master unit address and communications mode. In master communications, the master unit address becomes the master address field value. In slave communications, the master unit address is compared with the received slave address field.

Bit:	7	6	5	4	3	2	1	0
	IARL4				IMD		-	STE
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/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	IARL4	0000	R/W	Lower 4 Bits of IEBus Master Unit Address Set the lower 4 bits of the master unit address. This register becomes the master address field value. In slave communications, the master unit address is compared with the received slave address field.
3, 2	IMD	00	R/W	IEBus Communications Mode Set IEBus communications mode. 00: Communications mode 0 01: Communications mode 1 10: Communications mode 2 11: Setting prohibited
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	STE	0	R/W	Slave Transmission Setting Sets bit 4 in the slave status register. Transmitting the slave status register informs the master unit that the slave transmission enabled state is entered by setting this bit to 1. Note that this bit only sets the slave status register value and does not directly affect slave transmission. 0: Bit 4 in the slave status register is 0 (slave transmission stop state) 1: Bit 4 in the slave status register is 1 (slave transmission enabled state)

chanhvinh.com

21.3.5 IEBus Master Unit Address Register 2 (IEAR2)

IEAR2 sets the upper eight bits of the master unit address. In master communications, this register becomes the master address field value. In slave communications, this register is compared with the received slave address field.

Bit:	7	6	5	4	3	2	1	0
	IARU8							
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	IARU8	All 0	R/W	Upper 8 Bits of IEBus Master Unit Address Set the upper 8 bits of the master unit address. This register becomes the master address field value. In slave communications, the master unit address is compared with the received slave address field.

21.3.6 IEBus Slave Address Setting Register 1 (IESA1)

IESA1 sets the lower four bits of the communications destination slave unit address.

Bit:	7	6	5	4	3	2	1	0
	ISAL4				-	-	-	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	ISAL4	0000	R/W	Lower 4 Bits of IEBus Slave Address These bits set the lower 4 bits of the communication destination slave unit address.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

21.3.7 IEBus Slave Address Setting Register 2 (IESA2)

IESA2 sets the upper eight bits of the communications destination slave unit address.

Bit:	7	6	5	4	3	2	1	0
	ISAU8							
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	ISAU8	All 0	R/W	Upper 8 Bits of IEBus Slave Address Set upper 8 bits of the communications destination slave unit address

chanhvinh.com

21.3.8 IEBus Transmit Message Length Register (IETBFL)

IETBFL sets the message length for master or slave transmission.

Bit:	7	6	5	4	3	2	1	0
	IBFL							
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	IBFL	All 0	R/W	Transmit Message Length Set the message length for master transmission. Set the message length that does not exceed the maximum transmit bytes in communications mode. H'01: 1 byte H'02: 2 bytes : H'7F: 127 bytes H'80: 128 bytes H'81: Undefined* : H'FF: Undefined* H'00: Undefined*

Note: * Setting prohibited

chanhvinh.com

21.3.9 IEBus Reception Master Address Register 1 (IEMA1)

IEMA1 indicates the lower four bits of the communication destination master unit address in slave/broadcast reception.

Bit:	7	6	5	4	3	2	1	0
	IMAL4				-	-	-	-
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	IMAL4	0000	R	Lower Four Bits of IEBus Reception Master Address Indicates the lower four bits of the communication destination master unit address in slave/broadcast reception. This register is enabled when slave/broadcast reception starts, and the contents are changed at the time of setting the RXS flag. If a broadcast receive error interrupt is selected by the DEE bit in IECTR and the receive buffer is not in the receive enabled state at control field reception, a receive error interrupt is generated and the lower four bits of the master address are stored in IEMA1.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

21.3.10 IEBus Reception Master Address Register 2 (IEMA2)

IEMA2 indicates the upper eight bits of the communications destination master unit address in slave/broadcast reception. This register is enabled when slave/broadcast reception starts, and the contents are changed at the time of setting the RXS flag in IERSR.

If a broadcast receive error interrupt is selected with the DEE bit in IECTR and the receive buffer is not in the receive enabled state at control field reception, a receive error interrupt is generated and the upper eight bits of the master address are stored in IEMA2. This register cannot be modified.

Bit:	7	6	5	4	3	2	1	0
	IMAU8							
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	IMAU8	All 0	R	Upper Eight Bits of IEBus Reception Master Address Indicates the upper eight bits of the communications destination master unit address in slave/broadcast reception. This register is enabled when slave/broadcast reception starts, and the contents are changed at the time of setting the RXS flag. If a broadcast receive error interrupt is selected by the DEE bit in IECTR and the receive buffer is not in the receive enabled state at control field reception, a receive error interrupt is generated and the upper eight bits of the master address are stored in IEMA2.

chanhvinh.com

21.3.11 IEBus Receive Control Field Register (IERCTL)

IERCTL indicates the control field value in slave/broadcast reception. This register is enabled when slave/broadcast receive starts, and the contents are changed at the time of setting the RXS flag in IERSR. This register cannot be modified.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	RCTL			
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. The write value should always be 0.
3 to 0	RCTL	0000	R	IEBus Receive Control Field Indicates the control field value in slave/broadcast reception. This register is enabled when slave/broadcast reception starts, and the contents are changed at the time of setting the RXS flag.

chanhvinh.com

21.3.12 IEBus Receive Message Length Register (IERBFL)

IERBFL indicates the message length field in slave/broadcast reception. This register is enabled when slave/broadcast receive starts, and the contents are changed at the time of setting the RXS flag in IERSR.

This register cannot be modified.

Bit:	7	6	5	4	3	2	1	0
	RBFL							
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	RBFL	All 0	R	IEBus Receive Message Length Indicates the contents of the message length field in slave/broadcast reception.

21.3.13 IEBus Lock Address Register 1 (IELA1)

IELA1 specifies the lower eight bits of a locked address when a unit is locked.

Bit:	7	6	5	4	3	2	1	0
	ILAL8							
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	ILAL8	All 0	R	Lower Eight Bits of IEBus Lock Address Indicates the lower eight bits of the master unit address when a unit is locked. These bits are valid only when the LCK bit in IEFLG is set.

chanhvinh.com

21.3.14 IEBus Lock Address Register 2 (IELA2)

IELA2 specifies the upper four bits of a locked address when a unit is locked.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	ILAU4			
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. The write value should always be 0.
3 to 0	ILAU4	0000	R	Upper Four Bits of IEBus Locked Address Stores the upper four bits of the master unit address when a unit is locked. These bits are valid only when the LCK bit in IEFLG is set

chanhvinh.com

21.3.15 IEBus General Flag Register (IEFLG)

IEFLG indicates the command execution status, lock status and slave address match, and broadcast reception detection.

Bit:	7	6	5	4	3	2	1	0
	CMX	MRQ	SRQ	SRE	LCK	-	RSS	GG
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	CMX	0	R	Command Execution Status Indicates the command execution status. 0: Command execution is completed 1: A command is being executed [Setting condition] • When a master communications request or slave transmit request command is issued while the MRQ, SRQ, or SRE flag is set [Clearing condition] • When a command execution has been completed
6	MRQ	0	R	Master Communications Request Indicates whether the unit is in the communications request state as a master unit. 0: The unit is not in the communications request state as a master unit 1: The unit is in the communications request state as a master unit [Setting condition] • When the CMX flag is cleared to 0 after the master communications request command is issued [Clearing condition] • When the master communications have been completed

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
5	SRQ	0	R	Slave Transmission Request Indicates whether the unit is in the transmit request state as a slave unit. 0: The unit is not in the transmit request state as a slave unit 1: The unit is in the transmit request state as a slave unit [Setting condition] • When the CMX flag is cleared to 0 after the slave transmit request command is issued. [Clearing condition] • When a slave transmission has been completed.
4	SRE	0	R	Slave Receive Status Indicates the execution status in slave/broadcast reception. 0: Slave/broadcast reception is not being executed 1: Slave/broadcast reception is being executed [Setting condition] • When the slave/broadcast reception is started while the RE bit in IECTR is set to 1. [Clearing condition] • When the slave/broadcast reception has been completed.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
3	LCK	0	R	Lock Status Indication Set to 1 when a unit is locked by a lock request from the master unit. IELA1 and IELA2 values are valid only when this flag is set to 1. 0: A unit is unlocked 1: A unit is locked [Setting condition] • When data for the number of bytes specified by the message length is not received after the control bits that make the unit locked are received from the master unit. (The LCK flag is set to 1 only when the message length exceeds the maximum number of transfer bytes in one frame. This flag is not set by completion of other errors.) [Clearing condition] • When an unlock condition is satisfied or when an unlock command is issued.
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	RSS	0	R	Receive Broadcast Bit Status Indicates the received broadcast bit value. This flag is valid when the slave/broadcast reception is started. (This flag is changed at the time of setting the RXS flag.) The previous value remains unchanged until the next slave/broadcast reception is started. 0: Received broadcast bit is 0 1: Received broadcast bit is 1

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
0	GG	0	R	General Broadcast Reception Acknowledgement Set to 1 when the slave address is acknowledged as H'FFF in broadcast reception. Like the receive broadcast bit, this flag is valid when the slave/broadcast reception is started. (This flag is changed at the time of setting the RXS flag in IERSR.) The previous value remains unchanged until the next slave/broadcast reception is started. This flag is cleared to 0 in slave normal reception. 0: (1) A unit is in slave reception (2) When H'FFF is not acknowledged in the slave address field in broadcast reception 1: When H'FFF is acknowledged in the slave address field in broadcast reception

21.3.16 IEBus Transmit Status Register (IETSR)

IETSR detects events such as transmit start, transmit normal completion, and transmit error end. Each status flag in IETSR corresponds to a bit in the IEBus transmit interrupt enable register (IEIET) that enables or disables each interrupt. This register is cleared by writing 1 to each bit.

Bit:	7	6	5	4	3	2	1	0
	-	TXS	TXF	-	TXEAL	TXETTME	TXERO	TXEACK
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/(W)*	R/(W)*	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	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	TXS	0	R/(W)*	Transmit Start Indicates that this module starts transmission. [Setting condition] • During master transmission, the arbitration is won and the master address field transmission is completed [Clearing condition] • When 1 is written
5	TXF	0	R/(W)*	Transmit Normal Completion Indicates that data for the number of bytes specified by the message length bits has been transmitted with no error. [Setting condition] • When data for the number of bytes specified by the message length bits has been transmitted normally [Clearing condition] • When 1 is written
4	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
3	TXEAL	0	R/(W)*	Arbitration Loss This module retransmits from the start bit for the number of times specified by the RN bit in IEMCR if the arbitration has been lost in master communications. If the arbitration has been lost for the specified number of times, the TXEAL is set to enter the wait state. If the arbitration has been won within retransmit for the specified number of times, this flag is not set to 1. This flag is set only when the arbitration has been lost and the wait state is entered. [Setting condition] • When the arbitration has been lost during data transmission and the transmission has been terminated [Clearing condition] • When 1 is written

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
2	TXETTME	0	R/(W)*	Transmit Timing Error Set to 1 if data is not transmitted at the timing specified by the IEBus protocol during data transmission. This module sets this bit and enters the wait state. [Setting condition] • When a timing error occurs during data transmission [Clearing condition] • When 1 is written
1	TXERO	0	R/(W)*	Overflow of Maximum Number of Transmit Bytes in One Frame Indicates that the maximum number of bytes defined by the communications mode have been transmitted because a NAK has been received from the receive unit and retransmit has been performed, or that transmission has not been completed because the message length value exceeds the maximum number of transmit bytes in one frame. This module sets this bit and enters the wait state. [Setting condition] • When the transmit has not been completed although the maximum number of bytes defined by the communications mode have been transmitted [Clearing condition] • When 1 is written

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
0	TXEACK	0	R/(W)*	Acknowledge Bit Status Indicates the data received in the acknowledge bit of the data field. • Acknowledge bit other than in the data field This module terminates the transmission and enters the wait state if a NAK is received. In this case, this bit is set to 1. • Acknowledge bit in the data field This module retransmits data up to the maximum number of bytes defined by the communications mode until an ACK is received from the receive unit if a NAK is received from the receive unit during data field transmission. In this case, when an ACK is received from the receive unit during retransmission, this flag is not set and transmission will be continued. When transmission is terminated without receiving an ACK, this flag is set to 1. Note: This flag is invalid in broadcast communications. [Setting condition] • When the acknowledge bit of 1 (NAK) is detected [Clearing condition] • When 1 is written

Note: * only 1 can be written to clear the flag.

chanhvinh.com

21.3.17 IEBus Transmit Interrupt Enable Register (IEIET)

IEIET enables/disables interrupts for sources such as transmit start, transmit normal completion, and transmit error completion in IETSR.

Bit:	7	6	5	4	3	2	1	0
	-	TXSE	TXFE	-	TXEALE	TXE TTMEE	TXEROE	TXE ACKE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R	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	TXSE	0	R/W	Transmit Start Interrupt Enable Enables/disables a transmit start (TXS) interrupt. 0: Disables a transmit start (TXS) interrupt 1: Enables a transmit start (TXS) interrupt
5	TXFE	0	R/W	Transmit Normal Completion Interrupt Enable Enables/disables a transmit normal completion (TXF) interrupt. 0: Disables a transmit normal completion (TXF) interrupt 1: Enables a transmit normal completion (TXF) interrupt
4	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
3	TXEALE	0	R/W	Arbitration Loss Interrupt Enable Enables/disables an arbitration loss (TXEAL) interrupt. 0: Disables an arbitration loss (TXEAL) interrupt 1: Enables an arbitration loss (TXEAL) interrupt

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
2	TXETTMEE	0	R/W	Transmit Timing Error Interrupt Enable Enables/disables a transmit timing error (TXETTMEE) interrupt. 0: Disables a transmit timing error (TXETTMEE) interrupt 1: Enables a transmit timing error (TXETTMEE) interrupt
1	TXEROE	0	R/W	Overflow of Maximum Number of Transmit Bytes in One Frame Interrupt Enable Enables/disables an overflow of the maximum number of transmit bytes in one frame (TXEROE) interrupt. 0: Disables an overflow of the maximum number of transmit bytes in one frame (TXEROE) interrupt 1: Enables an overflow of the maximum number of transmit bytes in one frame (TXEROE) interrupt
0	TXEACKE	0	R/W	Acknowledge Bit Interrupt Enable Enables/disables an acknowledge bit (TXEACKE) interrupt. 0: Disables an acknowledge bit (TXEACKE) interrupt 1: Enables an acknowledge bit (TXEACKE) interrupt

chanhvinh.com

21.3.18 IEBus Receive Status Register (IERSR)

IERSR detects receive busy, receive start, receive normal completion, or receive completion with an error. Each status flag in IERSR corresponds to a bit in the IEIER that enables/disables each interrupt. This register is cleared by writing 1 to each bit.

Bit:	7	6	5	4	3	2	1	0
	RXBSY	RXS	RXF	RXEDE	RXEOVE	RXE RTME	RXEDLE	RXEPE
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	RXBSY	0	R/(W)*	Receive Busy Indicates that the receive data is stored in the receive data buffer (IERB001 to IERB128). Clear this bit after reading out all data. The next receive data cannot be received while this bit is set. [Setting condition] • When all receive data has been written to the receive data buffer. [Clearing condition] • When 1 is written
6	RXS	0	R/(W)*	Receive Start Detection Indicates that this module starts reception. [Setting condition] • When the data from the master unit to message length field has been received correctly in slave reception [Clearing condition] • When 1 is written

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
5	RXF	0	R/(W)*	Receive Normal Completion Indicates that data for the number of bytes specified by the message length bits has been received normally. [Setting condition] • When data for the number of bytes specified by the message length bits has been received normally. [Clearing condition] • When 1 is written
4	RXEDE	0	R/(W)*	Broadcast Receive Error Indicates that data could not be received because the receive buffer is not in the receive enabled state (when the RE bit is not set to 1 or the RXBSY flag is set.) during receiving control field broadcast reception. This bit functions when the DEE bit in IECTR is set to 1. [Setting condition] • When data could not be received during broadcast reception. [Clearing condition] • When 1 is written
3	RXEOVE	0	R/(W)*	Receive Overrun Flag Used to indicate the overrun during data reception. This module sets this flag when this module receives the next byte data while the receive data has not been read (the RXBSY flag is not cleared). If this case, this module assumes that an overrun error has occurred and returns a NAK to the communications destination unit. The communications destination unit retransmits data up to the maximum number of transmit bytes. This module, however, returns a NAK when the RXBSY flag remains set. If the RXBSY flag is cleared to 0, this module returns an ACK, and receives the next data. In broadcast reception, if the RXBSY flag is set during data receive start, this module immediately enters the wait state. This flag becomes enabled only after the receive start flag (RXS) is set. [Setting condition] • When the next byte data is received while the RXBSY flag is not cleared. [Clearing condition] • When 1 is written

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
2	RXERTME	0	R/(W)*	Receive Timing Error Set to 1 if data is not received at the time specified by the IEBus protocol during data reception. This module sets this bit and enters the wait state. This flag is enabled only after the receive start flag (RXS) is set. If this error occurs before the receive start flag (RXS) is set, this module stops communication and enters the wait state. This bit is not set in this case. [Setting condition] • When a timing error occurs during data reception [Clearing condition] • When 1 is written
1	RXEDLE	0	R/(W)*	Overflow of Maximum Number of Receive Bytes in One Frame Indicates that the data reception has not finished within the maximum number of bytes defined by the communications mode because of a parity error or overrun error causing the retransfer of data, or that reception has not been completed because the message length value exceeds the maximum number of receive bytes in one frame. This module sets the RXEDLE flag and enters the wait state. This flag is enabled only after the receive start flag (RXS) is set. If this error occurs before the receive start flag is set, this module stops communication and enters the wait state. This bit is not set in this case. [Setting condition] • When the reception has not been completed within the maximum number of bytes defined by communications mode. [Clearing condition] • When 1 is written

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
0	RXEPE	0	R/(W)*	Parity Error Indicates that a parity error has occurred during data field reception. If a parity error occurs before data field reception, this module immediately enters the wait state and the RXEPE flag is not set. If a parity error occurs when the maximum number of receive bytes in one frame have not been received, the RXEPE flag is not set yet. When a parity error occurs, this module returns a NAK to the communications destination unit via the acknowledge bit. In this case, the communications destination unit continues retransfer up to the maximum number of receive bytes in one frame and if the reception has been completed normally by clearing the parity error, the RXEPE flag is not set. If the parity error is not cleared when the reception is terminated before receiving data for the number of bytes specified by the message length, the RXEPE flag is set. In broadcast reception, if a parity error occurs during data field reception, this module enters the wait state immediately after setting the RXEPE flag. This flag is enabled only after the receive start flag (RXS) is set. If this error occurs before the receive start flag is set, this module stops communication and enters the wait state. This bit is not set in this case. [Setting condition] • When the parity bit of the last data of the data field is not correct after the maximum number of receive bytes have been received [Clearing condition] • When 1 is written

Note: * only 1 can be written to clear the flag.

chanhvinh.com

21.3.19 IEBus Receive Interrupt Enable Register (IEIER)

IEIER enables/disables interrupts for sources such as IERSR receive busy, receive start, receive normal completion, and receive error completion.

Bit:	7	6	5	4	3	2	1	0
	RXBYSE	RXSE	RXFE	RXEDEE	RXE OVEE	RXE RTMEE	RXE DLEE	RXEPEE
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	RXBSYE	0	R/W	Receive Busy Interrupt Enable Enables/disables a receive busy interrupt (RXBSY) 0: Disables a receive busy (RXBSY) interrupt 1: Enables a receive busy (RXBSY) interrupt
6	RXSE	0	R/W	Receive Start Interrupt Enable Enables/disables a receive start (RXS) interrupt 0: Disables a receive start (RXS) interrupt 1: Enables a receive start (RXS) interrupt
5	RXFE	0	R/W	Receive Normal Completion Enable Enables/disables a receive normal completion (RXF) interrupt 0: Disables a receive normal completion (RXF) interrupt 1: Enables a receive normal completion (RXF) interrupt
4	RXEDEE	0	R/W	Broadcast Receive Error Interrupt Enable Enables/disables a broadcast receive error (RXEDE) interrupt 0: Disables a broadcast receive error (RXEDE) interrupt 1: Enables a broadcast receive error (RXEDE) interrupt

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
3	RXEOVEE	0	R/W	Overrun Control Flag Interrupt Enable Enables/disables an overrun control flag (RXEOVE) interrupt 0: Disables an overrun control flag (RXEOVE) interrupt 1: Enables an overrun control flag (RXEOVE) interrupt
2	RXERTMEE	0	R/W	Receive Timing Error Interrupt Enable Enables/disables a receive timing error (RXERTME) interrupt. 0: Disables a receive timing error (RXERTME) interrupt 1: Enables a receive timing error (RXERTME) interrupt
1	RXEDLEE	0	R/W	Overflow of Maximum Number of Receive Bytes in One Frame Interrupt Enable Enables/disables an overflow of the maximum number of receive bytes in one frame (RXEDLE) interrupt 0: Disables an overflow of the maximum number of receive bytes in one frame (RXEDLE) interrupt 1: Enables an overflow of the maximum number of receive bytes in one frame (RXEDLE) interrupt
0	RXEPEE	0	R/W	Parity Error Interrupt Enable Enables/disables a parity error (RXEPE) interrupt 0: Disables a parity error (RXEPE) interrupt 1: Enables a parity error (RXEPE) interrupt

chanhvinh.com

21.3.20 IEBus Clock Selection Register (IECKSR)

IECKSR is a readable/writable 8-bit register that specifies the clock used in this module.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	CKS3	-	CKS[2:0]		
Initial value:	0	0	0	0	0	0	0	1
R/W:	R	R	R	R/W	R	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	CKS3	0	R/W	Input Clock Selection 3*1*2 Specifies the clock for this module 0: Peripheral clock (Pϕ) 1: AUDIO_X1, AUDIO_X2
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	CKS[2:0]	001	R/W	Input Clock Selection 2 to 0*1 Specifies the division ratio of the clock for this module 000: Setting prohibited 001: This module uses the 1/2 divided clock of IEBϕ specified by CKS3 (IEBϕ = 12 MHz, 12.58 MHz). 010: This module uses the 1/3 divided clock of IEBϕ specified by CKS3 (IEBϕ = 18 MHz, 18.87 MHz). 011: This module uses the 1/4 divided clock of IEBϕ specified by CKS3 (IEBϕ = 24 MHz, 25.16 MHz). 100: This module uses the 1/5 divided clock of IEBϕ specified by CKS3 (IEBϕ = 30 MHz, 31.45 MHz). 101: This module uses the 1/6 divided clock of IEBϕ specified by CKS3 (IEBϕ = 36 MHz). 110: Setting prohibited 111: Setting prohibited

Notes: 1. Do not change the setting of CKS3 and CKS[2:0] while IEBus is in transmit/receive operation.
2. When the CKS3 bit is set to 1, be sure to set the MSTP36 bit in STBCR3 to 0. For the setting of STBCR3, see section 33, Power-Down Modes.

chanhvinh.com

21.3.21 IEBus Transmit Data Buffer 001 to 128 (IETB001 to IETB128)

IETB001 to IETB128 are 128-byte (8 × 128) buffers to which data to be transmitted during master transmission is written.

The initial values in IETB001 to IETB128 are undefined.

Bit:	7	6	5	4	3	2	1	0
	TBn							
Initial value:	-	-	-	-	-	-	-	-
R/W:	W*	W*	W*	W*	W*	W*	W*	W*

[Legend]
n = 001 to 128

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	TBn	Undefined	W*	IEBus Transmit Data Buffer Data to be transmitted in the data field during master transmission is written to TB001 to TB128. Data is written starting with TB001 for the start 1-byte data, followed by TB002 and TB003 and so on according to the transmission order, and TB128 stores the last data.

Note: * Writing to these bits during master transmission (MRQ in IEFLG is 1) is prohibited

chanhvinh.com

21.3.22 IEBus Receive Data Buffer 001 to 128 (IERB001 to IERB128)

IERB001 to IERB128 are 128-byte (8 × 128) buffers to which data to be transmitted during slave transmission is written.

The initial values in IERB001 to IERB128 are undefined.

Bit:	7	6	5	4	3	2	1	0
	RBn							
Initial value:	-	-	-	-	-	-	-	-
R/W:	R*	R*	R*	R*	R*	R*	R*	R*

[Legend]
n = 001 to 128

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	RBn	Undefined	R*	IEBus Receive Data Buffer Data in RB001 to RB128 can be read when the RXBSY bit in the IEBus receive status register (IERSR) is set to 1. Data read from RB001 to RB128 is the field data during slave receive. Receive data is written starting with RB001 for the start 1-byte data, followed by RB002 and RB003 and so on, and RB128 stores the last data.

Note: * Reading these bits during slave reception (SRE in IEFLG is 1 and RXBSY in IERSR is 0) is prohibited. (Read value is undefined.)

chanhvinh.com

21.4 Data Format

21.4.1 Transmission Format

Figure 21.6 shows the relationship between the transfer format and each register during the IEBus data transmission.

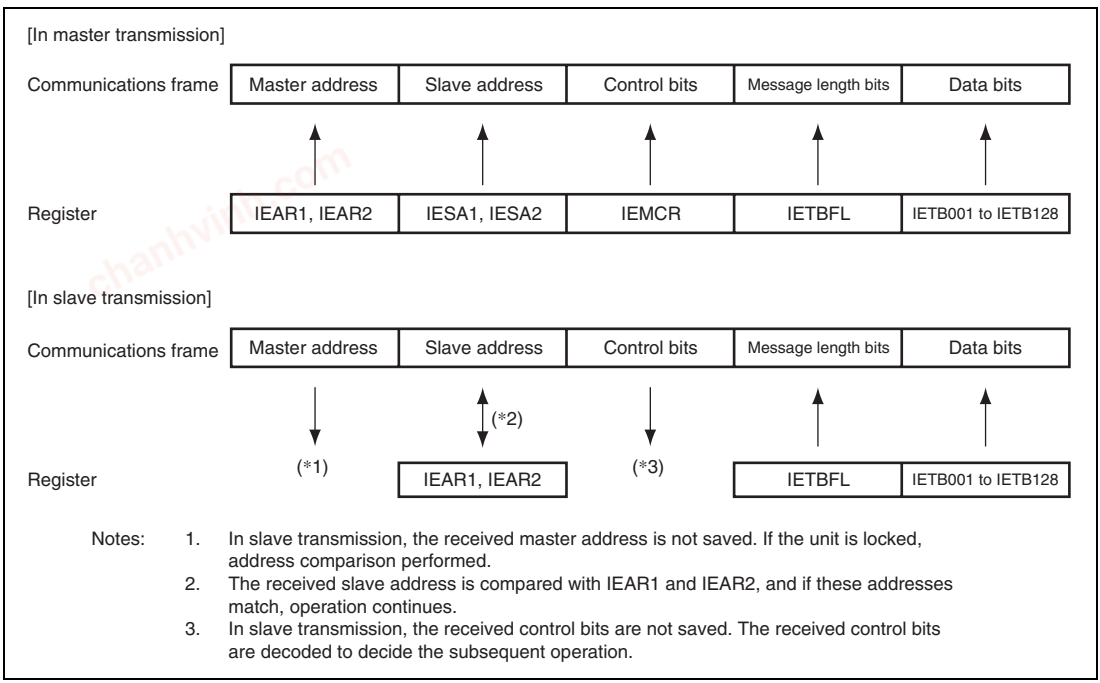


Figure 21.6 Relationship between Transfer Format and Each Register during IEBus Data Transmission

chanhvinh.com

21.4.2 Reception Format

Figure 21.7 shows the relationship between the transfer format and each register during the IEBus data reception.

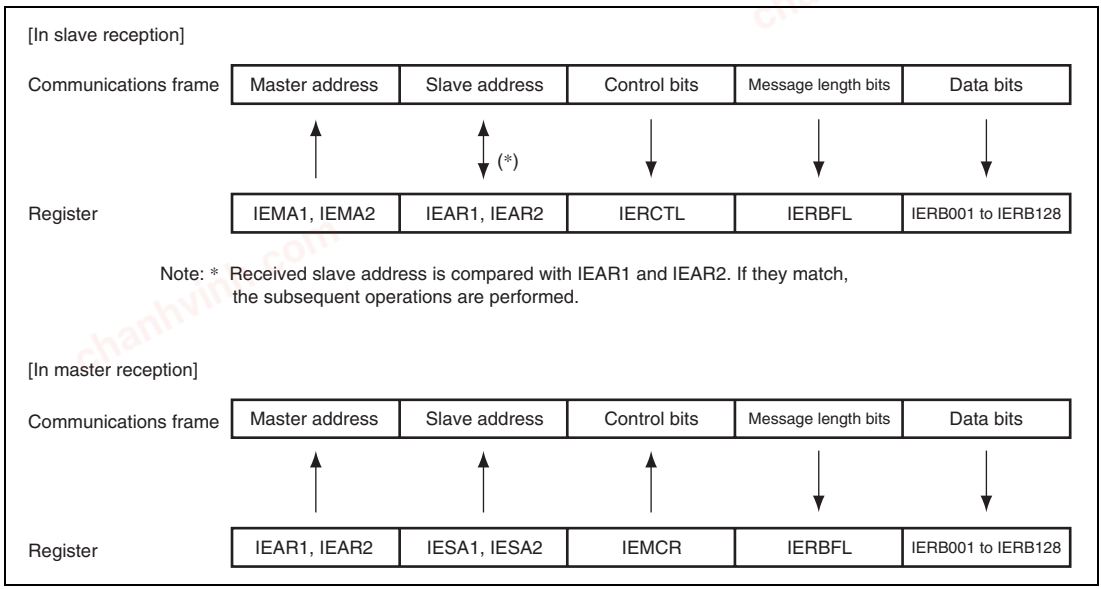


Figure 21.7 Relationship between Transfer Format and Each Register during IEBus Data Reception

chanhvinh.com

21.5 Software Control Flows

21.5.1 Initial Setting

Figure 21.8 shows the flowchart for the initial setting.

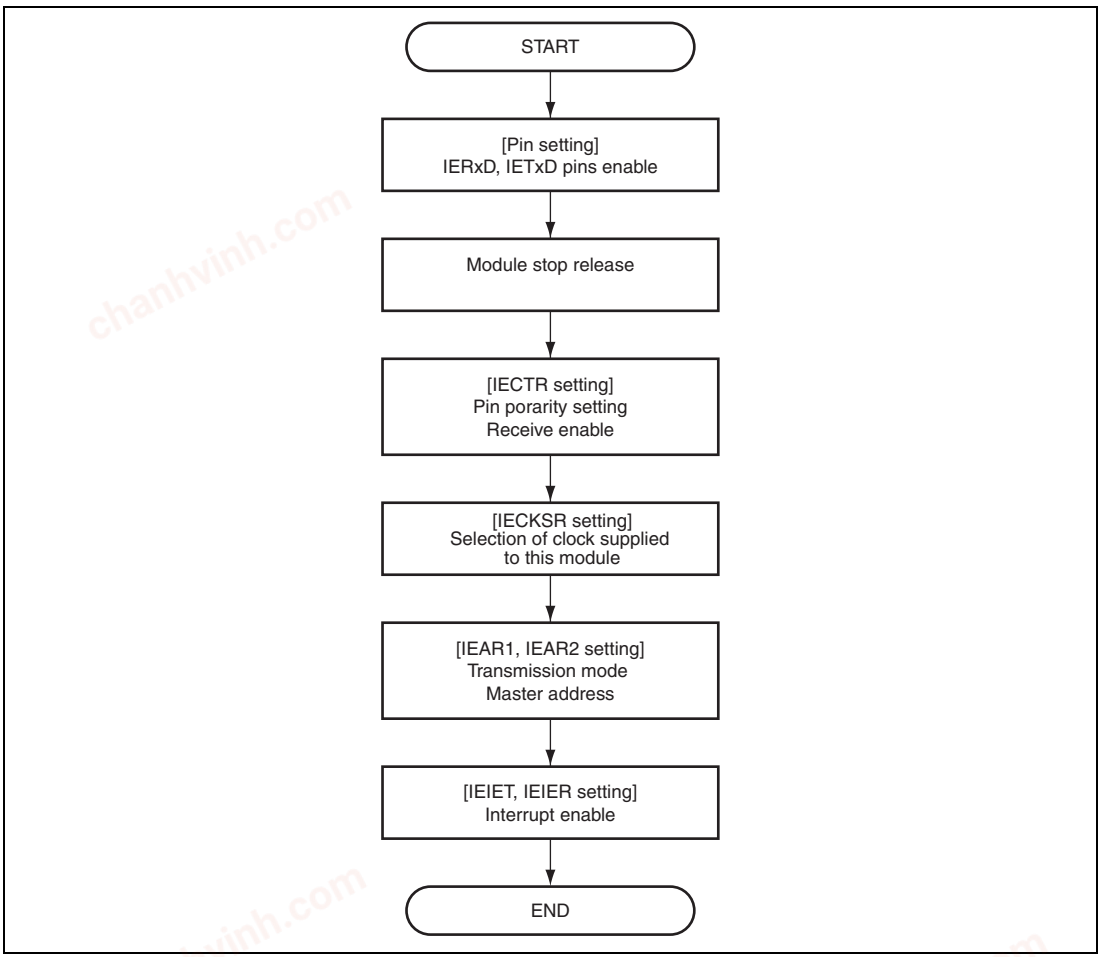


Figure 21.8 Flowchart for Initial Setting

chanhvinh.com

21.5.2 Master Transmission

Figure 21.9 shows the flowchart for master transmission.

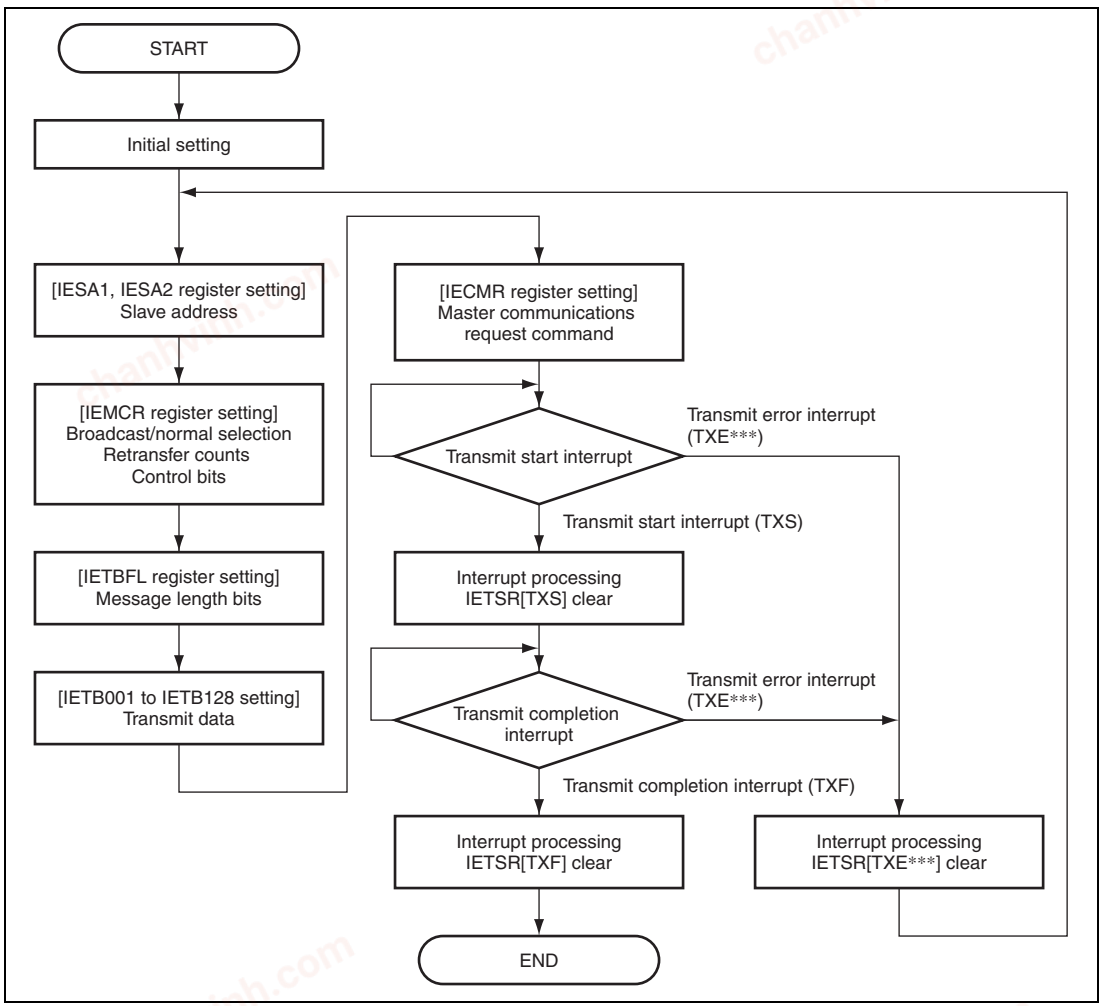


Figure 21.9 Flowchart for Master Transmission

chanhvinh.com

21.5.3 Slave Reception

Figure 21.10 shows the flowchart for slave reception.

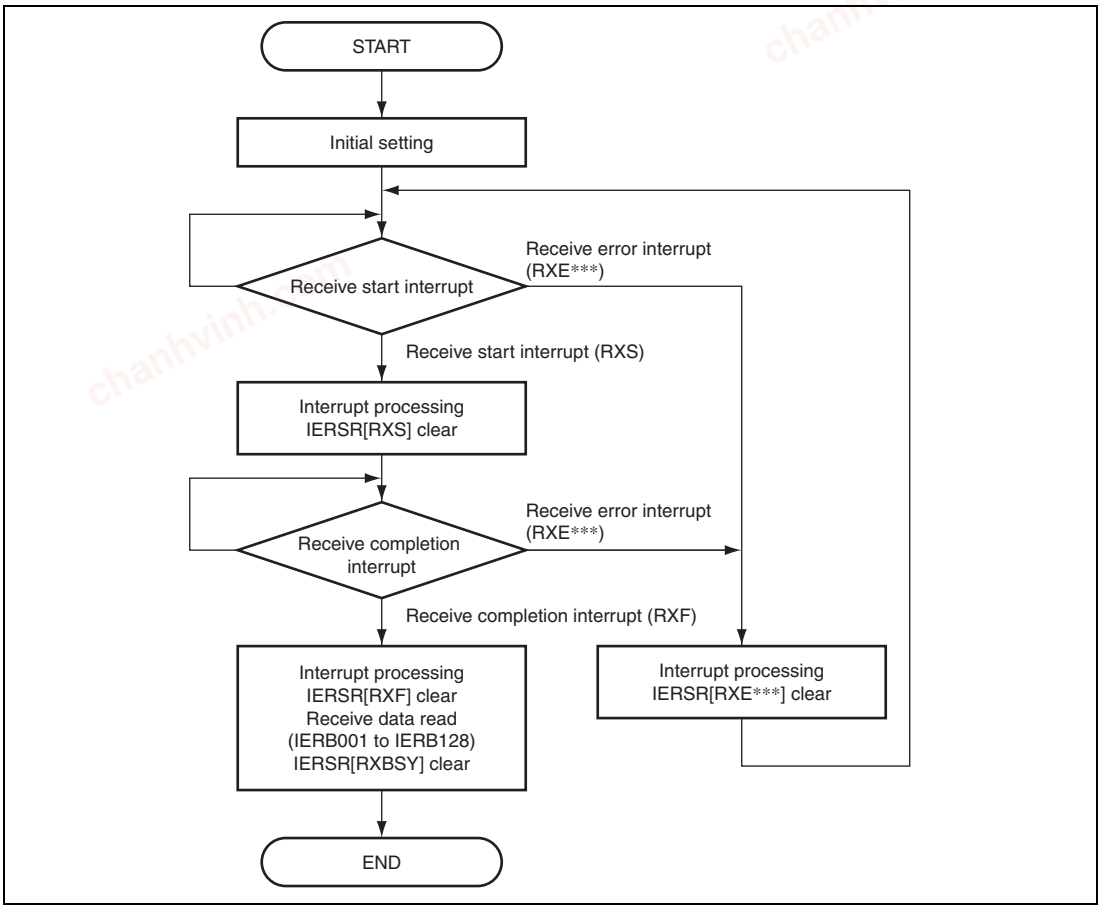


Figure 21.10 Flowchart for Slave Reception

chanhvinh.com

21.5.4 Master Reception

Figure 21.11 shows the flowchart for master reception.

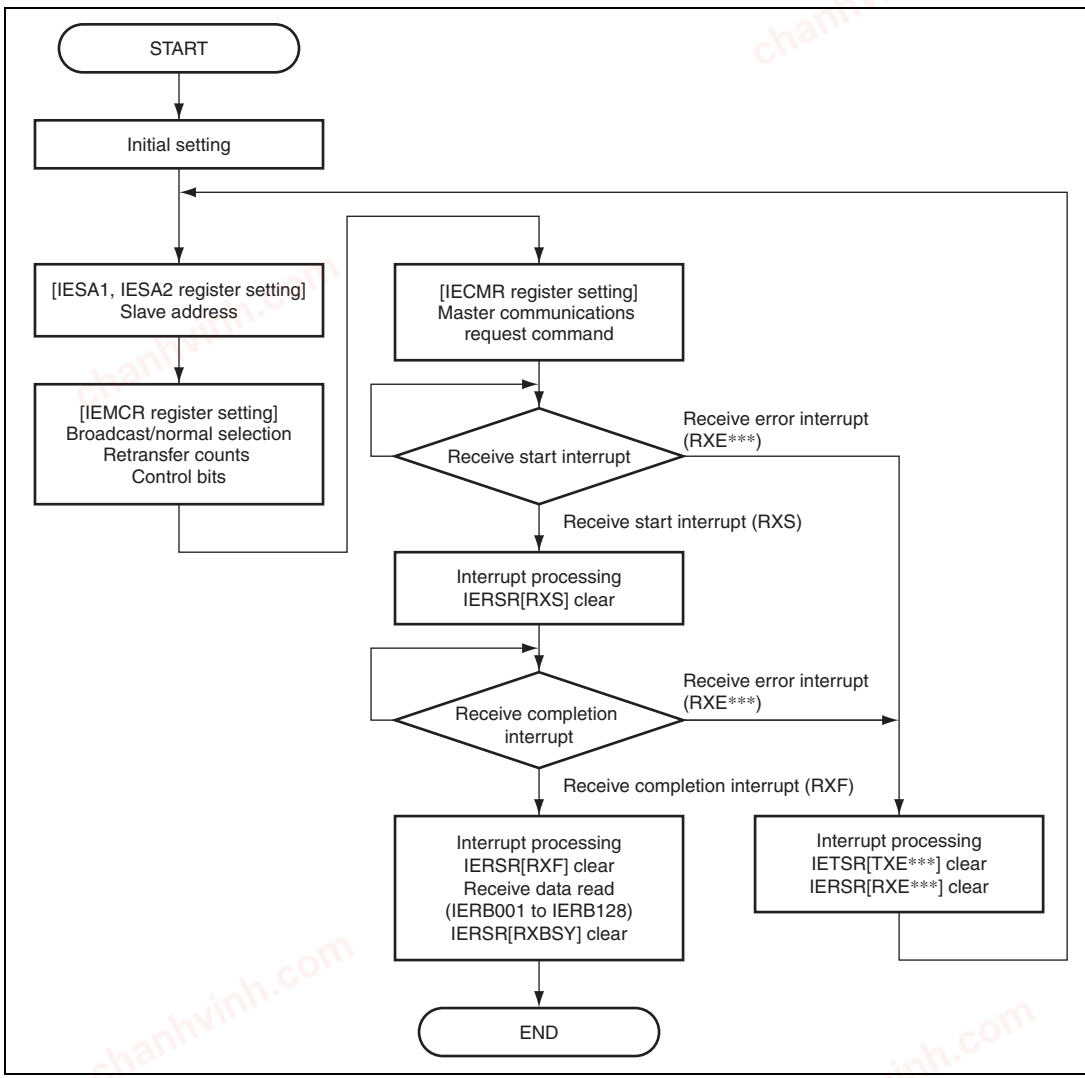


Figure 21.11 Flowchart for Master Reception

chanhvinh.com

21.5.5 Slave Transmission

Figure 21.12 shows the flowchart for slave transmission.

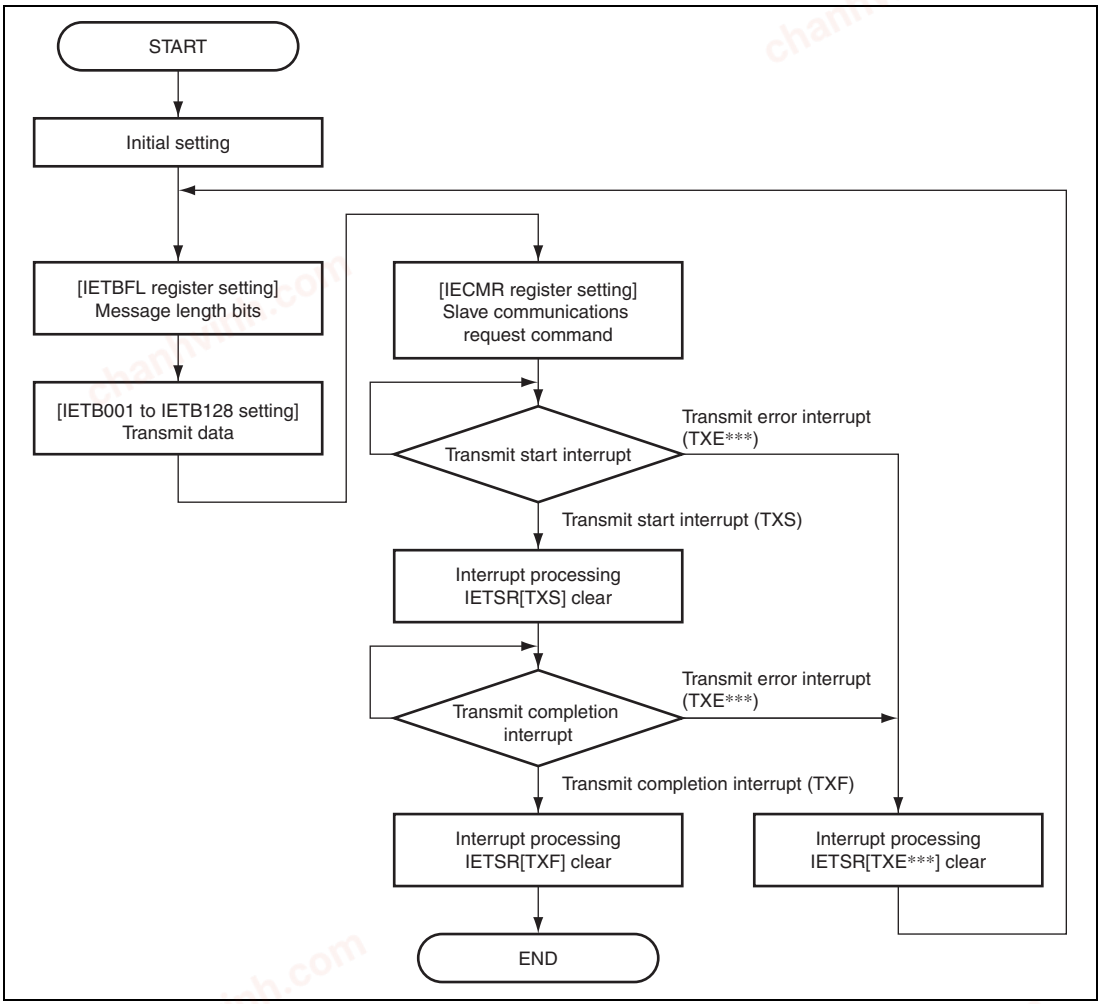


Figure 21.12 Flowchart for Slave Transmission

chanhvinh.com

21.6 Operation Timing

21.6.1 Master Transmit Operation

Figure 21.13 shows the timing for master transmit operation.

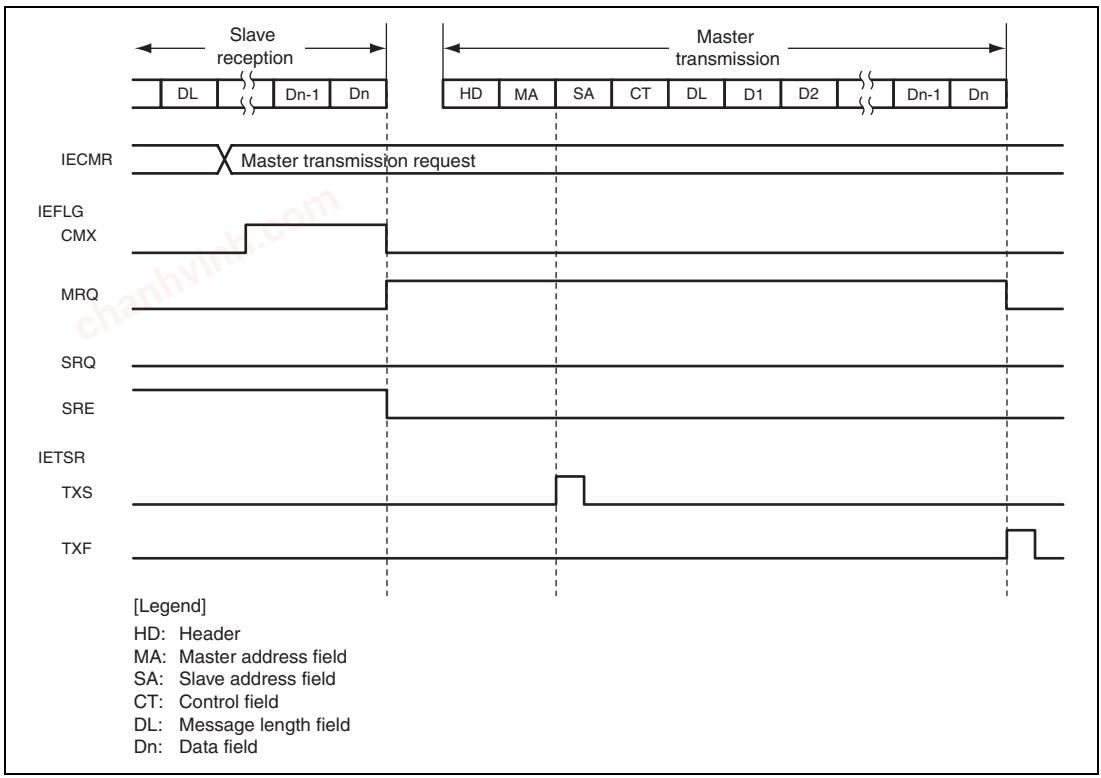


Figure 21.13 Master Transmit Operation Timing

chanhvinh.com

21.6.2 Slave Receive Operation

Figure 21.14 shows the timing for slave receive operation.

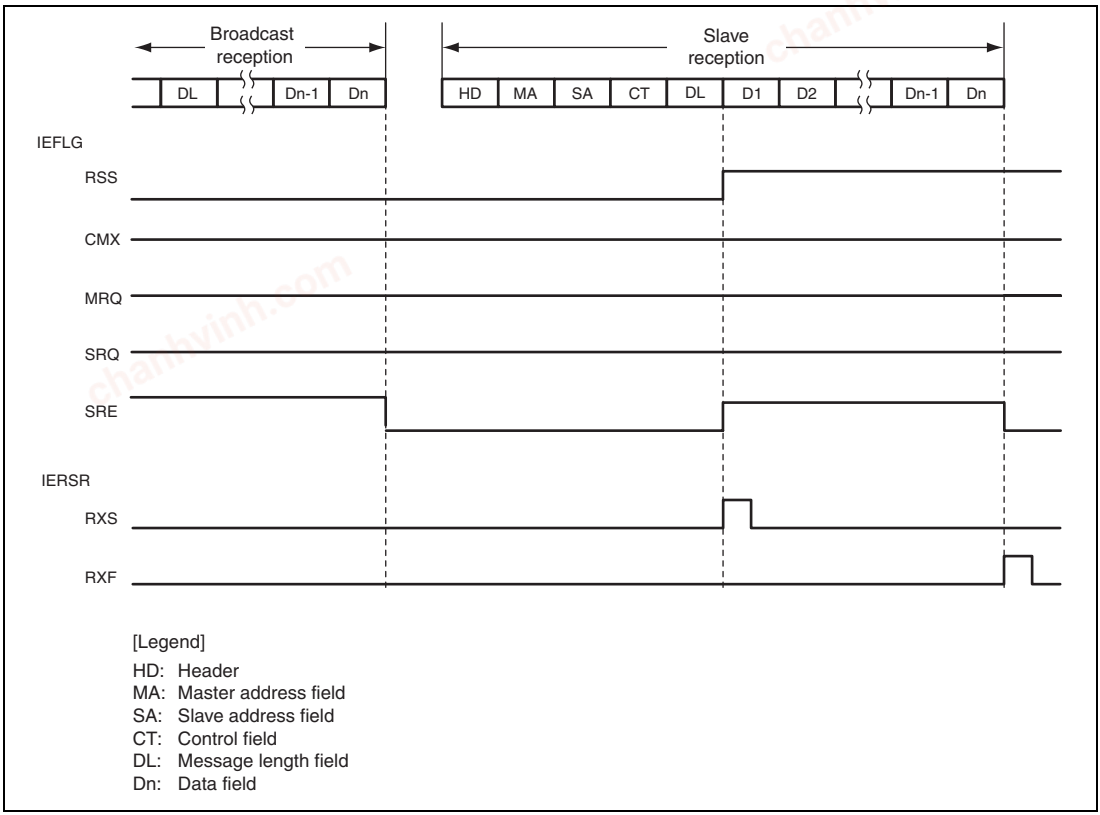


Figure 21.14 Slave Receive Operation Timing

chanhvinh.com

21.6.3 Master Receive Operation

Figure 21.15 shows the timing for master receive operation.

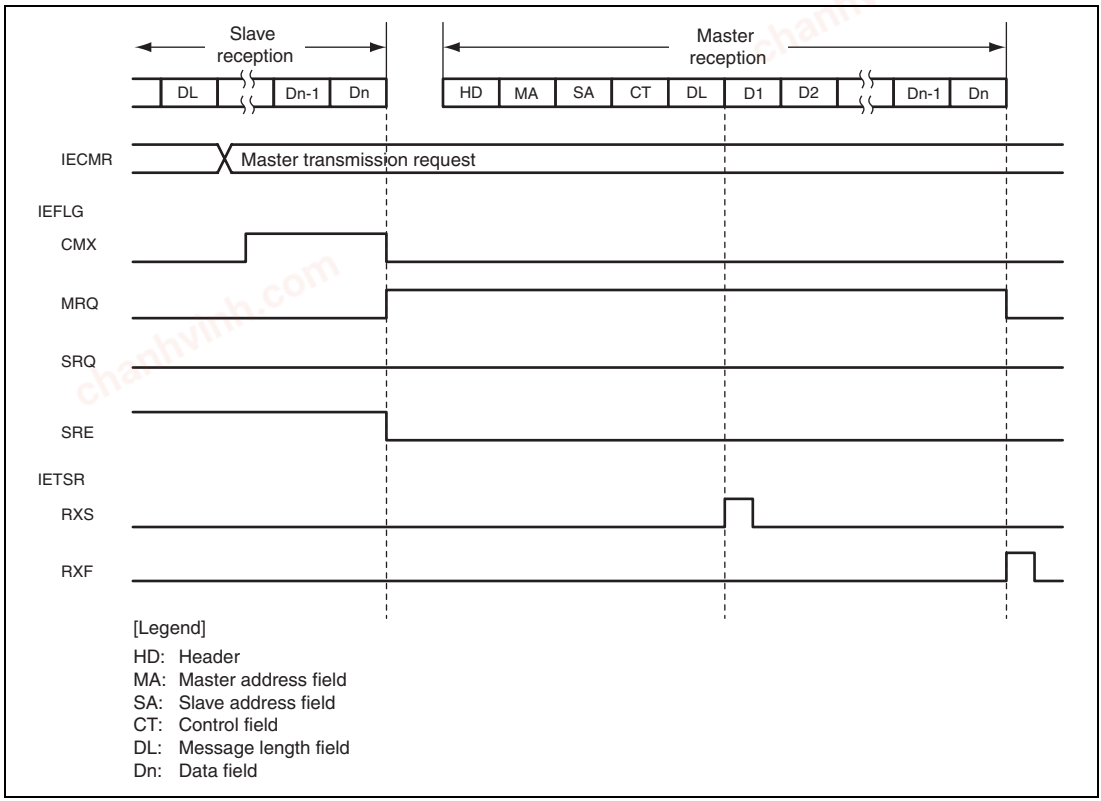


Figure 21.15 Master Receive Operation Timing

chanhvinh.com

21.6.4 Slave Transmit Operation

Figure 21.16 shows the timing for slave transmit operation.

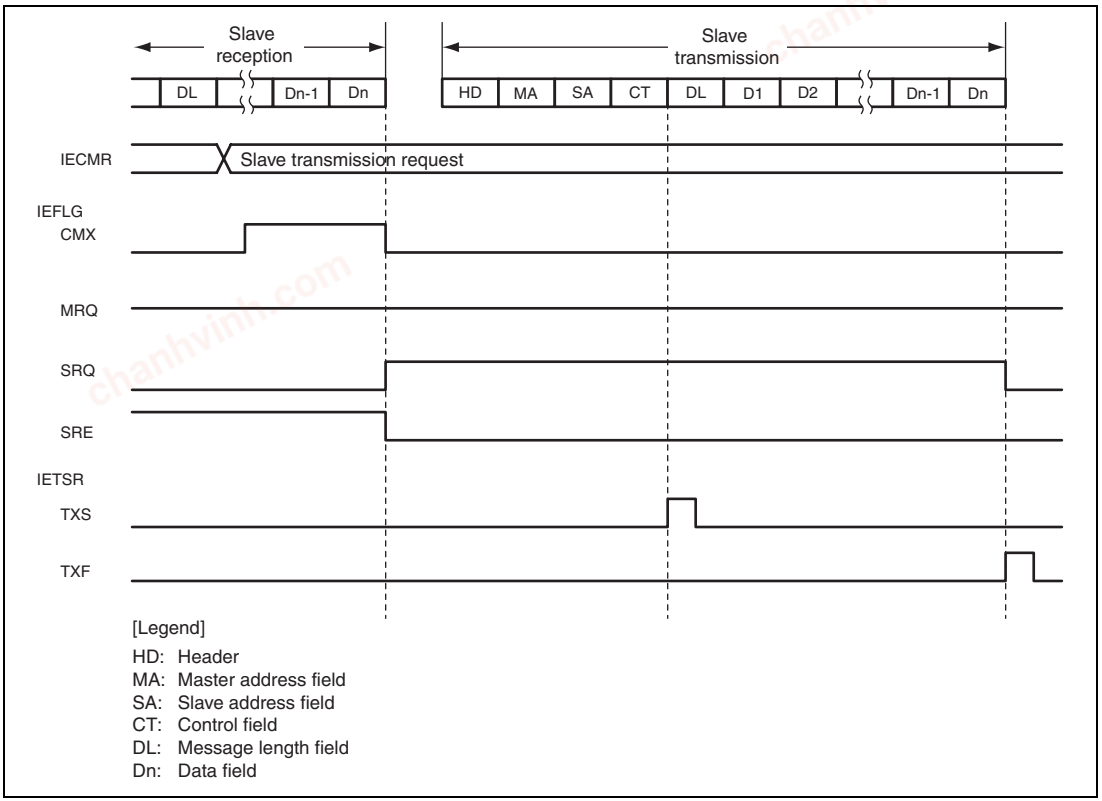


Figure 21.16 Slave Transmit Operation Timing

chanhvinh.com

21.7 Interrupt Sources

Interrupt sources for this module include the following:

- Transmit start (TXS)
- Transmit normal completion (TXF)
- Arbitration loss (TXEAL)
- Transmit timing error (TXETTME)
- Overflow of the maximum number of transmit bytes in one frame (TXERO)
- Acknowledge bits (TXEACK)
- Receive busy (RXBSY)
- Receive start (RXS)
- Receive normal completion (RXF)
- Broadcast Receive Error (RXEDE)
- Receive overrun flag (RXEOVE)
- Receive timing error (RXERTME)
- Overflow of the maximum number of receive bytes in one frame (RXEDLE)
- Parity error (RXEPE)

Each source has bits corresponding to the IEBus transmit interrupt enable register (IEIET) and the IEBus receive interrupt enable register (IEIER) and can enable/disable interrupts. Each source also has status flags corresponding to the IEBus transmit status register (IETSR) and IEBus receive status register (IERSR). Reading the status flags allows determination of the interrupt sources.

chanhvinh.com

Figure 21.17 shows the relations between the interrupt sources.

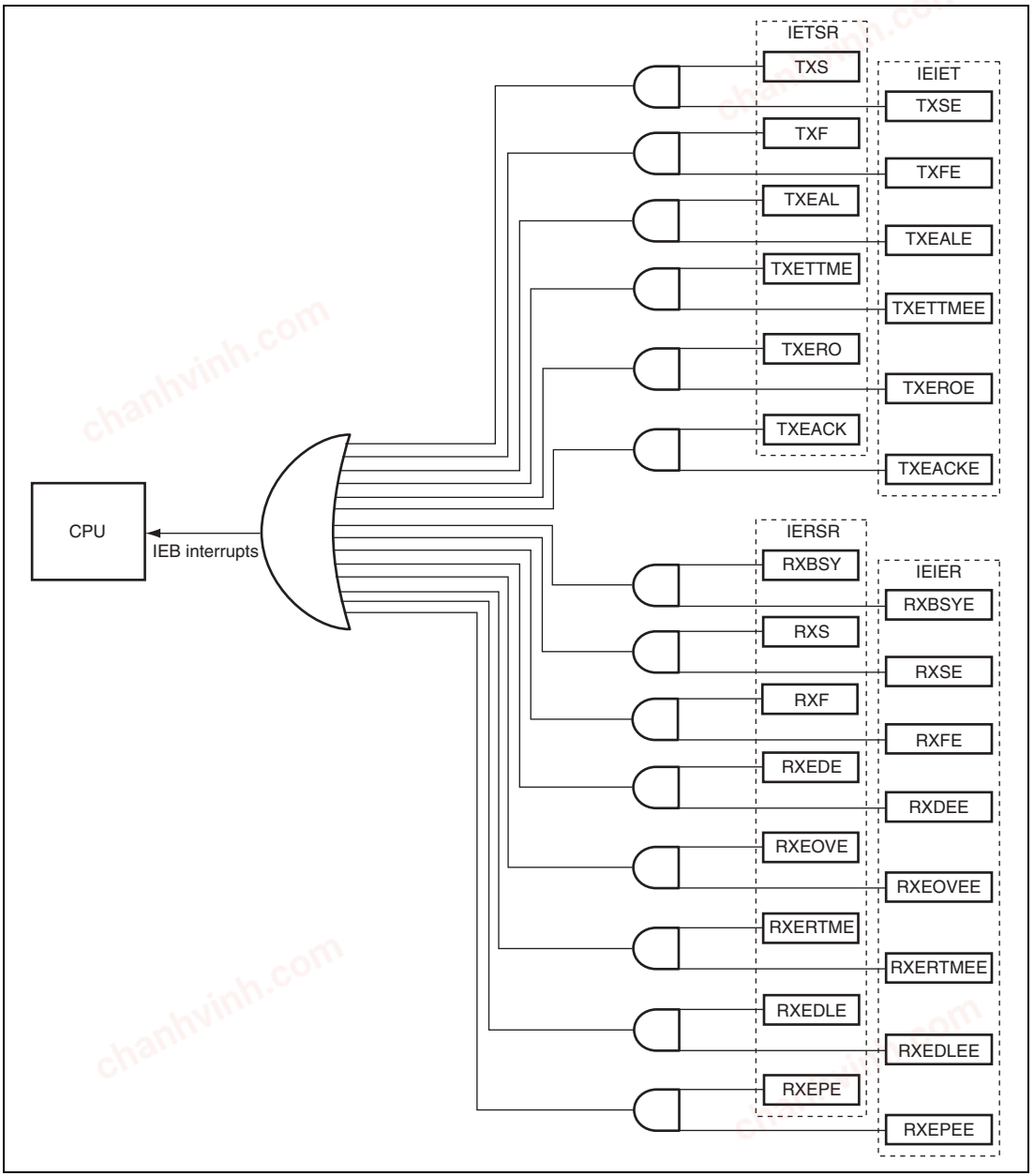


Figure 21.17 Relations between Interrupt Sources

chanhvinh.com

21.8 Usage Notes

21.8.1 Note on Operation when Transfer is Incomplete after Transfer of the Maximum Number of Bytes

(1) Data Transmission

When the maximum number of bytes defined by the communications mode have been transmitted because a NAK has been received from the receive unit or transmission has not been completed because the message length value exceeds the maximum number of transfer bytes in one frame, this module sets the error flag and enters a wait state. At this time, transfer proceeds until the (n + 1)th byte has been transmitted, where n is the maximum number of transfer bytes. Then, when NAK is received via the acknowledge bit of the (n + 1)th byte, the TXERO flag is set. If ACK is received rather than NAK, the TXF flag is set.

Figure 21.18 shows the timing of operations when the maximum number of transfer bytes is reached but transmission has not been completed.

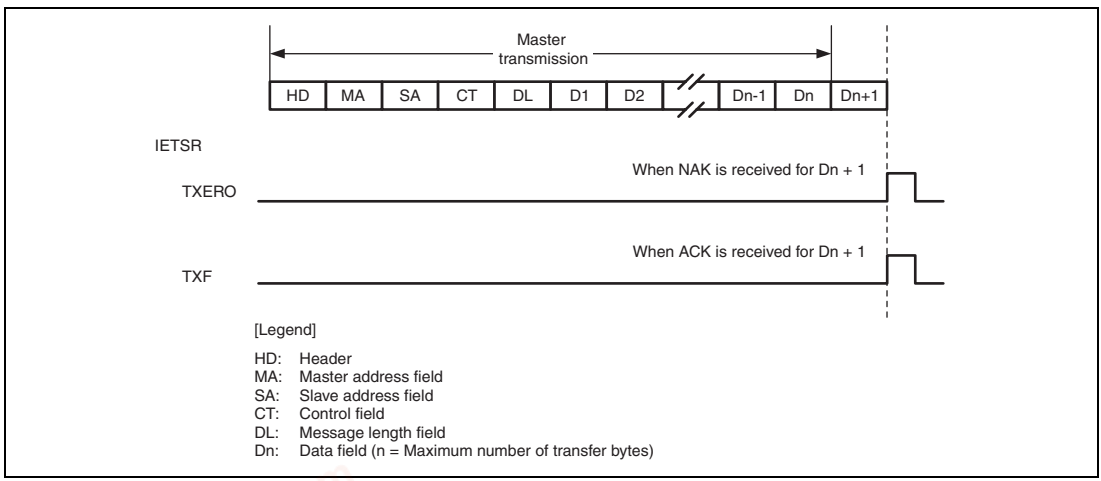


Figure 21.18 Timing of Operations when Transmission Has Not Been Completed Within the Maximum Number of Transfer Bytes

chanhvinh.com

(2) Data Reception

When the data reception has not finished within the maximum number of bytes defined by the communications mode because of a parity error or overrun error causing the retransfer of data, or reception has not been completed because the message length value exceeds the maximum number of transfer bytes in one frame, this module sets the error flag and enters a state of waiting for the (n + 1)th byte of data, where n is the maximum number of transfer bytes. Thus, when data of the (n + 1)th byte cannot be received, the receive timing error is detected and the RXERTME flag is set. At this time, the RXEDLE flag is not set. The RXEDLE flag is set when the (n + 1)th byte is received.

In the same way, when the maximum number of transfer bytes has been received and a parity error has not been cleared, and the (n + 1)th byte cannot be received, the RXERTME flag is set. At this time, the RXEPE flag is not set. The RXEPE flag is set when the (n + 1)th byte is received.

Figure 21.19 shows the timing of operations when the maximum number of transfer bytes has been reached but reception is not complete.

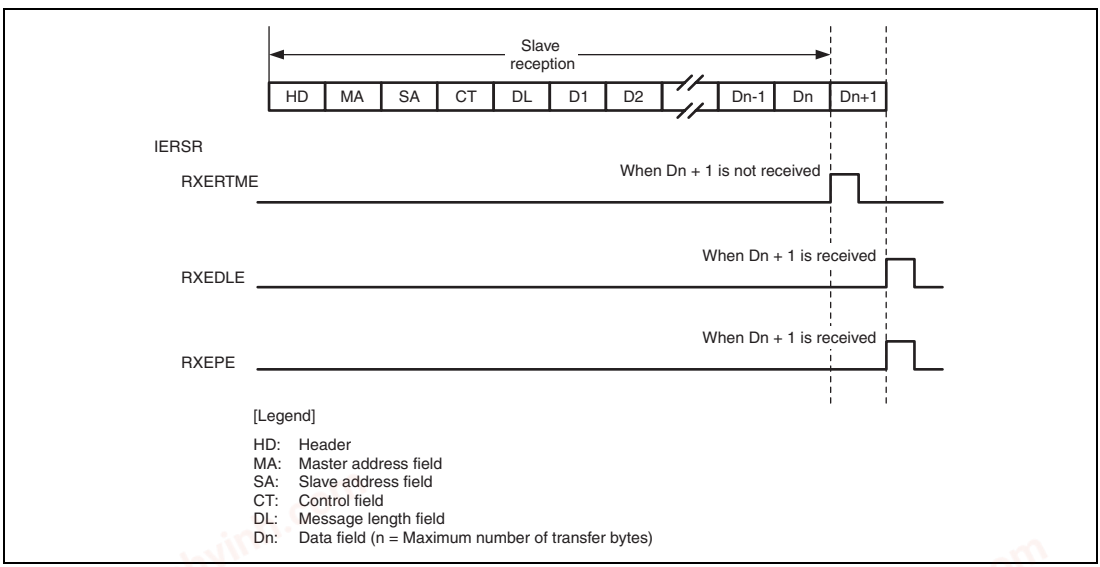


Figure 21.19 Timing of Operations when Reception Has Not Been Completed Within the Maximum Number of Transfer Bytes

chanhvinh.com

Section 22 Renesas SPDIF Interface

22.1 Overview

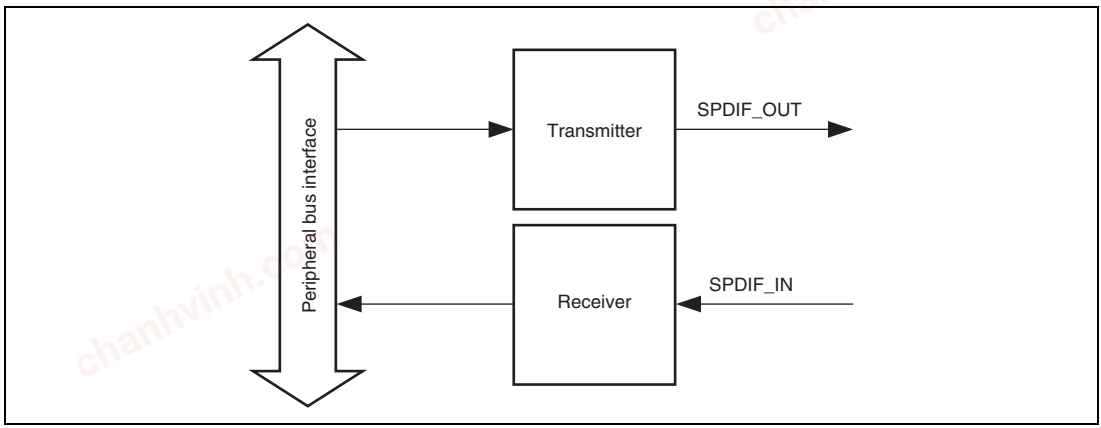


Figure 22.1 Overview Block Diagram

22.2 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

22.3 Functional Block Diagram

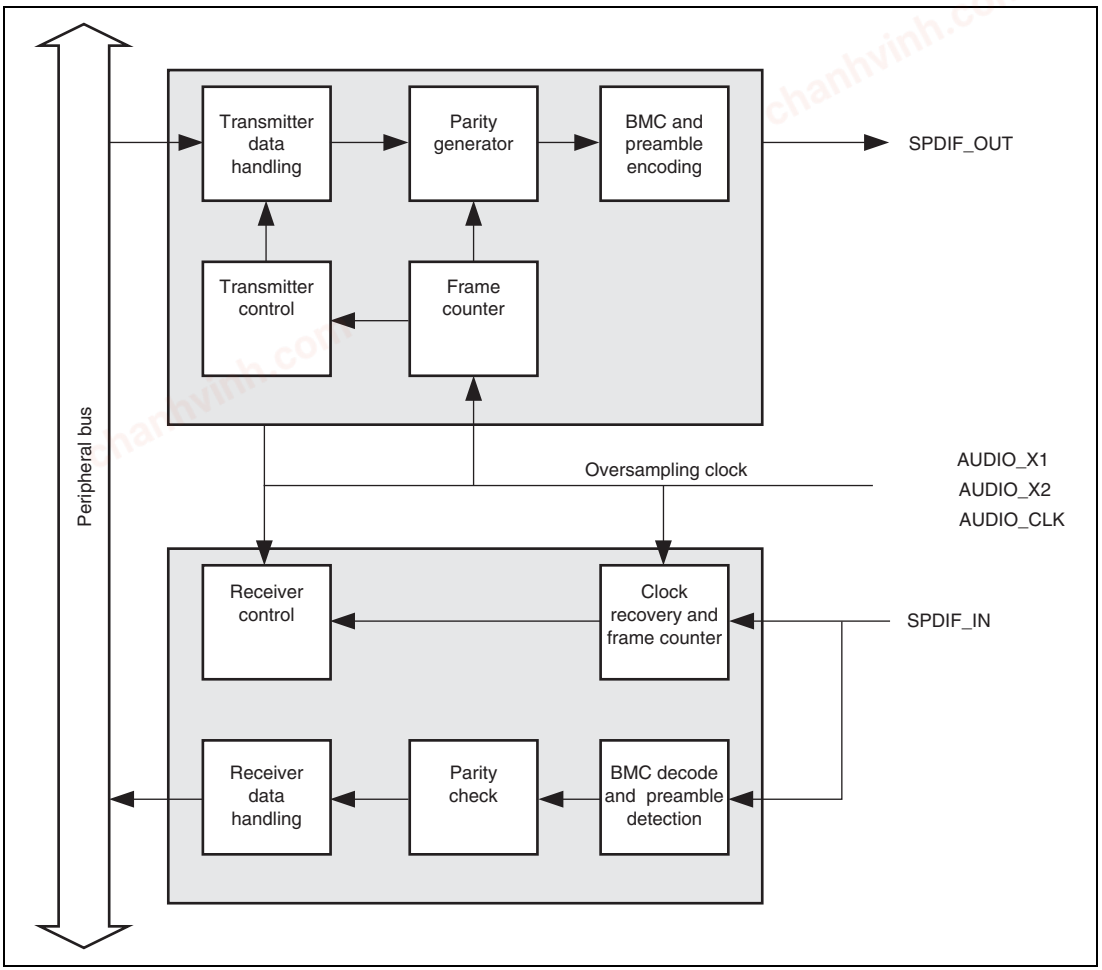


Figure 22.2 Functional Block Diagram

chanhvinh.com

22.4 Input/Output Pins

Table 22.1 shows the pin configuration.

Table 22.1 Pin Configuration

Channel	Pin Name	I/O	Description
0	SPDIF_IN	Input	Transmitter biphase-mark encoded SPDIF bitstream
1	SPDIF_OUT	Output	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	

22.5 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 22.3 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.

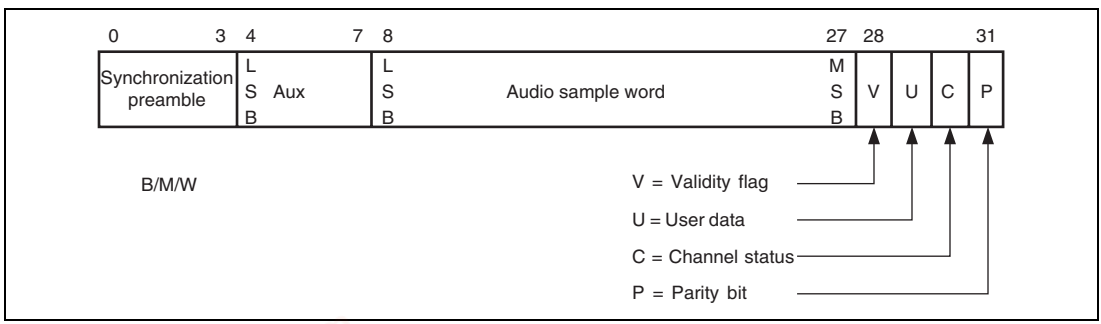


Figure 22.3 Subframe Format

chanhvinh.com

Figure 22.4 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.

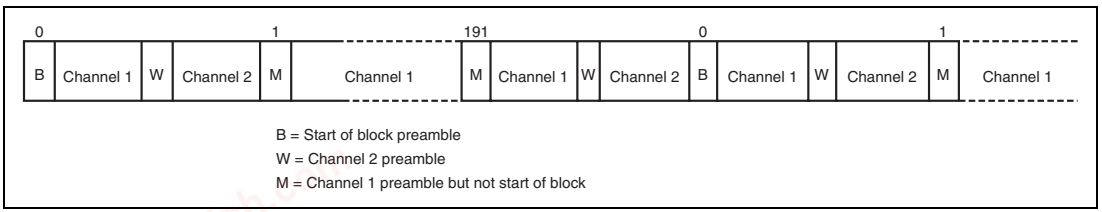


Figure 22.4 Block Format

Table 22.2 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 22.2 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 22.3, 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 22.2.

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

22.6 Register

Table 22.3 shows the register configuration.

Table 22.3 Register Configuration

Channel	Register Name	Abbreviation	Address	Access Size
0 (Transmit)	Transmitter channel 1 audio register	TLCA	H'FFFF 5800	32
	Transmitter channel 2 audio register	TRCA	H'FFFF 5804	32
	Transmitter channel 1 status register	TLCS	H'FFFF 5808	32
	Transmitter channel 2 status register	TRCS	H'FFFF 580C	32
	Transmitter user data register	TUI	H'FFFF 5810	32
1 (Receive)	Receiver channel 1 audio register	RLCA	H'FFFF 5814	32
	Receiver channel 2 audio register	RRCA	H'FFFF 5818	32
	Receiver channel 1 status register	RLCS	H'FFFF 581C	32
	Receiver channel 2 status register	RRCS	H'FFFF 5820	32
	Receiver user data register	RUI	H'FFFF 5824	32
0, 1 (Common)	Control register	CTRL	H'FFFF 5828	32
	Status register	STAT	H'FFFF 582C	32
0, 1 (Common)	Transmitter DMA audio data register	TDAD	H'FFFF 5830	32
	Receiver DMA audio data register	RDAD	H'FFFF 5834	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

22.7 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

22.7.1 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

chanhvinh.com

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
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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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
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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 empty 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 empty 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

22.7.2 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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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
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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
8	CSE	0	R/WC0	Channel Status Error* 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* 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* 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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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
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: * 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.

chanhvinh.com

22.7.3 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

22.7.4 Transmitter Channel 2 Audio Register (TRCA)

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

22.7.5 Transmitter DMA Audio Data Register (TDAD)

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

22.7.6 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

22.7.7 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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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

22.7.8 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

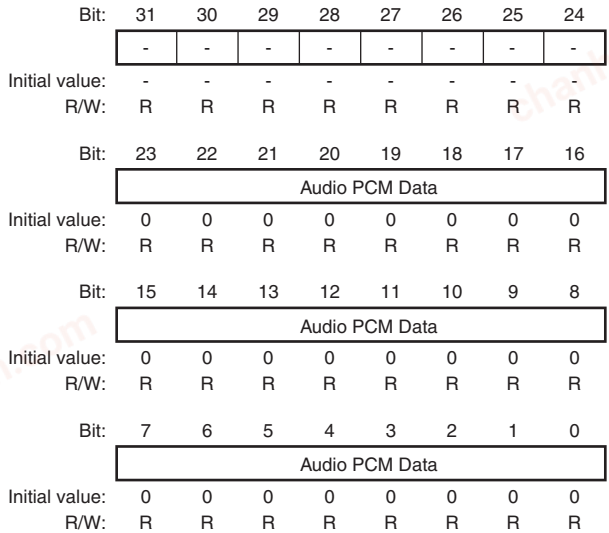
Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	—	W	Reserved
29, 28	CLAC[1:0]	All 0	W	Clock Accuracy 00: Level 2 01: Level 1 10: Level 3 11: Reserved
27 to 24	FS[3:0]	All 0	W	Sample Frequency (FS) 0000: 44.1 kHz 0010: 48 kHz 0011: 32 kHz

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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

22.7.9 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.

chanhvinh.com

22.7.10 Receiver Channel 2 Audio Register (RRCA)

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

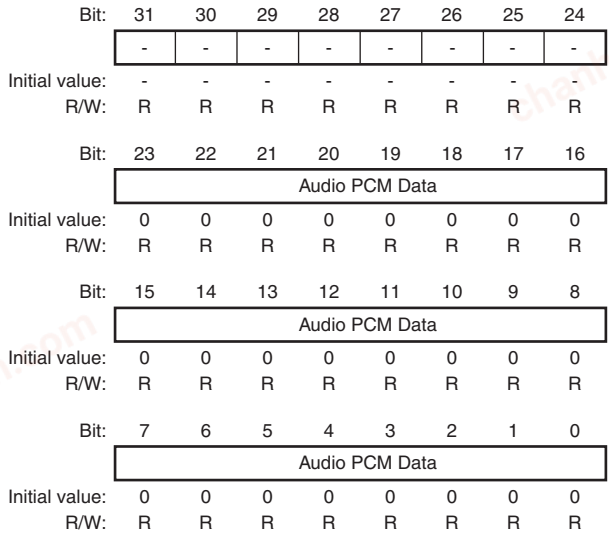
Bit:	15	14	13	12	11	10	9	8
	Audio PCM Data							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

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

22.7.11 Receiver DMA Audio Data (RDAD)



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

22.7.12 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

22.7.13 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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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

22.7.14 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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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

22.8 Functional Description—Transmitter

22.8.1 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.

chanhvinh.com

22.8.2 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.

22.8.3 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.

chanhvinh.com

22.8.4 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 22.5 shows a data transfer with an interrupt for the transmitter.

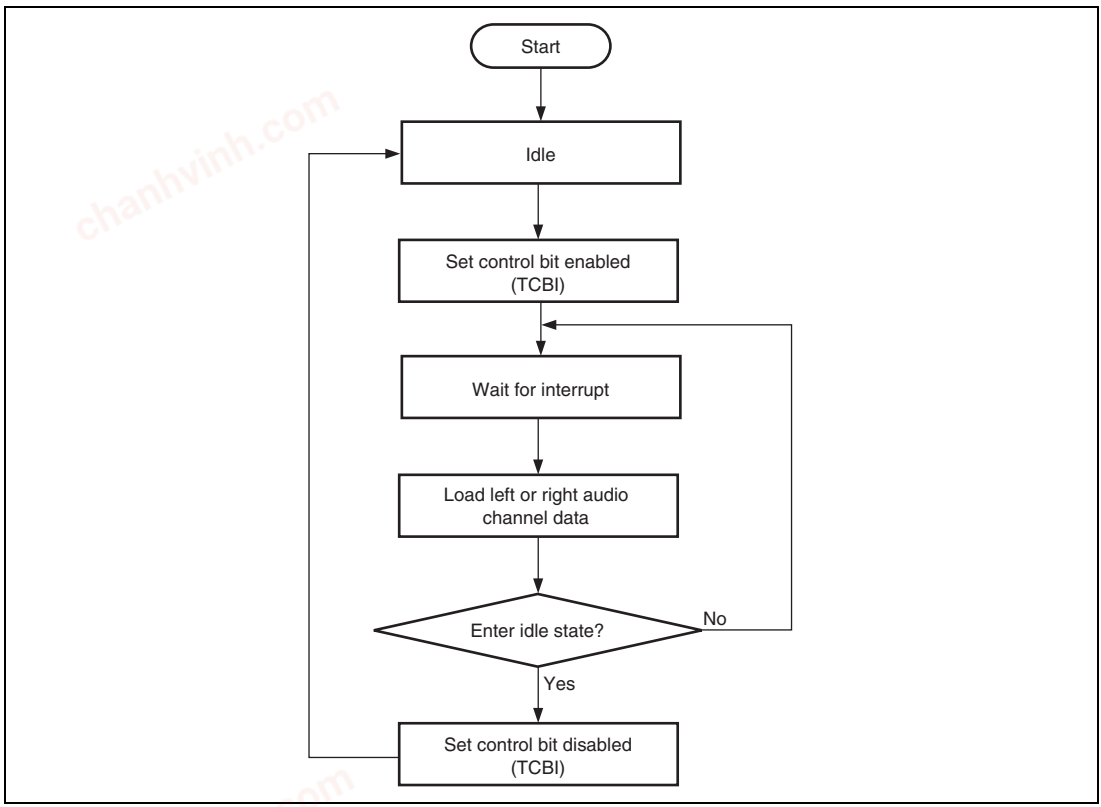


Figure 22.5 Transmitter Data Transfer Flow Diagram - Interrupt Driven

chanhvinh.com

Figure 22.6 shows a data transfer with a DMA transfer for the transmitter.

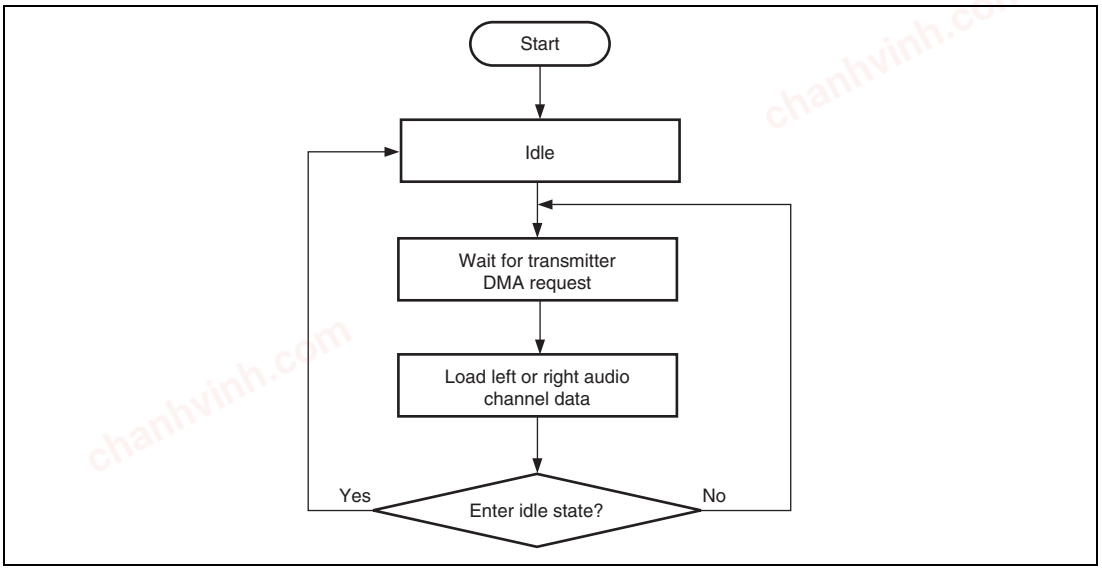


Figure 22.6 Transmitter Data Transfer Flow Diagram—DMA Request Driven

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

22.9 Functional Description—Receiver

22.9.1 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 22.2). 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.

chanhvinh.com

Notes: 1. Channel status data requests do not support DMA.
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.

22.9.2 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.

22.9.3 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:

$$M = \left| \left(0.5 - \frac{1}{2N} \right) - (L - 0.5)\, F - \frac{|D - 0.5|}{N} (1 + F) \right| \times 100\%$$

chanhvinh.com

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 22.7 indicates what the receive margin M represents

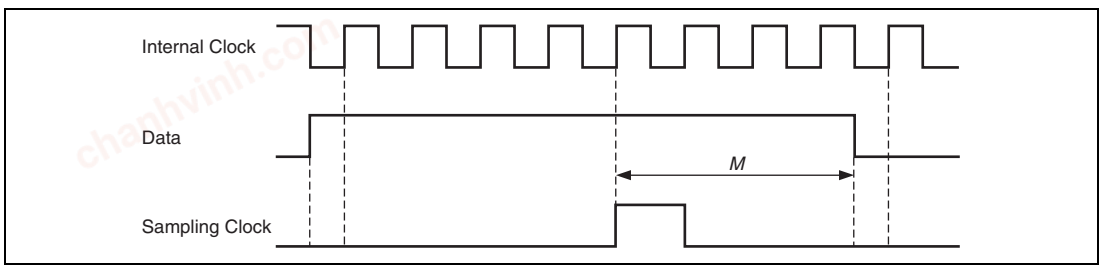


Figure 22.7 Receive Margin

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 22.8 illustrates the receiver data transfer using interrupts.

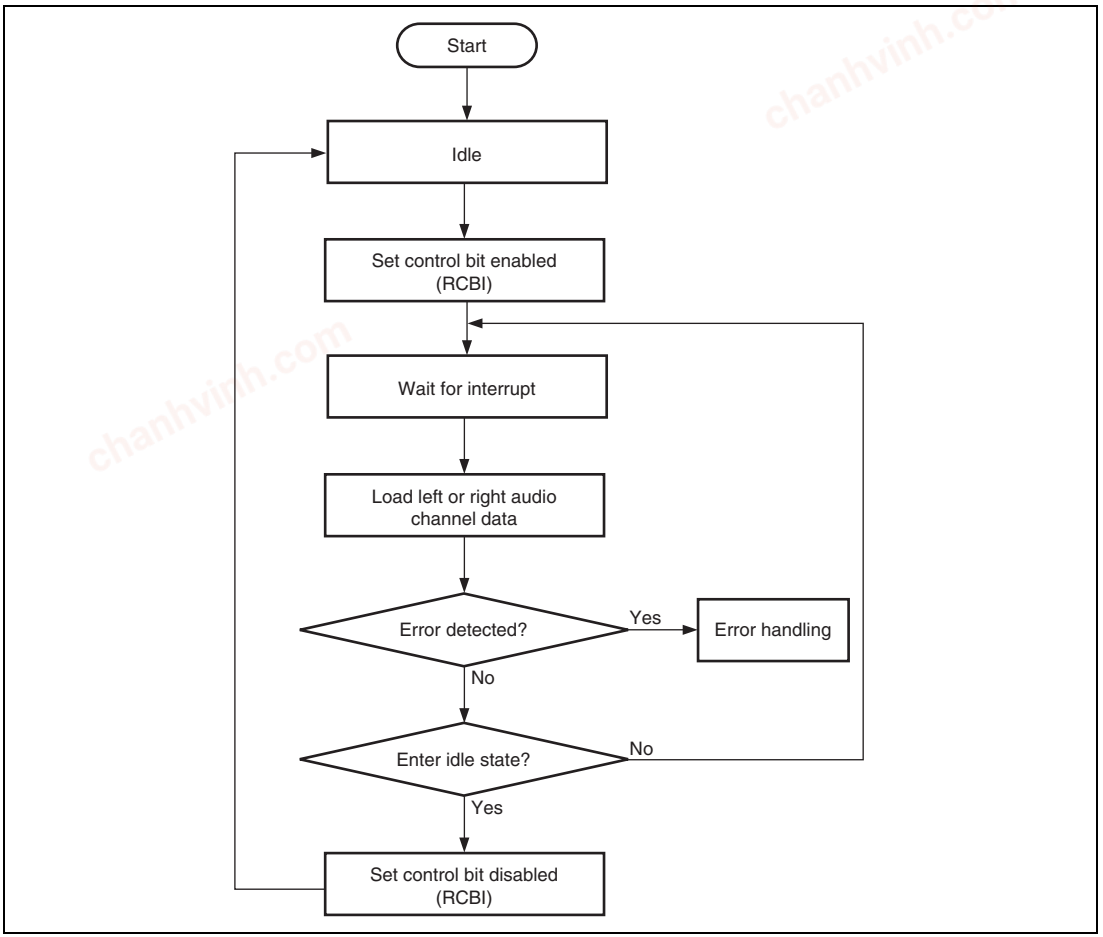


Figure 22.8 Receiver Data Transfer Flow Diagram - Interrupt Driven

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

22.10 Disabling the Module

22.10.1 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).

22.11 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.

22.12 References

IEC60958 Digital Audio Interface

IEC61937 Compressed Mode Digital Audio Interface

chanhvinh.com

22.13 Usage Notes

22.13.1 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.

22.13.2 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 Pϕ frequency.

chanhvinh.com

Section 23 CD-ROM Decoder

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.

Notes: 1. Compliant with JIS S 8605 (Red Book)
2. Compliant with JIS X 6281 (Yellow Book)

23.1 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 support

  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 correction based on the ECC. Furthermore, an operating mode is available in which, if the result of pre-correction EDC checking indicates no errors, ECC correction is not performed regardless of the result of syndrome calculation.

chanhvinh.com

- 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.

23.1.1 Formats Supported by CD-ROM Decoder

This module supports the five formats shown in figure 23.1.

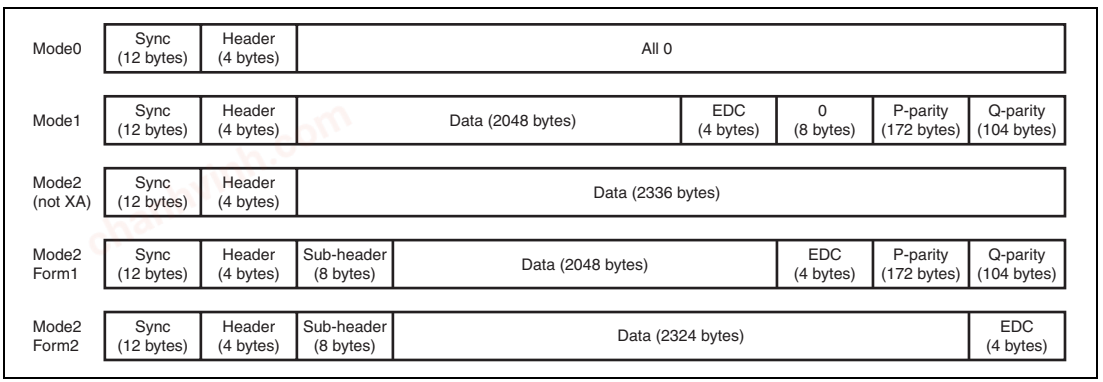


Figure 23.1 Formats Supported by CD-ROM Decoder

chanhvinh.com

23.2 Block Diagrams

Figure 23.2 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.

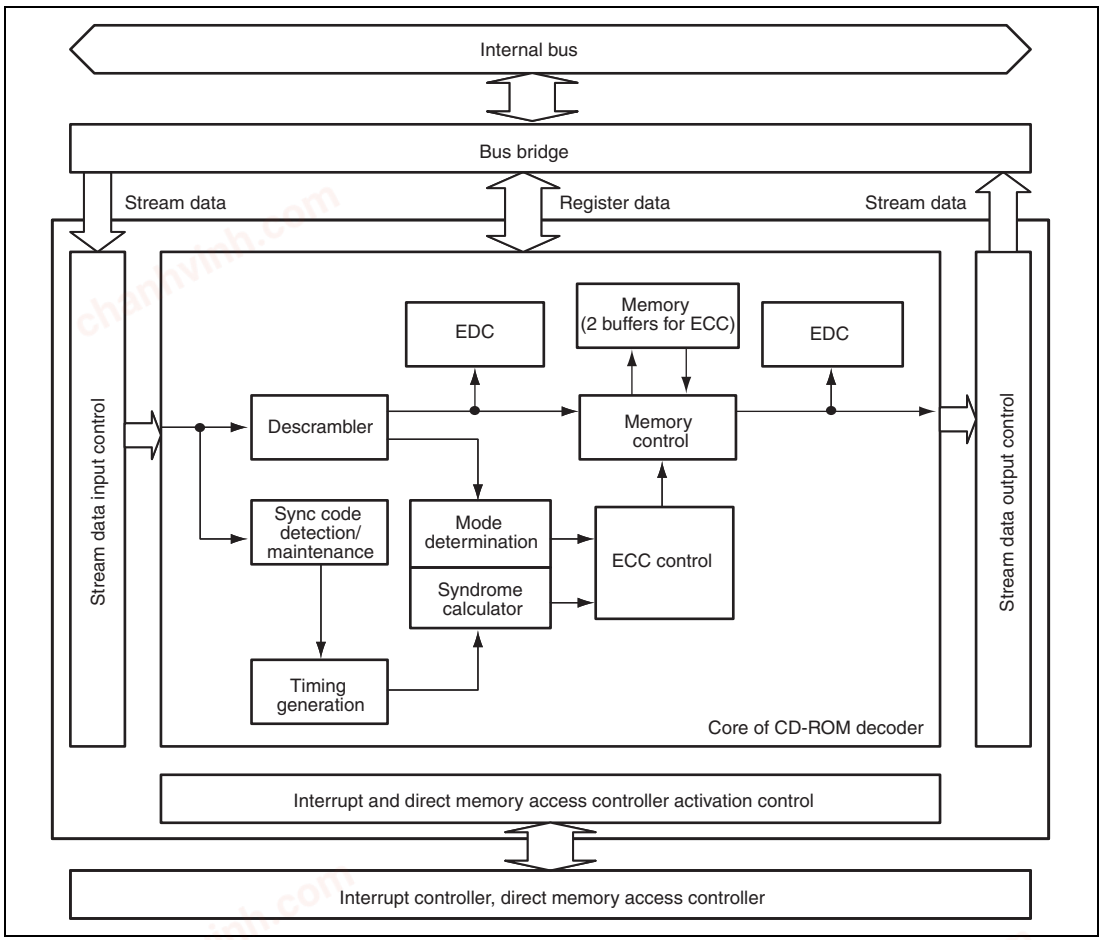


Figure 23.2 Block Diagram

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.

chanhvinh.com

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. Data can be transferred by either the direct memory access controller or the CPU.

Figure 23.3 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.

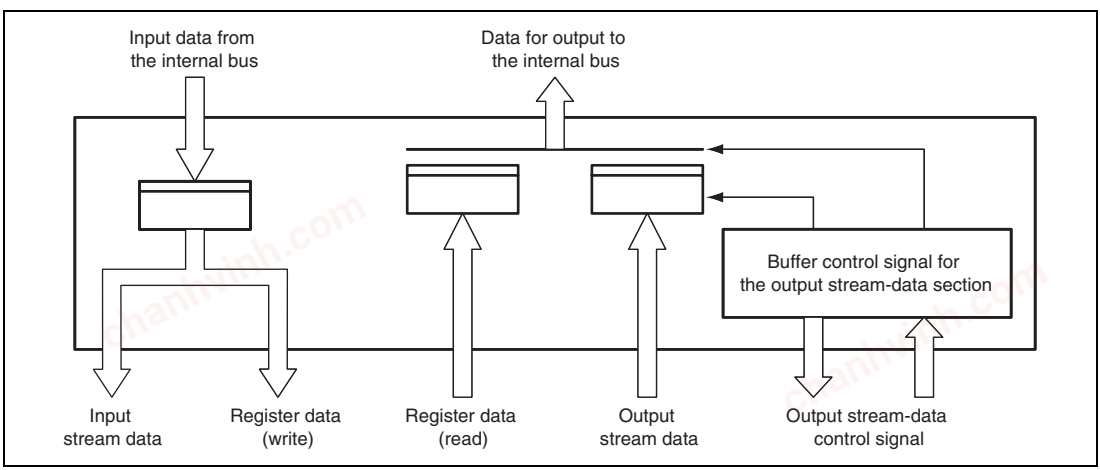


Figure 23.3 Schematic Diagram of the Bus Bridge

chanhvinh.com

Figure 23.4 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.

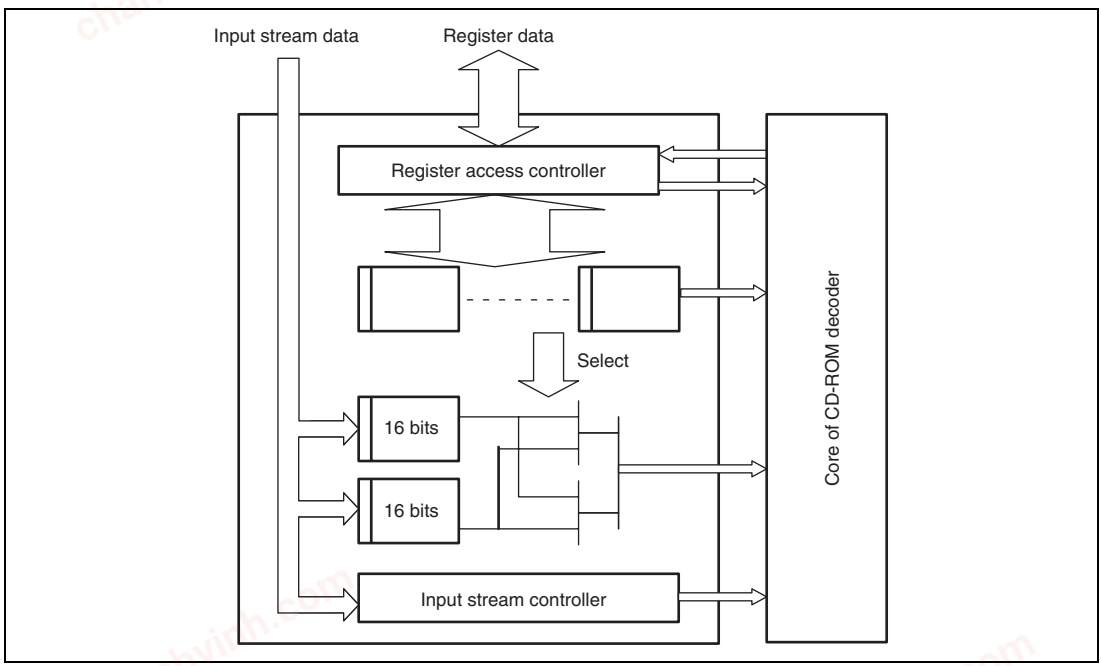


Figure 23.4 Schematic Diagram of the Stream-Data Input Control Block

chanhvinh.com

Figure 23.5 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.

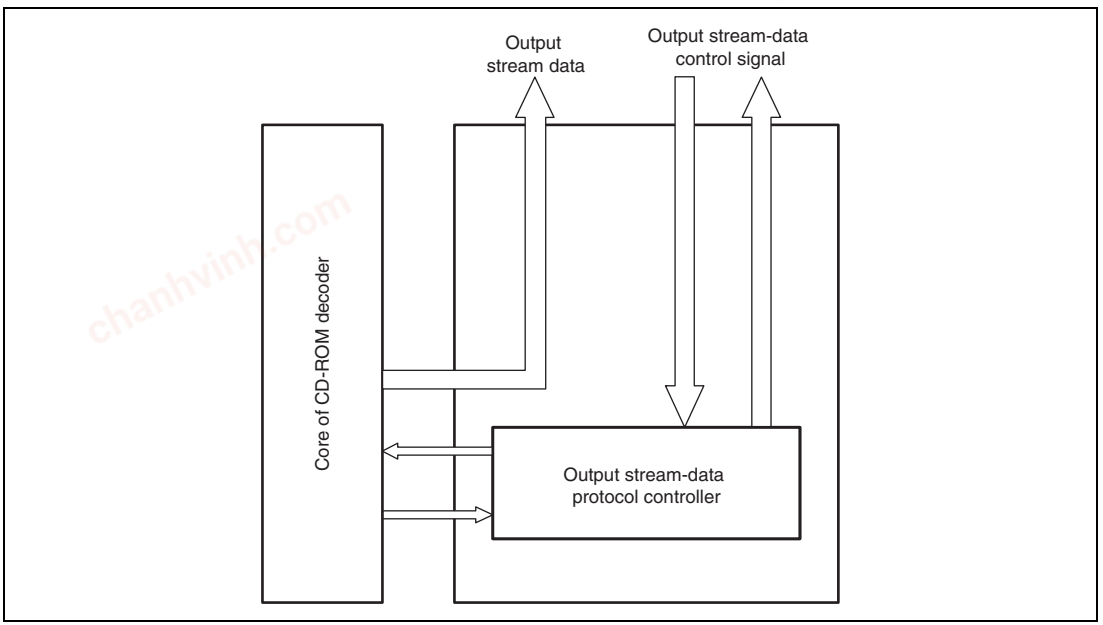


Figure 23.5 Schematic Diagram of the Stream-Data Output Control Block

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 memory access controller, and negating the activation signal according to the detected amount of data that has been transferred.

chanhvinh.com

23.3 Register Descriptions

This module has the following registers.

Table 23.1 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Enable control register	CROMEN	R/W	H'00	H'FFFF9000	8
Sync code-based synchronization control register	CROMSY0	R/W	H'89	H'FFFF9001	8
Decoding mode control register	CROMCTL0	R/W	H'82	H'FFFF9002	8
EDC/ECC check control register	CROMCTL1	R/W	H'D1	H'FFFF9003	8
Automatic decoding stop control register	CROMCTL3	R/W	H'00	H'FFFF9005	8
Decoding option setting control register	CROMCTL4	R/W	H'00	H'FFFF9006	8
HEAD20 to HEAD22 representation control register	CROMCTL5	R/W	H'00	H'FFFF9007	8
Sync code status register	CROMST0	R	H'00	H'FFFF9008	8
Post-ECC header error status register	CROMST1	R	H'00	H'FFFF9009	8
Post-ECC subheader error status register	CROMST3	R	H'00	H'FFFF900B	8
Header/subheader validity check status register	CROMST4	R	H'00	H'FFFF900C	8
Mode determination and link sector detection status register	CROMST5	R	H'00	H'FFFF900D	8
ECC/EDC error status register	CROMST6	R	H'00	H'FFFF900E	8
Buffer status register	CBUFST0	R	H'00	H'FFFF9014	8
Decoding stoppage source status register	CBUFST1	R	H'00	H'FFFF9015	8
Buffer overflow status register	CBUFST2	R	H'00	H'FFFF9016	8
Pre-ECC correction header: minutes data register	HEAD00	R	H'00	H'FFFF9018	8
Pre-ECC correction header: seconds data register	HEAD01	R	H'00	H'FFFF9019	8
Pre-ECC correction header: frames (1/75 second) data register	HEAD02	R	H'00	H'FFFF901A	8
Pre-ECC correction header: mode data register	HEAD03	R	H'00	H'FFFF901B	8

chanhvinh.com

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Pre-ECC correction subheader: file number (byte 16) data register	SHEAD00	R	H'00	H'FFFF901C	8
Pre-ECC correction subheader: channel number (byte 17) data register	SHEAD01	R	H'00	H'FFFF901D	8
Pre-ECC correction subheader: sub-mode (byte 18) data register	SHEAD02	R	H'00	H'FFFF901E	8
Pre-ECC correction subheader: data type (byte 19) data register	SHEAD03	R	H'00	H'FFFF901F	8
Pre-ECC correction subheader: file number (byte 20) data register	SHEAD04	R	H'00	H'FFFF9020	8
Pre-ECC correction subheader: channel number (byte 21) data register	SHEAD05	R	H'00	H'FFFF9021	8
Pre-ECC correction subheader: sub-mode (byte 22) data register	SHEAD06	R	H'00	H'FFFF9022	8
Pre-ECC correction subheader: data type (byte 23) data register	SHEAD07	R	H'00	H'FFFF9023	8
Post-ECC correction header: minutes data register	HEAD20	R	H'00	H'FFFF9024	8
Post-ECC correction header: seconds data register	HEAD21	R	H'00	H'FFFF9025	8
Post-ECC correction header: frames (1/75 second) data register	HEAD22	R	H'00	H'FFFF9026	8
Post-ECC correction header: mode data register	HEAD23	R	H'00	H'FFFF9027	8
Post-ECC correction subheader: file number (byte 16) data register	SHEAD20	R	H'00	H'FFFF9028	8
Post-ECC correction subheader: channel number (byte 17) data register	SHEAD21	R	H'00	H'FFFF9029	8
Post-ECC correction subheader: sub-mode (byte 18) data register	SHEAD22	R	H'00	H'FFFF902A	8
Post-ECC correction subheader: data type (byte 19) data register	SHEAD23	R	H'00	H'FFFF902B	8
Post-ECC correction subheader: file number (byte 20) data register	SHEAD24	R	H'00	H'FFFF902C	8

chanhvinh.com

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Post-ECC correction subheader: channel number (byte 21) data register	SHEAD25	R	H'00	H'FFFF902D	8
Post-ECC correction subheader: sub-mode (byte 22) data register	SHEAD26	R	H'00	H'FFFF902E	8
Post-ECC correction subheader: data type (byte 23) data register	SHEAD27	R	H'00	H'FFFF902F	8
Automatic buffering setting control register	CBUFCTL0	R/W	H'04	H'FFFF9040	8
Automatic buffering start sector setting: minutes control register	CBUFCTL1	R/W	H'00	H'FFFF9041	8
Automatic buffering start sector setting: seconds control register	CBUFCTL2	R/W	H'00	H'FFFF9042	8
Automatic buffering start sector setting: frames control register	CBUFCTL3	R/W	H'00	H'FFFF9043	8
ISY interrupt source mask control register	CROMST0M	R/W	H'00	H'FFFF9045	8
CD-ROM decoder reset control register	ROMDECRST	R/W	H'00	H'FFFF9100	8
CD-ROM decoder reset status register	RSTSTAT	R	H'00	H'FFFF9101	8
Serial sound interface data control register	SSI	R/W	H'18	H'FFFF9102	8
Interrupt flag register	INTHOLD	R/W	H'00	H'FFFF9108	8
Interrupt source mask control register	INHINT	R/W	H'00	H'FFFF9109	8
CD-ROM decoder stream data input register	STRMDIN0	R/W	H'0000	H'FFFF9200	Read: 16 Write: 16/32
CD-ROM decoder stream data input register	STRMDIN2	R/W	H'0000	H'FFFF9202	16
CD-ROM decoder stream data output register	STRMDOUT0	R	H'0000	H'FFFF9204	16, 32

chanhvinh.com

23.3.1 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 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

23.3.2 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 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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 23.2 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

23.3.3 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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 sector is taken to be in the same mode and form as the previous sector. 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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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.

23.3.4 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’.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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

23.3.5 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

chanhvinh.com

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-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 (1/75 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.

23.3.6 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
	LINKOFF	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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
7	LINKOFF	0	R/W	Buffering Control at Link Block Detection 0: ON 1: OFF When this bit is set to 1, buffering control is not performed when a link block is detected. The link block is processed as normal sectors. However, link-block detection processing does proceed, and the results are reflected in the values of bits 3 to 0 in the CROMST5 register.
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. When this bit is set to 1, buffering control for link blocks is disabled (link blocks are processed as normal sectors). 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-correction EDC checking is a ‘pass’, regardless of the results of syndrome calculation.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.

23.3.7 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)

chanhvinh.com

23.3.8 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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 currently being decoded sector.
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.

23.3.9 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 (1/75 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.

chanhvinh.com

23.3.10 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

23.3.11 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 (1/75 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 (1/75 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

23.3.12 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. When this bit is set to 1, buffering control is performed according to the setting of the CBUF_LINK bit in the CBUFCTL0 register.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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).

23.3.13 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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 non-0.
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’.

chanhvinh.com

23.3.14 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

23.3.15 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-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 sectors.
3	BUF_MIN	0	R	Indicates that decoding and buffering have been stopped because a non-sequential minutes, seconds, or frames (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.

chanhvinh.com

23.3.16 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, 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.

23.3.17 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.

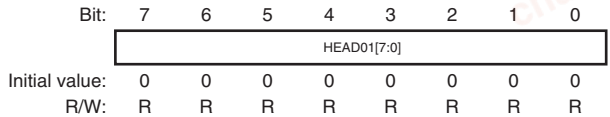
Bit:	7	6	5	4	3	2	1	0
	HEAD00[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	HEAD00 [7:0]	All 0	R	Minutes Value in Header Before ECC Correction

chanhvinh.com

23.3.18 Pre-ECC Correction Header: Seconds Data Register (HEAD01)

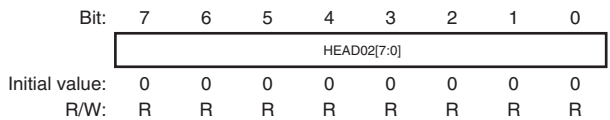
The pre-ECC correction header: seconds data register (HEAD01) indicates the seconds value in the header before ECC correction.



Bit	Bit Name	Initial Value	R/W	Description
7 to 0	HEAD01 [7:0]	All 0	R	Seconds Value in Header Before ECC Correction

23.3.19 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.

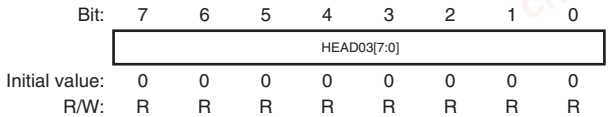


Bit	Bit Name	Initial Value	R/W	Description
7 to 0	HEAD02 [7:0]	All 0	R	Frames Value in Header Before ECC Correction

chanhvinh.com

23.3.20 Pre-ECC Correction Header: Mode Data Register (HEAD03)

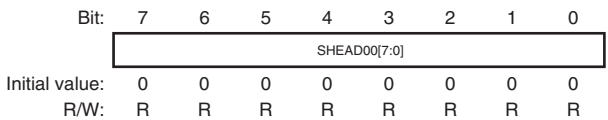
The pre-ECC correction header: mode data register (HEAD03) indicates the mode value in the header before ECC correction.



Bit	Bit Name	Initial Value	R/W	Description
7 to 0	HEAD03 [7:0]	All 0	R	Mode value in the header before ECC correction

23.3.21 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	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.

chanhvinh.com

23.3.22 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	Indicate 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.

23.3.23 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	Indicate 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

23.3.24 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	Indicate 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.

23.3.25 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	Indicate 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.

chanhvinh.com

23.3.26 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	Indicate 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.

23.3.27 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	Sub-Mode Value in Subheader Before ECC Correction (Byte 22) For sectors not in Mode 2, this register contains the byte of data at the corresponding position.

chanhvinh.com

23.3.28 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	Data Type Value in Subheader Before ECC Correction (Byte 23) For sectors not in Mode 2, this register contains the byte of data at the corresponding position.

23.3.29 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.

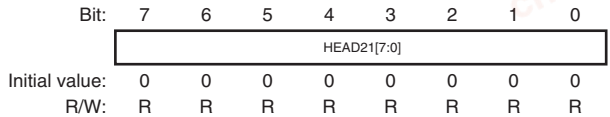
Bit:	7	6	5	4	3	2	1	0
	HEAD20[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	HEAD20 [7:0]	All 0	R	Minutes Value in Header After ECC Correction When MSF_LBA_SEL = 1, this register indicates the first byte of the total number of sectors calculated from M, S, and F.

chanhvinh.com

23.3.30 Post-ECC Correction Header: Seconds Data Register (HEAD21)

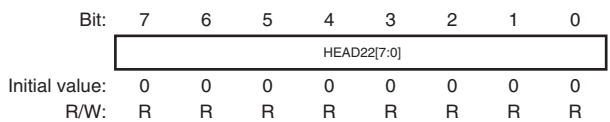
The post-ECC correction header: seconds data register (HEAD21) indicates the seconds value in the header after ECC correction.



Bit	Bit Name	Initial Value	R/W	Description
7 to 0	HEAD21 [7:0]	All 0	R	Seconds Value in Header After ECC Correction When MSF_LBA_SEL = 1, this register indicates the second byte of the total number of sectors calculated from M, S, and F.

23.3.31 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	Bit Name	Initial Value	R/W	Description
7 to 0	HEAD22 [7:0]	All 0	R	Frames Value in Header After ECC Correction When MSF_LBA_SEL = 1, this register indicates the third byte of the total number of sectors calculated from M, S, and F.

chanhvinh.com

23.3.32 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	Mode Value in Header After ECC Correction

23.3.33 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	Indicate file number value in the subheader after ECC correction (byte 16).

chanhvinh.com

23.3.34 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	Indicate channel number value in the subheader after ECC correction (byte 17).

23.3.35 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

23.3.36 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	Indicate data type value in the subheader after ECC correction (byte 19).

23.3.37 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	Indicate file number value in the subheader after ECC correction (byte 20).

chanhvinh.com

23.3.38 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	Indicate channel number value in the subheader after ECC correction (byte 21).

23.3.39 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	Indicate sub-mode value in the subheader after ECC correction (byte 22).

chanhvinh.com

23.3.40 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	Data Type Value in Subheader After ECC Correction (byte 23)

23.3.41 Automatic Buffering Setting Control Register 0 (CBUFCTL0)

Bit:	7	6	5	4	3	2	1	0
	CBUF_ AUT	CBUF_ EN	CBUF_ LINK	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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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	CBUF_ LINK	0	R/W	Buffering Control on Link Block Detection 0: Allocates area for seven sectors 1: Data are buffered, skipping the link block
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

23.3.42 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	Indicate setting of the minutes value in the header for the first sector to be buffered.

23.3.43 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	Indicate setting of the seconds value in the header for the first sector to be buffered.

chanhvinh.com

23.3.44 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	Indicate setting of the frames (1/75 second) value in the header for the first sector to be buffered.

23.3.45 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 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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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

23.3.46 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
	LOGI CRST	RAM RST	-	-	-	-	-	-
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

23.3.47 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.

23.3.48 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 23.4.1, 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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 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 is without 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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 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 is without 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

23.3.49 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

23.3.50 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	PREINH REQDM	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	PREINH IREADY	0	R/W	Inhibits setting of the IREADY interrupt flag. When this bit is set to 1, the IREADY interrupt source not retained.

chanhvinh.com

23.3.51 CD-ROM Decoder Stream Data Input Register (STRMDIN0)

The CD-ROM decoder stream data input register (STRDMIN0) 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	Indicate 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.

23.3.52 CD-ROM Decoder Stream Data Input Register (STRMDIN2)

The CD-ROM decoder stream data input register (STRDMIN2) 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	Indicate 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.

chanhvinh.com

23.3.53 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	Indicate 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

23.4 Operation

23.4.1 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 23.6 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.

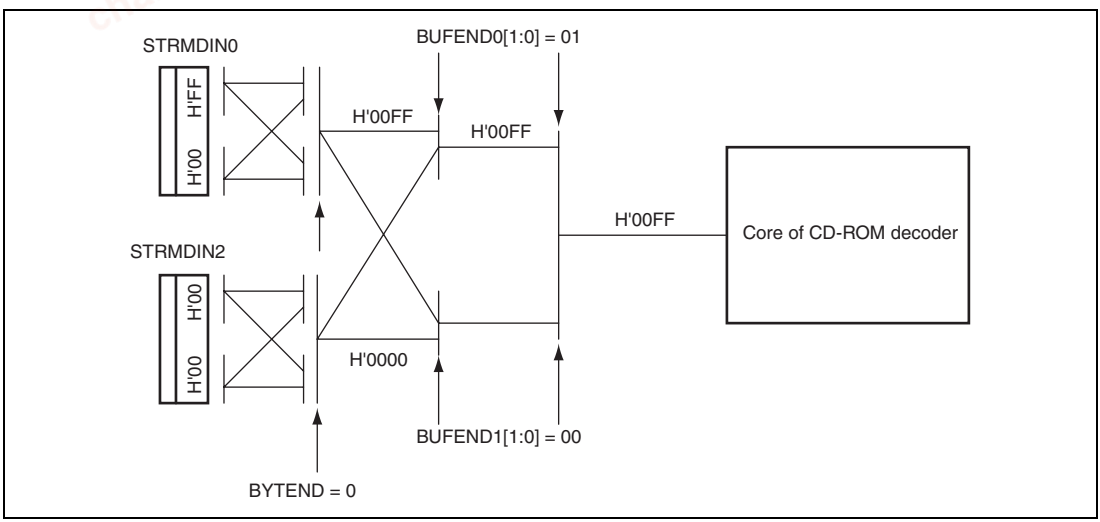


Figure 23.6 Example of Padded Stream Data Control by the SSI Register

chanhvinh.com

Figure 23.7 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.

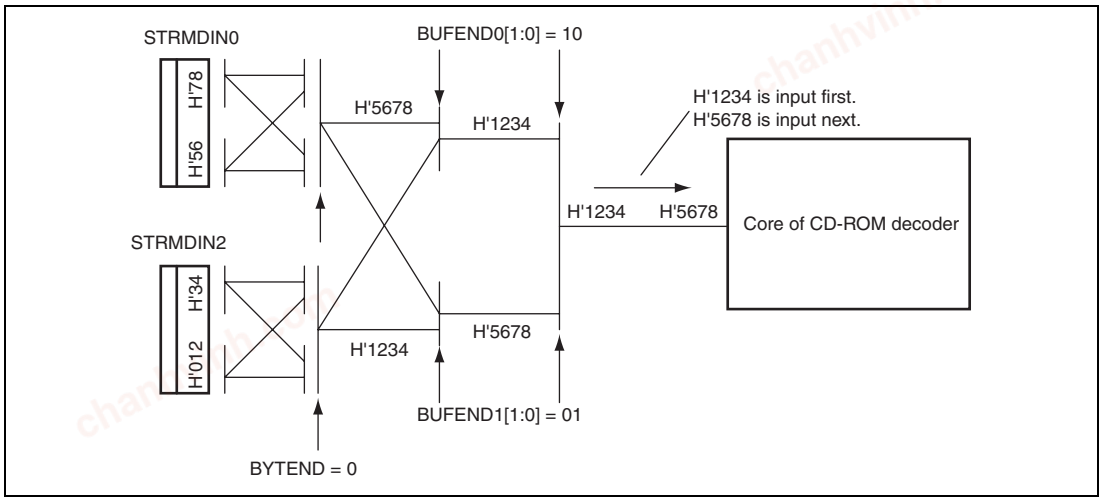


Figure 23.7 Example of Non-Padded Stream Data Control by the SSI Register

23.4.2 Sync Code Maintenance Function

Each sector of CD-ROM data consists of 2352 bytes starting with H'00FFFFFFFFFFFFFFFFFFFF00 (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 23.3.2, Sync Code-Based Synchronization Control Register (CROMSY0), and table 23.2.

- Automatic sync maintenance mode
- External sync mode
- Interpolated sync mode
- Interpolated sync plus external sync mode

chanhvinh.com

(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 23.8 shows operation in the case of normal sync-code detection, figure 23.9 shows a case where a sync code is detected before a current one-sector period has elapsed, and figure 23.10 shows the case where the actual sync code is only detected some time after a full one-sector period has elapsed.

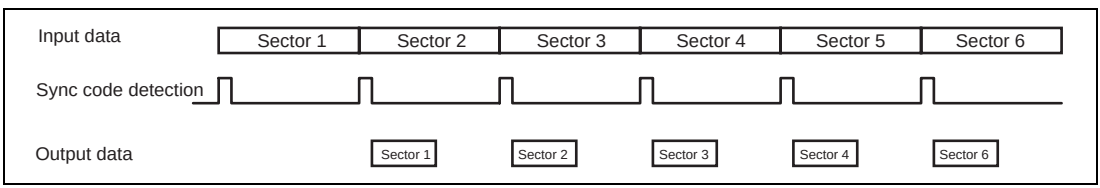


Figure 23.8 Operation in Automatic Sync Maintenance Mode (Normal Timing)

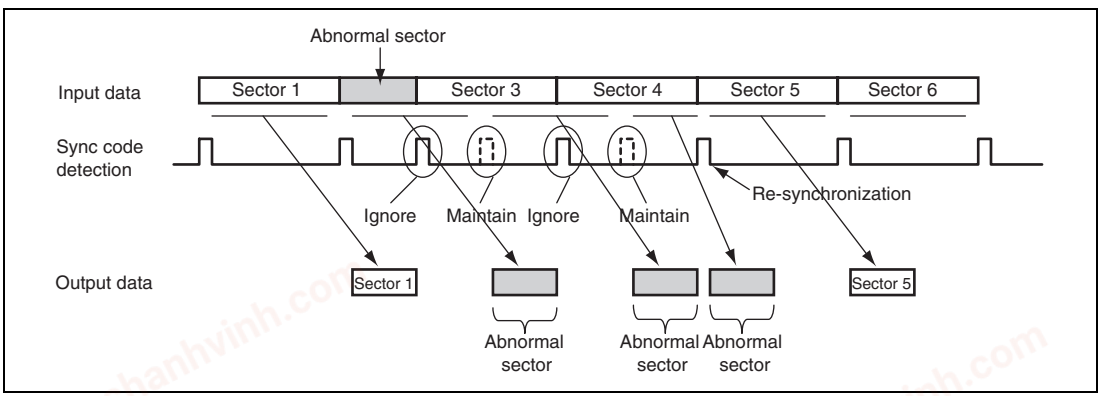


Figure 23.9 Operation in Automatic Sync Maintenance Mode (When an Abnormally Short Sector is Encountered)

chanhvinh.com

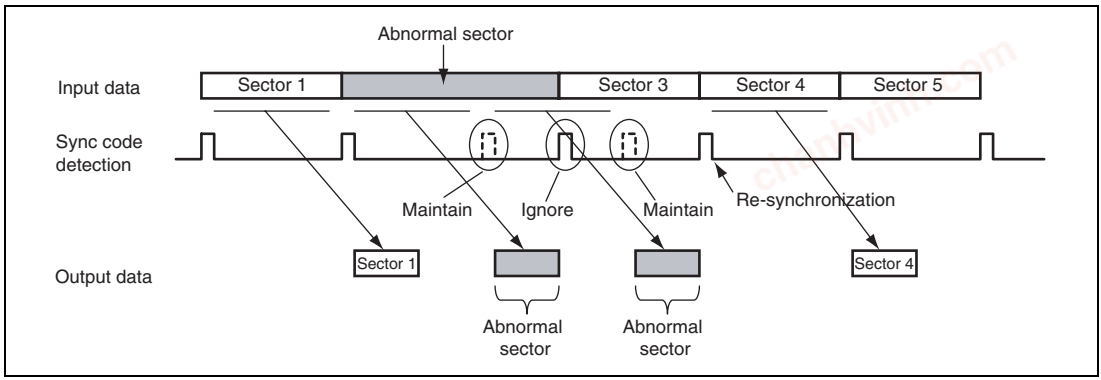


Figure 23.10 Operation in Automatic Sync Maintenance Mode (When an Abnormally Long Sector is Encountered)

(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 23.11 shows the operation in external sync mode.

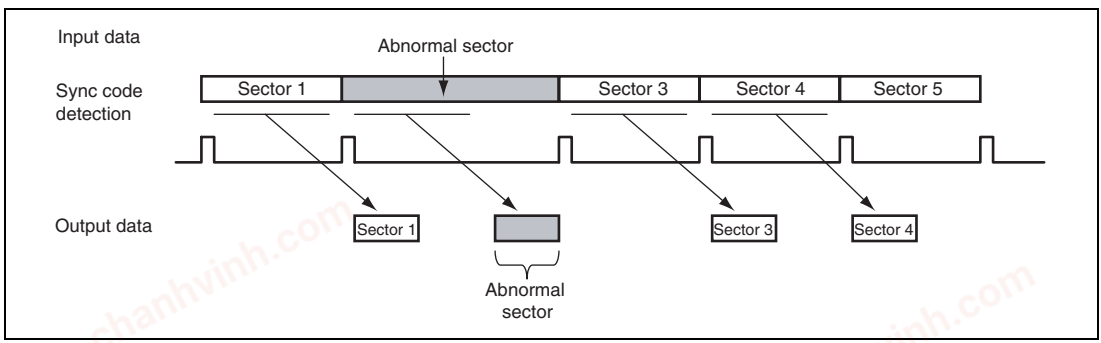


Figure 23.11 Operation in External Sync Mode

chanhvinh.com

(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 23.12 shows the operation in interpolated sync mode.

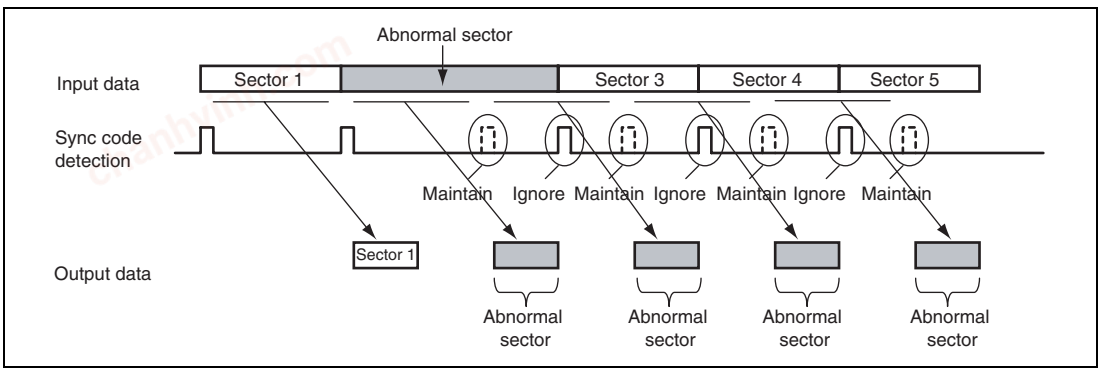


Figure 23.12 Operation in Interpolated Sync Mode

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.

Figures 23.13 and 23.14 show the operation in interpolated sync plus external sync mode in the cases of abnormally short and long sectors, respectively.

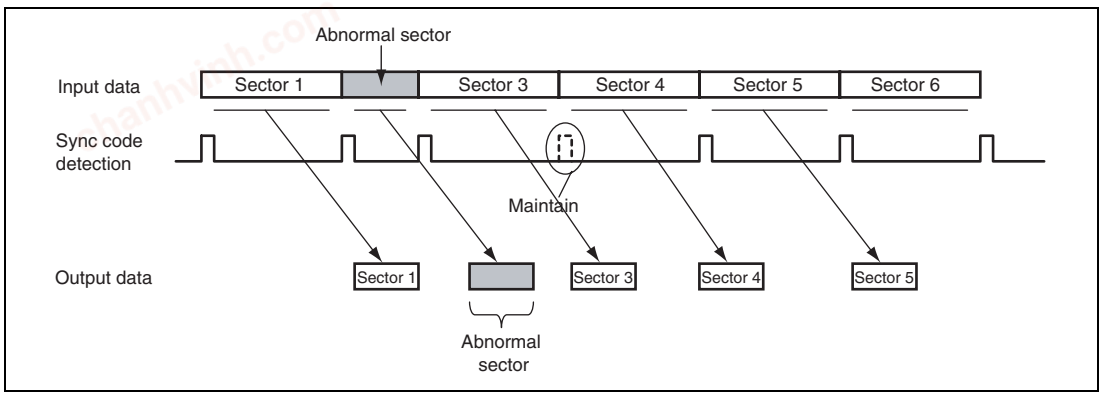


Figure 23.13 Operation in Interpolated Sync Plus External Sync Mode (When an Abnormally Short Sector is Encountered)

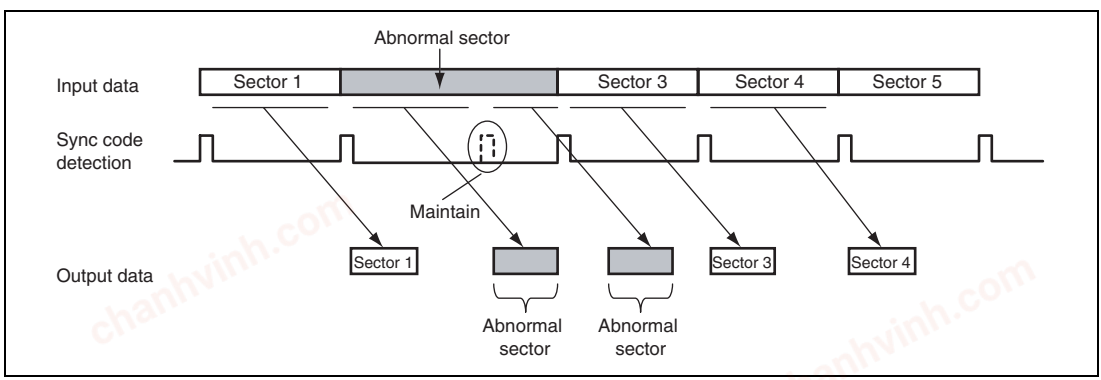


Figure 23.14 Operation in Interpolated Sync Plus External Sync Mode (When an Abnormally Long Sector is Encountered)

chanhvinh.com

23.4.3 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.

(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 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.

chanhvinh.com

While the NO_ECC bit of the CROMCTL4 register is set to 1, a ‘pass’ result in pre-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.

23.4.4 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-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

23.4.5 Buffering Format

Figure 23.15 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.

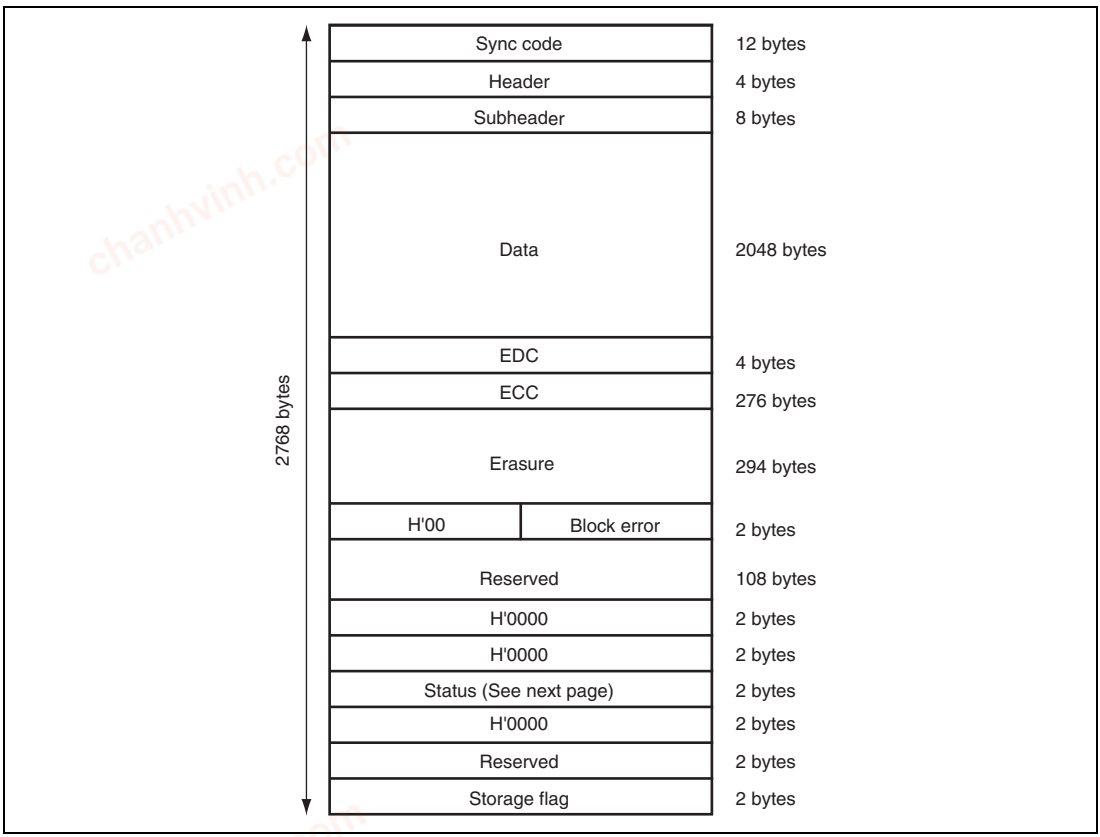


Figure 23.15 Output Data Stream Format

chanhvinh.com

The meanings of bits in the two-byte status field shown in figure 23.15 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-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 (1/75) are non-sequential)

The value of the storage flag field in figure 23.15 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 incrementation reaches H'FFFF. Note that the upper byte and lower byte in the storage flag are swapped.

chanhvinh.com

23.4.6 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 sectors, 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 23.16 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.

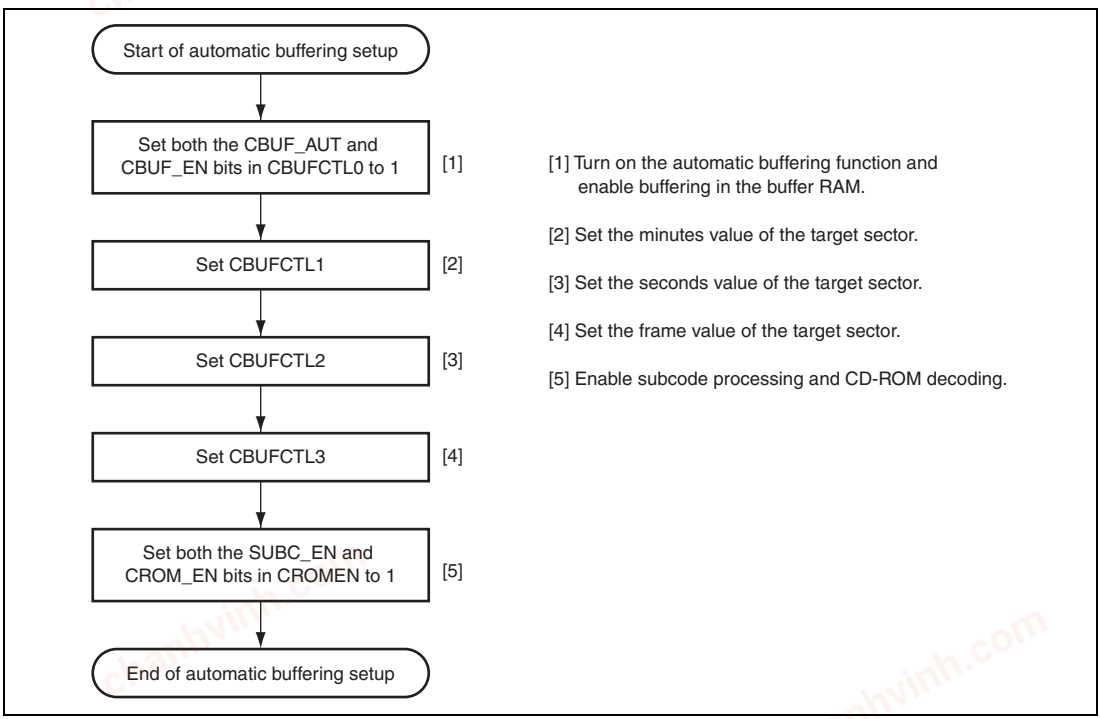


Figure 23.16 Example of Setting Up Automatic Buffering

chanhvinh.com

(2) Setting Up Manual Buffering

Figure 23.17 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.

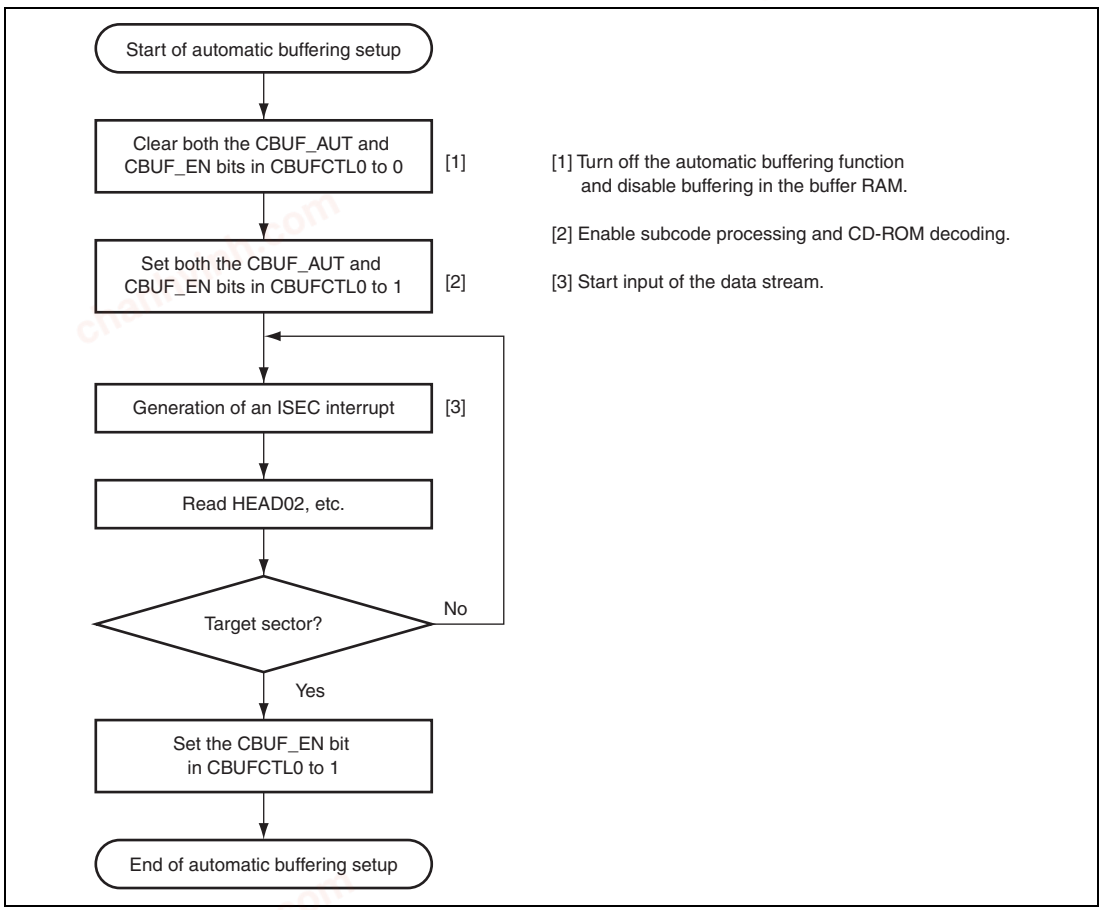


Figure 23.17 Example of Setting Up Manual Buffering

chanhvinh.com

23.5 Interrupt Sources

23.5.1 Interrupt and DMA Transfer Request Signals

Table 23.3 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 23.3 Interrupt and DMA Transfer Request Signals

Name	Condition	Connected To
ISEC	Transitions 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.

chanhvinh.com

(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

(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 23.15 have been transferred, the request signal is negated once. 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.

chanhvinh.com

23.5.2 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.

23.6 Usage Notes

23.6.1 Stopping and Resuming Buffering Alone during Decoding

When the data of the output stream are being not 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 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 at the same time.

23.6.2 When CROMST0 Status Register Bits are Set

1. When the ST_SECS bit in the CROMST0 register becomes set, 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 set, stop decoding immediately and retry from two sectors before the sector that was being decoded.

chanhvinh.com

23.6.3 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 above + 7 as the MSF value for the target sector. The start sector address will be the address where RUN_OUT is stored + 7 when CBUF_LINK = 0, and the address where RUN_OUT is stored when CBUF_LINK = 1.

23.6.4 Stopping and Resuming CD-DSP Operation

When stopping and resuming the stream data input to the CD-ROM decoder, note that the input data stream does not stop immediately before a sync code and that the CD-ROM decoder may recognize the data as incorrect when the input stream is resumed. This happens because the system holds a combination of the data up to the point where input was stopped and data that is input from the point of resumption. Take care on this point when stopping and resuming input.

23.6.5 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.

chanhvinh.com

23.6.6 Note on Stream Data Transfer (1)

When reading of the stream data is slower than writing of the stream data, the buffer of the CD-ROM decoder will overflow. This causes the CD-ROM decoder to be abnormally stopped. Caution is required in writing and reading of the stream data. Sample combinations of stream data transfer settings are shown below.

Table 23.4 Sample Combinations of Stream Data Transfer Settings

Stream Input	Stream Output
LW/cycle-stealing transfer by direct memory access controller (without padding)	(1) 16-byte/cycle-stealing transfer by direct memory access controller (16 bytes*) (2) Burst transfer by direct memory access controller (16 bytes*, longword, word)
LW/cycle-stealing transfer by direct memory access controller (with padding)	(1) Cycle-stealing transfer by direct memory access controller (16 bytes*, longword) (2) Burst transfer by direct memory access controller (16 bytes*, longword, word)
LW write by CPU	(1) Cycle-stealing transfer by direct memory access controller (16 bytes*, longword, word) (2) Burst transfer by direct memory access controller (16 bytes*, longword, word)

Note: * Set bit 25 in the DMA channel control register (CHCR_n) to 1, as well as making the regular settings for 16-byte transfer.

23.6.7 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.

chanhvinh.com

Section 24 A/D Converter

This LSI includes a 10-bit successive-approximation A/D converter allowing selection of up to eight analog input channels.

24.1 Features

- Resolution: 10 bits
- Input channels: four channels in the SH7262 Group and eight channels in the SH7264 Group
- Minimum conversion time: 4.0 μs per channel
- Absolute accuracy: ±4 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 (one to four channels in the SH7262 Group)
  - — Scan mode: Continuous A/D conversion on one to four channels or on one to eight channels (one to four channels in the SH7262 Group)
- 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
  - — Software
  - — Conversion start trigger from the multi-function timer pulse unit 2
  - — External trigger signal
- Interrupt source

  An A/D conversion end interrupt (ADI) request can be generated on completion of A/D conversion.
- Module standby mode can be set

chanhvinh.com

Figure 24.1 shows a block diagram of the A/D converter.

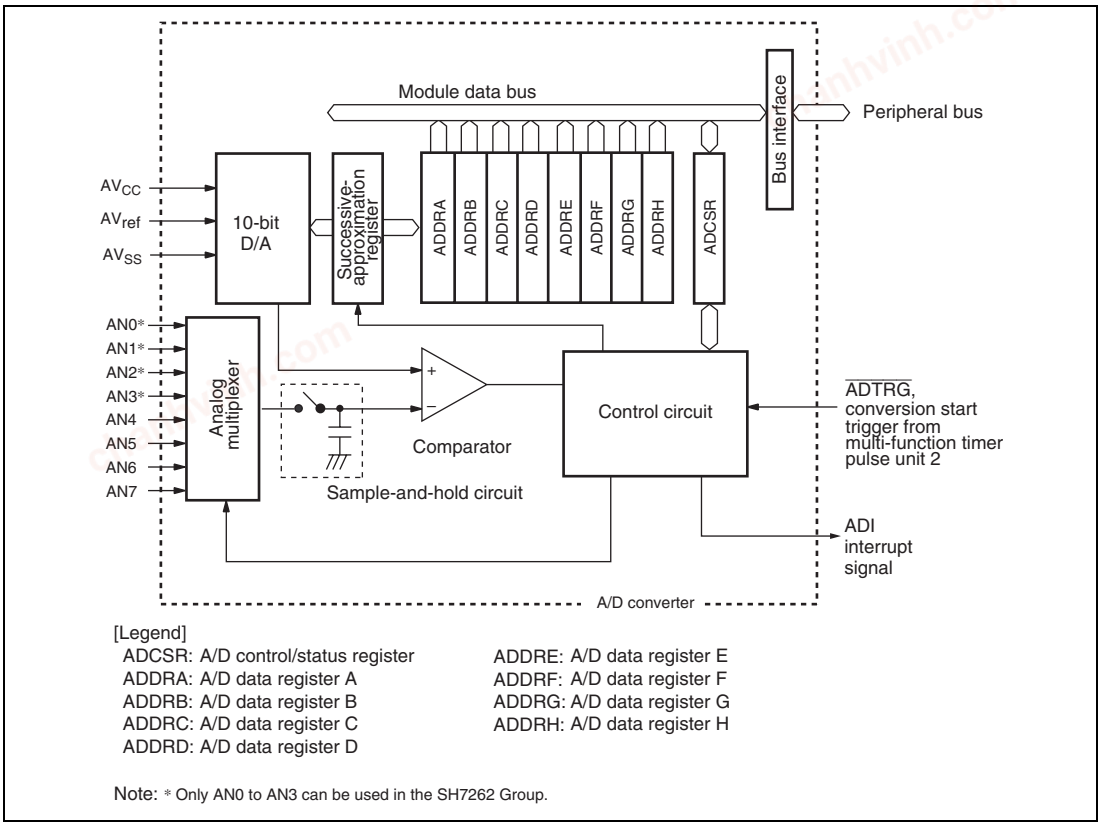


Figure 24.1 Block Diagram of A/D Converter

chanhvinh.com

24.2 Input/Output Pins

Table 24.1 shows the A/D converter pins.

Table 24.1 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

Note: * Only analog input pins 0 to 3 (AN0 to AN3) can be used in the SH7262 Group.

chanhvinh.com

24.3 Register Descriptions

The A/D converter has the following registers.

Table 24.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
A/D data register A	ADDRA	R	H'0000	H'FFFF9800	16
A/D data register B	ADDRB	R	H'0000	H'FFFF9802	16
A/D data register C	ADDRC	R	H'0000	H'FFFF9804	16
A/D data register D	ADDRD	R	H'0000	H'FFFF9806	16
A/D data register E	ADDRE	R	H'0000	H'FFFF9808	16
A/D data register F	ADDRF	R	H'0000	H'FFFF980A	16
A/D data register G	ADDRG	R	H'0000	H'FFFF980C	16
A/D data register H	ADDRH	R	H'0000	H'FFFF980E	16
A/D control/status register	ADCSR	R/W	H'0000	H'FFFF9820	16

chanhvinh.com

24.3.1 A/D Data Registers A to H (ADDRA to ADDRH)

The sixteen 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 10-bit data, which is transferred for storage into the ADDR corresponding to the selected channel. The 10 bits of the result are stored in the upper bits (bits 15 to 6) of ADDR. Bits 5 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 24.3 indicates the pairings of 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 6		All 0	R	Bit data (10 bits)
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Table 24.3 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

Note: * Only AN0 to AN3 can be used in the SH7262 Group.

chanhvinh.com

24.3.2 A/D Control/Status Register (ADCSR)

ADCSR is a 16-bit readable/writable register that selects the 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 the selected channels in multi mode • A/D conversion ends for 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 made. 0: A/D end interrupt request (ADI) is disabled 1: A/D end interrupt request (ADI) is enabled

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
13	ADST	0	R/W	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, software standby mode or module 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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
8 to 6	CKS[2:0]	000	R/W	Clock Select These bits select the A/D conversion time.*2 Set the A/D conversion time while A/D conversion is halted (ADST = 0). 000: Conversion time = 212 t_{cyc} (maximum) 001: Conversion time = 276 t_{cyc} (maximum) 010: Conversion time = 308 t_{cyc} (maximum) 011: Conversion time = 340 t_{cyc} (maximum) 100: Conversion time = 404 t_{cyc} (maximum) 101: Conversion time = 548 t_{cyc} (maximum) 110, 111: Setting prohibited
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: AD conversion on 1 to 8 channels*3 110: Scan mode: A/D conversion on 1 to 4 channels 111: Scan mode: AD conversion on 1 to 8 channels*3

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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*3	100: AN4*3	100: AN0 to AN4*3
101: AN5*3	101: AN4, AN5*3	101: AN0 to AN5*3
110: AN6*3	110: AN4 to AN6*3	110: AN0 to AN6*3
111: AN7*3	111: AN4 to AN7*3	111: AN0 to AN7*3

[Legend]

x: Don't care

Notes: 1. Only 0 can be written to clear the flag after 1 is read.
Please note that ADF flag becomes "0" in the following cases, too.
(1) Reading the state of ADF = 1 with CPU.
(2) Clearing ADF flag by having read ADDR with DMAC
(3) Set of ADF flag according to A/D conversion end
(4) Writing 0 in the ADF flag with CPU
2. Set the A/D conversion time to minimum or more values to meet the absolute accuracy of the A/D conversion characteristics.
3. Settings prohibited in the SH7262 Group.

chanhvinh.com

24.4 Operation

The A/D converter uses the successive-approximation method, and the resolution is 10 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.

24.4.1 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.

chanhvinh.com

Typical operations when a single channel (AN1) is selected in single mode are described next. Figure 24.2 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

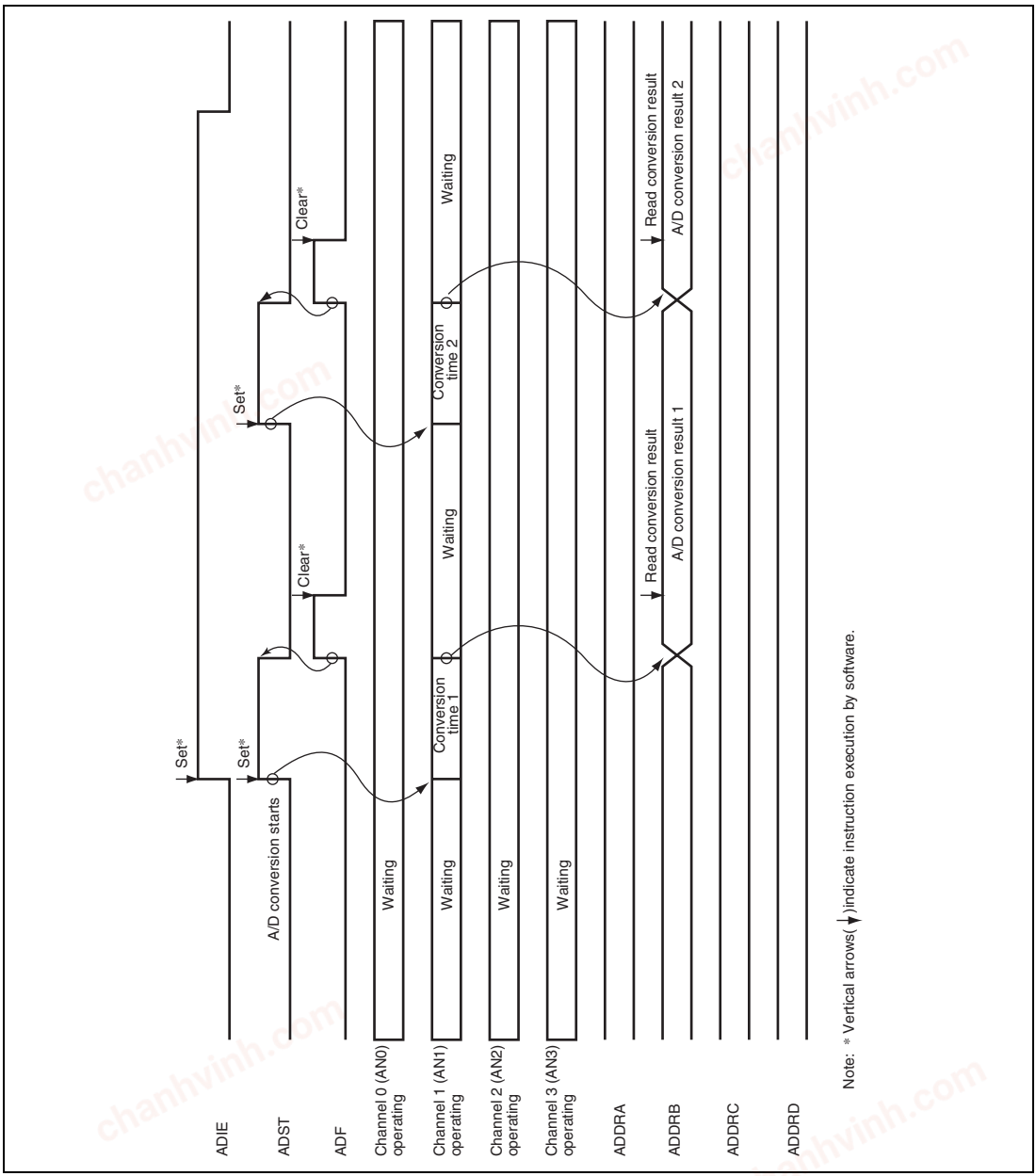


Figure 24.2 Example of A/D Converter Operation (Single Mode, One Channel (AN1) Selected)

chanhvinh.com

24.4.2 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 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 24.3 shows a timing diagram for this example.

1. Multi mode is selected (MDS2 = 1, MDS1 = 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 cleared to 0.
6. If the ADIE bit is set to 1 at this time, an ADI interrupt is requested.

chanhvinh.com

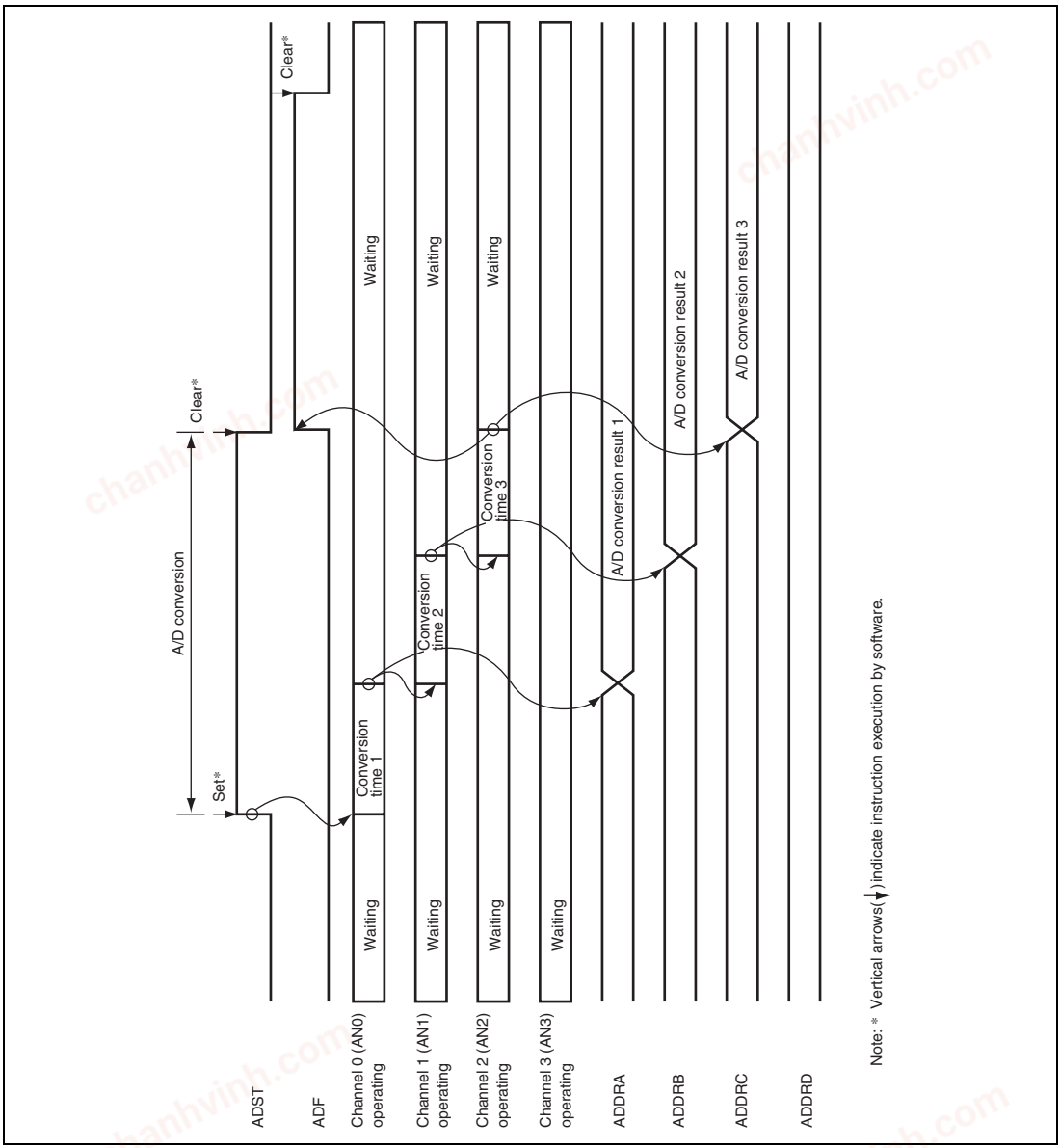


Figure 24.3 Example of A/D Converter Operation (Multi Mode, Three Channels (AN0 to AN2) Selected)

chanhvinh.com

24.4.3 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 24.4 shows a timing diagram for this example.

1. Scan mode is selected (MDS2 = 1, MDS1 = 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.

chanhvinh.com

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

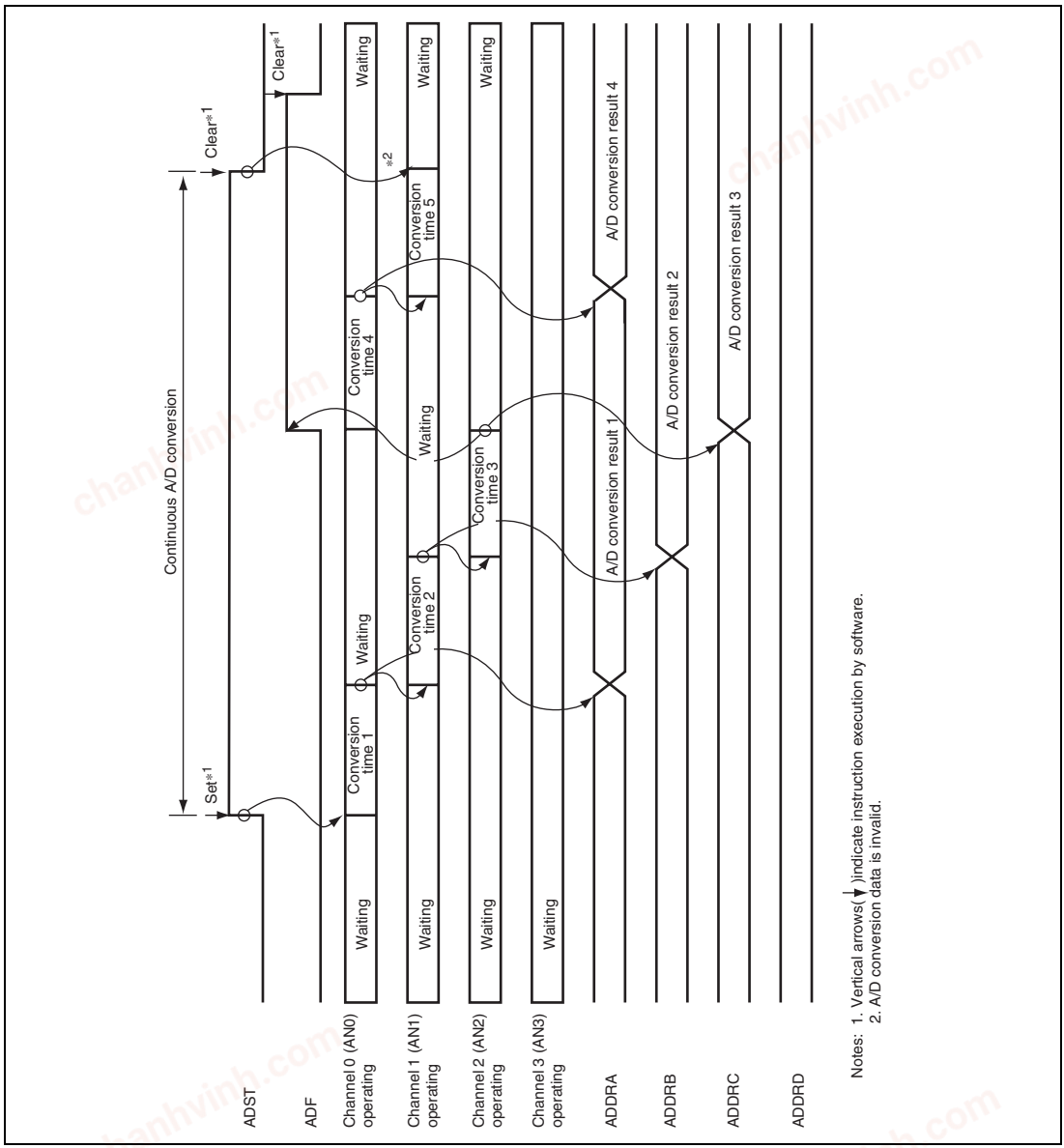


Figure 24.4 Example of A/D Converter Operation (Scan Mode, Three Channels (AN0 to AN2) Selected)

chanhvinh.com

24.4.4 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.

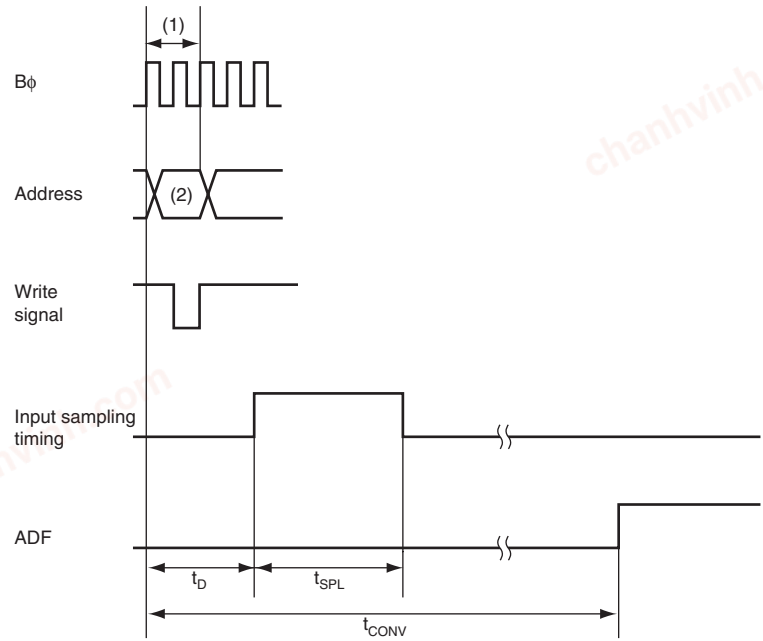
24.4.5 Input Sampling and A/D Conversion Time

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 24.5 shows the A/D conversion timing. Table 24.4 indicates the A/D conversion time.

As indicated in figure 24.5, 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 24.4.

In multi mode and scan mode, the values given in table 24.4 apply to the first conversion. In the second and subsequent conversions, time is the values given in table 24.5.

chanhvinh.com



[Legend]
(1): ADCSR write cycle
(2): ADCSR address
t_D: A/D conversion start delay time
t_{SPL}: Input sampling time
t_{CONV}: A/D conversion time

Figure 24.5 A/D Conversion Timing

chanhvinh.com

Table 24.4 A/D Conversion Time (Single Mode)

		CKS2 = 0												CKS2 = 1					
		CKS1 = 0						CKS1 = 1						CKS1 = 0					
		CKS0 = 0			CKS0 = 1			CKS0 = 0			CKS0 = 1			CKS0 = 0			CKS0 = 1		
Item	**Symbol**	**Min.**	**Typ.**	**Max.**	**Min.**	**Typ.**	**Max.**	**Min.**	**Typ.**	**Max.**	**Min.**	**Typ.**	**Max.**	**Min.**	**Typ.**	**Max.**	**Min.**	**Typ.**	**Max.**
A/D conversion start delay time	t_D	12	—	18	22	—	28	24	—	30	26	—	32	28	—	34	38	—	52
Input sampling time	t_{SPL}	—	50	—	—	66	—	—	74	—	—	82	—	—	98	—	—	130	—
A/D conversion time	t_{CONV}	206	—	212	270	—	276	300	—	304	330	—	340	392	—	404	534	—	548

Note: Values in the table are represented in terms of t_{cyc} (CKIO clock output cycle time).

Table 24.5 A/D Conversion Time (Multi Mode and Scan Mode)

CKS2	CKS1	CKS0	Conversion Time (t_{cyc})
0	0	0	192 (constant)
		1	256 (constant)
	1	0	288 (constant)
		1	320 (constant)
1	0	0	384 (constant)
		1	512 (constant)

Note: Values in the table are represented in terms of t_{cyc} (CKIO clock output cycle time).

chanhvinh.com

24.4.6 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 24.6 shows the timing.

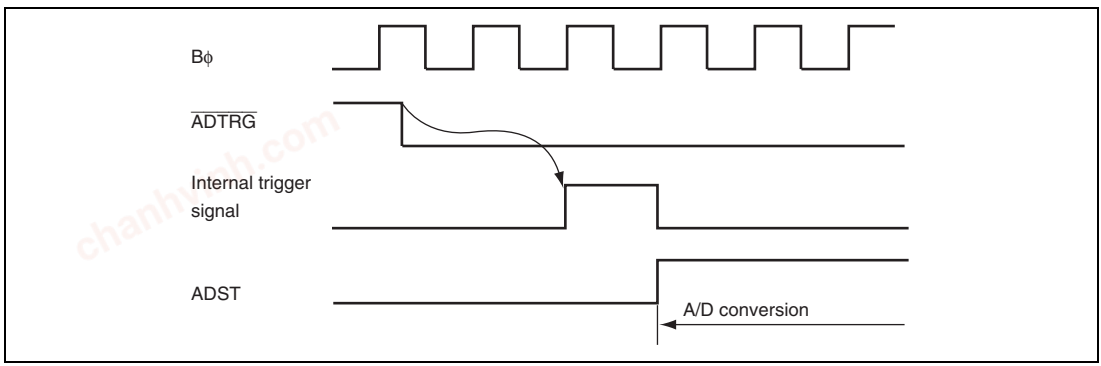


Figure 24.6 External Trigger Input Timing

chanhvinh.com

24.5 Interrupt Sources and DMA Transfer Request

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 is performed only once. In the case of A/D conversion on multiple channels in scan mode or multi mode, setting the DMA transfer count 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 transfer count.

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.

Table 24.6 Relationship 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

chanhvinh.com

24.6 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 10-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 24.7. In the figure, the 10-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'0000000000 (000 in the figure) to B'000000001 (001 in the figure)(figure 24.7, item (1)). Full-scale error is the deviation between actual and ideal A/D conversion characteristics when the digital output value changes from B'1111111110 (110 in the figure) to the maximum B'1111111111 (111 in the figure)(figure 24.7, item (2)). Quantization error is the intrinsic error of the A/D converter and is expressed as 1/2 LSB (figure 24.7, item (3)). Nonlinearity error is the deviation between actual and ideal A/D conversion characteristics between zero voltage and full-scale voltage (figure 24.7, item (4)). Note that it does not include offset, full-scale, or quantization error.

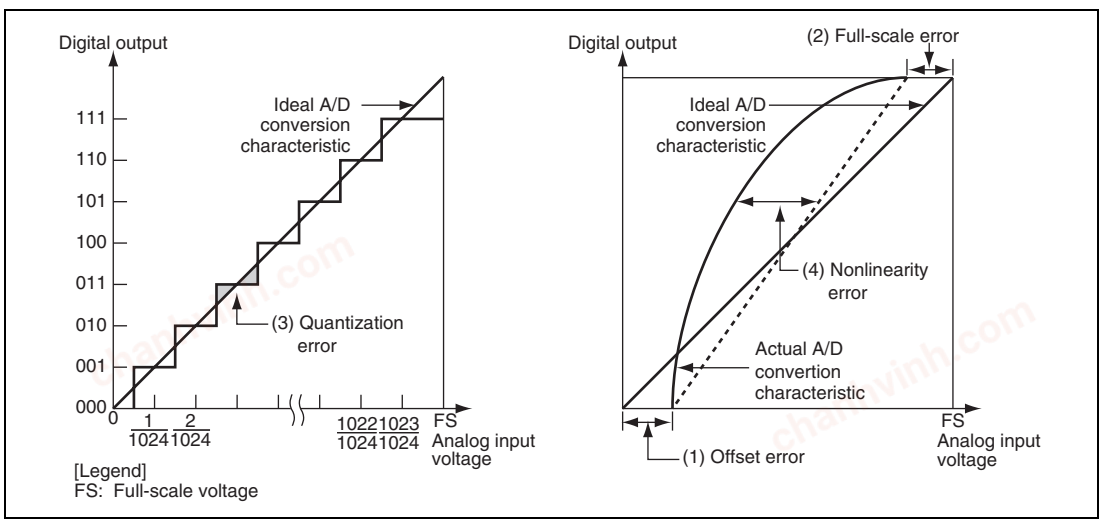


Figure 24.7 Definitions of A/D Conversion Accuracy

chanhvinh.com

24.7 Usage Notes

When using the A/D converter, note the following points.

24.7.1 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 33, Power-Down Modes.

24.7.2 Setting Analog Input Voltage

Permanent damage to the LSI may result if the following voltage ranges are exceeded.

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 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.

24.7.3 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.

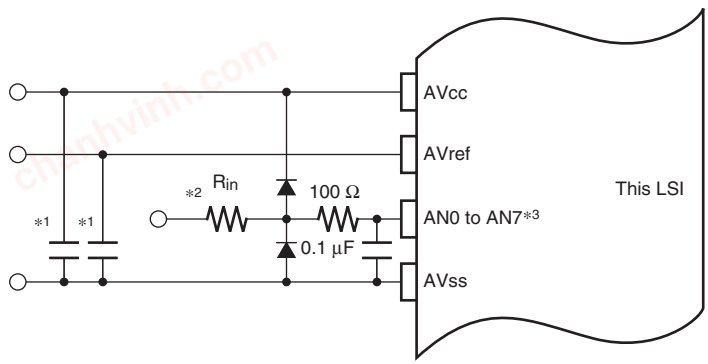
Digital circuitry must be isolated from the analog input signals (AN0 to AN3), analog reference voltage (AVref), and analog power supply (AVcc) by the analog ground (AVss). Also, the analog ground (AVss) should be connected at one point to a stable digital ground (Vss) on the board.

chanhvinh.com

24.7.4 Processing of Analog Input Pins

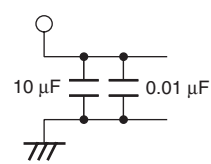
To prevent damage from voltage surges at the analog input pins (AN0 to AN7), connect an input protection circuit like the one shown in figure 24.8. The circuit shown also includes a CR filter to suppress noise. This circuit is shown as an example; the circuit constants should be selected according to actual application conditions.

Figure 24.9 shows an equivalent circuit diagram of the analog input ports and table 24.7 lists the analog input pin specifications.



Notes: Values are reference values.

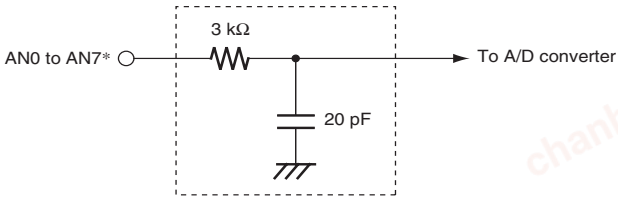
1.



2. R_{in}: Input impedance
3. Only AN0 to AN3 can be used in the SH7262 Group.

Figure 24.8 Example of Analog Input Protection Circuit

chanhvinh.com



Note: Values are reference values.
* Only AN0 to AN3 can be used in the SH7262 Group.

Figure 24.9 Analog Input Pin Equivalent Circuit

Table 24.7 Analog Input Pin Ratings

Item	Min.	Max.	Unit
Analog input capacitance	—	20	pF
Allowable signal-source impedance	—	5	kΩ

24.7.5 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 5 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 5 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 3 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 24.10). When converting a high-speed analog signal, a low-impedance buffer should be inserted.

chanhvinh.com

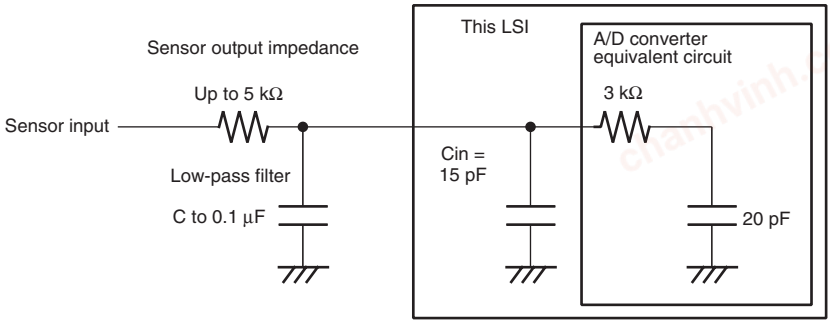


Figure 24.10 Example of Analog Input Circuit

24.7.6 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 mounted on the board do not pick up interference from digital signals (i.e., by acting as antennae).

24.7.7 A/D Conversion in Deep Standby Mode

Before entering deep standby mode, disable A/D conversion by clearing the ADST bit to 0. If the LSI enters deep standby mode with A/D conversion enabled, the states on the A/D converter pins are not guaranteed.

24.7.8 Note on Usage in Scan Mode and Multi Mode

Starting conversion immediately after having stopped scan mode or multi mode operation may lead to erroneous results of conversion.

To perform continuous conversion in such cases, set ADST to 0, wait for at least the A/D conversion time for a single channel to elapse, and then start conversion (ADST = 1). (The A/D conversion time for a single channel will vary according to the settings of the ADC registers.)

chanhvinh.com

chanhvinh.com
chanhvinh.com
chanhvinh.com
chanhvinh.com

chanhvinh.com

Section 25 NAND Flash Memory Controller

The NAND flash memory controller provides interfaces for an external NAND-type flash memory. To take measures for errors specific to flash memory, the NAND flash memory controller supports the ECC generation and error detection functions.

Up to 4-symbol ECC generator, error detector, and hardware error pattern generator have been provided in addition to the 3-symbol ECC detector of the earlier products.

25.1 Features

(1) NAND-Type Flash Memory Interface

- Interface directly connectable to NAND-type flash memory
- Read or write in sector units (512 + 16 bytes) and ECC processing executed
- Read or write in byte units
- Supports large-block (2048 + 64 bytes) flash memory*
- Supports addresses for 2 Gbits and more by extension to 5-byte addresses

Note: * This module handles 512 + 16 bytes as a sector. For products with 2048 + 64 byte-pages, this module divides a page into 512 +16 bytes units (i.e. four sectors per page) for processing.

(2) Access Modes: This module can select one of the following two access modes.

- Command access mode: Performs an access by specifying a command to be issued from this module to flash memory, address, and data size to be input or output. Read, write, or erasure of data without ECC processing can be achieved.
- Sector access mode: Performs a read or write in sector units by specifying a sector address and controls ECC generation and check. By specifying the number of sectors, the continuous physical sectors can be read or written.

chanhvinh.com

(3) Sectors and Control Codes

- A sector is the basic unit of access and comprised of 512-byte data and 16-byte control code fields. The control code field includes 8-byte ECC when the 3-symbol ECC circuit is used, and 10-byte ECC when the 4-symbol ECC circuit is used.
- The position of the ECC in the control code field can be specified in 4-byte units when the 3-symbol ECC circuit is used, and in 1-byte units when the 4-symbol ECC circuit is used.
- User information can be written to the part of the control code field where ECC is not placed.

(4) 3-Symbol ECC

- 64 bits (8 bytes) of ECC is added to a sector, which consists of 512-byte data + 0/4/8-byte control code.
- Error correction and detection is up to three errors (30 bits at maximum) at random positions.
- In a write operation, ECC is generated for the data and control code preceding the ECC. The control code following the ECC is not considered.
- In a read operation, an ECC error is checked for data and control code preceding the ECC. The ECC on the control code in the FIFO are the results of checking replaced by the ECC circuit, not the ECC read from flash memory.
- Error correction is not performed even when an ECC error occurs. Error corrections must be performed by software.

(5) 4-Symbol ECC

- 80 bits (10 bytes) of ECC is added to a sector, which consists of 512-byte data + 1-to 6-byte control code.
- Error correction and detection is up to four errors (40 bits at maximum) at random positions.
- In a write operation, ECC is generated for the data and control code preceding the ECC. The control code following the ECC is not considered.
- In a read operation, an ECC error is checked for data and control code preceding the ECC. The ECC on the control code in the FIFO are the results of checking replaced by the ECC circuit, not the ECC read from flash memory.
- The 4-symbol ECC circuit of this module has the capability of error correction pattern generation by hardware, which is executed on a sector-by-sector basis.
- In the error correction by hardware, addresses indicating the error positions and an error pattern for correcting the errors are output. Data replacement must be performed by software.

chanhvinh.com

(6) Data Error

- When a program error or erase error occurs, the error is reflected on the error source flags. Interrupts for each source can be specified.
- When a read error occurs, an ECC in the control code is other than 0. This read error is reflected on the ECC error source flag.
- When an ECC error occurs, perform an error correction, specify another sector to be replaced, and copy the contents of the block to another sector as required.

(7) Data Transfer FIFO and Data Register

- The 224-byte data FIFO register (FLDTFIFO) is incorporated for data transfer of flash memory.
- The 32-byte control code FIFO register (FLECFIFO) is incorporated for data transfer of control code.

(8) DMA Transfer

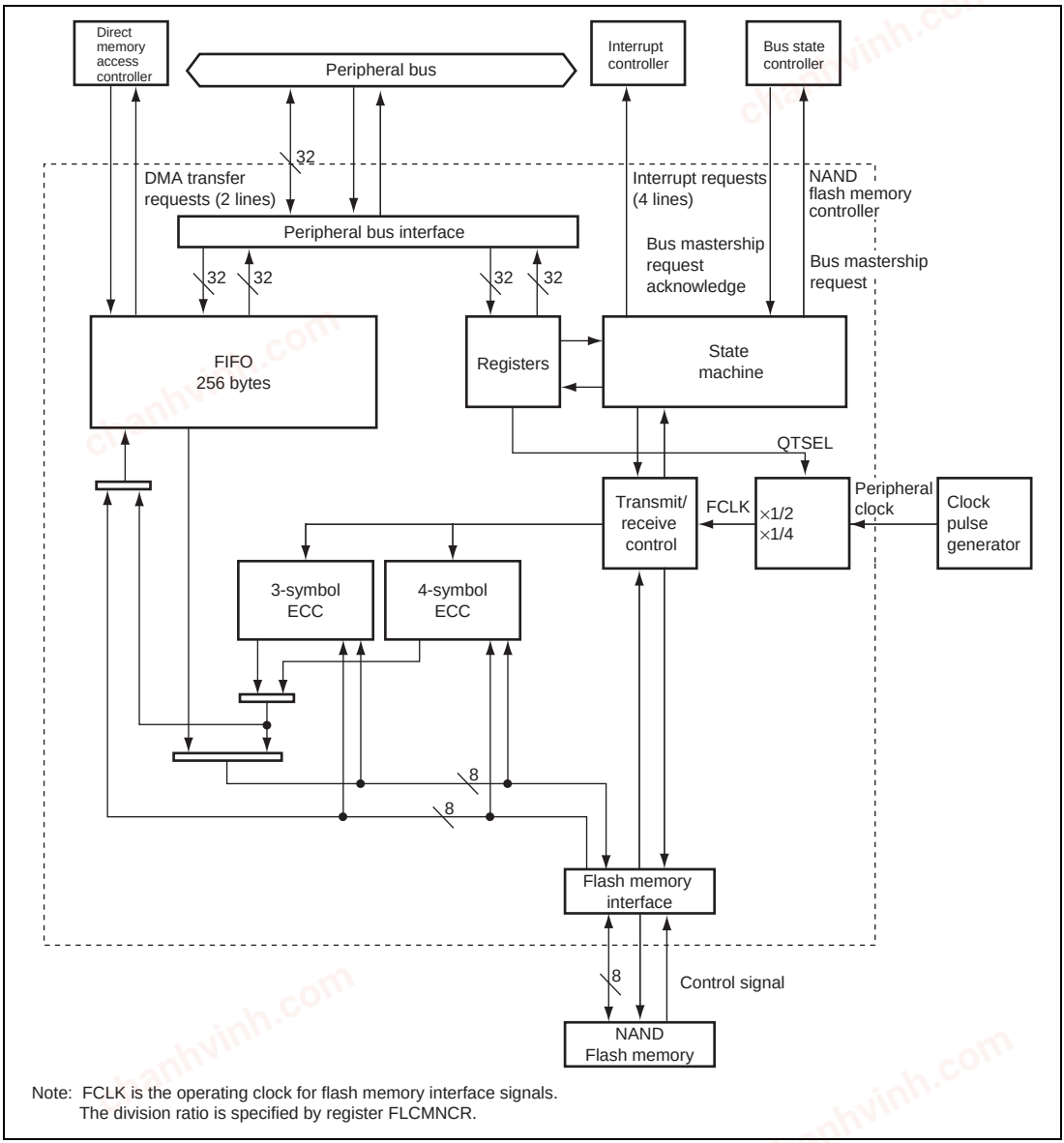
- By individually specifying the destinations of data and control code of flash memory to the direct memory access controller, data and control code can be sent to different areas.

(9) Access Time

- The operating clock (FCLK) on the pins for the NAND-type flash memory is generated by dividing the peripheral clock (Pϕ). The division ratio can be specified by the QTSEL bit in the common control register (FLCMNCR).
- Before changing the clock pulse generator configuration, this module must be placed in a module stop state.
- In NAND-type flash memory, the $\overline{\text{FRE}}$ and $\overline{\text{FWE}}$ pins operate at the frequency of FCLK. The operating frequency must be specified within the maximum operating frequency of memory to be connected.

chanhvinh.com

Figure 25.1 shows a block diagram.



Note: FCLK is the operating clock for flash memory interface signals.
The division ratio is specified by register FLCMNCR.

Figure 25.1 Block Diagram

chanhvinh.com

25.2 Input/Output Pins

The pin configuration of is listed in table 25.1.

Table 25.1 Pin Configuration

Pin Name	I/O	Corresponding Flash Memory Pin: NAND Type	Function
$\overline{\text{FCE}}$	Output	$\overline{\text{CE}}$	Flash Memory Chip Enable Enables flash memory connected to this LSI.
NAF7 to NAF0	I/O	I/O7 to I/O0	Flash Memory Data I/O pins for command, address, and data.
FCLE	Output	CLE	Flash Memory Command Latch Enable Asserted when a command is output.
FALE	Output	ALE	Flash Memory Address Latch Enable Asserted when an address is output and negated when data is input or output.
$\overline{\text{FRE}}$	Output	$\overline{\text{RE}}$	Flash Memory Read Enable Reads data at the falling edge of $\overline{\text{RE}}$.
$\overline{\text{FWE}}$	Output	$\overline{\text{WE}}$	Flash Memory Write Enable Flash memory latches a command, address, and data at the rising edge of $\overline{\text{WE}}$.
FRB	Input	R/$\overline{\text{B}}$	Flash Memory Ready/Busy Indicates ready state at high level; indicates busy state at low level.
—*	—	$\overline{\text{WP}}$	Write Protect/Reset When this pin goes low, erroneous erasure or programming at power on or off can be prevented.
—*	—	$\overline{\text{SE}}$	Spare Area Enable Used to access spare area. This pin must be fixed at low in sector access mode.

Note: * Not supported in this LSI.

chanhvinh.com

25.3 Register Descriptions

Table 25.2 shows the register configuration.

Table 25.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Common control register	FLCMNCR	R/W	H'00100001	H'FFFF4000	32
Command control register	FLCMDCR	R/W	H'00000000	H'FFFF4004	32
Command code register	FLCMCDR	R/W	H'00000000	H'FFFF4008	32
Address register	FLADR	R/W	H'00000000	H'FFFF400C	32
Address register 2	FLADR2	R/W	H'00000000	H'FFFF403C	32
Data register	FLDATAR	R/W	H'00000000	H'FFFF4010	32
Data counter register	FLDTCNTR	R/W	H'00000000	H'FFFF4014	32
Interrupt DMA control register	FLINTDMACR	R/W	H'00000000	H'FFFF4018	32
Ready busy timeout setting register	FLBSYTMR	R/W	H'00000000	H'FFFF401C	32
Ready busy timeout counter	FLBSYCNT	R	H'00000000	H'FFFF4020	32
Data FIFO register	FLDTFIFO	R/W	H'xxxxxxxx	H'FFFF4050	32
Control code FIFO register	FLECFIFO	R/W	H'xxxxxxxx	H'FFFF4060	32
Transfer control register	FLTRCR	R/W	H'00	H'FFFF402C	8
Bus hold time setting register	FLHOLDCR	R/W	H'00000000	H'FFFF4038	32
4-symbol ECC processing result register 1	FL4ECCRES1	R	H'xxxxxxxx	H'FFFF4080	32
4-symbol ECC processing result register 2	FL4ECCRES2	R	H'xxxxxxxx	H'FFFF4084	32
4-symbol ECC processing result register 3	FL4ECCRES3	R	H'xxxxxxxx	H'FFFF4088	32
4-symbol ECC processing result register 4	FL4ECCRES4	R	H'xxxxxxxx	H'FFFF408C	32
4-symbol ECC control register	FL4ECCCR	R/W	H'00000000	H'FFFF4090	32
4-symbol ECC error count register	FL4ECCCNT	R/W	H'00000000	H'FFFF4094	32

chanhvinh.com

25.3.1 Common Control Register (FLCMNCR)

FLCMNCR is a 32-bit readable/writable register that specifies access mode, and other items.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	ECCPOS [2]	4ECCCN TEN	4ECCEN	4ECCCO RRECT	BUSYON	-	-	SNAND	QT SEL	-
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/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	ECCPOS[1:0]		ACM[1:0]		NAND WF	-	-	-	-	-	CE	-	-	-
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/W	R/W	R/W	R	R	R	R	R	R/W	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	ECCPOS [2]	0	R/W	See the description of ECCPOS[1:0] at bits 13 and 12.
24	4ECCCN TEN	0	R/W	4-Symbol ECC Error Count Selects whether to output to the FL4ECCCNT register the total number of errors found in the sectors that have been read and the maximum number of errors found in a single sector. 0: Error counting is not performed. 1: When 4-symbol ECC circuit is used, the total number of errors found in the read sectors and the maximum number of errors in a sector are output to FL4ECCCNT. Note: When this bit is set to 1, the 4ECCCORRECT bit must be cleared to 0.
23	4ECCEN	0	R/W	4-Symbol ECC Circuit Enable Enables the 4-symbol ECC circuit by setting this bit to 1 in sector access mode. 0: 3-symbol ECC circuit is enabled. 1: 4-symbol ECC circuit is enabled.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
22	4ECCCORRECT	0	R/W	4-Symbol ECC Circuit Correction Execution Specifies to execute error correction for a single sector when the 4-symbol ECC circuit is used. This module suspends sector reading on detection of an ECC error and starts error pattern generation by the 4-symbol ECC circuit. 0: Error pattern is not output but only ECC is output. 1: Reading of sectors is suspended on detection of an ECC error.
21	BUSYON	0	R/W	Busy Select Specifies whether to release the external bus mastership while the FRB pin is busy. The $\overline{\text{FCE}}$ pin, however, is negated regardless of the busy/ready state upon completion of a necessary process. For details, see section 25.7.1, External Bus Mastership Release Timing. 0: Holds the bus mastership while the FRB pin is busy. 1: Releases the bus mastership while the FRB pin is busy. Note: Some flash memory devices do not allow the $\overline{\text{FCE}}$ pin to be negated during the busy state.
20	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
18	SNAND	0	R/W	Large-Capacity NAND Flash Memory Select This bit is used to specify 1-Gbit or larger NAND flash memory with the page configuration of 2048 + 64 bytes. 0: When flash memory with the page configuration of 512 + 16 bytes is used. 1: When NAND flash memory with the page configuration of 2048 + 64 is used.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
17	QTSEL	0	R/W	Select Dividing Rates for Flash Clock Selects the dividing rate of clock FCLK in the flash memory. 0: Divides a clock (Pϕ) provided from the clock pulse generator by two and uses it as FCLK. 1: Divides a clock (Pϕ) provided from the clock pulse generator by four and uses it as FCLK.
16 to 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	ECCPOS [1:0]	00	R/W	ECC Embedding Position Specification ECCPOS[2:0] (bits 25, 13, and 12 of this register) specifies the position to place ECC in the control code field when 3- or 4-symbol ECC circuit is used • When 4ECCEN = 0 (ECC is eight bytes) 000: Places ECC with offset of 512 bytes in a sector 001: Places ECC with offset of 516 bytes in a sector 010: Places ECC with offset of 520 bytes in a sector Other than above: Setting prohibited • When 4ECCEN = 1 (ECC is ten bytes) 000: Places ECC with offset of 518 bytes in a sector 001: Places ECC with offset of 517 bytes in a sector 010: Places ECC with offset of 516 bytes in a sector 011: Places ECC with offset of 515 bytes in a sector 100: Places ECC with offset of 514 bytes in a sector 101: Places ECC with offset of 513 bytes in a sector 110: Places ECC with offset of 512 bytes in a sector 111: Setting prohibited

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
11, 10	ACM[1:0]	00	R/W	Access Mode Specification 1 and 0 Specify access mode. 00: Command access mode 01: Sector access mode 10: Setting prohibited 11: Setting prohibited
9	NANDWF	0	R/W	NAND Wait Insertion Operation 0: Performs address or data input/output in one FCLK cycle 1: Performs address or data input/output in two FCLK cycles
8 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	CE	0	R/W	Chip Enable 0: Disables the chip (Outputs high level to the $\overline{\text{FCE}}$ pin) 1: Enables the chip (Outputs low level to the $\overline{\text{FCE}}$ pin)
2, 1	—	All 0	R	Reserved These bits are 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.

chanhvinh.com

25.3.2 Command Control Register (FLCMDCR)

FLCMDCR is a 32-bit readable/writable register that issues a command in command access mode, specifies address issue, and specifies source or destination of data transfer. In sector access mode, FLCMDCR specifies the number of sector transfers.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADR CNT2	SCTCNT[19:16]				ADR MD	CDS RC	DOSR	-	-	SEL RW	DOA DR	ADRCNT[1:0]		DOC MD2	DOC MD1
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
	SCTCNT[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	ADRCNT2	0	R	Address Issue Byte Count Specification 2 Specifies the number of bytes for the address data to be issued in address stage. This bit is used together with ADRCNT[1:0]. 0: Issue the address of byte count, specified by ADRCNT[1:0]. 1: Issue 5-byte address. ADRCNT[1:0] should be set to 00.
30 to 27	SCTCNT [19:16]	0000	R/W	Sector Transfer Count Specification [19:16] These bits are extended bits of the sector transfer count specification bits (SCTCNT) 15 to 0. SCTCNT[19:16] and SCTCNT[15:0] are used together to operate as SCTCNT[19:0], the 20-bit counter.
26	ADRMD	0	R/W	Sector Access Address Specification This bit is invalid in command access mode. This bit is valid only in sector access mode. 0: The value of the address register is handled as a sector address. Use this value usually in sector access. 1: The value of the address register is output as the address of flash memory. Note: Clear this bit to 0 in continuous sector access.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
25	CDSRC	0	R/W	Data Buffer Specification Specifies the data buffer to be read from or written to in the data stage in command access mode. 0: Specifies FLDATAR as the data buffer. 1: Specifies FLDTFIFO as the data buffer.
24	DOSR	0	R/W	Status Read Check Specifies whether or not the status read is performed after the second command has been issued in command access mode. 0: Performs no status read 1: Performs status read
23, 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21	SELRW	0	R/W	Data Read/Write Specification Specifies the direction of read or write in data stage. 0: Read 1: Write
20	DOADR	0	R/W	Address Stage Execution Specification Specifies whether or not the address stage is executed in command access mode. 0: Performs no address stage 1: Performs address stage
19, 18	ADRCNT [1:0]	00	R/W	Address Issue Byte Count Specification [1:0] Specify the number of bytes for the address data to be issued in address stage. 00: Issue 1-byte address 01: Issue 2-byte address 10: Issue 3-byte address 11: Issue 4-byte address

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
17	DOCMD2	0	R/W	Second Command Stage Execution Specification Specifies whether or not the second command stage is executed in command access mode. 0: Does not execute the second command stage 1: Executes the second command stage
16	DOCMD1	0	R/W	First Command Stage Execution Specification Specifies whether or not the first command stage is executed in command access mode. 0: Does not execute the first command stage 1: Executes the first command stage
15 to 0	SCTCNT [15:0]	H'0000	R/W	Sector Transfer Count Specification [15:0] Specify the number of sectors to be read continuously in sector access mode. These bits are counted down for each sector transfer end and stop when they reach 0. These bits are used together with SCTCNT[19:16]. In command access mode, these bits are H'0 0001.

chanhvinh.com

25.3.3 Command Code Register (FLCMCDR)

FLCMCDR is a 32-bit readable/writable register that specifies a command to be issued in command access or sector access.

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
	CMD2[7:0]								CMD1[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	CMD2[7:0]	H'00	R/W	Second Command Data Specify a command code to be issued in the second command stage.
7 to 0	CMD1[7:0]	H'00	R/W	First Command Data Specify a command code to be issued in the first command stage.

chanhvinh.com

25.3.4 Address Register (FLADR)

FLADR is a 32-bit readable/writable register that specifies the value to be output as an address.

The address of the size specified by ADRCNT[1:0] in the command control register is output sequentially from ADR1 in byte units. By the sector access address specification bit (ADRMD) of the command control register, it is possible to specify whether the sector number set in the address data bits is converted into an address to be output to the flash memory.

- When ADRMD = 1

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADR4[7:0]								ADR3[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
	ADR2[7:0]								ADR1[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	ADR4[7:0]	H'00	R/W	Fourth Address Data Specify 4th data to be output to flash memory as an address when ADRMD = 1.
23 to 16	ADR3[7:0]	H'00	R/W	Third Address Data Specify 3rd data to be output to flash memory as an address when ADRMD = 1.
15 to 8	ADR2[7:0]	H'00	R/W	Second Address Data Specify 2nd data to be output to flash memory as an address when ADRMD = 1.
7 to 0	ADR1[7:0]	H'00	R/W	First Address Data Specify 1st data to be output to flash memory as an address when ADRMD = 1.

chanhvinh.com

- When ADRMD = 0

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	ADR[25:16]									
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/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 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25 to 0	ADR[25:0]	H'0000 000	R/W	Sector Address Specification Specify a sector number to be accessed when ADRMD = 0. The sector number is converted into an address and is output to flash memory. When the ADRCNT2 bit in FLCMDCR = 1, the ADR[25:0] bits are valid. When the ADRCNT2 bit in FLCMDCR = 0, the ADR[17:0] bits are valid. See figure 25.11 for details. • Large-block products (2048 + 64 bytes) ADR[25:2] specifies the page address and ADR[1:0] specifies the column address in sector units. ADR[1:0] = 00: 0th byte (sector 0) ADR[1:0] = 01: (512 + 16)th byte (sector 1) ADR[1:0] = 00: (1024 + 32)th byte (sector 2) ADR[1:0] = 00: (1536 + 48)th byte (sector 3) • Small-block products (512 + 16 bytes) Only the page address can be specified.

chanhvinh.com

25.3.5 Address Register 2 (FLADR2)

FLADR2 is a 32-bit readable/writable register, and is valid when the ADRCNT2 bit in FLCMDCR is set to 1. FLADR2 specifies an address to be output in command access 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
	-	-	-	-	-	-	-	-	ADR5[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 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	ADR5[7:0]	H'00	R/W	Fifth Address Data Specify 5th data to be output to flash memory as an address when ADRMD = 1.

chanhvinh.com

25.3.6 Data Counter Register (FLDTCNTR)

FLDTCNTR is a 32-bit readable/writable register that specifies the number of bytes to be read or written in command access mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECFLW[7:0]								DTFLW[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
	-	-	-	-	DTCNT[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 24	ECFLW [7:0]	H'00	R	FLECFIFO Access Count Specify the number of longwords in FLECFIFO to be read or written. These bit values are used when the CPU reads from or writes to FLECFIFO. In FLECFIFO read, these bits specify the number of longwords of the data that can be read from FLECFIFO. In FLECFIFO write, these bits specify the number of longwords of unoccupied area that can be written in FLECFIFO.
23 to 16	DTFLW [7:0]	H'00	R	FLDTFIFO Access Count Specify the number of longwords in FLDTFIFO to be read or written. These bit values are used when the CPU reads from or writes to FLDTFIFO. In FLDTFIFO read, these bits specify the number of longwords of the data that can be read from FLDTFIFO. In FLDTFIFO write, these bits specify the number of longwords of unoccupied area that can be written in FLDTFIFO.
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	DTCNT [11:0]	H'000	R/W	Data Count Specification Specify the number of bytes of data to be read or written in command access mode. (Up to 2048 + 64 bytes can be specified.)

chanhvinh.com

25.3.7 Data Register (FLDATAR)

FLDATAR is a 32-bit readable/writable register. It stores input/output data used when 0 is written to the CDSRC bit in FLCMDCR in command access mode. FLDATAR cannot be used for reading or writing of five or more bytes of contiguous data.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DT4[7:0]								DT3[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
	DT2[7:0]								DT1[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	DT4[7:0]	H'00	R/W	Fourth Data Specify the 4th data to be input or output via the NAF7 to NAF0 pins. In write: Specify write data In read: Store read data
23 to 16	DT3[7:0]	H'00	R/W	Third Data Specify the 3rd data to be input or output via the NAF7 to NAF0 pins. In write: Specify write data In read: Store read data
15 to 8	DT2[7:0]	H'00	R/W	Second Data Specify the 2nd data to be input or output via the NAF7 to NAF0 pins. In write: Specify write data In read: Store read data
7 to 0	DT1[7:0]	H'00	R/W	First Data Specify the 1st data to be input or output via the NAF7 to NAF0 pins. In write: Specify write data In read: Store read data

chanhvinh.com

25.3.8 Interrupt DMA Control Register (FLINTDMACR)

FLINTDMACR is a 32-bit readable/writable register that enables or disables DMA transfer requests or interrupts. A transfer request from this module to the direct memory access controller is issued after each access mode has been started.

Bits 9 to 5 are the flag bits that indicate various errors occurred in flash memory access and whether there is a transfer request from the FIFO. Only 0 can be written to these bits. To clear a flag, write 0 to the target flag bit and 1 to the other flag bits.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	4ECE INTE	ECER INTE	-	-	FIFOTRG [1:0]		AC1 CLR	AC0 CLR	DREQ1 EN	DREQ0 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/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
	-	-	-	-	-	-	EC ERB	ST ERB	BTO ERB	TRR EQF1	TRR EQF0	STER INTE	RBER INTE	TE INTE	TR INTE1	TR INTE0
Initial value:	0	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 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25	4ECEINTE	0	R/W	4-Symbol ECC Pattern Generation End Interrupt Enable Enables or disables an interrupt to CPU by 4-symbol ECC pattern generation end. 0: Disables an interrupt to CPU by 4-symbol ECC pattern generation end. 1: Enables an interrupt to CPU by 4-symbol ECC pattern generation end.
24	ECERINTE	0	R/W	ECC Error Interrupt Enable Enables or disables an interrupt to CPU when ECC error occurs. 0: Disables an interrupt to CPU when an ECC error occurs 1: Enables an interrupt to CPU when an ECC error occurs

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
23, 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21, 20	FIFOTRG [1:0]	00	R/W	FIFO Trigger Setting Specify the condition (the byte number) for generation of FLDTFIFO and FLECFIFO transfer requests. • In flash-memory read Issue an interrupt to the CPU or issue a DMA transfer request when FLDTFIFO (FLECFIFO) stores the following number of bytes or more: 00: 4 (4) 01: 16 (16) 10: 128 (4) 11: 128 (16) • In flash-memory programming Issue an interrupt to the CPU or issue a DMA transfer request when FLDTFIFO (FLECFIFO) has the following empty area of bytes or more: 00: 4 (4) 01: 16 (16) 10: 128 (4) 11: 128 (16) Note: For DMA transfer from/to FLDTFIFO, setting 10 and 11 are prohibited.
19	AC1CLR	0	R/W	FLECFIFO Clear Clears FLECFIFO. 0: Retains the FLECFIFO value. In flash-memory access, this bit should be cleared to 0. 1: Clears FLECFIFO. After FLECFIFO has been cleared, this bit should be cleared to 0.
18	AC0CLR	0	R/W	FLDTFIFO Clear Clears FLDTFIFO. 0: Retains the FLDTFIFO value. In flash-memory access, this bit should be cleared to 0. 1: Clears FLDTFIFO. After FLDTFIFO has been cleared, this bit should be cleared to 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
17	DREQ1EN	0	R/W	FLECFIFODMA Request Enable Enables or disables the DMA transfer request issued from FLECFIFO. 0: Disables the DMA transfer request issued from FLECFIFO 1: Enables the DMA transfer request issued from FLECFIFO
16	DREQ0EN	0	R/W	FLDTFIFODMA Request Enable Enables or disables the DMA transfer request issued from FLDTFIFO. 0: Disables the DMA transfer request issued from the FLDTFIFO 1: Enables the DMA transfer request issued from the FLDTFIFO
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	ECERB	0	R/(W)*	ECC Error Indicates the result of ECC error detection. This bit is set to 1 if an ECC error occurs while flash memory is read in sector access mode. This bit is a flag. 1 cannot be written to this bit. Only 0 can be written to clear the flag. 0: Indicates that no ECC error occurs (Latched ECC is all 0.) 1: Indicates that an ECC error occurs

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
8	STERB	0	R/(W)*	Status Error Indicates the result of status read. This bit is set to 1 if the specific bit in the bits STAT[7:0] in FLBSYCNT is set to 1 in status read. This bit is a flag. 1 cannot be written to this bit. Only 0 can be written to clear the flag. 0: Indicates that no status error occurs (the specific bit in the bits STAT[7:0] in FLBSYCNT is 0.) 1: Indicates that a status error occurs For details on the specific bit in STAT7 to STAT0 bits, see section 25.4.7, Status Read.
7	BTOERB	0	R/(W)*	R/$\overline{B}$ Timeout Error This bit is set to 1 if an R/$\overline{B}$ timeout error occurs (the bits RBTIMCNT[19:0] in FLBSYCNT are decremented to 0). This bit is a flag. 1 cannot be written to this bit. Only 0 can be written to clear the flag. 0: Indicates that no R/$\overline{B}$ timeout error occurs 1: Indicates that an R/$\overline{B}$ timeout error occurs
6	TRREQF1	0	R/(W)*	FLECFIFO Transfer Request Flag Indicates that a transfer request is issued from FLECFIFO. This bit is a flag. 1 cannot be written to this bit. Only 0 can be written to clear the flag. 0: Indicates that no transfer request is issued from FLECFIFO 1: Indicates that a transfer request is issued from FLECFIFO

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
5	TRREQF0	0	R/(W)*	FLDTFIFO Transfer Request Flag Indicates that a transfer request is issued from FLDTFIFO. This bit is a flag. 1 cannot be written to this bit. Only 0 can be written to clear the flag. 0: Indicates that no transfer request is issued from FLDTFIFO 1: Indicates that a transfer request is issued from FLDTFIFO
4	STERINTE	0	R/W	Interrupt Enable at Status Error Enables or disables an interrupt request to the CPU when a status error has occurred. 0: Disables the interrupt request to the CPU by a status error 1: Enables the interrupt request to the CPU by a status error
3	RBERINTE	0	RW	Interrupt Enable at R/$\overline{B}$ Timeout Error Enables or disables an interrupt request to the CPU when a timeout error has occurred. 0: Disables the interrupt request to the CPU by an R/$\overline{B}$ timeout error 1: Enables the interrupt request to the CPU by an R/$\overline{B}$ timeout error
2	TEINTE	0	R/W	Transfer End Interrupt Enable Enables or disables an interrupt request to the CPU when a transfer has been ended (TREND bit in FLTRCR). 0: Disables the transfer end interrupt request to the CPU 1: Enables the transfer end interrupt request to the CPU

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
1	TRINTE1	0	R/W	FLECFIFO Transfer Request Enable to CPU Enables or disables an interrupt request to the CPU by a transfer request issued from FLECFIFO. 0: Disables an interrupt request to the CPU by a transfer request from FLECFIFO. 1: Enables an interrupt request to the CPU by a transfer request from FLECFIFO. When the DMA transfer is enabled, this bit should be cleared to 0.
0	TRINTE0	0	R/W	FLDTFIFO Transfer Request Enable to CPU Enables or disables an interrupt request to the CPU by a transfer request issued from FLDTFIFO. 0: Disables an interrupt request to the CPU by a transfer request from FLDTFIFO 1: Enables an interrupt request to the CPU by a transfer request from FLDTFIFO When the DMA transfer is enabled, this bit should be cleared to 0.

Note: * Only 0 can be written to these bits.

25.3.9 Ready Busy Timeout Setting Register (FLBSYTMR)

FLBSYTMR is a 32-bit readable/writable register that specifies the timeout time when the FRB pin is busy.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RBTMOUT[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

chanhvinh.com

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 to 0	RBTMOUT[19:0]	H'00000	R/W	Ready Busy Timeout Specify timeout time (the number of Pϕ clocks) in busy state. When these bits are set to 0, timeout is not generated.

25.3.10 Ready Busy Timeout Counter (FLBSYCNT)

FLBSYCNT is a 32-bit read-only register.

The status of flash memory obtained by the status read is stored in the bits STAT[7:0].

The timeout time set in the bits RBTMOUT[19:0] in FLBSYTMR is copied to the bits RBTIMCNT[19:0] and counting down is started when the FRB pin is placed in a busy state. When values in the RBTIMCNT[19:0] become 0, 1 is set to the BTOERB bit in FLINTDMACR, thus notifying that a timeout error has occurred. In this case, an FLSTE interrupt request can be issued if an interrupt is enabled by the RBERINTE bit in FLINTDMACR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	STAT[7:0]								-	-	-	-	RBTIMCNT[19:16]			
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
	RBTIMCNT[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 24	STAT[7:0]	All 0	R	Indicate the flash memory status obtained by the status read.
23 to 20	—	All 0	R	Reserved These bits are always read as 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
19 to 0	RBTIMCNT[19:0]	H'00000	R	Ready Busy Timeout Counter When the FRB pin is placed in a busy state, the values of the bits RBTMOUT[19:0] in FLBSYTMR are copied to these bits. These bits are counted down while the FRB pin is busy. A timeout error occurs when these bits are decremented to 0.

25.3.11 Data FIFO Register (FLDTFIFO)

FLDTFIFO is used to read or write the data FIFO area.

In DMA transfer, this register must be specified as the destination or source.

Note that the direction of read or write specified by the SELRW bit in FLCMDCR must match that specified in this register. When changing the read/write direction, FLDTFIFO should be cleared by setting the AC0CLR bit in FLINTDMACR before use.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DTFO[31:16]															
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
	DTFO[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 to 0	DTFO[31:0]	H'xxxxxxxx	R/W	Data FIFO Area Read/Write Data In write: Data in this register is written to the data FIFO area. In read: Data read from the data FIFO area is stored in this register.

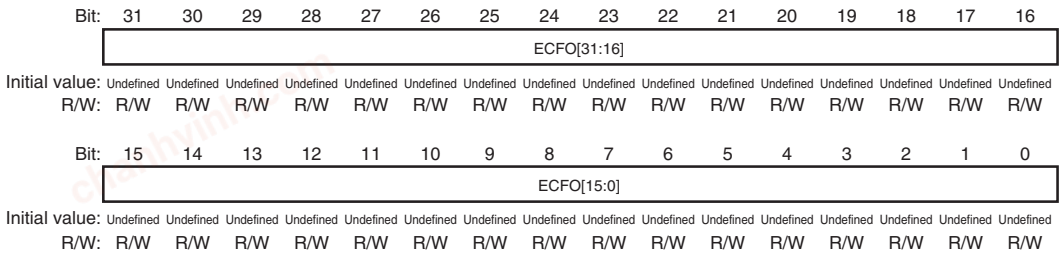
chanhvinh.com

25.3.12 Control Code FIFO Register (FLECFIFO)

FLECFIFO is used to read or write the control code FIFO area.

In DMA transfer, data in this register must be specified as the destination (source).

Note that the direction of read or write specified by the SELRW bit in FLCMDCR must match that specified in this register. When changing the read/write direction, FLECFIFO should be cleared by setting the AC1CLR bit in FLINTDMACR before use.



Bit	Bit Name	Initial Value	R/W	Description
31 to 0	ECFO [31:0]	H'xxxxxxxx	R/W	Control Code FIFO Area Read/Write Data In write: Data in this register is written to the control code FIFO area. In read: Data read from the control code FIFO area is stored in this register.

chanhvinh.com

25.3.13 Transfer Control Register (FLTRCR)

Setting the TRSTRT bit to 1 initiates access to flash memory. Access completion can be checked by the TREND bit. During the transfer (from when the TRSTRT bit is set to 1 until the TREND bit is set to 1), the processing should not be forcibly ended (by setting the TRSTRT bit to 0).

When reading from flash memory, TREND is set when reading from flash memory have been finished. However, if there is any read data remaining in the FIFO, the processing should not be forcibly ended until all data has been read from the FIFO. While this module has the external bus mastership and transfer is in progress, the SLEEP instruction should not be executed until the TREND bit is set and transfer is completed.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	TR STAT	TR END	TR STRT
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 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	TRSTAT	0	R	Transfer State Indicates that this module has acquired the external bus mastership and that transfer is actually being performed. 0: Transfer has not been started. 1: Transfer is in progress or transfer has ended.
1	TREND	0	R/W	Processing End Flag Bit Indicates that the processing performed in the specified access mode has been completed. The write value should always be 0.
0	TRSTRT	0	R/W	Transfer Start By setting this bit from 0 to 1 when the TREND bit is 0, processing in the access mode specified by the access mode specification bits ACM[1:0] is initiated. 0: Stops transfer 1: Starts transfer

chanhvinh.com

25.3.14 Bus Hold Time Setting Register (FLHOLDCR)

FLHOLDCR specifies the external bus release frequency if any other module (including the CPU) accesses a memory under the control of the bus state controller while this module is writing to or reading from the flash memory in sector access mode. With the HOLDEN bit = 0 in this register, this module holds the external bus during transfers between the flash memory and this LSI. Note that this may cause a deadlock depending on the program code and transfer data location and usage.

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
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	HOLDEN
Initial value:	0	0	0	0	0	0	0	0	0	0	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	HOLDEN	0	R/W	Bus Hold Enable Specifies whether to release the external bus mastership during write to or read from the flash memory in sector access mode. 0: Holds the bus mastership during transfers. 1: Releases the bus mastership during transfers if the FIFO empty or full state is entered in sector access mode. Note: When using the FIFO in command access mode, store the control program for this module and transfer data in the on-chip RAM.

chanhvinh.com

25.3.15 4-Symbol ECC Processing Result Register n (FL4ECCRESn) (n = 1 to 4)

FL4ECCRESn is a 32-bit read-only register that stores the error correction pattern for the nth error generated by the 4-symbol ECC circuits and the address for the nth error. The contents of this register become valid when bits 23 (4EECEN) and 22 (4ECCCORRECT) are set to 1 and a correction pattern has been generated by the setting of the 4-symbol ECC control register (FL4ECCCR).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	LOCn[9:0]									
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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PATn[9:0]									
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 26	—	Undefined	R	Reserved These bits are always read as 0. The write value should always be 0.
25 to 16	LOCn[9:0]	Undefined	R	nth Error Address Indication Indicates the address of the nth error of the four errors. Since one sector is handled as 528 bytes, the valid address range is from H'000 to H'20F. Addresses beyond the range from H'000 to H'20F are invalid (and indicate that generation of an error pattern was not possible or that there were no errors). The initial value is H'3FF. The values of these bits that are set after the 4ECCEND bit in the 4-symbol ECC control register is set to 1 are valid. Note that starting to read out the data for the next sector before reading these bits will destroy the data.
15 to 10	—	Undefined	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
9 to 0	PATn[9:0]	Undefined	R	nth Error Correction Pattern Indication Indicates the pattern for the correction of the nth error of the four errors. Patterns for which PAT[9:8] = B'11 and patterns for which all bits of PAT[9:0] are 0 are invalid (and indicate that generation of an error pattern was not possible or that there were no errors). The initial value is H'3FF. The values of these bits that are set after the 4ECCEND bit in the 4-symbol ECC control register is set to 1 are valid. Note that starting to read out the data for the next sector before reading these bits will destroy the data.

Note: n = 1 to 4

25.3.16 4-Symbol ECC Control Register (FL4ECCCR)

FL4ECCCR is a 32-bit readable register that indicates the processing states of the 4-symbol ECC circuit. This register consists of flag bits to which only 0 can be written. To clear a flag, write 0 to the target flag bit and 1 to the other flag bits.

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
	-	-	-	-	-	-	-	-	-	-	-	-	-	4ECCFA	4ECCEND	4ECCEXST
Initial value:	0	0	0	0	0	0	0	0	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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
2	4ECCFA	0	R/(W)*	4-Symbol ECC Uncorrectable Error Only 0 can be written to this bit. If five or more errors have been detected, it is regarded that the errors are uncorrectable and this bit is set to 1. 0: Indicates that the error is correctable. 1: Indicates that the error is uncorrectable
1	4ECCEND	0	R/(W)*	4-Symbol ECC Error Counting/Correction Pattern Generation End Only 0 can be written to this bit. When set, it indicates that counting of errors or generation of correction pattern has ended. If both of bits 4ECCFA and 4ECCEND are set to 1, it indicates that five or more errors were detected and the processing has therefore ended without generating a correction pattern.
0	4ECCEXST	0	R/(W)*	4-Symbol ECC Correction Execution When an ECC error is detected, this bit is set and error counting or generation of correction pattern is executed. Generation of correction pattern is executed for a sector. 0: Error counting and correction pattern generation is stopped. 1: Error counting or correction pattern generation is executed. If the 4ECCCORRECT bit in FLCMNCR is set to 1, reading is stopped while 4ECCEXST is set to 1 and reading is restarted when 4ECCEXST is cleared. Do not write 0 to this bit until 4ECCEND bit is set to 1.

Note: * Only writing 0 is valid.

chanhvinh.com

25.3.17 4-Symbol ECC Error Count Register (FL4ECCCNT)

FL4ECCCNT is a 32-bit readable register that indicates the number of errors detected by the 4-symbol ECC circuit. Only 0 can be written to this register. To clear this register, write 0 to all bits.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	ERRCNT[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
	-	-	-	-	-	-	-	-	-	-	-	-	-	ERRMAX[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 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	ERRCNT [10:0]	H'000	R/W	Error Counter Only 0 can be written to these bits. Indicates the total number of errors found in a series of sectors that have been read (for one block at maximum, which consists of 64 pages × 4 sectors). If a sector contains five or more errors, they are counted as five.
15 to 3	—	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
2 to 0	ERRMAX [2:0]	000	R/W	Maximum Number of Errors Only 0 can be written to these bits. Indicates the maximum number of errors found in a sector among the series of sectors that have been read. 000: Maximum number of errors was 0 001: Maximum number of errors was 1 010: Maximum number of errors was 2 011: Maximum number of errors was 3 100: Maximum number of errors was 4 101: Maximum number of errors was 5 or more 110: Not set 111: Not set

chanhvinh.com

25.4 Operation

25.4.1 Access Sequence

This module performs accesses in several independent stages.

For example, NAND-type flash memory programming consists of the following five stages.

- First command issue stage (program setup command)
- Address issue stage (program address)
- Data stage (output)
- Second command issue stage (program start command)
- Status read stage

NAND-type flash memory programming access is achieved by executing these five stages sequentially. An access to flash memory is completed at the end of the final stage (status read stage).

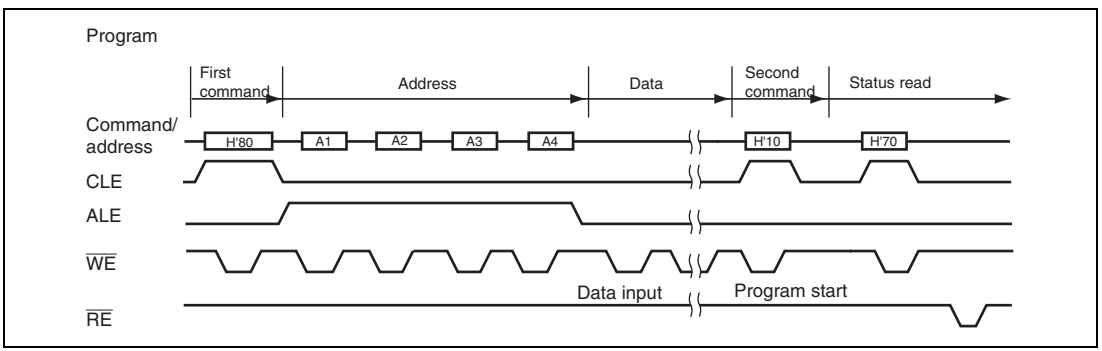


Figure 25.2 Programming Operation for NAND-Type Flash Memory and Stages

For details on NAND-type flash memory read operation, see section 25.4.4, Command Access Mode.

25.4.2 Operating Modes

Two operating modes are supported.

- Command access mode
- Sector access mode

The ECC generation and error check are performed in sector access mode.

chanhvinh.com

25.4.3 Register Setting Procedure

Figure 25.3 shows the register setting flow required for accessing the flash memory.

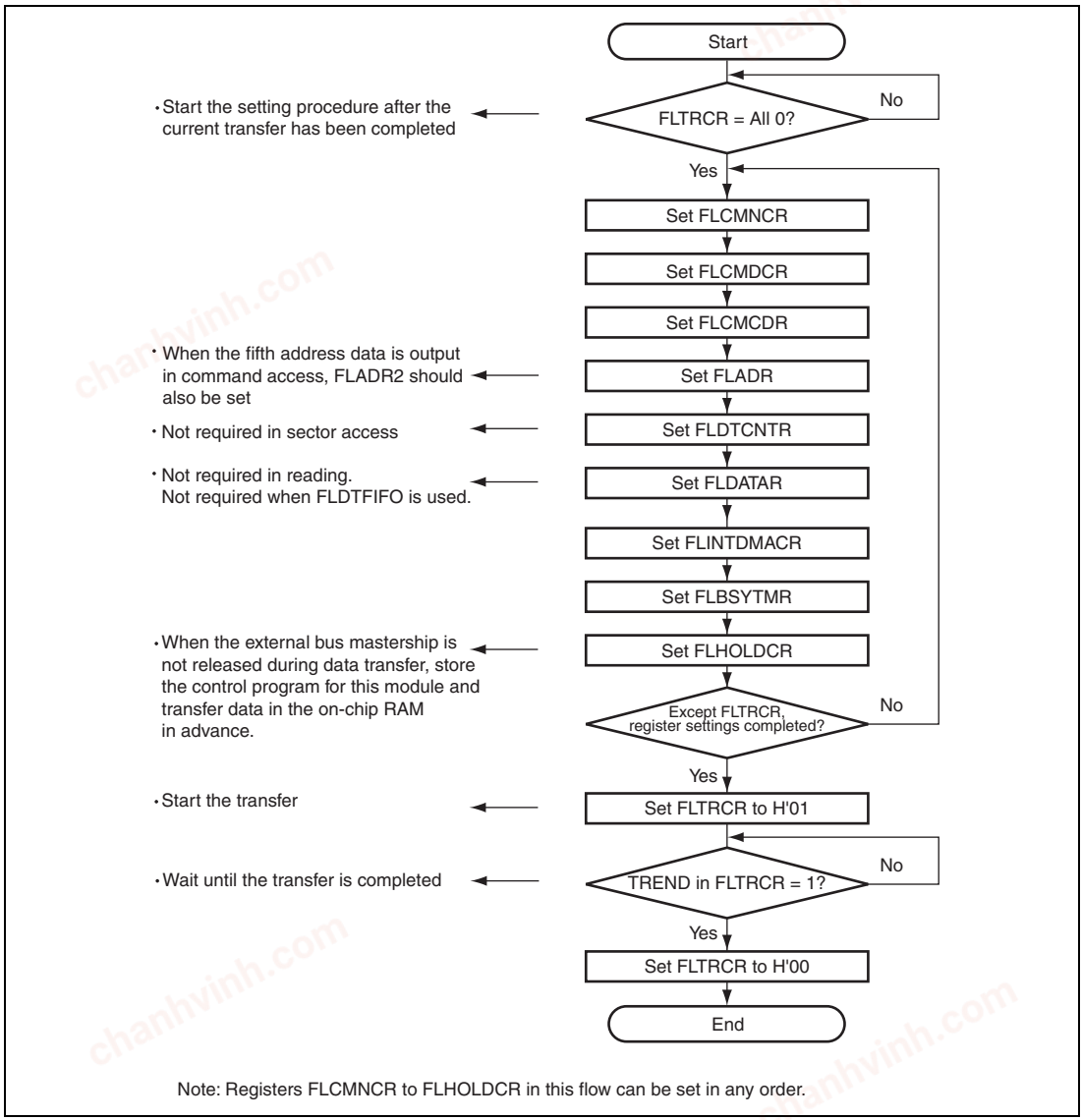


Figure 25.3 Register Setting Flow

chanhvinh.com

25.4.4 Command Access Mode

Command access mode accesses flash memory by specifying a command to be issued to flash memory, address, data, read/write direction, and number of times to the registers. In this mode, I/O data can be transferred by the DMA via FLDTFIFO.

(1) NAND-Type Flash Memory Access

Figure 25.4 shows an example of read operation for NAND-type flash memory. In this example, the first command is specified as H'00, address data length is specified as 3 bytes, and the number of read bytes is specified as 8 bytes in the data counter.

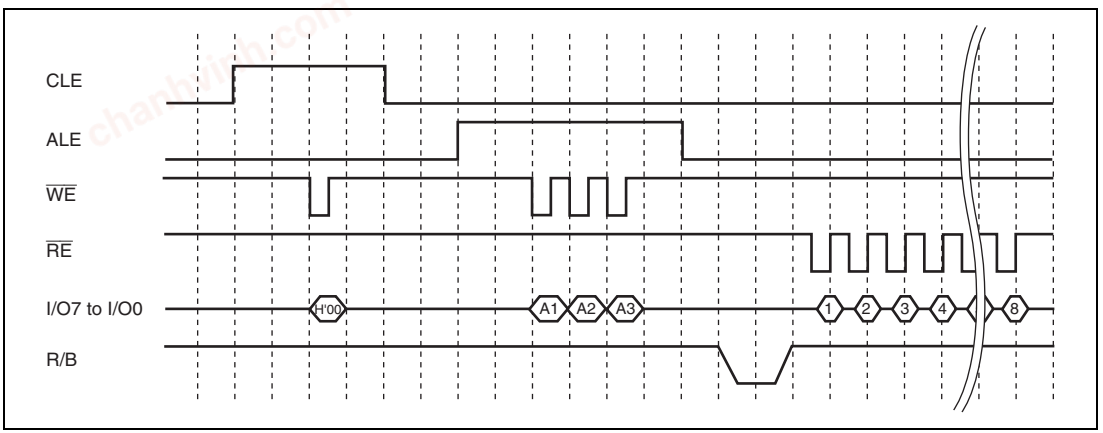


Figure 25.4 Read Operation Timing for NAND-Type Flash Memory

chanhvinh.com

Figures 25.5 and 25.6 show examples of programming operation for NAND-type flash memory.

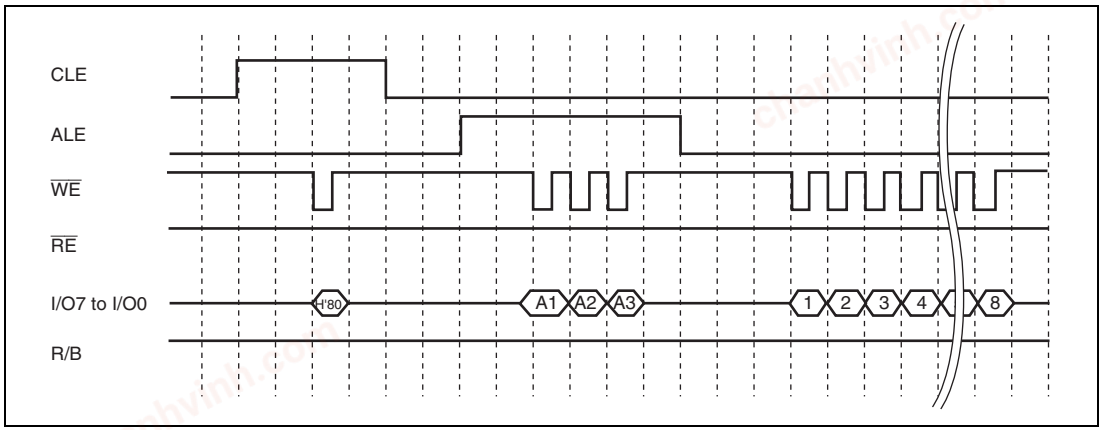


Figure 25.5 Programming Operation Timing for NAND-Type Flash Memory (1)

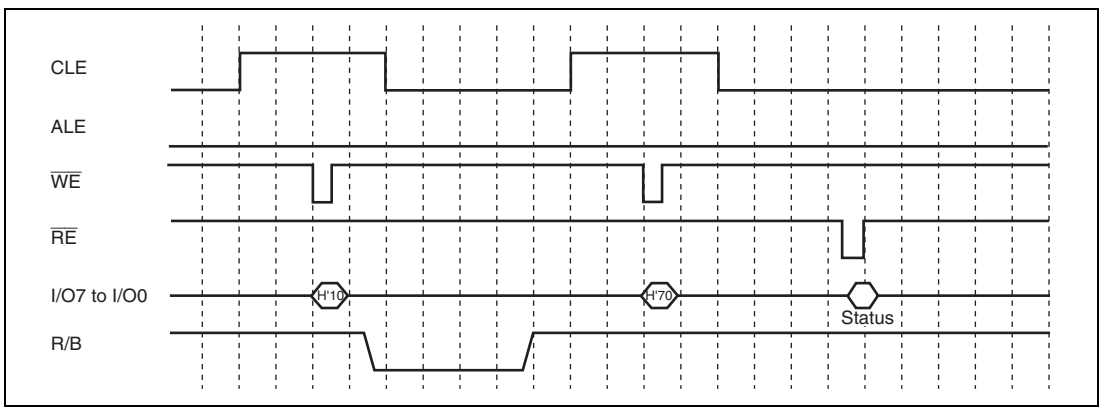


Figure 25.6 Programming Operation Timing for NAND-Type Flash Memory (2)

chanhvinh.com

(2) NAND-Type Flash Memory (2048 + 64 Bytes) Access

Figure 25.7 shows an example of read operation for NAND-type flash memory (2048 + 64 bytes). In this example, the first command is specified as H'00, the second command is specified as H'30, and address data length is specified as 4 bytes. The number of read bytes is specified as 4 bytes in the data counter.

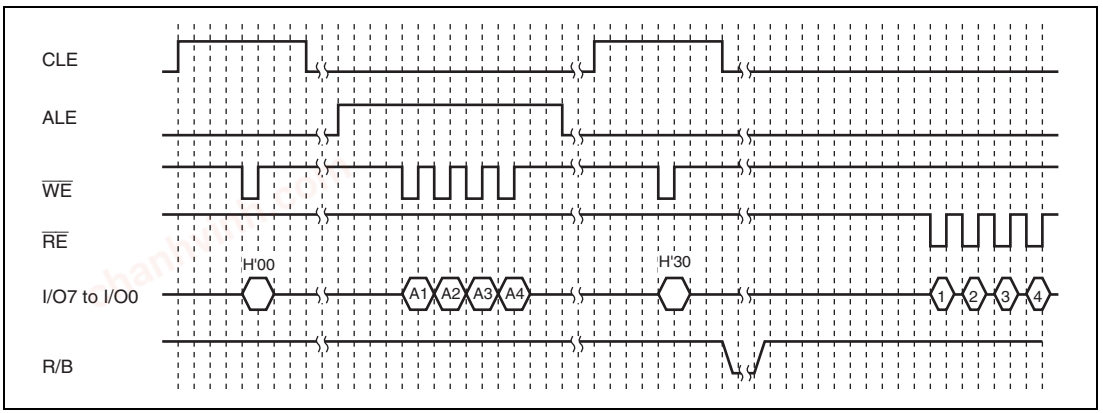


Figure 25.7 Read Operation Timing for NAND-Type Flash Memory

Figures 25.8 and 25.9 show examples of programming operation for NAND-type flash memory (2048 + 64 bytes).

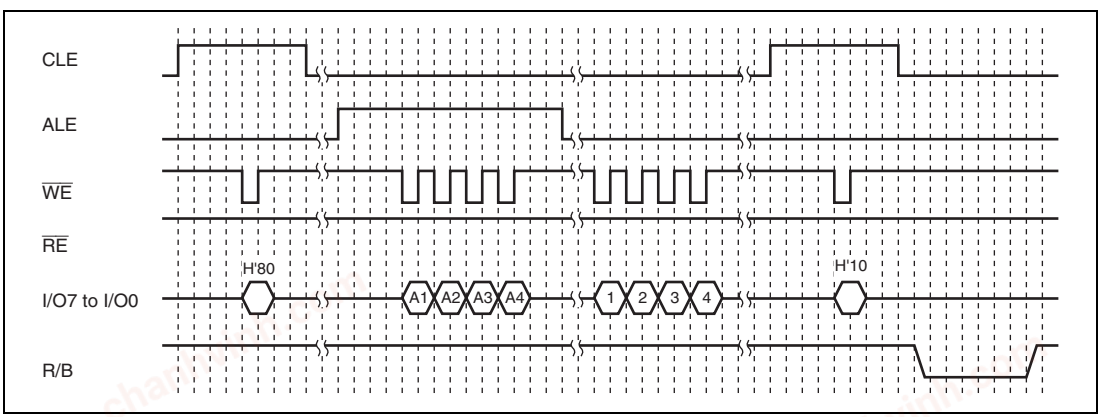


Figure 25.8 Programming Operation Timing for NAND-Type Flash Memory (1)

chanhvinh.com

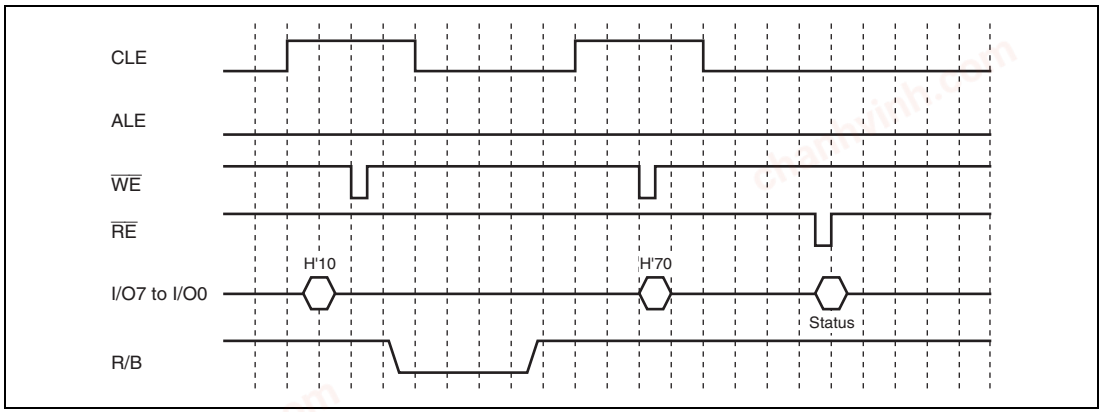


Figure 25.9 Programming Operation Timing for NAND-Type Flash Memory (2)

25.4.5 Sector Access Mode

In sector access mode, flash memory can be read or programmed in sector units by specifying the sector number of the sector to be accessed. In programming, an ECC is added. In read, an ECC error check (detection) is performed.

Since 512-byte data is stored in FLDTFIFO and 16-byte control code is stored in FLECFIFO, the DREQ1EN and DREQ0EN bits in FLINTDMACR can be set to transfer by the DMA.

chanhvinh.com

Figure 25.10 shows the relationship of DMA transfer between sectors in flash memory (data and control code) and memory on the address space.

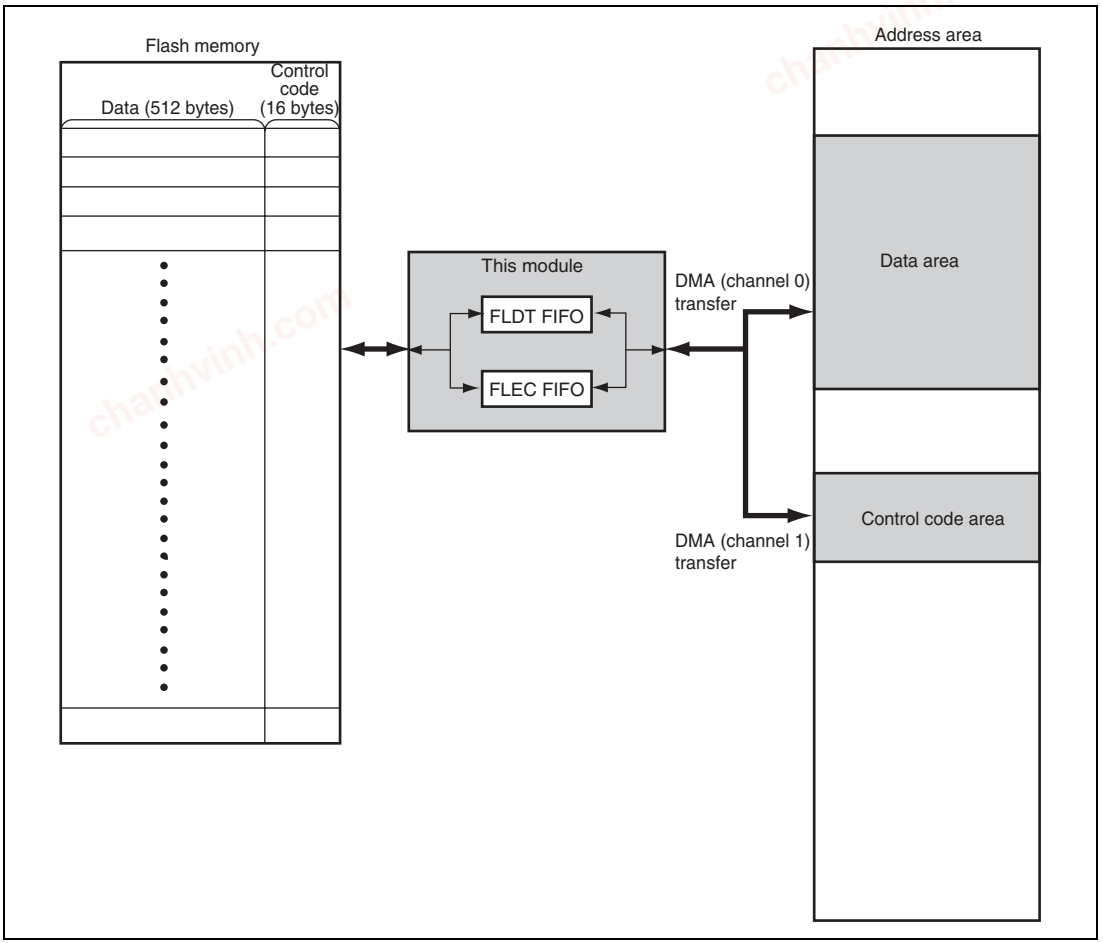


Figure 25.10 Relationship between DMA Transfer and Sector (Data and Control Code), and Memory and DMA Transfer

chanhvinh.com

(1) Sector Address

Figure 25.11 shows the relationship between the physical sector address of NAND-type flash memory and the address of flash memory.

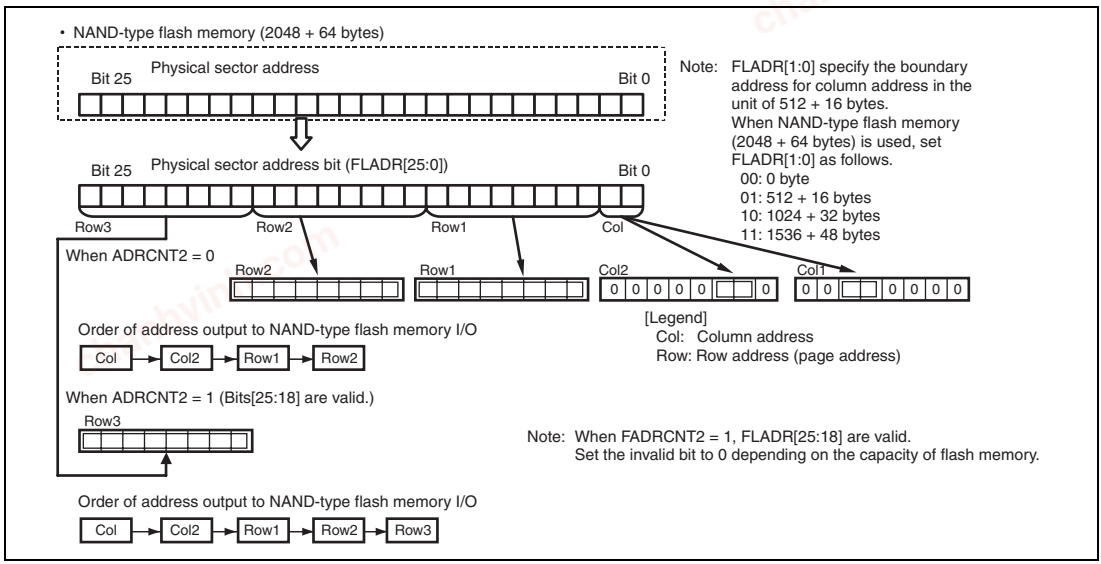


Figure 25.11 Relationship between Sector Number and Address Expansion of NAND-Type Flash Memory

chanhvinh.com

(2) Continuous Sector Access

A series of sectors can be read or written by specifying the start sector address of NAND-type flash memory and the number of sectors to be transferred. Figure 25.12 shows an example of physical sector specification register and transfer count specification register settings when transferring logical sectors 0 to 40, which are not contiguous because of an unusable sector in NAND-type flash memory.

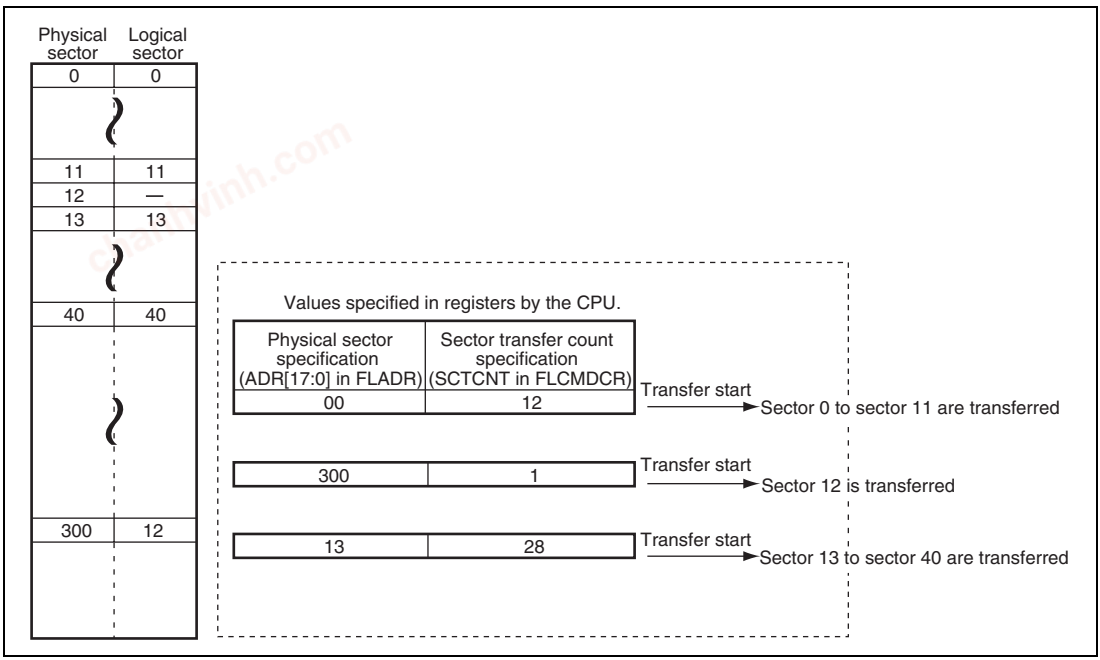


Figure 25.12 Sector Access when Unusable Sector Exists in Continuous Sectors

chanhvinh.com

(3) Flash Memory Access in Sector Access Mode

Figures 25.13 and 25.14 show the timing of writing to and reading from the NAND-type flash memory in sector access mode.

Figure 25.13 shows the timing of writing to the 1-Gbit large-block flash memory. During the execution of sequential sector access spanning multiple pages, data are written to the flash memory with the timing shown in the figure for every page (2048 + 64 bytes).

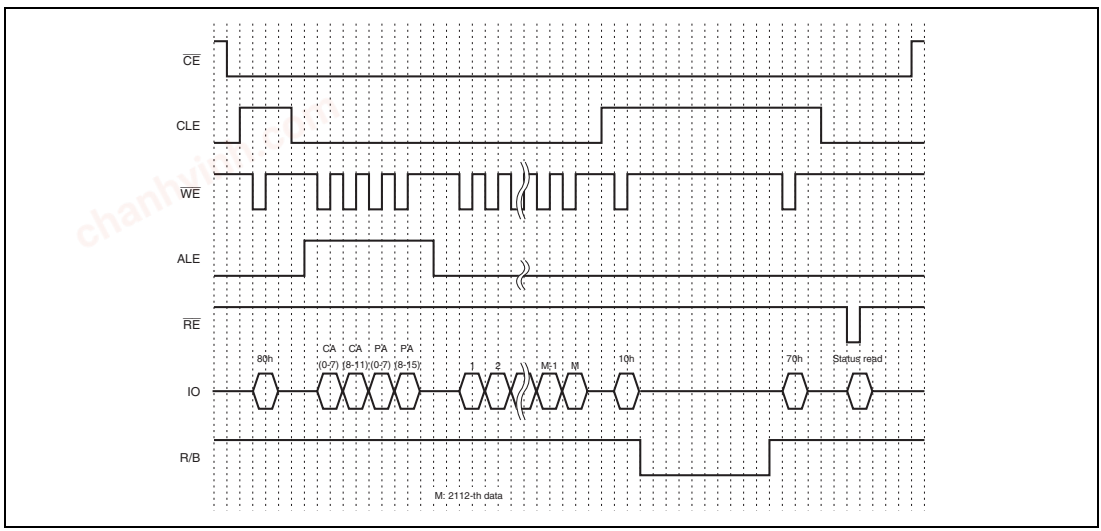


Figure 25.13 Programming Operation Timing for NAND-Type Flash Memory

chanhvinh.com

Figure 25.14 shows the timing of reading from the 1-Gbit large-block flash memory. During the execution of sequential sector access spanning multiple pages, data are read from the flash memory with the timing shown in the figure for every page (2048 + 64 bytes).

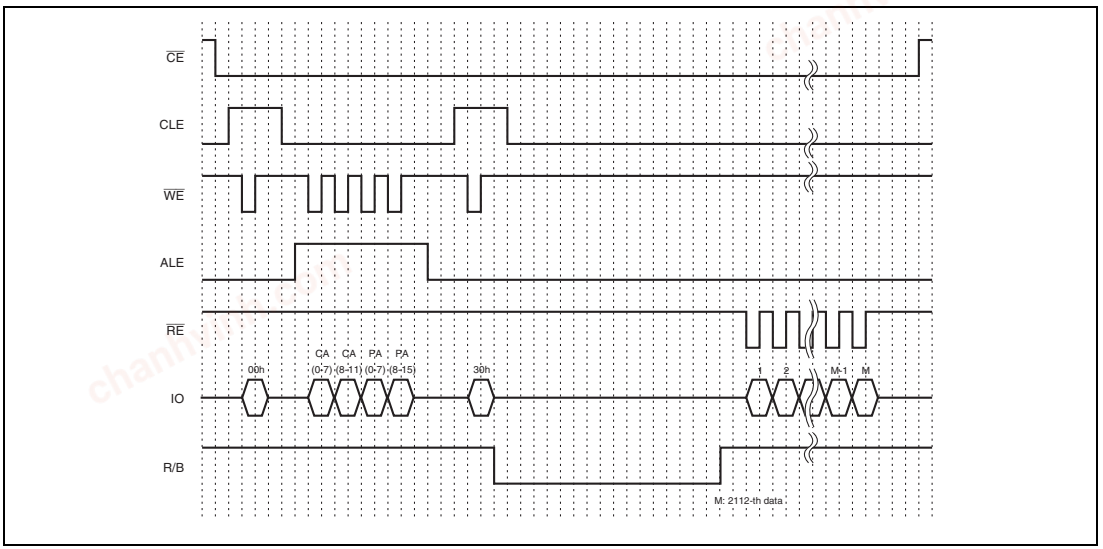


Figure 25.14 Read Timing from NAND-Type Flash Memory (Sector Access Mode)

25.4.6 ECC Error Correction

This module generates and adds ECC during write operation in sector access mode and performs ECC error check during read operation in sector access mode. ECC processing is selectable between 3-symbol ECC, the function provided in the earlier products, and 4-symbol ECC.

With 3-symbol ECC, only ECC generation and error detection are performed and error correction is not performed. So, errors must be corrected by software. On the other hand, 4-symbol ECC is capable of ECC generation, error detection, and error correction pattern generation by hardware.

chanhvinh.com

(1) Overview of 4-Symbol ECC Circuit

The 4-symbol ECC circuit in this module is capable of correcting up to 10 bits per symbol, which makes a maximum of 40 bits for four symbols. However, the circuit corrects up to 32 bits because the data in the flash memory data area is counted as eight bits per symbol.

Error correction pattern generation means generation of information necessary for correcting errors, not execution of error correction. For details, see section 25.4.6, (3) 4-Symbol ECC Error Correction Pattern Generation.

The 4-symbol ECC circuit is roughly divided into three stages (figure 25.15).

1. ECC generator
2. Error count detector
3. Error correction pattern generator

ECC generation and error count detection can be executed continuously while error correction pattern generation is executed on a sector-by-sector basis.

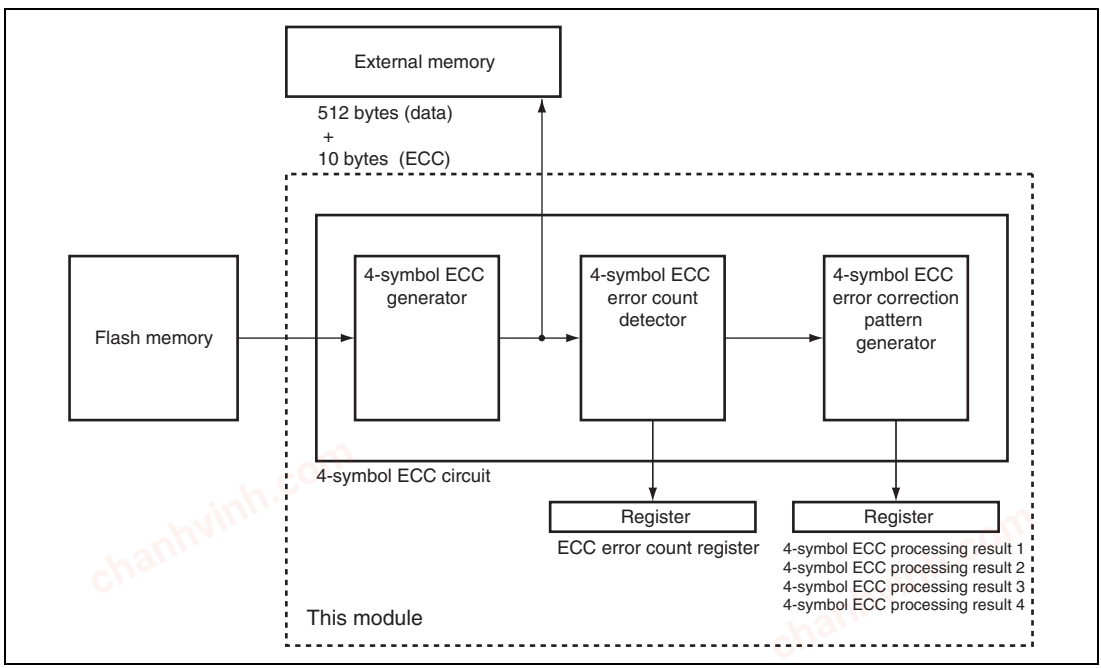


Figure 25.15 4-Symbol ECC Circuit

chanhvinh.com

(2) 4-Symbol ECC Operation

Figure 25.16 shows a flowchart of the operation when the 4-symbol ECC circuit is used. Setting the 4ECCEN bit in FLCMNCR enables the 4-symbol ECC circuit and ECC is generated and output for each sector. If the 4ECCCORRECT bit in FLCMNCR is also set to 1, information necessary for correction pattern generation is accumulated in the 4-symbol ECC circuit.

In the case when the NAND flash memory controller is reading data from flash memory by continuous sector access, the reading operation stops when an error-containing sector has been read regardless of the number of remaining sectors. After reading of the error-containing sector has ended, generation of error correction pattern is started by setting the FL4ECCCR register. If the sector contains five or more errors, that sector is regarded as uncorrectable. Note that a sector may be uncorrectable for some error patterns even if it contains four or less errors. In such a case, invalid data are placed in the FL4ECCRES1 to FL4ECCRES4 registers.

chanhvinh.com

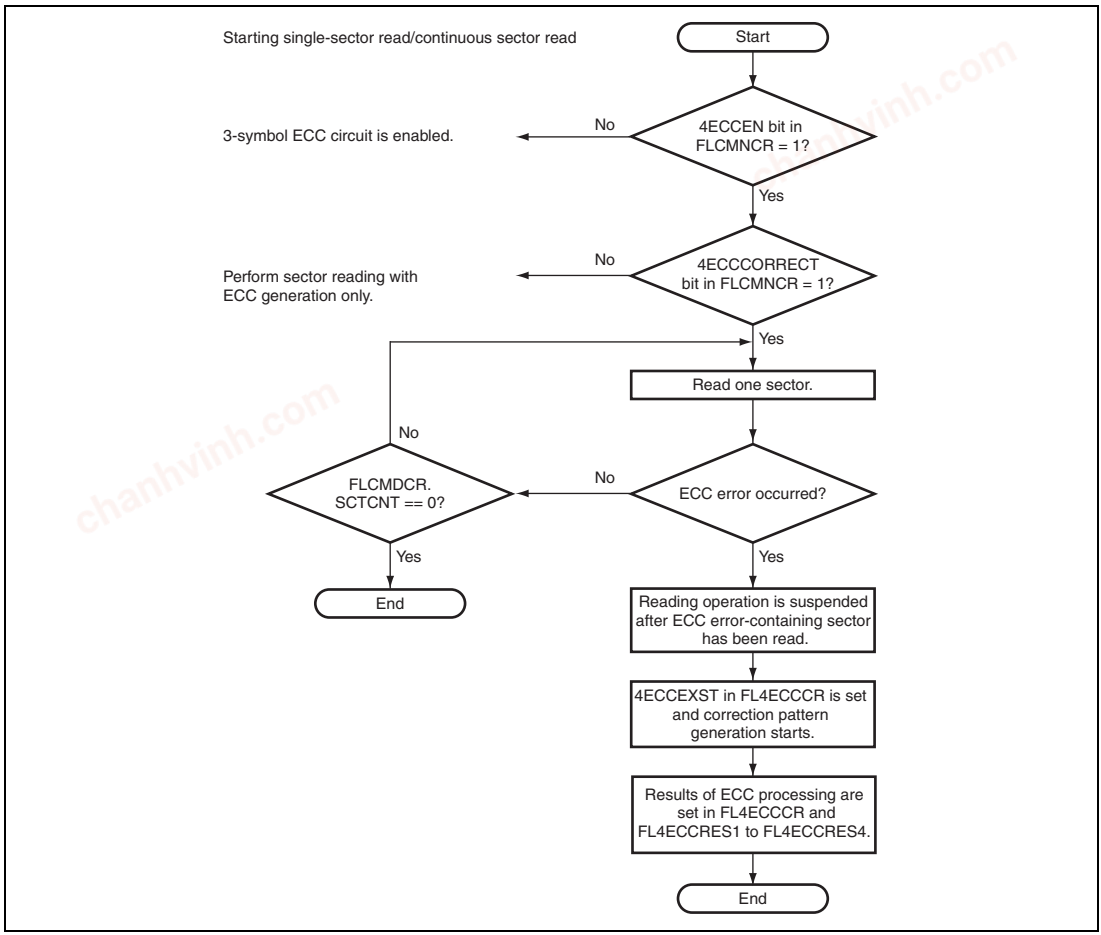


Figure 25.16 Flow of 4-Symbol ECC Operation

chanhvinh.com

(3) 4-Symbol ECC Error Correction Pattern Generation

The 4-symbol ECC circuit of this module can generate error correction patterns by hardware. The original data can be restored by using the error correction patterns. Since the hardware processing only covers generation of error correction patterns, processing for data restoration must be provided by software.

The error correction patterns are output in the following format. The bits in a correction pattern at error bit positions are set to 1, so the original data is restored by taking EOR of error data and error correction pattern.

- Example 1
  Original data: B'00000000
  Erroneous data: B'11111111
  Correction pattern: B'0011111111 (higher two bits are unnecessary data)
  Recovered data: B'00000000
  (EOR of error pattern and correction pattern)

- Example 2
  Original data: B'10101010
  Erroneous data: B'01010101
  Correction pattern: B'0011111111 (higher two bits are unnecessary data)
  Recovered data: B'10101010
  (EOR of error pattern and correction pattern)

- Example 3
  Original data: B'11110000
  Erroneous data: B'00000000
  Correction pattern: B'0011110000 (higher two bits are unnecessary data)
  Recovered data: B'11110000
  (EOR of error pattern and correction pattern)

chanhvinh.com

25.4.7 Status Read

This module can read the status register of an AND/NAND-type flash memory. The data in the status register is input through the I/O7 to I/O0 pins and stored in the bits STAT[7:0] in FLBSYCNT, which can be read by the CPU. If a program error or erase error is detected when the status register value is stored in the bits STAT[7:0] in FLBSYCNT, the STERB bit in FLINTDMACR is set to 1 and generates an interrupt to the CPU if the STERINTE bit in FLINTDMACR is enabled. If a status error occurs during continuous sector access, the TREND bit in FLTRCR is set to 1 and the procedure stops.

(1) Status Read of NAND-Type Flash Memory

The status register of NAND-type flash memory can be read by inputting command H'70 to NAND-type flash memory. If programming is executed in command access mode or sector access mode while the DOSR bit in FLCMDCR is set to 1, this module automatically inputs command H'70 to NAND-type flash memory and reads the status register of NAND-type flash memory. When the status register of NAND-type flash memory is read, the I/O7 to I/O0 pins indicate the following information as described in table 25.3.

Table 25.3 Status Read of NAND-Type Flash Memory

I/O	Status (definition)	Description
I/O7	Program protection	0: Cannot be programmed 1: Can be programmed
I/O6	Ready/busy	0: Busy state 1: Ready state
I/O5 to I/O1	Reserved	—
I/O0	Program/erase	0: Pass 1: Fail

chanhvinh.com

25.5 Interrupt Sources

This module has seven interrupt sources: Status error, ready/busy timeout error, ECC error, 4-symbol ECC pattern generation end, transfer end, FIFO0 transfer request, and FIFO1 transfer request. Each of the interrupt sources has its corresponding interrupt flag and the interrupt can be requested independently to the CPU if the interrupt is enabled by the interrupt enable bit. Note that the status error, ready/busy timeout error, ECC error, and 4-symbol ECC pattern generation end, use the common FLSTE interrupt to the CPU.

Table 25.4 NAND Flash Memory Controller Interrupt Requests

Interrupt Source	Interrupt Flag	Enable Bit	Description	Priority
FLSTE interrupt	STERB	STERINTE	Status error	High
	BTOERB	RBERINTE	Ready/busy timeout error	
	ECERB	ECERINTE	ECC error	
	4ECCEND	4ECEINTE	4-symbol ECC pattern generation end	
FLTEND interrupt	TREND	TEINTE	Transfer end	
FLTRQ0 interrupt	TRREQF0	TRINTE0	FIFO0 transfer request	
FLTRQ1 interrupt	TRREQF1	TRINTE1	FIFO1 transfer request	Low

25.6 DMA Transfer Specifications

This module can request DMA transfers separately to the data area FLDTFIFO and control code area FLECFIFO. Table 25.5 summarizes DMA transfer enable or disable states in each access mode.

Table 25.5 DMA Transfer Specifications

	Sector Access Mode	Command Access Mode
FLDTFIFO	DMA transfer enabled	DMA transfer enabled
FLECFIFO	DMA transfer enabled	DMA transfer disabled

For details on settings of the direct memory access controller, see section 10, Direct Memory Access Controller.

chanhvinh.com

25.7 Usage Notes

25.7.1 External Bus Mastership Release Timing

This module negates $\overline{FCE}$ regardless of the busy/ready state when having completed a necessary process. With bit 21 (BUSYON) set to 0 in the common control register (FLCMNCR), this module negates $\overline{FCE}$ and releases the bus mastership even during the busy state upon completion of the process. With BUSYON = 0, setting bit 24 (DOSR) in the command control register (FLCMDCR) to 1 to read the status enables acquiring the bus mastership even during the busy state.

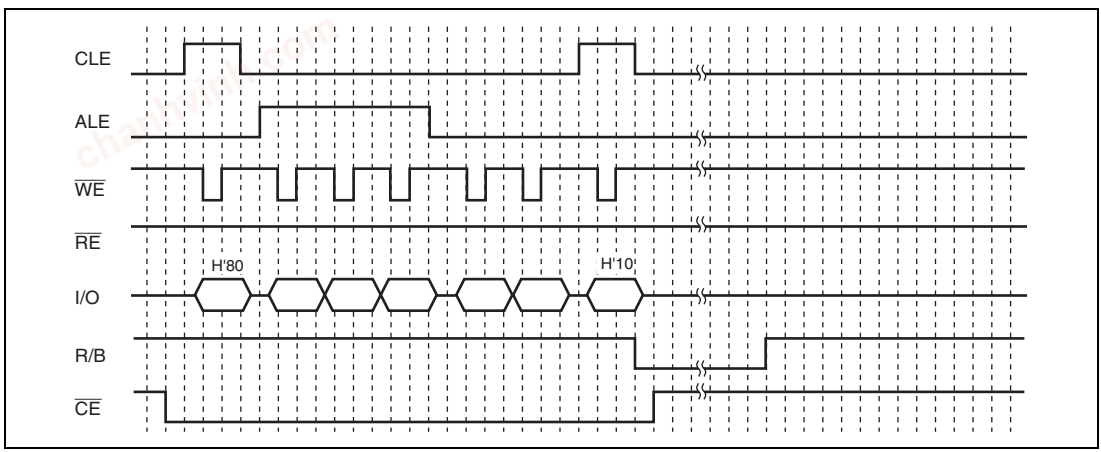


Figure 25.17 BUSYON = 0, DOSR = 0 (Writing to Flash Memory)

chanhvinh.com

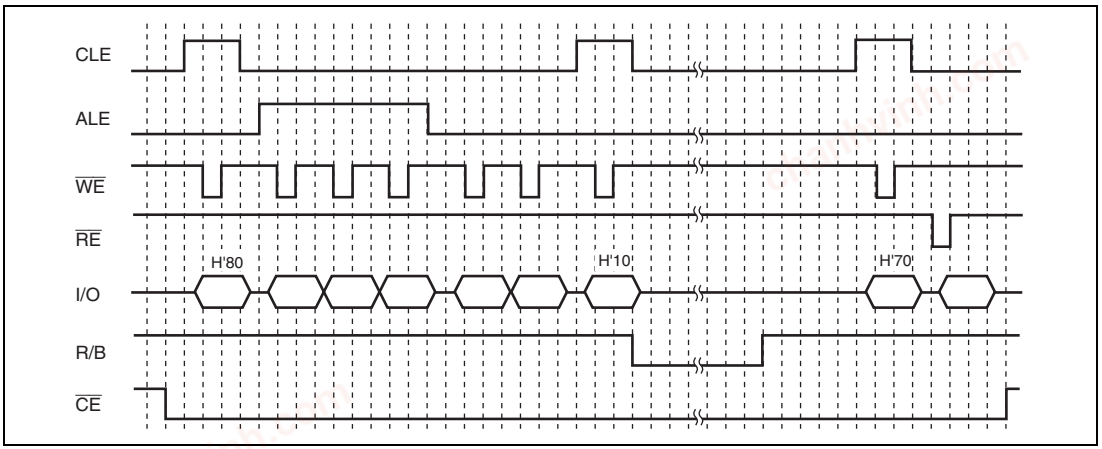


Figure 25.18 BUSYON = 0, DOSR = 1 (Writing to Flash Memory)

chanhvinh.com

25.7.2 Writing to the Control-Code Area when 4-Symbol ECC Circuit is in Use

Follow the procedure given below to write to the control-code area when the 4-symbol ECC circuit is in use. If this procedure is not followed, correct writing to the control-code area of the flash memory will not be possible.

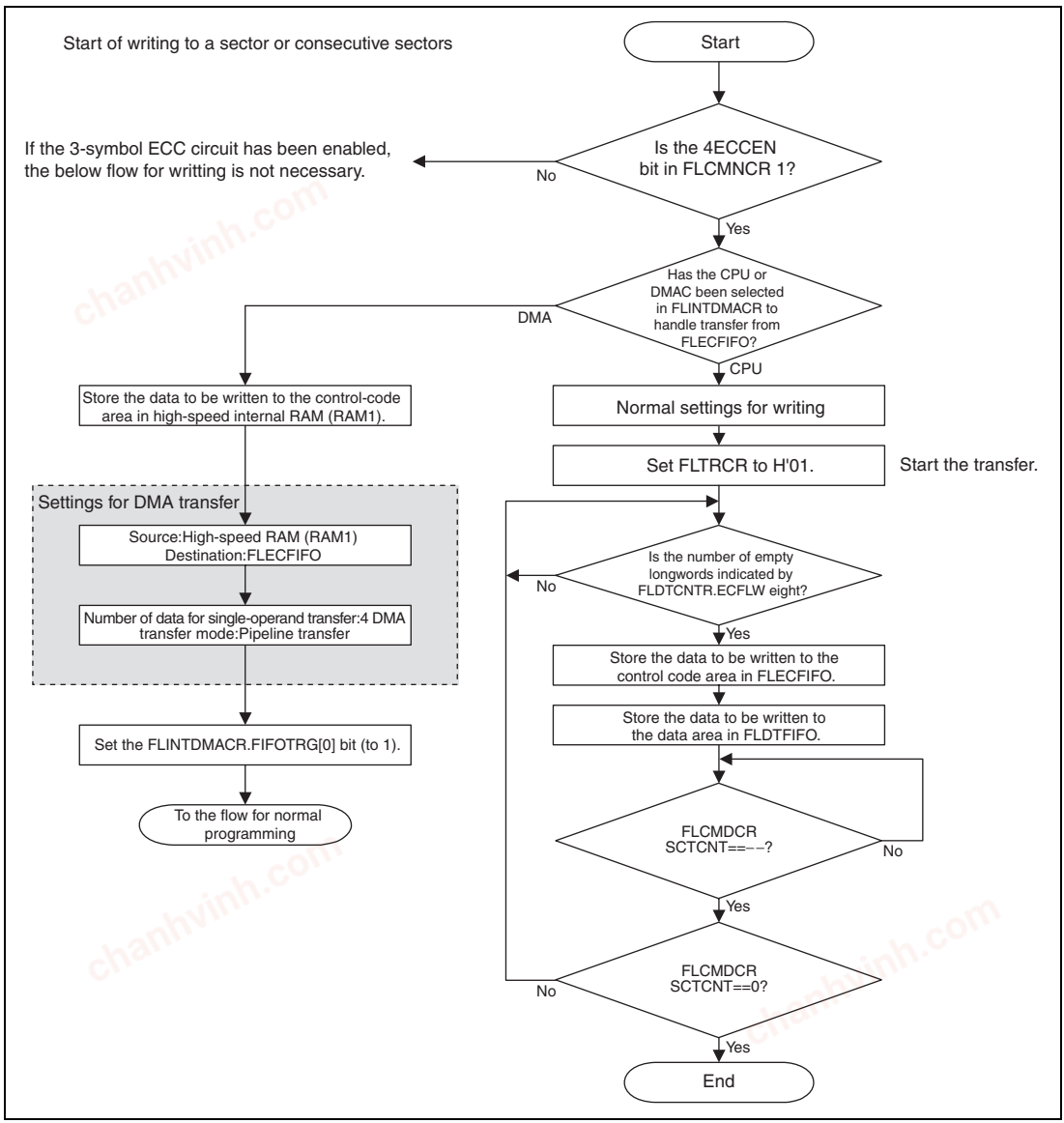


Figure 25.19 Writing Procedure to the Control-Code Area when 4-Symbol ECC is Used

chanhvinh.com

25.7.3 Usage Notes for the SNAND Bit

When using the SNAND bit in FLCMNCR, only the first command or the second command is corresponded in spite of the setting of the DOCMD1 or DOCMD2 bit in FLCMDCR.

When no command or only the first command is issued, 0 should be written in the SNAND bit.

chanhvinh.com

Section 26 USB 2.0 Host/Function Module

The USB 2.0 host/function module is a USB controller which provides capabilities as a USB host controller and USB function controller function. This module supports high-speed transfer defined by USB (universal serial bus) Specification 2.0, full-speed transfer, and low-speed transfer when used as the host controller, and supports high-speed transfer and full-speed transfer when used as the function controller. This module has a USB transceiver* and 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 ten pipes. Any endpoint numbers can be assigned to PIPE1 to PIPE9, based on the peripheral devices or user system for communication.

Note: * Before using this module, set up the internal transceiver. For details, refer to section 26.5.1, Procedure for Setting the USB Transceiver.

26.1 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.
- USB transceiver is incorporated.

(2) Reduced Number of External Pins and Space-Saving Installation

- On-chip D+ pull-up resistor (during USB function operation)
- On-chip D+ and D- pull-down resistor (during USB host operation)
- On-chip D+ and D- terminal resistor (during high-speed operation)
- On-chip D+ and D- output resistor (during full-speed operation)

(3) All Types of USB Transfers Supported

- Control transfer
- Bulk transfer
- Interrupt transfer (high bandwidth transfers not supported)
- Isochronous transfer (high bandwidth transfers not supported)

chanhvinh.com

(4) Internal Bus Interfaces

- Two DMA interface channels are incorporated.

(5) Pipe Configuration

- On-chip 8-Kbyte buffer memory for USB communications
- Up to ten pipes can be selected (including the default control pipe)
- Programmable pipe configuration
- Endpoint numbers can be assigned flexibly to PIPE1 to PIPE9.
- Transfer conditions that can be set for each pipe:

PIPE0:	Control transfer (default control pipe: DCP), 64-byte fixed single buffer
PIPE1 and PIPE2:	Bulk transfers/isochronous transfer, 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 PIPE9:	Interrupt transfer, 64-byte fixed single buffer

(6) 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

(7) Features of the USB Function Controller

- Both 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 control function
- Device state control function
- Auto response function for SET_ADDRESS request
- NAK response interrupt function (NRDY)
- SOF interpolation function

chanhvinh.com

(8) 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 (n = 0 or 1) port has been read (DCLRM)
- NAK setting function for response PID generated by end of transfer (SHTNAK)

26.2 Input/Output Pins

Table 26.1 shows the pin configuration of the USB.

Table 26.1 USB Pin Configuration

Category	Name	Pin Name	I/O	Function
USB bus interface	USB D+ data	DP	I/O	D+ I/O of the USB on-chip transceiver This pin should be connected to the D+ pin of the USB bus.
	USB D- data	DM	I/O	D− I/O of the USB on-chip transceiver This pin should be connected to the D- pin of the USB bus.
VBUS monitor input	VBUS input	VBUS	Input	USB cable connection monitor pin This pin should be connected directly to the VBUS of the USB bus. Whether the VBUS is connected or disconnected can be detected. If this pin is not connected with the VBUS of the USB bus, it should be supplied with 5 V. It should be supplied with 5 V also when the host controller function is selected. Note: The VBUS cannot be supplied to connected peripheral devices.
Reference resistor	Reference input	REFRIN	Input	Reference resistor connection pin This pin should be connected to USBAPVss through a 5.6 kΩ ±1% resistor.
Clock	Crystal resonator for USB/external clock	USB_X1	Input	This pin should be connected to crystal resonator for the USB. This pin can also be used for external clock input.
		USB_X2	Output	

chanhvinh.com

Category	Name	Pin Name	I/O	Function
Power supply	Transceiver block analog pin power supply	USBAPVcc	Input	Power supply for pins
	Transceiver block analog pin ground	USBAPVss	Input	Ground for pins
	Transceiver block digital pin power supply	USBDPVcc	Input	Power supply for pins
	Transceiver block digital pin ground	USBDPVss	Input	Ground for pins
	Transceiver block analog core power supply	USBAVcc	Input	Power supply for the core
	Transceiver block analog core ground	USBAVss	Input	Ground for the core
	Transceiver block digital core power supply	USBDVcc	Input	Power supply for the core
	Transceiver block digital core ground	USBDVss	Input	Ground for the core
	USB 480 MHz power supply	USBUVcc	Input	Power supply for 480-MHz operation block
	USB 480 MHz ground	USBUVss	Input	Ground for 480-MHz operation block

chanhvinh.com

26.3 Register Description

Table 26.2 shows the register configuration.

Table 26.2 Register Configuration

Register Name	Abbreviation	R/W	Address	Access Size
System configuration control register	SYSCFG	R/W	H'FFFF C000	16
CPU bus wait setting register	BUSWAIT	R/W	H'FFFF C002	16
System configuration status register	SYSSTS	R	H'FFFF C004	16
Device state control register	DVSTCTR	R/W	H'FFFF C008	16
Test mode register	TESTMODE	R/W	H'FFFF C00C	16
DMA0-FIFO bus configuration register	D0FBCFG	R/W	H'FFFF C010	16
DMA1-FIFO bus configuration register	D1FBCFG	R/W	H'FFFF C012	16
CFIFO port register	CFIFO	R/W	H'FFFF C014	8, 16, 32
D0FIFO port register	D0FIFO	R/W	H'FFFF C018	8, 16, 32
D1FIFO port register	D1FIFO	R/W	H'FFFF C01C	8, 16, 32
CFIFO port select register	CFIFOSEL	R/W	H'FFFF C020	16
CFIFO port control register	CFIFOCTR	R/W	H'FFFF C022	16
D0FIFO port select register	D0FIFOSEL	R/W	H'FFFF C028	16
D0FIFO port control register	D0FIFOCTR	R/W	H'FFFF C02A	16
D1FIFO port select register	D1FIFOSEL	R/W	H'FFFF C02C	16
D1FIFO port control register	D1FIFOCTR	R/W	H'FFFF C02E	16
Interrupt enable register 0	INTENB0	R/W	H'FFFF C030	16
Interrupt enable register 1	INTENB1	R/W	H'FFFF C032	16
BRDY interrupt enable register	BRDYENB	R/W	H'FFFF C036	16
NRDY interrupt enable register	NRDYENB	R/W	H'FFFF C038	16
BEMP interrupt enable register	BEMPENB	R/W	H'FFFF C03A	16
SOF output configuration register	SOFCFG	R/W	H'FFFF C03C	16
Interrupt status register 0	INTSTS0	R/W	H'FFFF C040	16
Interrupt status register 1	INTSTS1	R/W	H'FFFF C042	16

chanhvinh.com

Register Name	Abbreviation	R/W	Address	Access Size
BRDY interrupt status register	BRDYSTS	R/W	H'FFFF C046	16
NRDY interrupt status register	NRDYSTS	R/W	H'FFFF C048	16
BEMP interrupt status register	BEMPSTS	R/W	H'FFFF C04A	16
Frame number register	FRMNUM	R/W	H'FFFF C04C	16
μFrame number register	UFRMNUM	R	H'FFFF C04E	16
USB address register	USBADDR	R	H'FFFF C050	16
USB request type register	USBREQ	R/W	H'FFFF C054	16
USB request value register	USBVAL	R/W	H'FFFF C056	16
USB request index register	USBINDX	R/W	H'FFFF C058	16
USB request length register	USBLENG	R/W	H'FFFF C05A	16
DCP configuration register	DCPCFG	R/W	H'FFFF C05C	16
DCP maximum packet size register	DCPMAXP	R/W	H'FFFF C05E	16
DCP control register	DCPCTR	R/W	H'FFFF C060	16
Pipe window select register	PIPESEL	R/W	H'FFFF C064	16
Pipe configuration register	PIPECFG	R/W	H'FFFF C068	16
Pipe buffer setting register	PIPEBUF	R/W	H'FFFF C06A	16
Pipe maximum packet size register	PIPEMAXP	R/W	H'FFFF C06C	16
Pipe cycle control register	PIPEPERI	R/W	H'FFFF C06E	16
Pipe 1 control register	PIPE1CTR	R/W	H'FFFF C070	16
Pipe 2 control register	PIPE2CTR	R/W	H'FFFF C072	16
Pipe 3 control register	PIPE3CTR	R/W	H'FFFF C074	16
Pipe 4 control register	PIPE4CTR	R/W	H'FFFF C076	16
Pipe 5 control register	PIPE5CTR	R/W	H'FFFF C078	16
Pipe 6 control register	PIPE6CTR	R/W	H'FFFF C07A	16
Pipe 7 control register	PIPE7CTR	R/W	H'FFFF C07C	16
Pipe 8 control register	PIPE8CTR	R/W	H'FFFF C07E	16
Pipe 9 control register	PIPE9CTR	R/W	H'FFFF C080	16
Pipe 1 transaction counter enable register	PIPE1TRE	R/W	H'FFFF C090	16
Pipe 1 transaction counter register	PIPE1TRN	R/W	H'FFFF C092	16
Pipe 2 transaction counter enable register	PIPE2TRE	R/W	H'FFFF C094	16

chanhvinh.com

Register Name	Abbreviation	R/W	Address	Access Size
Pipe 2 transaction counter register	PIPE2TRN	R/W	H'FFFF C096	16
Pipe 3 transaction counter enable register	PIPE3TRE	R/W	H'FFFF C098	16
Pipe 3 transaction counter register	PIPE3TRN	R/W	H'FFFF C09A	16
Pipe 4 transaction counter enable register	PIPE4TRE	R/W	H'FFFF C09C	16
Pipe 4 transaction counter register	PIPE4TRN	R/W	H'FFFF C09E	16
Pipe 5 transaction counter enable register	PIPE5TRE	R/W	H'FFFF C0A0	16
Pipe 5 transaction counter register	PIPE5TRN	R/W	H'FFFF C0A2	16
USB AC characteristics switching register 1	USBACSWR1	R/W	H'FFFF C0C2	16
Device address 0 configuration register	DEVADD0	R/W	H'FFFF C0D0	16
Device address 1 configuration register	DEVADD1	R/W	H'FFFF C0D2	16
Device address 2 configuration register	DEVADD2	R/W	H'FFFF C0D4	16
Device address 3 configuration register	DEVADD3	R/W	H'FFFF C0D6	16
Device address 4 configuration register	DEVADD4	R/W	H'FFFF C0D8	16
Device address 5 configuration register	DEVADD5	R/W	H'FFFF C0DA	16
Device address 6 configuration register	DEVADD6	R/W	H'FFFF C0DC	16
Device address 7 configuration register	DEVADD7	R/W	H'FFFF C0DE	16
Device address 8 configuration register	DEVADD8	R/W	H'FFFF C0E0	16
Device address 9 configuration register	DEVADD9	R/W	H'FFFF C0E2	16
Device address A configuration register	DEVADDA	R/W	H'FFFF C0E4	16

chanhvinh.com

26.3.1 System Configuration Control Register (SYSCFG)

SYSCFG is a register that enables high-speed operation, selects the host controller function or function controller function, 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
	—	—	—	—	—	SCKE	—	—	HSE	DCFM	DRPD	DPRPU	—	—	—	USBE
Initial value:	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/W	R/W	R/W	R	R	R	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	SCKE	0	R/W	USB Module Clock Enable Stops or enables supplying 48-MHz clock signal to this module. 0: Stops supplying the clock signal to the USB module. 1: Enables supplying the clock signal to the USB module. When this bit is 0, only this register and the BUSWAIT register allow both writing and reading; the other registers in the USB module allows reading only.
9, 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	HSE	0	R/W	High-Speed Operation Enable 0: High-speed operation is disabled When the function controller function is selected: Only full-speed operation is enabled. When the host controller function is selected: Full-speed or low-speed operation is enabled. 1: High-speed operation is enabled (detected by this module) (1) When the host controller function 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 USBRESET to 1). (2) When the function controller function 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 Function Select Selects the host controller function or function controller function. 0: Function controller function is selected. 1: Host controller function is selected. This bit should be modified while DPRPU and DPRD are 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
5	DRPD	0	R/W	D+/D− Line Resistor Control Enables or disables pulling down D+ and D- lines when the host controller function 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 function is selected, and should be set to 0 if the function controller function is selected.
4	DPRPU	0	R/W	D+ Line Resistor Control Enables or disables pulling up D+ line when the function controller function 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 function is selected allows this module to pull up the D+ line to 3.3 V, 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 function is selected, and should be set to 0 if the host controller function is selected. Note: Set this bit to 0 when the USB is disconnected. Include the following processing when this bit is changed from 1 to 0. (1) Set the DPRPU bit to 0. (2) Wait for at least 1 μs. (3) Set the DCFM bit to 1. (4) Wait for at least 200 ns. (5) Set the DCFM bit to 0.
3 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	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 tables 26.3 and 26.4. This bit should be modified while SCKE is 1. When the host controller function 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.

Table 26.3 Register Bits Initialized by Writing USBE = 0 (when Function Controller Function is Selected)

Register Name	Bit Name	Remarks
SYSSTS	LNST	The value is retained when the host controller function is selected.
DVSTCTR	RHST	
INTSTS0	DVSQ	The value is retained when the host controller function is selected.
USBADDR	USBADDR	The value is retained when the host controller function is selected.
USEREQ	BRequest, bmRequestType	The values are retained when the host controller function is selected.
USBVAL	wValue	The value is retained when the host controller function is selected.
USBINDX	wIndex	The value is retained when the host controller function is selected.
USBLENG	wLength	The value is retained when the host controller function is selected.

chanhvinh.com

Table 26.4 Register Bits Initialized by Writing USBE = 0 (when Host Controller Function is Selected)

Register Name	Bit Name	Remarks
DVSTCTR	RHST	
FRMNUM	FRNM	The value is retained when the function controller function is selected.
UFRMNUM	UFRNM	The value is retained when the function controller function is selected.

26.3.2 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 SCKE bit in SYSCFG 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[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	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 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	BWAIT[3:0]	1111	R/W	CPU Bus Access Wait Specifies the number of wait cycles to be inserted during an access to a register (the same number applies to an access to a FIFO port). For details, see section 26.4.1 (5), Register Access Wait Control. 0000: 0 wait cycles (2 access cycles) : 0010: 2 wait cycles (4 access cycles) : 0100: 4 wait cycles (6 access cycles) : 1111: 15 wait cycles (17 access cycles)

chanhvinh.com

26.3.3 System Configuration Status Register (SYSSTS)

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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LNST[1:0]	
Initial value:	0	0	0	0	0	Undefined	0	0	0	0	0	0	0	0	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
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
9 to 2	—	All 0	R	Reserved These bits are always read as 0. 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 26.5. These bits should be read after setting DPRPU to 1 to notify connection when the function controller function is selected; whereas after setting DRPD to 1 to enable pulling down the lines when the host controller function is selected.

Note: * Depends on the DP and DM pin status.

chanhvinh.com

Table 26.5 USB Data Bus Line Status

LNST[1]	LNST[0]	During Low-Speed Operation (only when Host Controller Function 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

26.3.4 Device State Control Register (DVSTCTR)

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	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	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 9	—	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
8	WKUP	0	R/W*	Wakeup Output Enables or disables outputting the remote wakeup signal (resume signal) to the USB bus when the function controller function is selected. 0: Remote wakeup signal is not output. 1: Remote wakeup signal is output. The 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. Note: Do not write 1 to this bit, unless the device state is in the suspended state (the DVSQ bit in the INTSTS0 register is 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 SCKE is 1). This bit should be set to 0 if the host controller function is selected.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 function 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 remote wakeup signal, this module detects the resume signal (K-state for 2.5 μs) from the downstream port device and 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 (SCKE 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 function 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 function is selected. 0: USB bus reset signal is not output. 1: USB bus reset signal is output. When the host controller function 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). USBRST should be 1 (= USB bus reset period) for the time defined by USB Specification 2.0. Writing 1 to this bit during communication (UACT = 1) or during the resume process (RESUME = 1) prevents this module from starting 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 function is selected.
5	RESUME	0	R/W	Resume Output Controls the resume signal output when the host controller function 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. This module continues outputting K-state while RESUME is 1 (until 0 is written to RESUME). RESUME should be 1 (= resume period) for the time defined by USB Specification 2.0. This bit should 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 function is selected.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 function 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 has been 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 function is selected.
3	—	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
2 to 0	RHST[2:0]	000	R	Reset Handshake Indicates the status of the reset handshake. (1) When the host controller function 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 has been 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 the UTST bits are set to 1xxx (when a host test mode is specified), the RHST bits indicate 011. (2) When the function controller function is selected 000: Communication speed not determined 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: * Only 1 can be written.

chanhvinh.com

26.3.5 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	0	0	0	1	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
15 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
7 to 4	—	All 0	R	Reserved These bits are always read as 0. 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. Table 26.6 shows test mode operation of this module.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	UTST[3:0]	0000	R/W	(1) When the host controller function is selected These bits can be set after writing 1 to DRPD. This module outputs waveforms when both DPRD and UACT are set to 1. This module also performs high-speed termination after the UTST bits are written to. • Procedure for setting the UTST bits 1. Power-on reset. 2. Start the clock supply (Set SCKE to 1 after the crystal oscillation and the PLL for USB are settled). 3. Set DCFM and DPRD to 1 (setting HSE to 1 is not required). 4. Set USBE to 1. 5. Set the UTST bits to the appropriate value according to the test specifications. 6. Set the UACT bit to 1. • Procedure for modifying the UTST bits 1. (In the state after executing step 6 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 is set to UACT. When these bits are set to Test_Force_Enable (1101), this module outputs the SOF packet when 1 is 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 and executed, a power-on reset should be applied.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	UTST[3:0]	0000	R/W	(2) When the function controller function 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.

Table 26.6 Test Mode Operation

Test Mode	UTST Bit Setting	
	When Function Controller Function is Selected	When Host Controller Function 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

chanhvinh.com

26.3.6 DMA-FIFO Bus Configuration Registers (D0FBCFG, D1FBCFG)

D0FBCFG is a register that controls DMA0-FIFO bus accesses. D1FBCFG is a register that controls DMA1-FIFO bus accesses.

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
	—	—	—	—	—	—	—	—	—	—	—	TENDE	—	—	—	—
Initial value:	0	0	0	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	0	Undefined	Undefined	Undefined	Undefined
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 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
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 output from the direct memory access controller on completion of a DMA transfer. For details, see section 26.4.4 (3), DMA Transfers (D0FIFO/D1FIFO Port). 0: DMA transfer end signal is not sampled. 1: DMA transfer end signal is sampled. For a DMA transfer size of 16 bytes, clear the TENDE bit to 0.
3 to 0	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.

chanhvinh.com

26.3.7 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 writing data to the FIFO buffer memory.

There are three FIFO ports: the CFIFO, D0FIFO and D1FIFO ports. Each FIFO port is configured of a port register (CFIFO, D0FIFO, 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, D1FIFOSEL) that is used to select the pipe assigned to the FIFO port, and a control register (CFIFOCTR, D0FIFOCTR, D1FIFOCTR).

Each FIFO port has the following features.

- The DCP FIFO buffer should be accessed through the CFIFO port.
- Accessing the FIFO buffer using DMA transfer should be performed through the D0FIFO or D1FIFO port.
- The D1FIFO and D0FIFO ports can be accessed also 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

chanhvinh.com

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 control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR) is 1. 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 tables 26.7 to 26.9.

Table 26.7 Endian Operation in 32-Bit Access (when MBW = 10)

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 26.8 Endian Operation in 16-Bit Access (when MBW = 01)

BIGEND Bit	Bits 31 to 24	Bits 23 to 16	Bits 15 to 8	Bits 7 to 0
0	Writing: invalid, reading: prohibited*		N + 1 address	N + 0 address
1	N + 0 address	N + 1 address	Writing: invalid, reading: prohibited*	

Note: * Reading data from the invalid bits in a word or byte unit is prohibited.

Table 26.9 Endian Operation in 8-Bit Access (when MBW = 00)

BIGEND Bit	Bits 31 to 24	Bits 23 to 16	Bits 15 to 8	Bits 7 to 0
0	Writing: invalid, reading: prohibited*			N + 0 address
1	N + 0 address		Writing: invalid, reading: prohibited*	

Note: * Reading data from the invalid bits in a word or byte unit is prohibited.

chanhvinh.com

26.3.8 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'000, 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]		—	BIGEND	—	—	ISEL	—	CURPIPE[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	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 bit is 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 bit is decremented when the receive data is read from the CFIFO.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 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 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	—	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
11, 10	MBW[1:0]	00	R/W	CFIFO 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 Once reading data is started after setting these bits, these bits should not be modified until all the data has been read. When the selected pipe is in the receiving direction, these bits should be set in the following timing: • Set the CURPIPE and MBW bits simultaneously. • When the DCP is selected (CURPIPE = B'000), set the ISEL and MBW bits simultaneously. For details, see section 26.4.4, FIFO Buffer Memory. When the selected 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	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8	BIGEND	0	R/W	CFIFO Port Endian Control Specifies the byte endian for the CFIFO port. 0: Little endian 1: Big endian
7, 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	ISEL	0	R/W	CFIFO Port Access Direction When DCP is Selected 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. Even if an attempt is made to modify the setting of this bit during access to the FIFO buffer, the current access setting is retained until the access is completed. Then, the modification becomes effective thus enabling continuous access. Set this bit and the CURPIPE bits simultaneously.
4	—	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
3 to 0	CURPIPE[3:0]	0000	R/W	CFIFO Port Access Pipe Specification Specifies the pipe number for reading or writing data through the CFIFO port. 0000: DCP 0001: Pipe 1 0010: Pipe 2 0011: Pipe 3 0100: Pipe 4 0101: Pipe 5 0110: Pipe 6 0111: Pipe 7 1000: Pipe 8 1001: Pipe 9 Other than above: Setting prohibited 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 an attempt is made to modify the setting of these bits during access to the FIFO buffer, the current access setting is retained until the access is completed. Then, the modification becomes effective thus enabling continuous access.

Note: * Only 0 can be read and 1 can be written.

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]		—	BIG END	—	—	—	—	CURPIPE[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/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 bit is 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 bit is decremented when the receive data is read from the DnFIFO. When accessing DnFIFO with the BFRE bit set to 1, set this bit to 0.
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 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 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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 DMA transfer request to be issued. 0: Request disabled 1: Request enabled Before setting this bit to 1 to enable the DMA transfer request to be issued, set the CURPIPE bits. Before modifying the CURPIPE bit setting, 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 DnFIFO port. 00: 8-bit width 01: 16-bit width 10: 32-bit width 11: Setting prohibited Once reading data is started after setting these bits, these bits should not be modified until all the data has been read. When the selected pipe is in the receiving direction, set the CURPIPE and MBW bits simultaneously. For details, see section 26.4.4, FIFO Buffer Memory. When the selected 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	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8	BIGEND	0	R/W	FIFO Port Endian Control Specifies the byte endian for the DnFIFO port. 0: Little endian 1: Big endian
7 to 4	—	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
3 to 0	CURPIPE[3:0]	0000	R/W	FIFO Port Access Pipe Specification Specifies the pipe number for reading or writing data through the D0FIFO/D1FIFO port. 0000: No pipe specified 0001: Pipe 1 0010: Pipe 2 0011: Pipe 3 0100: Pipe 4 0101: Pipe 5 0110: Pipe 6 0111: Pipe 7 1000: Pipe 8 1001: Pipe 9 Other than above: Setting prohibited 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 an attempt is made to modify the setting of these bits during access to the FIFO buffer, the current access setting is retained until the access is completed. Then, the modification becomes effective thus enabling continuous access.

Note: * Only 0 can be read and 1 can be written.

26.3.9 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.

chanhvinh.com

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	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 This bit should be set to 1 when data has been completely written to the FIFO buffer on the CPU side for the pipe selected using the CURPIPE bits (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 data is written 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 integer 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. When the selected pipe is in the transmitting direction, if 1 is written to BVAL and BCLR bits simultaneously, this module clears the data that has been written before it, enabling transmission of a zero-length packet. Writing 1 to this bit should be done while FRDY indicates 1 (set by this module). When the selected pipe is in the receiving direction, do not set this bit to 1.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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. To clear 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 (set by this module).
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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
12	—	0	R	Reserved This bit is always read as 0. 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. • RCNT = 0: This module sets these bits to indicate the length of the receive data until all the 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. • 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 former plane when all the data has been read from the latter plane. Note: When RCNT is 1, it takes 10 bus cycles for these bits to be updated after the FIFO port has been read.

Notes: 1. Only 0 can be read and 1 can be written.
2. Only 1 can be written.

chanhvinh.com

26.3.10 Interrupt Enable Register 0 (INTENB0)

INTENB0 is a register that enables or disables the various interrupts. On detecting the interrupt corresponding to the bit that 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 modified 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	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 request disabled 1: Interrupt request enabled
14	RSME	0	R/W	Resume Interrupt Enable* Enables or disables the USB interrupt request when the RESM interrupt is detected. 0: Interrupt request disabled 1: Interrupt request 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 request disabled 1: Interrupt request enabled

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
12	DVSE	0	R/W	Device State Transition Interrupt Enable* Enables or disables the USB interrupt request when the DVST interrupt is detected. 0: Interrupt request disabled 1: Interrupt request enabled
11	CTRE	0	R/W	Control Transfer Stage Transition Interrupt Enable* Enables or disables the USB interrupt request when the CTRT interrupt is detected. 0: Interrupt request disabled 1: Interrupt request 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 request disabled 1: Interrupt request 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 request disabled 1: Interrupt request 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 request disabled 1: Interrupt request enabled
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * The RSME, DVSE, and CTRE bits can be set to 1 only when the function controller function is selected; do not set these bits to 1 to enable the corresponding interrupt output when the host controller function is selected.

chanhvinh.com

26.3.11 Interrupt Enable Register 1 (INTENB1)

INTENB1 is a register that enables or disables the various interrupts when the host controller function is selected. On detecting the interrupt corresponding to the bit in this register that 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 modified from 0 to 1.

When the function controller function 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	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 request disabled 1: Interrupt request enabled
13	—	0	R	Reserved This bit is always read as 0. 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 request disabled 1: Interrupt request enabled

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
11	ATTCHE	0	R/W	Connection Detection Interrupt Enable Enables or disables the USB interrupt request when the ATTCH interrupt is detected. 0: Interrupt request disabled 1: Interrupt request enabled
10 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
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 request disabled 1: Interrupt request 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 request disabled 1: Interrupt request 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 request disabled 1: Interrupt request enabled
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: The INTENB1 register bits can be set to 1 only when the host controller function is selected; do not set these bits to 1 to enable the corresponding interrupt output when the function controller function is selected.

chanhvinh.com

26.3.12 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 that 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 modified 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
	—	—	—	—	—	—	PIPE9 BRDYE	PIPE8 BRDYE	PIPE7 BRDYE	PIPE6 BRDYE	PIPE5 BRDYE	PIPE4 BRDYE	PIPE3 BRDYE	PIPE2 BRDYE	PIPE1 BRDYE	PIPE0 BRDYE
Initial value:	0	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
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	PIPE9BRDYE	0	R/W	BRDY interrupt Enable for PIPE9 0: Interrupt output disabled 1: Interrupt output enabled
8	PIPE8BRDYE	0	R/W	BRDY interrupt Enable for PIPE8 0: Interrupt output disabled 1: Interrupt output enabled
7	PIPE7BRDYE	0	R/W	BRDY interrupt Enable for PIPE7 0: Interrupt output disabled 1: Interrupt output enabled
6	PIPE6BRDYE	0	R/W	BRDY interrupt Enable for PIPE6 0: Interrupt output disabled 1: Interrupt output enabled

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
5	PIPE5BRDYE	0	R/W	BRDY interrupt Enable for PIPE5 0: Interrupt output disabled 1: Interrupt output enabled
4	PIPE4BRDYE	0	R/W	BRDY interrupt Enable for PIPE4 0: Interrupt output disabled 1: Interrupt output enabled
3	PIPE3BRDYE	0	R/W	BRDY interrupt Enable for PIPE3 0: Interrupt output disabled 1: Interrupt output enabled
2	PIPE2BRDYE	0	R/W	BRDY interrupt Enable for PIPE2 0: Interrupt output disabled 1: Interrupt output enabled
1	PIPE1BRDYE	0	R/W	BRDY interrupt Enable for PIPE1 0: Interrupt output disabled 1: Interrupt output enabled
0	PIPE0BRDYE	0	R/W	BRDY interrupt Enable for PIPE0 0: Interrupt output disabled 1: Interrupt output enabled

26.3.13 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 that 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 modified from 0 to 1.

This register is initialized by a power-on reset.

chanhvinh.com

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PIPE9 NRDYE	PIPE8 NRDYE	PIPE7 NRDYE	PIPE6 NRDYE	PIPE5 NRDYE	PIPE4 NRDYE	PIPE3 NRDYE	PIPE2 NRDYE	PIPE1 NRDYE	PIPE0 NRDYE
Initial value:	0	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
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	PIPE9NRDYE	0	R/W	NRDY Interrupt Enable for PIPE9 0: Interrupt output disabled 1: Interrupt output enabled
8	PIPE8NRDYE	0	R/W	NRDY Interrupt Enable for PIPE8 0: Interrupt output disabled 1: Interrupt output enabled
7	PIPE7NRDYE	0	R/W	NRDY Interrupt Enable for PIPE7 0: Interrupt output disabled 1: Interrupt output enabled
6	PIPE6NRDYE	0	R/W	NRDY Interrupt Enable for PIPE6 0: Interrupt output disabled 1: Interrupt output enabled
5	PIPE5NRDYE	0	R/W	NRDY Interrupt Enable for PIPE5 0: Interrupt output disabled 1: Interrupt output enabled
4	PIPE4NRDYE	0	R/W	NRDY Interrupt Enable for PIPE4 0: Interrupt output disabled 1: Interrupt output enabled
3	PIPE3NRDYE	0	R/W	NRDY Interrupt Enable for PIPE3 0: Interrupt output disabled 1: Interrupt output enabled
2	PIPE2NRDYE	0	R/W	NRDY Interrupt Enable for PIPE2 0: Interrupt output disabled 1: Interrupt output enabled

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
1	PIPE1NRDYE	0	R/W	NRDY Interrupt Enable for PIPE1 0: Interrupt output disabled 1: Interrupt output enabled
0	PIPE0NRDYE	0	R/W	NRDY Interrupt Enable for PIPE0 0: Interrupt output disabled 1: Interrupt output enabled

26.3.14 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 that 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 modified 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
	—	—	—	—	—	—	PIPE9 BEMPE	PIPE8 BEMPE	PIPE7 BEMPE	PIPE6 BEMPE	PIPE5 BEMPE	PIPE4 BEMPE	PIPE3 BEMPE	PIPE2 BEMPE	PIPE1 BEMPE	PIPE0 BEMPE
Initial value:	0	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
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	PIPE9BEMPE	0	R/W	BEMP Interrupt Enable for PIPE9 0: Interrupt output disabled 1: Interrupt output enabled

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
8	PIPE8BEMPE	0	R/W	BEMP Interrupt Enable for PIPE8 0: Interrupt output disabled 1: Interrupt output enabled
7	PIPE7BEMPE	0	R/W	BEMP Interrupt Enable for PIPE7 0: Interrupt output disabled 1: Interrupt output enabled
6	PIPE6BEMPE	0	R/W	BEMP Interrupt Enable for PIPE6 0: Interrupt output disabled 1: Interrupt output enabled
5	PIPE5BEMPE	0	R/W	BEMP Interrupt Enable for PIPE5 0: Interrupt output disabled 1: Interrupt output enabled
4	PIPE4BEMPE	0	R/W	BEMP Interrupt Enable for PIPE4 0: Interrupt output disabled 1: Interrupt output enabled
3	PIPE3BEMPE	0	R/W	BEMP Interrupt Enable for PIPE3 0: Interrupt output disabled 1: Interrupt output enabled
2	PIPE2BEMPE	0	R/W	BEMP Interrupt Enable for PIPE2 0: Interrupt output disabled 1: Interrupt output enabled
1	PIPE1BEMPE	0	R/W	BEMP Interrupt Enable for PIPE1 0: Interrupt output disabled 1: Interrupt output enabled
0	PIPE0BEMPE	0	R/W	BEMP Interrupt Enable for PIPE0 0: Interrupt output disabled 1: Interrupt output enabled

26.3.15 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.

chanhvinh.com

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	0	0	0	0	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	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	TRNENSEL	0	R/W	Transaction-Enabled Time Select Selects the transaction-enabled time either for full- or low-speed communication, during 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 function is selected. Even when the host controller function 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 function is selected.
7	—	0	R	Reserved This bit is always read as 0. 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. Set the BRDYM bit during the initial settings of the USB 2.0 host/function module (before performing data communication). Do not change the setting of the BRDYM bit after data communication starts. 0: Writing 0 clears the status. 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 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

26.3.16 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 on detecting a level change (high to low or low to high) in the VBUS pin input value. This module sets the VBSTS bit to indicate the VBUS pin input value. When the VBUS interrupt is generated, repeat reading the VBSTS bit until the same value is read several times to eliminate chattering.
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 function 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 function is selected, the read value is invalid.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 function is selected This module sets this bit to 1 on updating the frame number when the UACT bit has been set to 1. (This interrupt is detected every 1 ms.) (2) When the function controller function 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 function 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 function is selected, the read value is invalid.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 function 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 function is selected, the read value is invalid.
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 output is enabled). For the conditions for PIPEBEMP status assertion, refer to section 26.4.2 (3), 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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 output is enabled). For the conditions for PIPENRDY status assertion, refer to section 26.4.2 (2), 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 Indicates the BRDY 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 output is enabled). For the conditions for PIPEBRDY status assertion, refer to section 26.4.2 (1), 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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 function 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 function 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 function is selected, the read value is invalid.

chanhvinh.com

Notes:
1. This bit is initialized to B'0 by a power-on reset and B'1 by a USB bus reset.
2. These bits are initialized to B'000 by a power-on reset and B'001 by a USB bus reset.
3. This bit is initialized to 0 when the level of the VBUS pin input is high and 1 when low.
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.
5. This module can detect a change in the status indicated by the VBINT and RESM bits even while the clock supply is stopped (while SCKE is 0), and outputs interrupts when the corresponding interrupt enable bits are enabled. Clearing the status should be done after enabling the clock supply.
6. A change in the status of the RESM, DVST, and CTRT bits occur only when the function controller function is selected; disable the corresponding interrupt enable bits (set to 0) when the host controller function is selected.
7. Only 0 can be written.

26.3.17 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 function 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	—	—	—	—	EOF ERR	SIGN	SACK	—	—	—	—
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/W*1	R/W*1	R	R	R	R	R/W*1	R/W*1	R/W*1	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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
14	BCHG	0	R/W*[1]	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 has been set to 1, this module generates the interrupt. This module sets the LNST bits in SYSSTS0 to indicate the current input state of the USB port. When the BCHG interrupt is generated, repeat reading the LNST bits until the same value is read several times, and eliminate chattering. A change in the USB bus state can be detected even while the internal clock supply is stopped. When the function controller function is selected, the read value is invalid.
13	—	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
12	DTCH	0	R/W*[1]	USB Disconnection Detection Interrupt Status Indicates the status of the USB disconnection detection interrupt when the host controller function 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 has been 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). 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) Causes a transition to the idle state. When the function controller function is selected, the read value is invalid.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
11	ATTCH	0	R/W*[1]	ATTCH Interrupt Status Indicates the status of the ATTCH interrupt when the host controller function is selected. 0: ATTCH interrupts not generated 1: ATTCH interrupts generated When this module has generated J-state or K-state of the full-speed or low-speed level signal for 2.5 μs, this module detects the ATTCH interrupt and sets this bit to 1. Here, if the corresponding interrupt enable bit has been set to 1, this module generates the interrupt. Specifically, this module detects the ATTCH interrupt on any of the following conditions. • K-stateSE0, 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 function is selected, the read value is invalid.
10 to 7	—	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
6	EOFERR	0	R/W*[1]	EOF Error Detection Interrupt Status Indicates the status of the EOFERR interrupt when the host controller function 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 has been 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). 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) Causes a transition to the idle state. When the function controller function is selected, the read value is invalid.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
5	SIGN	0	R/W*[1]	Setup Transaction Error Interrupt Status Indicates the status of the setup transaction error interrupt when the host controller function 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 has been 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 function is selected, the read value is invalid.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
4	SACK	0	R/W*[1]	Setup Transaction Normal Response Interrupt Status Indicates the status of the setup transaction normal response interrupt when the host controller function 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 has been set to 1, this module generates the SACK interrupt.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Notes: 1. Only 0 can be written.
2. This module can detect a change in the status indicated by the BCHG bit even while the clock supply is stopped (while SCKE is 0), and outputs an interrupt 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 (while SCKE is 0).

26.3.18 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
	—	—	—	—	—	—	PIPE9 BRDY	PIPE8 BRDY	PIPE7 BRDY	PIPE6 BRDY	PIPE5 BRDY	PIPE4 BRDY	PIPE3 BRDY	PIPE2 BRDY	PIPE1 BRDY	PIPE0 BRDY
Initial value:	0	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*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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	PIPE9BRDY	0	R/W*[1]	BRDY Interrupt Status for PIPE9*[2] 0: Interrupts not generated 1: Interrupts generated
8	PIPE8BRDY	0	R/W*[1]	BRDY Interrupt Status for PIPE8*[2] 0: Interrupts not generated 1: Interrupts generated
7	PIPE7BRDY	0	R/W*[1]	BRDY Interrupt Status for PIPE7*[2] 0: Interrupts not generated 1: Interrupts generated
6	PIPE6BRDY	0	R/W*[1]	BRDY Interrupt Status for PIPE6*[2] 0: Interrupts not generated 1: Interrupts generated
5	PIPE5BRDY	0	R/W*[1]	BRDY Interrupt Status for PIPE5*[2] 0: Interrupts not generated 1: Interrupts generated
4	PIPE4BRDY	0	R/W*[1]	BRDY Interrupt Status for PIPE4*[2] 0: Interrupts not generated 1: Interrupts generated
3	PIPE3BRDY	0	R/W*[1]	BRDY Interrupt Status for PIPE3*[2] 0: Interrupts not generated 1: Interrupts generated
2	PIPE2BRDY	0	R/W*[1]	BRDY Interrupt Status for PIPE2*[2] 0: Interrupts not generated 1: Interrupts generated

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
1	PIPE1BRDY	0	R/W*[1]	BRDY Interrupt Status for PIPE1*[2] 0: Interrupts not generated 1: Interrupts generated
0	PIPE0BRDY	0	R/W*[1]	BRDY Interrupt Status for PIPE0*[2] 0: Interrupts not generated 1: Interrupts generated

Notes: 1. Only 0 can be written.
2. When BRDYM is 0, clearing this bit should be done before accessing the FIFO.

26.3.19 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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PIPE9 NRDY	PIPE8 NRDY	PIPE7 NRDY	PIPE6 NRDY	PIPE5 NRDY	PIPE4 NRDY	PIPE3 NRDY	PIPE2 NRDY	PIPE1 NRDY	PIPE0 NRDY
Initial value:	0	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
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	PIPE9NRDY	0	R/W*	NRDY Interrupt Status for PIPE9 0: Interrupts not generated 1: Interrupts generated
8	PIPE8NRDY	0	R/W*	NRDY Interrupt Status for PIPE8 0: Interrupts not generated 1: Interrupts generated

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
7	PIPE7NRDY	0	R/W*	NRDY Interrupt Status for PIPE7 0: Interrupts not generated 1: Interrupts generated
6	PIPE6NRDY	0	R/W*	NRDY Interrupt Status for PIPE6 0: Interrupts not generated 1: Interrupts generated
5	PIPE5NRDY	0	R/W*	NRDY Interrupt Status for PIPE5 0: Interrupts not generated 1: Interrupts generated
4	PIPE4NRDY	0	R/W*	NRDY Interrupt Status for PIPE4 0: Interrupts not generated 1: Interrupts generated
3	PIPE3NRDY	0	R/W*	NRDY Interrupt Status for PIPE3 0: Interrupts not generated 1: Interrupts generated
2	PIPE2NRDY	0	R/W*	NRDY Interrupt Status for PIPE2 0: Interrupts not generated 1: Interrupts generated
1	PIPE1NRDY	0	R/W*	NRDY Interrupt Status for PIPE1 0: Interrupts not generated 1: Interrupts generated
0	PIPE0NRDY	0	R/W*	NRDY Interrupt Status for PIPE0 0: Interrupts not generated 1: Interrupts generated

Note: * Only 0 can be written.

chanhvinh.com

26.3.20 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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PIPE9 BEMP	PIPE8 BEMP	PIPE7 BEMP	PIPE6 BEMP	PIPE5 BEMP	PIPE4 BEMP	PIPE3 BEMP	PIPE2 BEMP	PIPE1 BEMP	PIPE0 BEMP
Initial value:	0	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
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	PIPE9BEMP	0	R/W*	BEMP Interrupts for PIPE9 0: Interrupts not generated 1: Interrupts generated
8	PIPE8BEMP	0	R/W*	BEMP Interrupts for PIPE8 0: Interrupts not generated 1: Interrupts generated
7	PIPE7BEMP	0	R/W*	BEMP Interrupts for PIPE7 0: Interrupts not generated 1: Interrupts generated
6	PIPE6BEMP	0	R/W*	BEMP Interrupts for PIPE6 0: Interrupts not generated 1: Interrupts generated
5	PIPE5BEMP	0	R/W*	BEMP Interrupts for PIPE5 0: Interrupts not generated 1: Interrupts generated
4	PIPE4BEMP	0	R/W*	BEMP Interrupts for PIPE4 0: Interrupts not generated 1: Interrupts generated

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
3	PIPE3BEMP	0	R/W*	BEMP Interrupts for PIPE3 0: Interrupts not generated 1: Interrupts generated
2	PIPE2BEMP	0	R/W*	BEMP Interrupts for PIPE2 0: Interrupts not generated 1: Interrupts generated
1	PIPE1BEMP	0	R/W*	BEMP Interrupts for PIPE1 0: Interrupts not generated 1: Interrupts generated
0	PIPE0BEMP	0	R/W*	BEMP Interrupts for PIPE0 0: Interrupts not generated 1: Interrupts generated

Note: * Only 0 can be written.

26.3.21 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	0	0	0
R/W:	R/W*	R/W*	R	R	R	R	R	R	R	R	R	R	R	R	R	R

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
15	OVRN	0	R/W*	Overrun/Underrun Detection Status Indicates whether an overrun/underrun error has been detected in the pipe during isochronous transfer. 0: No error 1: An error occurred This bit can be cleared to 0 by writing 0 to the bit. (1) When the host controller function 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 function 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 is provided for debugging. The system should be designed so that no overrun/underrun should occur.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
14	CRCE	0	R/W*	Receive Data Error Indicates whether a CRC error or bit stuffing error has been detected in the pipe during isochronous transfer. Simultaneously with error detection, the internal NRDY interrupt request is generated. For details, see section 26.4.2, Interrupt Functions. 0: No error 1: An error occurred
13 to 11	—	All 0	R	Reserved These bits are always read as 0. 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) Read these bits twice to check that the same value is read.

Note: * Only 0 can be written

chanhvinh.com

26.3.22 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	0	0	0	0	0	0	0	0	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 3	—	All 0	R	Reserved These bits are always read as 0. 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. Read these bits twice to check that the same value is read.

chanhvinh.com

26.3.23 USB Address Register (USBADDR)

USBADDR is a register that indicates the USB address. This register is valid only when the function controller function is selected. When the host controller function 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	0	0	0	0	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	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6 to 0	USBADDR [6:0]	H'00	R	USB Address When the function controller function is selected, these bits indicate the USB address assigned by the host when the SET_ADDRESS request is successfully processed.

chanhvinh.com

26.3.24 USB Request Type Register (USBREQ)

USBREQ is a register that stores setup requests for control transfers. When the function controller function is selected, the values of bRequest and bmRequestType that have been received are stored. When the host controller function 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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BREQUEST[7:0]								BMREQUESTTYPE[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
15 to 8	BREQUEST [7:0]	H'00	R/W*	Request These bits store the USB request bRequest value. (1) When the host controller function is selected The USB request data value for the setup transaction to be transmitted should be set in these bits. After setting SUREQ to 1, do not modify these bits until 0 is read from SUREQ. (2) When the function controller function 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*	Request Type These bits store the USB request bmRequestType value. (1) When the host controller function is selected The USB request type value for the setup transaction to be transmitted should be set in these bits. After setting SUREQ to 1, do not modify these bits until 0 is read from SUREQ. (2) When the function controller function is selected Indicates the USB request type value received during the setup transaction. Writing to these bits is invalid.

Note: * When the function controller function is selected, these bits can only be read, and writing to these bits is invalid. When the host controller function is selected, these bits can be read and written to.

chanhvinh.com

26.3.25 USB Request Value Register (USBVAL)

USBVAL is a register that stores setup requests for control transfers. When the function controller function is selected, the value of wValue that has been received is stored. When the host controller function 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*	R/W*	R/W*	R/W*	R/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	WVALUE[15:0]	H'0000	R/W*	Value These bits store the USB request wValue value. (1) When the host controller function is selected The USB request wValue value for the setup transaction to be transmitted should be set in these bits. After setting SUREQ to 1, do not modify these bits until 0 is read from SUREQ. (2) When the function controller function is selected Indicates the USB request wValue value received during the setup transaction. Writing to these bits is invalid.

Note: * When the function controller function is selected, these bits can only be read, and writing to these bits is invalid. When the host controller function is selected, these bits can be read and written to.

chanhvinh.com

26.3.26 USB Request Index Register (USBINDX)

USBINDEX is a register that stores setup requests for control transfers. When the function controller function is selected, the value of wIndex that has been received is stored. When the host controller function 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*	R/W*	R/W*	R/W*	R/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	WINDEX[15:0]	H'0000	R/W*	Index These bits store the USB request wIndex value. (1) When the host controller function is selected The USB request wIndex value for the setup transaction to be transmitted should be set in these bits. After setting SUREQ to 1, do not modify these bits until 0 is read from SUREQ. (2) When the function controller function is selected Indicates the USB request wIndex value received during the setup transaction. Writing to these bits is invalid.

Note: * When the function controller function is selected, these bits can only be read, and writing to these bits is invalid. When the host controller function is selected, these bits can be read and written to.

chanhvinh.com

26.3.27 USB Request Length Register (USBLENG)

USBLENG is a register that stores setup requests for control transfers. When the function controller function is selected, the value of wLength that has been received is stored. When the host controller function 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*	R/W*	R/W*	R/W*	R/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	WLENGTH [15:0]	H'0000	R/W*	Length These bits store the USB request wLength value. (1) When the host controller function is selected The USB request wLength value for the setup transaction to be transmitted should be set in these bits. After setting SUREQ to 1, do not modify these bits until 0 is read from SUREQ. (2) When the function controller function is selected Indicates the USB request wLength value received during the setup transaction. Writing to these bits is invalid.

Note: * When the function controller function is selected, these bits can only be read, and writing to these bits is invalid. When the host controller function is selected, these bits can be read and written to.

chanhvinh.com

26.3.28 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
	—	—	—	—	—	—	—	—	—	—	—	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/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	DIR	0	R/W	Transfer Direction When the host controller function is selected, this bit sets the transfer direction of data stage. 0: Data receiving direction 1: Data transmitting direction When the function controller function is selected, this bit should be cleared to 0.
3 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 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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DEVSEL[3:0]				—	—	—	—	—	MXPS[6:0]						
Initial value:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	DEVSEL[3:0]	0000	R/W	Device Select When the host controller function is selected, these bits specify the communication target peripheral device address. 0000: Address 0000 0001: Address 0001 : : 1001: Address 1001 1010: Address 1010 Other than above: 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. 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, checking PBUSY is not necessary. When the function controller function is selected, these bits should be set to B'0000.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
11 to 7	—	All 0	R	Reserved These bits are always read as 0. 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 and PID is NAK. 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, checking PBUSY is not necessary. While MXPS is 0, do not write to the FIFO buffer or do not set PID to BUF.

26.3.30 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	SUREQ CLR	—	—	SQCLR	SQSET	SQMON	PBUSY	PINGE	—	CCPL	PID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	1	0	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*1	R/W	R/W

chanhvinh.com

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 function is selected. 0: 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 until the setup transaction is completed (SUREQ = 1). Write 1 to this bit only when transmitting the setup token; for the other purposes, write 0. When the function controller function is selected, be sure to write 0 to this bit.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
13	CSCLR	0	R/W*[1]	C-SPLIT Status Clear for Split Transaction When the host controller function 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: 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 function is selected, be sure to write 0 to this bit.
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 function 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 function is selected, the read value is invalid.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
11	SUREQCLR	0	R/W*[1]	SUREQ Bit Clear When the host controller function is selected, setting this bit to 1 clears the SUREQ bit to 0. 0: 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 function is selected, be sure to write 0 to this bit.
10, 9	—	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
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: 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 CSCTS 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, checking PBUSY is not necessary.
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: 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, checking PBUSY is not necessary.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 function 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 function 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.
5	PBUSY	0	R	Pipe Busy This bit indicates whether or not a transaction using the pertinent pipe is currently in progress. 0: The pertinent pipe is not in use by a transaction. 1: The pertinent pipe is in use by a transaction. The USB 2.0 host/function module changes the setting of the PBUSY bit from 0 to 1 when a USB transaction using the DCP starts. It clears the PBUSY bit from 1 to 0 when one transaction completes successfully. Reading this bit after setting PID to NAK allows checking that modification of the pipe settings is possible. For details, refer to section 26.4.3 (1), Pipe Control Register Switching Procedures.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
4	PINGE	0	R/W	PING Token Issue Enable When the host controller function 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 during OUT transactions, this module performs the PING transaction as the next transaction. When the host controller function 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. These bits should be modified while CSSTS is 0 and PID is NAK. Before changing the value of this bit after changing the PID bits for the DCP from BUF to NAK, confirm that CSSTS and PBUSY are both cleared to 0. However, if the PID bits have been modified to NAK by this module, checking PBUSY is not necessary. When the function controller function is selected, be sure to write 0 to this bit.
3	—	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
2	CCPL	0	R/W*[1]	Control Transfer End Enable When the function controller function is selected, setting this bit to 1 enables the status stage of the control transfer to be completed. 0: Invalid 1: Completion of control transfer is enabled. When this bit is set to 1 while the corresponding PID bits are set to BUF, this module completes the control transfer 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. A 1 cannot be written to this bit while VALID is 1. When the host controller function 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 function 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 receive error such as a CRC error three consecutive times. • This module also sets PID to STALL (11) on receiving the STALL handshake.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
1, 0	PID[1:0]	00	R/W	Even if the PID bits have been 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. On completion of C-SPLIT, this module sets PID to NAK. (2) When the function controller function 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. 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).

Notes: 1. This bit is always read as 0. Only 1 can be written.
2. Only 1 can be written.

chanhvinh.com

26.3.31 Pipe Window Select Register (PIPESEL)

PIPE1 to PIPE 9 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	0	0	0	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
15 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	PIPESEL[3:0]	0000	R/W	Pipe Window Select Setting 0001 to 1001to these bits, the PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI registers, these registers indicate the information or set values of the corresponding pipe. 0000: No pipe selected 0001: PIPE1 0010: PIPE2 0011: PIPE3 0100: PIPE4 0101: PIPE5 0110: PIPE6 0111: PIPE7 1000: PIPE8 1001: PIPE9 Other than above: Setting prohibited Setting 0000 to these bits, the PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI registers all indicate 0. Here, writing to these registers are invalid.

chanhvinh.com

26.3.32 Pipe Configuration Register (PIPECFG)

PIPECFG is a register that specifies the transfer type, buffer memory access direction, and endpoint numbers for PIPE1 to PIPE9. 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	SHT NAK	—	—	DIR	EPNUM[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	R	R	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
15, 14	TYPE[1:0]	00	R/W	Transfer Type Selects the transfer type for the pipe selected by the PIPESEL bits (selected pipe) • PIPE1 and PIPE2 00: Pipe cannot be used 01: Bulk transfer 10: Setting prohibited 11: Isochronous transfer • PIPE3 to PIPE5 00: Pipe cannot be used 01: Bulk transfer 10: Setting prohibited 11: Setting prohibited • PIPE6 and PIPE7 00: Pipe cannot be used 01: Setting prohibited 10: Interrupt transfer 11: Setting prohibited 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, checking PBUSY is not necessary.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
13 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
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 of data 1: BRDY interrupt upon completion of reading of data This bit is valid when any of pipes 1 to 5 is selected. When this bit has been 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 has been 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 26.4.2 (1), BRDY Interrupt. Modify these bits while CSSTS is 0 and PID is NAK and before the pipe is selected by the CURPIPE bits. To modify these bits 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 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, checking PBUSY is not necessary.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 are selected. When this bit has been 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 these bits while CSSTS is 0 and PID is NAK and before the pipe is selected by the CURPIPE bits. To modify these bits 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 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, checking PBUSY is not necessary.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 are selected by the PIPESEL bits and bulk transfer is selected (TYPE = 01). Modify these bits while CSSTS is 0 and PID is NAK and before the pipe is selected by the CURPIPE bits. To modify these bits 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 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, checking PBUSY is not necessary.

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 in the receiving direction. When this bit has been 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 are successfully received. Modify these bits while CSSTS is 0 and PID is 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, checking PBUSY is not necessary. This bit should be cleared to 0 for the pipe in the transmitting direction.
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	DIR	0	R/W	Transfer Direction Specifies the transfer direction for the selected pipe. 0: Receiving direction 1: Sending direction When this bit has been set to 0, this module uses the selected pipe in the receiving direction, and when this bit has been set to 1, this module uses the selected pipe in the transmitting direction. Modify these bits when the value of CSSTS is 0, the PID bits are set to NAK, and no pipe is specified by the CURPIPE bits. To modify these bits 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 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, checking PBUSY is not necessary.
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 and PID is 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, checking PBUSY is not necessary. 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

26.3.33 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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	BUFSIZE[4:0]					—	—	—	BUFNMB[6:0]						
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	R	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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 has been 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: Any value from H'00 to H'1F is valid. • PIPE6 to PIPE9: H'00 should be set. When used with CNTMD = 1, set an integer multiple of the maximum packet size to the BUFSIZE bits. Modify these bits when the value of CSSTS is 0, the PID bits are set to NAK, and no pipe is specified 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, checking PBUSY is not necessary.
9 to 7	—	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
6 to 0	BUFNMB[6:0]	H'00	R/W	Buffer Number These bits specify the start block number of the FIFO buffers to be assigned to the selected pipe. The following blocks of FIFO buffers are assigned to the selected pipe. Block number BUFNMB to block number BUFNMB + (BUFSIZE + 1) × (DBLB + 1) - 1 Specify a value from H'04 to H'7F. When the selected pipe is one of PIPE1 to PIPE5, any value can be set to these bits according to the user system. BUFNUMB = H'00 to H'03 are 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. BUFNUMB = 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. BUFNUMB = 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 invalid and H'07 is automatically assigned by this module. Modify these bits when the value of CSSTS is 0, the PID bits are set to NAK, and no pipe is specified 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, checking PBUSY is not necessary.

chanhvinh.com

26.3.34 Pipe Maximum Packet Size Register (PIPEMAXP)

PIPEMAXP is a register that specifies the maximum packet size for PIPE1 to PIPE9.

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
	DEVSEL[3:0]				—	MXPS[10:0]										
Initial value:	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	Bit Name	Initial Value	R/W	Description
15 to 12	DEVSEL[3:0]	0000	R/W	Device Select When the host controller function 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 Other than above: Setting prohibited These bits should be set after setting the address to the DEVADDn register corresponding to the value to be set in these bits. For example, before setting DEVSEL to 0010, the address should be set to the DEVADD2 register. 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, checking PBUSY is not necessary. When the function controller function is selected, these bits should be set to B'0000.
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
10 to 0	MXPS[10:0]	*	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 to 0 are not provided.) PIPE6 to PIPE9: 1 byte (H'001) to 64 bytes (H'040) These bits should be set to the appropriate 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 when the value of CSSTS is 0, the PID bits are set to NAK, and no pipe is specified 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, checking PBUSY is not necessary. While MXPS is 0, do not write to the FIFO buffer or set PID to BUF.

Note: * 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 bit in PIPESEL.

chanhvinh.com

26.3.35 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 PIPE9.

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	0	0	0	0	0	0	0	0	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	—	All 0	R	Reserved These bits are always read as 0. 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 (selected pipe) is used for isochronous IN transfers. 0: The buffer is not flushed. 1: The buffer is flushed. When the function controller function 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 function is selected, set this bit to 0. When the selected pipe is not for the isochronous transfer, set this bit to 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
11 to 3	—	All 0	R	Reserved These bits are always read as 0. 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 when the value of CSSTS is 0, the PID bits are set to NAK, and no pipe is specified 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, checking PBUSY is not necessary. 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; set these bits to 000 for these pipes.

chanhvinh.com

26.3.36 PIPEn Control Registers (PIPEnCTR) (n = 1 to 9)

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 PIPE9. This register can be set regardless of the pipe selection in PIPESEL.

This register is initialized by a power-on reset. PID[1:0] are initialized by a USB bus reset.

(1) PIPEnCTR (n = 1 to 5)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BSTS	INBUFM	CSCLR	CSSTS	—	AT REPM	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[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	R/W	R/W	R/W*	R/W*	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	BSTS	0	R	Buffer Status Indicates the FIFO buffer status for the pertinent pipe. 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 26.10.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
14	INBUFM	0	R	IN Buffer Monitor Indicates the pertinent FIFO buffer status when the pertinent pipe is in the transmitting direction. 0: There is no data to be transmitted in the buffer memory. 1: There is data to be transmitted in the buffer memory. When the pertinent pipe is in the transmitting direction (DIR = 1), this module sets this bit to 1 when at least one FIFO buffer plane of data has been written. 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 when this module completes transmitting the data from the two FIFO buffer planes before one FIFO buffer plane of data has been written. This bit indicates the same value as the BSTS bit when the pertinent pipe is in the receiving direction (DIR = 0).

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
13	CSCLR	0	R/W*	C-SPLIT Status Clear Bit When the host controller function 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 function is selected, be sure to write 0 to this bit.
12	CSSTS	0	R	CSSTS Status Bit Indicates the C-SPLIT status of the split transaction when the host controller function 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. If the USB device is detached when processing of a C-SPLIT transaction is in progress, the CSSTS bit may remain set to 1. In such a case (DTCH = 1 detected), use the CSCLR bit to clear the CSSTS bit to 0. Indicates the valid value only when the host controller function is selected.
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
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 function 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 (allows toggling of) 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 (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, checking PBUSY is not necessary.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
10	ATREPM	0	R/W	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 function 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 all the information assigned to the pertinent pipe from the FIFO buffer, write 1 and 0 in succession to the ACLRM bit. Table 26.11 (1) lists the information related to the USB 2.0 host/function module that is cleared by setting the ACLRM bit to 1 and then to 0 in succession. Table 26.11 (2) lists the cases in which it is necessary to do this. Modify this bit while CSSTS is 0, PID is NAK, and the pertinent pipe is not specified in 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, checking PBUSY is not necessary.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
8	SQCLR	0	R/W*	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: 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 module always sets this bit to 0. When the host controller function 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, checking PBUSY is not necessary.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
7	SQSET	0	R/W*	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: 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, checking PBUSY is not necessary.
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 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 receiving transfer.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
5	PBUSY	0	R	Pipe Busy This bit indicates whether the relevant pipe is used or not for the transaction. 0: The relevant pipe is not used for the transaction. 1: The relevant pipe is used for the transaction. The USB 2.0 host/function module switches the PBUSY bit from 0 to 1 when a USB transaction using the pertinent pipe starts. It clears the PBUSY bit from 1 to 0 when one transaction completes successfully. Reading this bit after PID has been set to NAK allows checking that modification of the pipe settings is possible. For details, refer to section 26.4.3 (1), Pipe Control Register Switching Procedures.
4 to 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	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. Tables 26.12 and 26.13 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. 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, checking PBUSY is not necessary. After S-SPLIT is issued (CSSTS = 1) for split transaction in the pertinent pipe, the transaction continues until C-SPLIT is completed even when the PID bits are set to NAK. 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 function is selected.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
1, 0	PID[1:0]	00	R/W	• This module sets PID to NAK on detecting a receive error such as a CRC error three consecutive times when the host controller function is selected. • This module sets PID to STALL (11) on receiving the STALL handshake when the host controller function 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: * Only 0 can be read and 1 can be written.

chanhvinh.com

Table 26.10 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: BCLR 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 26.11(1) Information Cleared by this Module by Setting ACLRM = 1

No.	Information Cleared by ACLRM Bit Manipulation
1	All the information 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

Table 26.11(2) Cases That Require Setting ACLRM to 1

No.	Cases in which Clearing the Information is Necessary
1	When it is necessary to clear all the information assigned to the pertinent pipe from the FIFO buffer
2	When the interval count value is to be reset
3	When the BFRE setting is modified
4	When the DBLB setting is modified
5	When the transaction count function is forcibly terminated

chanhvinh.com

Table 26.12 Operation of This Module depending on PID Setting (when Host Controller Function is Selected)

PID	Transfer Type	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 or interrupt	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	Operation does not depend on the setting.	Issues tokens irrespective of the status of the FIFO buffer corresponding to the pertinent pipe.
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 26.13 Operation of This Module depending on PID Setting (when Function Controller Function is Selected)

PID	Transfer Type	Transfer Direction (DIR Bit)	Operation of This Module
00 (NAK)	Bulk or interrupt	Operation does not depend on the setting.	Returns NAK in response to the token from the USB host. For the operation when ATREPM is 1, refer to the description of the ATREPM bit.
	Isochronous	Receiving direction (DIR = 0)	Returns nothing in response to a token from the USB host.
		Transmitting direction (DIR = 1)	Returns a zero-length packet in response to a token from the USB host.
01 (BUF)	Bulk	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. 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 NYET if not ready.
	Interrupt	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 or interrupt	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	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.

chanhvinh.com

PID	Transfer Type	Transfer Direction (DIR Bit)	Operation of This Module
01 (BUF)	Isochronous	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 or interrupt	Operation does not depend on the setting.	Returns STALL in response to the token from the USB host.
	Isochronous	Operation does not depend on the setting	Returns nothing in response to the token from the USB host.

(2) PIPEnCTR (n = 6 to 9)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BSTS	—	CSCLR	CSSTS	—	—	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[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/W*	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	BSTS	0	R	Buffer Status Indicates the FIFO buffer status for the pertinent pipe. 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 26.10.
14	—	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
13	CSCLR	0	R/W*	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 function 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 function 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 function is selected.
11, 10	—	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
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 all the information assigned to the pertinent pipe from the FIFO buffer, write 1 and 0 in succession to the ACLRM bit. Table 26.14 (1) lists the information related to the USB 2.0 host/function module that is cleared by setting the ACLRM bit to 1 and then to 0 in succession. Table 26.14 (2) lists the cases in which it is necessary to do this. Modify this bit while CSSTS is 0 and PID is NAK and before the pipe is 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, checking PBUSY is not necessary.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
8	SQCLR	0	R/W*	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: 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 module always sets this bit to 0. When the host controller function 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, checking PBUSY is not necessary.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
7	SQSET	0	R/W*	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: 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, checking PBUSY is not necessary.
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 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 receiving transfer.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
5	PBUSY	0	R	Pipe Busy This bit indicates whether the relevant pipe is used or not for the transaction. 0: The relevant pipe is not used for the transaction. 1: The relevant pipe is used for the transaction. This module modifies this bit from 0 to 1 upon start of the USB transaction for the pertinent pipe, and clears it from 1 to 0 when one transaction completes successfully. Reading this bit after PID has been set to NAK allows checking that modification of the pipe settings is possible. For details, refer to section 26.4.3 (1), Pipe Control Register Switching Procedures.
4 to 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	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. Tables 26.12 and 26.13 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. 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, checking PBUSY is not necessary. After S-SPLIT is issued (CSSTS = 1) for split transaction in the pertinent pipe, the transaction continues until C-SPLIT is completed even when the PID bits are set to NAK. 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 function is selected.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
1, 0	PID[1:0]	00	R/W	• This module sets PID to NAK on detecting a receive error such as a CRC error three consecutive times when the host controller function is selected. • This module sets PID to STALL (11) on receiving the STALL handshake when the host controller function 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: * Only 0 can be read and 1 can be written.

chanhvinh.com

Table 26.14 (1) Information Cleared by this Module by Setting ACLRM = 1

No.	Information Cleared by ACLRM Bit Manipulation
1	All the information in the FIFO buffer assigned to the pertinent pipe
2	When the host controller function is selected, the interval count value when the pertinent pipe is for isochronous transfer

Table 26.14 (2) Cases That Require Setting ACLRM to 1

No.	Cases in which Clearing the Information is Necessary
1	When it is necessary to clear all the information assigned to the pertinent pipe from the FIFO buffer
2	When the interval count value is to be reset
3	When the BFRE setting is modified
4	When the transaction count function is forcibly terminated

26.3.37 PIPEn Transaction Counter Enable Registers (PIPEnTRE) (n = 1 to 5)

PIPEnTRE is a register that enables or disables the transaction counter corresponding to PIPE1 to PIPE5, and clears the transaction counter.

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
	—	—	—	—	—	—	TRENB	TRCLR	—	—	—	—	—	—	—	—
Initial value:	0	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*1	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	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
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*[1]	Transaction Counter Clear Clears the current 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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Notes: 1. Only 0 can be read and 1 can be written.
2. 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, checking PBUSY is not necessary.

26.3.38 PIPEn Transaction Counter Registers (PIPEnTRN) (n = 1 to 5)

PIPEnTRN is a transaction counter corresponding to PIPE1 to PIPE5.

This register is initialized by a power-on reset, but retains the set value by a USB bus reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TRNCNT[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	TRNCNT[15:0]	All 0	R/W	Transaction Counter When written to: Specifies the number of transactions to be transferred through DMA. When read from: Indicates the specified number of transactions if TRENB is 0. Indicates the number of currently counted transaction if TRENB is 1.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	TRNCNT[15:0]	All 0	R/W	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 set value ≠ 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 set value = 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. The TRCLR bit has been set to 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, checking PBUSY is not necessary. To modify the value of these bits, set TRNCNT to 1 before setting TRENB to 1.

chanhvinh.com

26.3.39 Device Address n Configuration Registers (DEVADDn) (n = 0 to A)

DEVADDn is a register that specifies the address and port number of the hub to which the communication target peripheral device is connected and that also specifies the transfer speed of the peripheral device for PIPE0 to PIPEA.

When the host controller function is selected, this register should be set before starting communication using each pipe.

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 condition 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 selected pipe or the selected pipe is the DCP with SUREQ being 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
	—	UPPHUB[3:0]				HUBPORT[2:0]			USBSPD[1:0]		—	—	—	—	—	—
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	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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 LSI. 0001 to 1010: USB address of the hub 1011 to 1111: Setting prohibited When the host controller function is selected, this module refers to the setting of these bits to generate packets for split transactions. When the function controller function is selected, set these bits to 0000.
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 LSI. 001 to 111: Port number of the hub When the host controller function is selected, this module refers to the setting of these bits to generate packets for split transactions. When the function controller function is selected, set these bits to 000.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 is not used. 01: Low speed 10: Full speed 11: High speed When the host controller function is selected, this module refers to the setting of these bits to generate packets. When the function controller function is selected, set these bits to 00.
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

26.3.40 USB AC Characteristics Switching Register 1 (USBACSWR1)

USBACSWR1 makes the necessary settings for the USB transceiver built in 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
	—	—	—	—	—	—	UACS25	—	—	—	—	—	—	—	—	—
Initial value:	0	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	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	UACS25	0	R/W	USB AC Characteristics Switching 25 This bit adjusts the eye-pattern characteristics in high speed.*
8 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * This bit should always be 1 when using this module.
For details, refer to section 26.5.1, Procedure for Setting the USB Transceiver.

chanhvinh.com

26.4 Operation

26.4.1 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 26.15 lists the types of controller resets. For the initialized states of the registers following the reset operations, see section 26.3, Register Description.

Table 26.15 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 function is selected

(2) Controller Function Selection

This module can select the host controller function or function controller function using the DCFM bit in SYSCFG. 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) Enabling High-Speed Operation

This module can select a USB communication speed (communication bit rate). When the host controller function is selected, the high-speed operation or full-speed/low-speed operation can be set. When the function controller function is selected, either the high-speed operation or full-speed operation can be selected. In order to enable the high-speed operation for this module, the HSE bit in SYSCFG should be set to 1. If high-speed mode has been enabled, this module executes the reset handshake protocol, and the USB communication speed is set automatically. The results of the reset handshake can be confirmed using the RHST bit in DVSTCTR.

If high-speed operation has been disabled, this module operates at full-speed or low-speed. If the function controller function is also selected, this module operates at full-speed.

Changing the HSE bit should be done between the ATTCH interrupt detection and bus reset execution when the host controller function is selected, or with the D+ line pull-up disabled (DPRPU = 0) when the host controller function is selected.

chanhvinh.com

(4) USB Data Bus Resistor Control

Figure 26.1 shows a diagram of the connections between this module and the USB connectors.

This module incorporates a pull-up resistor for the D+ signal and a pull-down resistor for the D+ and D- signals. These signals can be pulled up or down using the DPRPU and DRPD bits in SYSCFG.

This module controls the terminal 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 by means of reset handshake, suspended state and resume detection.

When the function controller function is selected and the DPRPU bit in SYSCFG is cleared to 0 during communication with the host controller, the pull-up resistor (or the terminal resistor) of the USB data line is disabled, making it possible to notify the USB host of the device disconnection.

chanhvinh.com

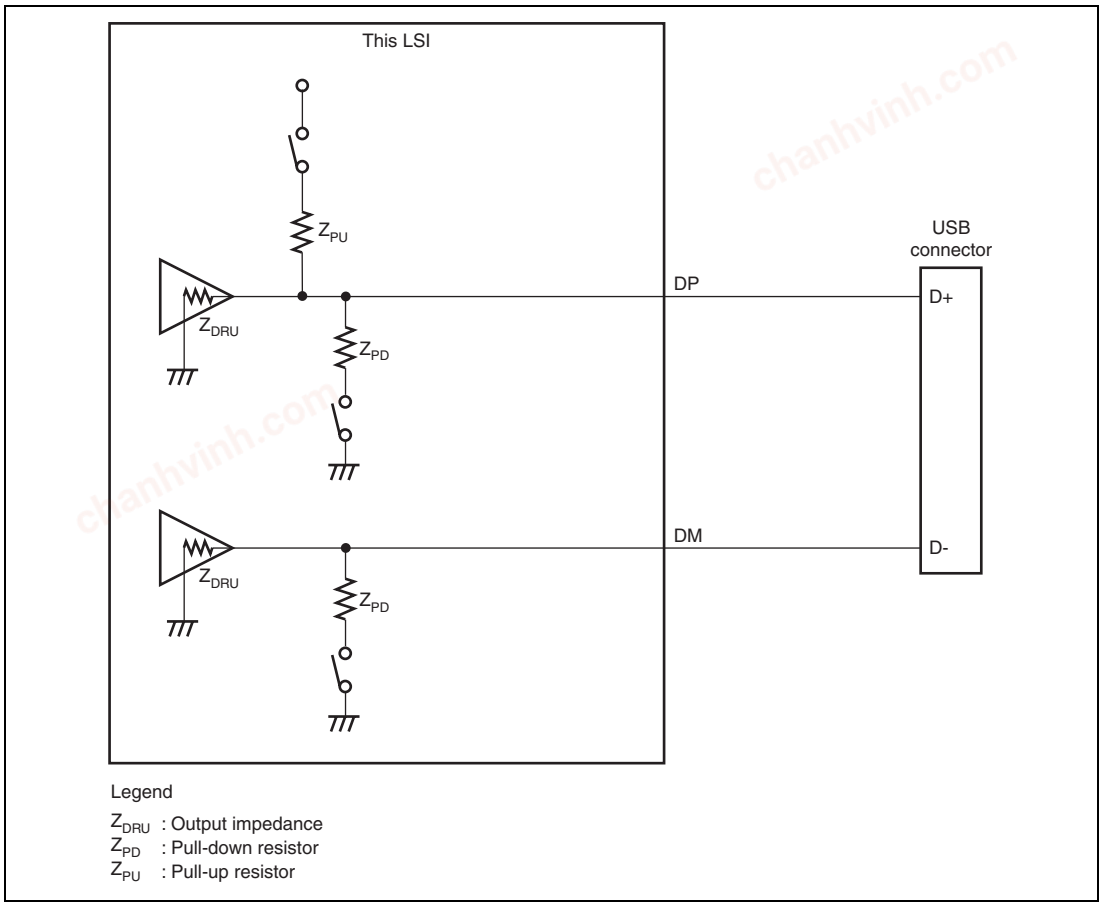


Figure 26.1 UBS Connector Connection

chanhvinh.com

(5) Register Access Wait Control

There is a restriction on the cycle time for accessing the registers of this module except for SYSSTS as given below.

Wait-related restriction: The cycle time for successive accesses to the USB 2.0 host/function module must be a duration of at least four USB clock (48 MHz) cycles (83.33 ns). To fulfill the above restriction, a register access wait control is necessary using the BWAIT[3:0] bits in BUSWAIT. The initial value is the maximum value (access cycles = 17 clock cycles). The optimum value should be found and set.

Setting example 1: When successively accessing the registers of this module

Bus clock frequency: 72 MHz

Calculation: (2 cycles (access cycles for the registers of this module) + 1 cycle (interval between successive accesses) + BWAIT) × 1/72 MHz ≥ 83.33 ns

BWAIT = 3

Setting example 2: When sending data from the on-chip high-speed RAM to the FIFO port register through DMA transfer

Bus clock frequency: 72 MHz

Calculation: (2 cycles (access cycles for the registers of this module) + 2 cycles (access cycles for the on-chip high-speed RAM) + BWAIT) × 1/72 MHz ≥ 83.33 ns

BWAIT = 2

chanhvinh.com

26.4.2 Interrupt Functions

Table 26.16 lists the interrupt generation conditions for this module.

When an interrupt generation condition is satisfied and the interrupt output is enabled using the corresponding interrupt enable register, this module issues a USB interrupt request to the interrupt controller.

Table 26.16 Interrupt Generation Conditions

Bit	Interrupt Name	Cause of Interrupt	Function 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 function is selected: • When an SOF packet with a different frame number has been transmitted When the function controller function is selected: • When an SOF packet with a different frame number is received	Host, function	—
DVST	Device state transition interrupt	When a device state transition is detected • A USB bus reset detected • The suspend state detected • SET_ADDRESS request received • SET_CONFIGURATION request received	Function	DVSQ

chanhvinh.com

Bit	Interrupt Name	Cause of Interrupt	Function That Generates the Interrupt	Related Status
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 • When an excessive maximum packet size error has been detected	Host, Function	BEMPSTS. PIPEBEMP

chanhvinh.com

Bit	Interrupt Name	Cause of Interrupt	Function That Generates the Interrupt	Related Status
NRDY	Buffer not ready interrupt	When the host controller function 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 function is selected: • When a token is received while the PID bits are set to BUF and transmission is not enabled for the buffer memory • 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	NRDYSTS.PIPENRDY
BRDY	Buffer ready interrupt	When the buffer is ready (reading or writing is enabled)	Host, function	BRDYSYS PIPEBRDY
BCHG	Bus change interrupt	When a change of USB bus state is detected	Host	—

chanhvinh.com

Bit	Interrupt Name	Cause of Interrupt	Function That Generates the Interrupt	Related Status
DTCH	Device disconnection	When disconnection of a peripheral device 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	—
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 26.2 shows a diagram relating to interrupts of this module.

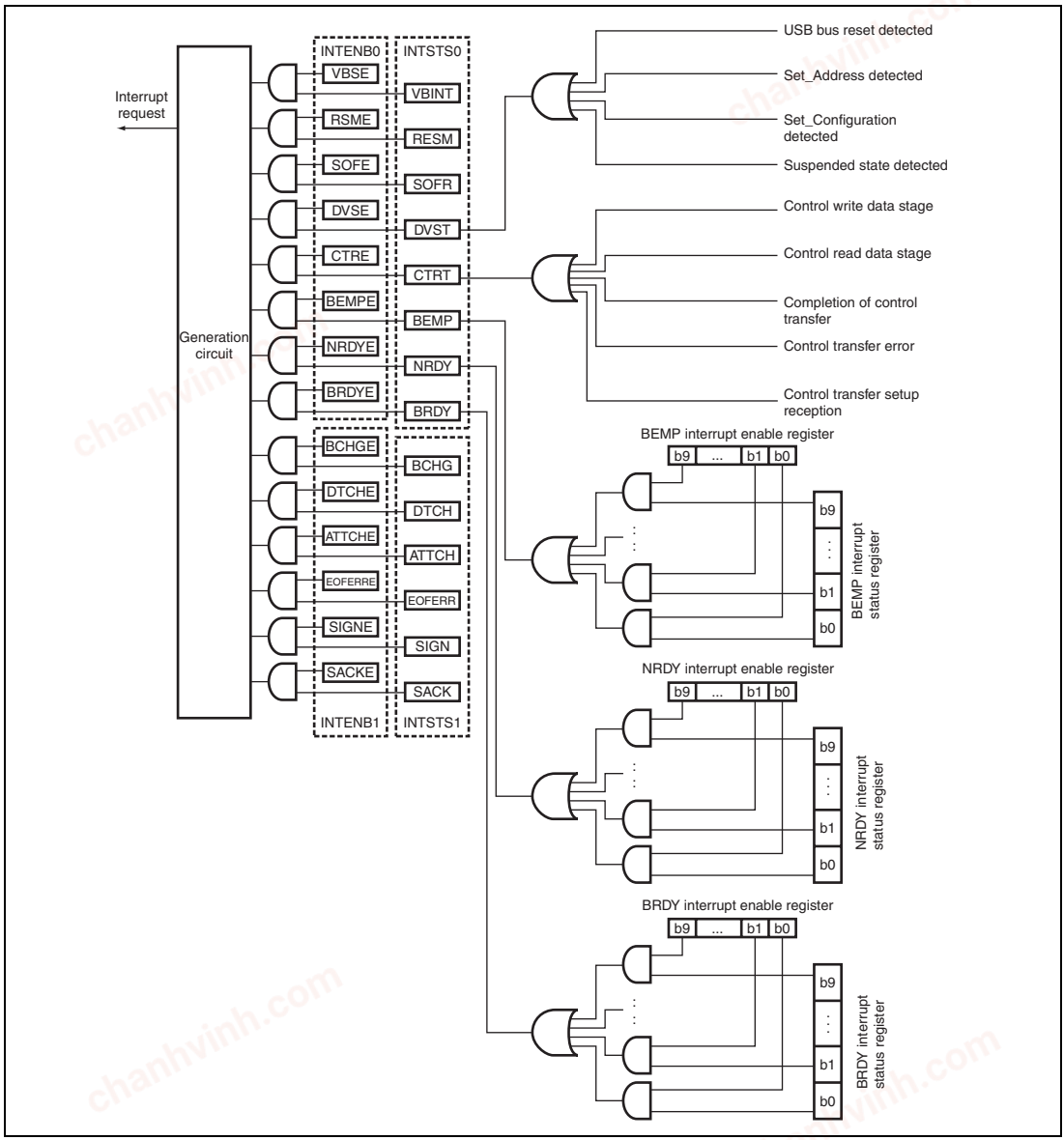


Figure 26.2 Items Relating to Interrupts

chanhvinh.com

(1) BRDY Interrupt

The BRDY interrupt is generated when either of the host controller function or function controller function is selected. The following shows the conditions under which this module sets 1 to a corresponding bit in BRDYSTS. Under this condition, this module generates BRDY interrupt, if the PIPEBRDYE bit in BRDYENB that corresponds to the pipe to 1 and the BRDYE bit in INTENB0 have been set to 1.

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 the 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.

(i) 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).

chanhvinh.com

(ii) 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 function 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 the BRDYM bit is 0 and the 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 are 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 while the FIFO buffer is empty, the USB 2.0 host/function module determines that all the data for a single transfer has been read at the point at which the FRDY bit is set to 1 and the DTLN bit cleared to 0 in the FIFO port control register. 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.

chanhvinh.com

(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.

(i) For the pipe in the transmitting direction:

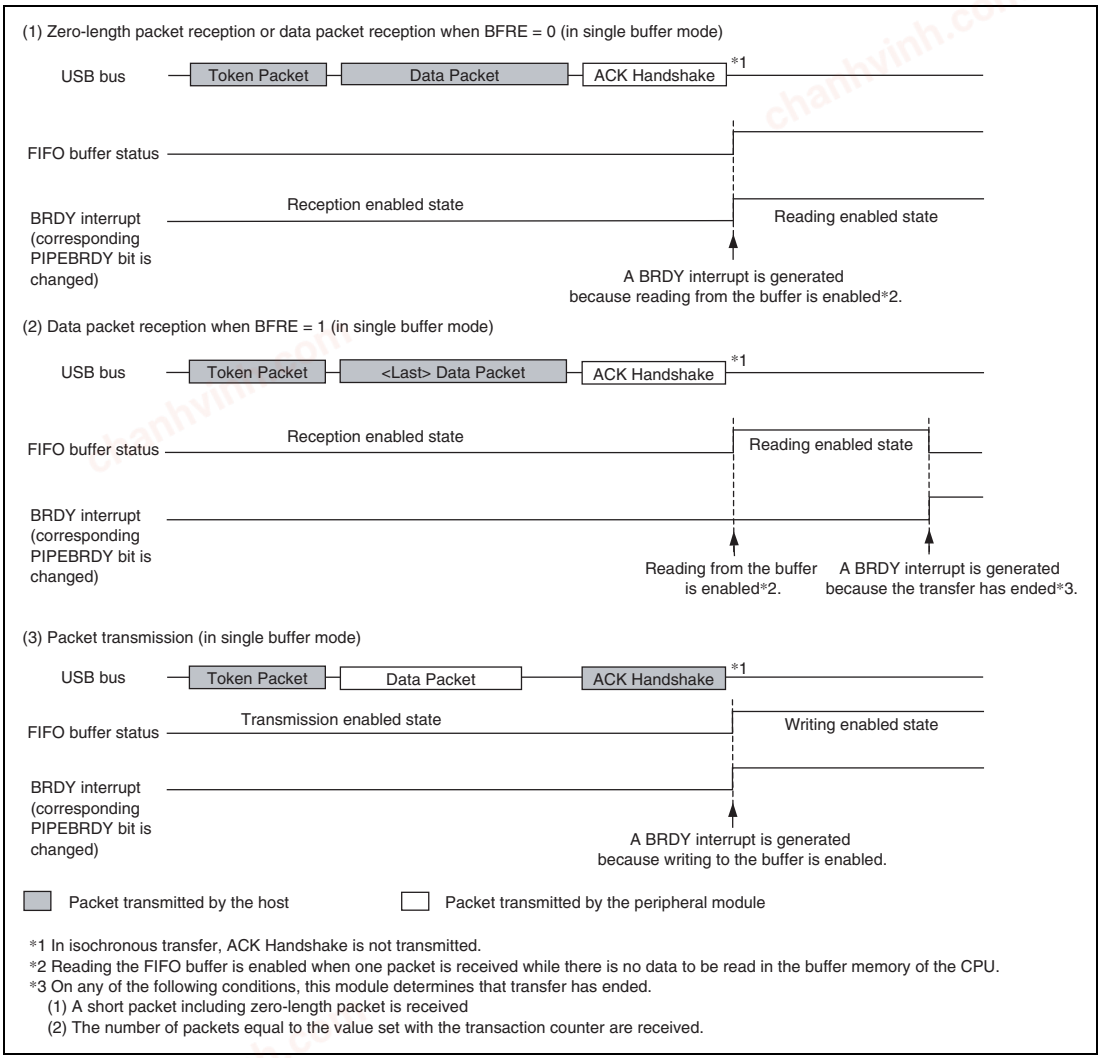
The BRDY interrupt status bits are set to 1 when the FIFO buffer is write-enabled and are set to 0 when write-disabled.
However, the BRDY interrupt is not generated if the DCP in the transmitting direction is write-enabled.

(ii) 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 is continuously generated 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

Figure 26.3 shows the timing at which the BRDY interrupt is generated.



*1 In isochronous transfer, ACK Handshake is not transmitted.
*2 Reading the FIFO buffer is enabled when one packet is received while there is no data to be read in the buffer memory of the CPU.
*3 On any of the following conditions, this module determines that transfer has ended.
(1) A short packet including zero-length packet is received
(2) The number of packets equal to the value set with the transaction counter are received.

Figure 26.3 Timing at which a BRDY Interrupt is Generated

chanhvinh.com

(2) 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 USB interrupt to be generated.

The following describes the conditions on which this module generates the internal NRDY interrupt request for a given pipe.

However, the internal NRDY interrupt request is not generated during setup transaction execution when the host controller function is selected. During setup transactions when the host controller function 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 function is selected.

(a) When the host controller function is selected and when the connection is used in which no split transactions occur

(i) 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).

chanhvinh.com

(ii) 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 to NAK is not modified.)

— 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.

chanhvinh.com

(b) When the host controller function is selected and when the connection is used in which split transactions occur

(i) 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.

— 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).

(ii) 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.

— During bulk-pipe 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.

chanhvinh.com

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 bit to 1 and CRCE bit to 1, and clears the CSSTS bit to 0 (does not modify the setting of the PID bits).

— 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 to 1 and CRCE bit to 1, and clears the CSSTS bit to 0 (does not modify the setting of the PID bits).

(c) When the function controller function is selected

(i) For the pipe in the transmitting direction:

— On receiving an IN token when there is no data to be transmitted in the FIFO buffer.

In this case, this module generates a 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.

(ii) For the pipe in the receiving direction:

— On receiving an OUT token when 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 a NRDY interrupt request, 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 a NRDY interrupt request when a NAK handshake is

chanhvinh.com

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 there is no space available in the FIFO buffer.

In this case, this module generates a NRDY interrupt request at the reception of the PING token, setting the PIPENRDY bit to 1.

— For the pipe for isochronous transfers, when a token is not received normally within an interval frame.

In this case, this module generates a NRDY interrupt request, setting the PIPENRDY bit to 1.

chanhvinh.com

Figure 26.4 shows the timing at which an NRDY interrupt is generated when the function controller function is selected.

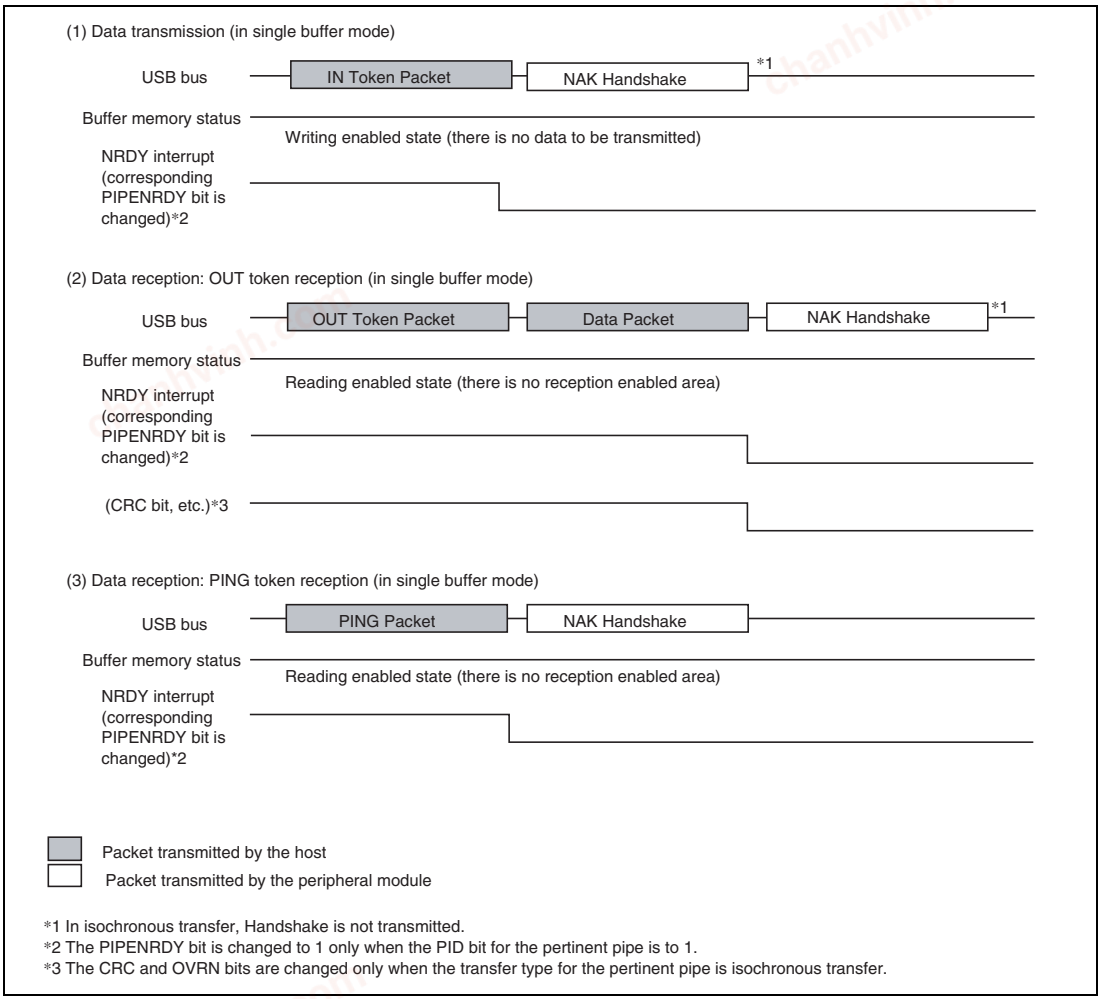


Figure 26.4 Timing at which NRDY Interrupt is Generated when Function Controller Function is Selected

chanhvinh.com

(3) BEMP Interrupt

On generating 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.

(a) 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 of the CPU has already been 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.

(b) 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

Figure 26.5 shows the timing at which a BEMP interrupt is generated when the function controller function has been selected.

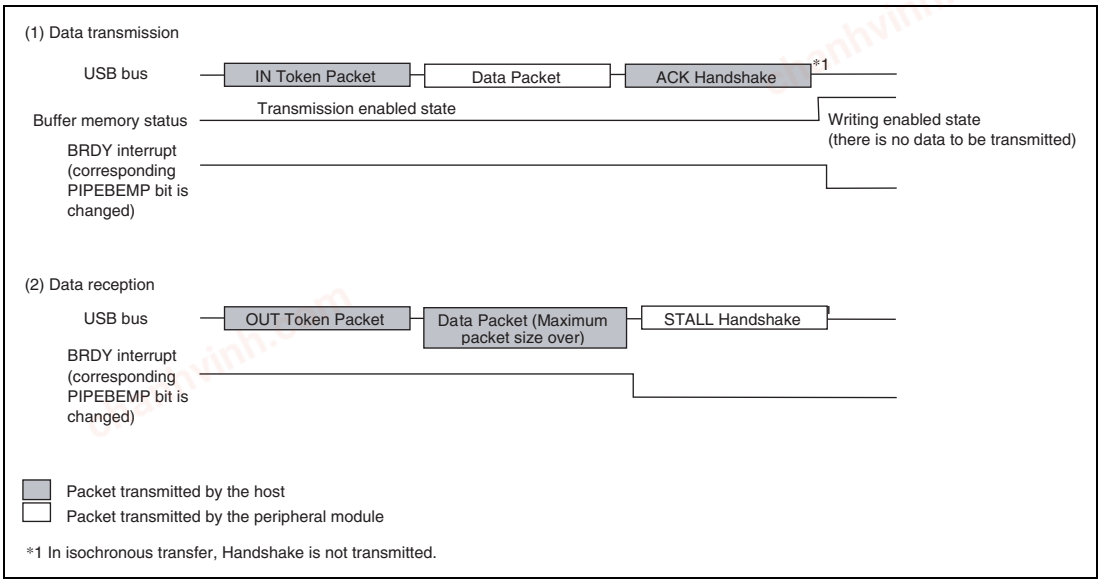


Figure 26.5 Timing at which BEMP Interrupt is Generated when Function Controller Function is Selected

(4) Device State Transition Interrupt

Figure 26.6 shows a diagram of this module device state transitions. This module controls 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 using INTENB0. The device state that made a transition can be confirmed using the DVSQ bit in INTSTS0.

To make 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 controlled only when the function controller function is selected. Also, the device state transition interrupts can be generated only when the function controller function is selected.

chanhvinh.com

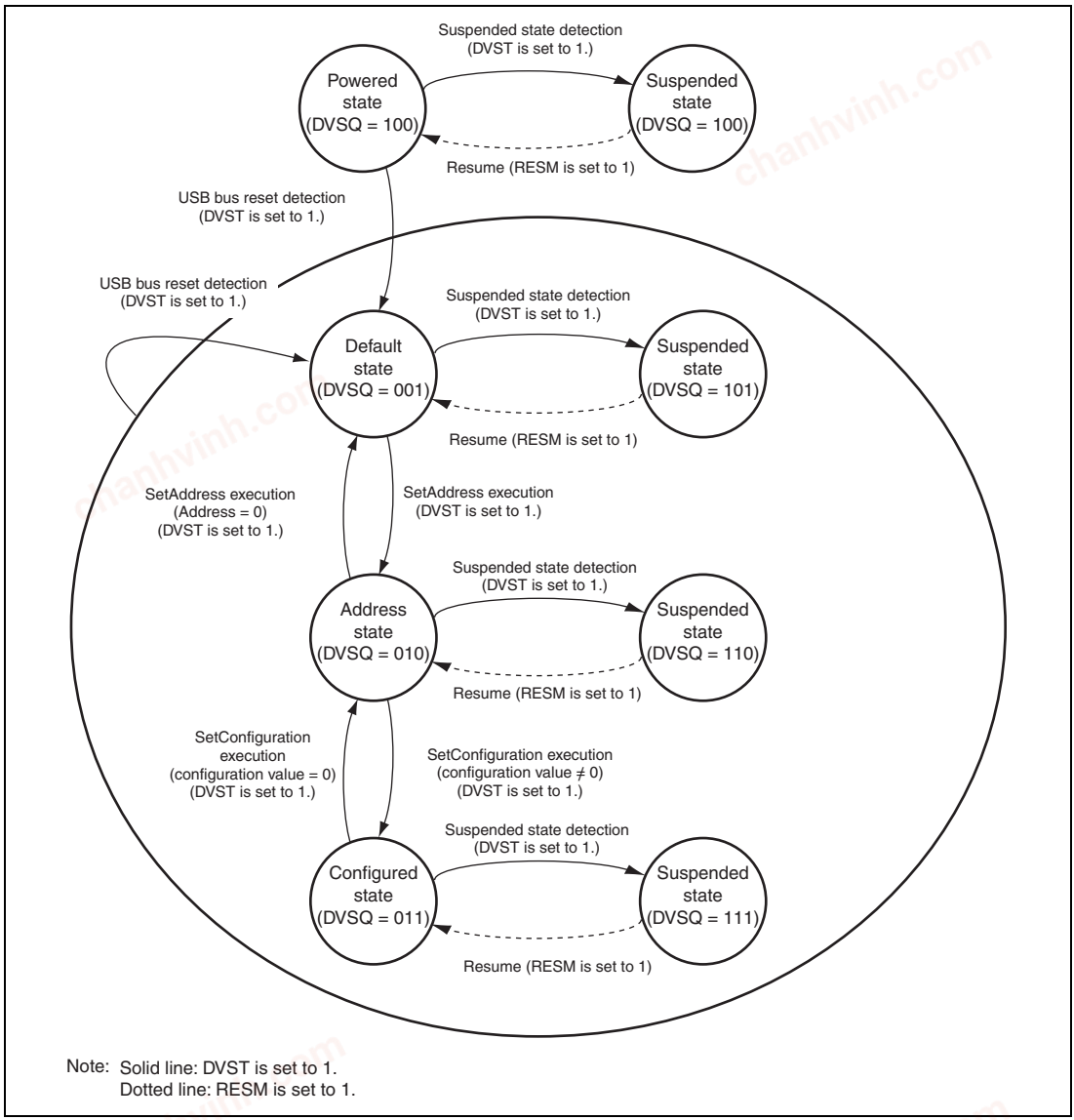


Figure 26.6 Device State Transitions

chanhvinh.com

(5) Control Transfer Stage Transition Interrupt (Function Controller Function)

Figure 26.7 shows a diagram of how this module handles the control transfer stage transition. This module controls 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 transfer stage that made a transition can be confirmed using the CTSQ bit in INTSTS0.

The control transfer stage transition interrupts are generated only when the function controller function is selected.

The control transfer sequence errors are described below. If an error occurs, the PID bit in DCPCTR is set to B'1x (STALL).

(a) 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

(b) 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

(c) During no-data control transfers

- At the status stage, an OUT or PING token is received

At the control write transfer 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, 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 held, the CTRT interrupt that ends the setup stage will not be generated even if a new USB request is received. (The USB 2.0 host/function module retains the setting of the CTSQ

chanhvinh.com

bits when the setup stage ends, and it generates a CTRT interrupt after the interrupt status is cleared.)

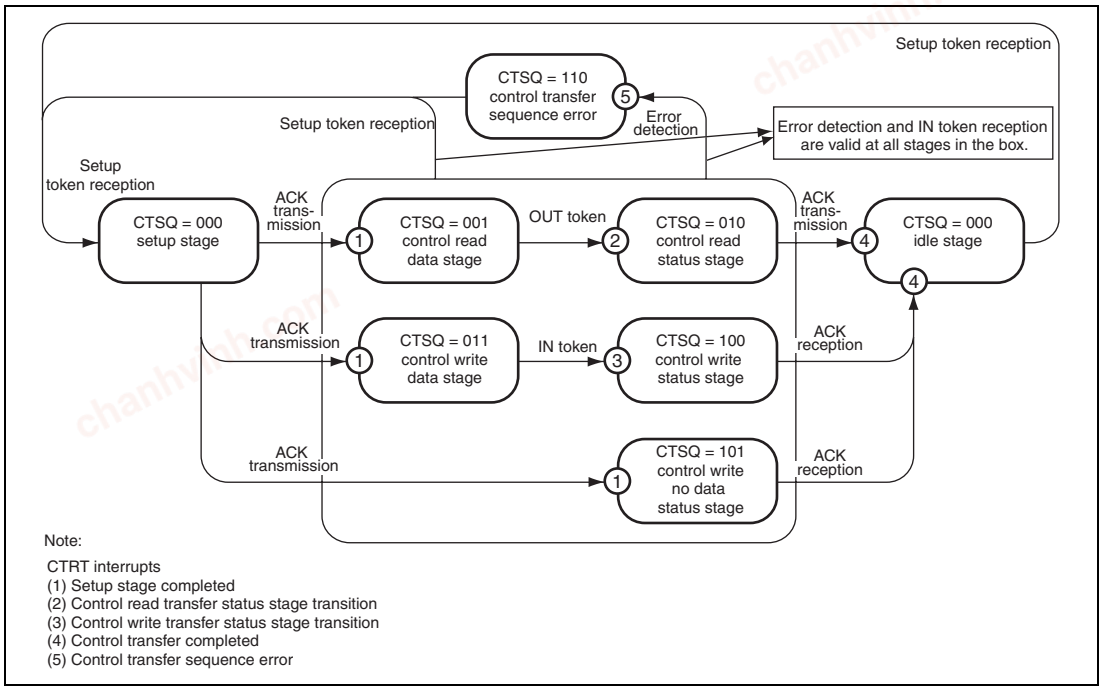


Figure 26.7 Control Transfer Stage Transitions

(6) Frame Update Interrupt

Figure 26.8 shows an example of the SOFR interrupt output timing of this module. With the host controller function selected, an interrupt is generated at the timing at which the frame number is updated. With the function controller function selected, the SOFR interrupt is generated when the frame number is updated.

When the function controller function is selected, this module updates the frame number and generates an SOFR interrupt if it detects a new SOF packet during full-speed operation. During high-speed operation, however, this module does not update the frame number, or generates no SOFR interrupt until the module enters the µSOF locked state. Also, the SOF interpolation function is not activated. The µSOF lock state is the state in which µSOF packets with different frame numbers are received twice continuously without error occurrence.

The conditions under which the µSOF lock monitoring begins and stops are as follows.

chanhvinh.com

1. Conditions under which µSOF lock monitoring begins
   USBE = 1
2. Conditions under which µSOF lock monitoring stops
   USBE = 0, a USB bus reset is received, or suspended state is detected.

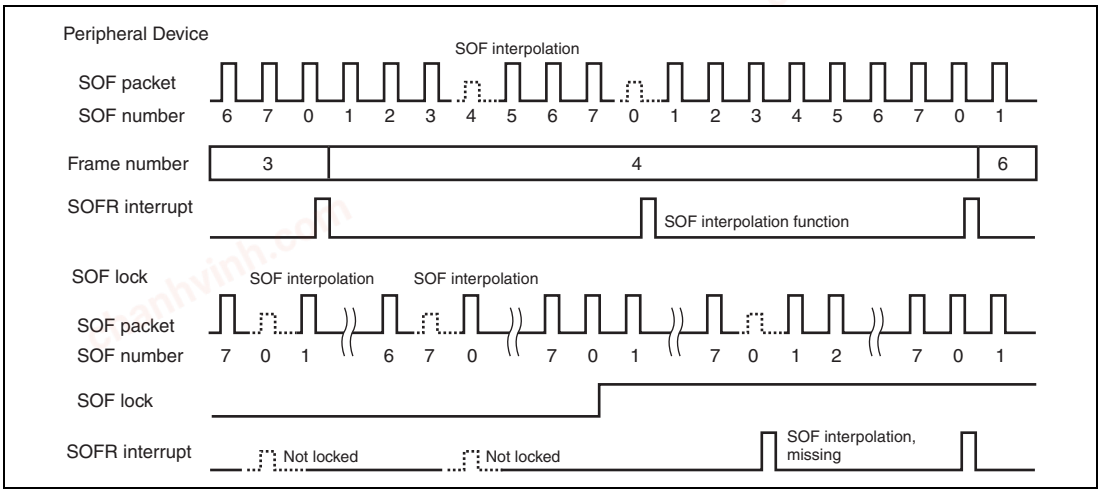


Figure 26.8 Example of SOFR Interrupt Output Timing

(7) VBUS Interrupt

If there has been a change in the VBUS pin, the VBUS interrupt is generated. The level of the VBUS pin can be checked with the VBSTS bit in INTSTS0. Whether the host controller is connected or disconnected can be confirmed using the VBUS interrupt. However, if the system is activated with the host controller connected, the first VBUS interrupt is not generated because there is no change in the VBUS pin.

(8) Resume Interrupt

With the function controller function selected, the resume interrupt is generated when the USB bus state changes (from J-state to K-state, or from J-state to SE0) while the device state is the suspended state.. Recovery from the suspended state is detected by means of the resume interrupt.

With the host controller function selected, the resume interrupt is not generated; use the BCHG interrupt to detect the change of the USB bus state.

chanhvinh.com

(9) BCHG Interrupt

The BCHG interrupt is generated when the USB bus state has changed. The BCHG interrupt can be used to detect whether or not the peripheral device is connected when the host controller function has been selected and can also be used to detect a remote wakeup. The BCHG interrupt is generated regardless of whether the host controller function or function controller function has been selected.

(10) DTCH Interrupt

The DTCH interrupt is generated if disconnection of the USB bus is detected when the host controller function has been selected. 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). 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).

(a) Modifies the UACT bit to 0.
(b) Puts the port into the idle state.

(11) SACK Interrupt

The SACK interrupt is generated when an ACK response for the transmitted setup packet has been received from the peripheral device with the host controller function selected. The SACK interrupt can be used to confirm that the setup transaction has been completed successfully.

(12) SIGN Interrupt

The SIGN interrupt is generated when an ACK response for the transmitted setup packet has not been correctly received from the peripheral device three consecutive times with the host controller function selected. The SIGN interrupt can be used to detect no ACK response transmitted from the peripheral device or corruption of an ACK packet.

chanhvinh.com

(13) ATTCH Interrupt

The ATTCH interrupt is generated when J-state or K-state of the full-speed or low-speed level signal is detected on the USB port for 2.5 µs in host controller mode. To be more specific, the ATTCH interrupt is detected on any of the following conditions.

(a) When K-state, SE0, or SE1 changes to J-state, and J-state continues 2.5 µs.
(b) When J-state, SE0, or SE1 changes to K-state, and K-state continues 2.5 µs.

(14) EOFERR Interrupt

The EOFERR interrupt is generated when it is detected that communication is not completed at the EOF2 timing prescribed by USB Specification 2.0.

After detecting the EOFERR interrupt, this module controls hardware as described below (irrespective of the set value of the corresponding interrupt enable bit). Terminate all the pipes in which communications are currently carried out for the pertinent port and perform re-enumeration of the pertinent port.

(a) Modifies the UACT bit for the port in which an EOFERR interrupt has been detected to 0.
(b) Puts the port in which an EOFERR interrupt has been generated into the idle state.

26.4.3 Pipe Control

Table 26.17 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 ten pipes that are used for data transfer.

Settings should be entered for each of the pipes in conjunction with the specifications of the system.

chanhvinh.com

Table 26.17 Pipe Setting Items

Register Name	Bit Name	Setting Contents	Remarks
DCPCFG PIPECFG	TYPE	Specifies the transfer type	PIPE1 to PIPE9: Can be set
	BFRE	Selects the BRDY interrupt mode	PIPE1 to PIPE5: Can be set
	DBLB	Selects a double buffer	PIPE1 to PIPE5: Can be set
	CNTMD	Selects continuous transfer or non-continuous transfer	PIPE1 and PIPE2: Can be set (only when bulk transfer has been selected). PIPE3 to PIPE5: Can be set
	DIR	Selects transfer direction	IN or OUT can be set
	EPNUM	Endpoint number	PIPE1 to PIPE9: 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	PIPE1 and PIPE2: Can be set (only when bulk transfer has been selected) PIPE3 to PIPE5: Can be set
PIPEBUF	BUFSIZE	Buffer memory size	DCP: Cannot be set (fixed at 256 bytes) PIPE1 to PIPE5: Can be set (a maximum of 2 Kbytes can be specified in 64-byte units) PIPE6 to PIPE9: 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: Can be set (can be specified in areas H'8 to H'7F) PIPE6 to PIPE9: Cannot be set (areas fixed at H'4 to H'7)

chanhvinh.com

Register Name	Bit Name	Setting Contents	Remarks
DCPMAXP PIPEMAXP	DEVSEL	Selects a device	Referenced only when the host controller function 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 PIPE5: Cannot be set PIPE6 to PIPE9: Can be set (only when the host controller function has been selected)
	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 function has been selected)
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 PIPE3 to PIPE5.
	SUREQ	SETUP request	Can be set only for the DCP. Can be controlled only when the host controller function has been selected.
	SUREQCLR	SUREQ clear	Can be set only for the DCP. Can be controlled only when the host controller function has been selected.
	CSCLR	CSSTS clear	Can be controlled only when the host controller function has been selected.
	CSSTS	SPLIT status indication	Can be referenced only when the host controller function has been selected.
	ATREPM	Auto response mode	PIPE1 to PIPE5: Can be set Can be controlled only when the function controller function has been selected.

chanhvinh.com

Register Name	Bit Name	Setting Contents	Remarks
DCPCTR	ACLRM	Auto buffer clear	PIPE1 to PIPE9: Can be set
PIPEnCTR	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 status	
	PID	Response PID	See section 26.4.3 (6), Response PID
PIPEnTRE	TRENB	Transaction counter enable	PIPE1 to PIPE5: Can be set
	TRCLR	Current transaction counter clear	PIPE1 to PIPE5: Can be set
PIPEnTRN	TRNCNT	Transaction counter	PIPE1 to PIPE5: Can be set

(1) Pipe Control Register Switching Procedures

The following bits in the pipe control registers can be modified only when USB communication is disabled (PID = NAK):

Registers that Should Not be Set in the USB Communication Enabled (PID = BUF) State

- Bits in DCPMAXP
- The SQCLR, SQSET, and PINGE bits in DCPCTR
- Bits in PIPECFG, PIPEBUF, PIPEMAXP and PIPEPERI
- The ATREPM, ACLRM, SQCLR and SQSET bits in PIPEnCTR
- Bits in PIPEnTRE and PIPEnTRN
- Bits in DEVADDn

Note: In addition to the above, observe the setting procedures described in the register descriptions regarding the settings of the CSCLR bit and DEVADDn register.

In order to modify the above bits from the USB communication enabled (PID = BUF) state, follow the procedure shown below:

1. Generate a bit modification request with the pipe control register.
2. Modify the PID corresponding to the pipe to NAK.

chanhvinh.com

3. Wait until the corresponding CSSTS bit is cleared to 0 (only when the host controller function has been selected).
4. Wait until the corresponding PBUSY bit is cleared to 0.
5. Modify the bits in the pipe control register.

The following bits in the pipe control registers can be modified only when the pertinent information has not been set by the CURPIPE bits in CFIFOSEL, D0FIFOSEL and D1FIFOSEL.

Registers that Should Not be Set When CURPIPE in FIFO-PORT is set.

- Bits in DCPCFG and DCPMAXP
- Bits in PIPECFG, PIPEBUF, PIPEMAXP and PIPEPERI
- 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.

(2) Transfer Types

The TYPE bit in PIPEPCFG is used to specify the transfer type for each pipe. The transfer types that can be set for the pipes are as follows.

1. DCP: No setting is necessary (fixed at control transfer).
2. PIPE1 and PIPE2: These should be set to bulk transfer or isochronous transfer.
3. PIPE3 to PIPE5: These should be set to bulk transfer.
4. PIPE6 to PIPE9: These should be set to interrupt transfer.

(3) Endpoint Number

The EPNUM bit in PIPEPCFG is used to set the endpoint number for each pipe. The DCP is fixed at endpoint 0. The other pipes can be set from endpoint 1 to endpoint 15.

1. DCP: No setting is necessary (fixed at end point 0).
2. PIPE1 to PIPE9: The endpoint numbers from 1 to 15 should be selected and set.
   These should be set so that the combination of the DIR bit and EPNUM bit is unique.

chanhvinh.com

(4) Maximum Packet Size Setting

The MXPS bit in DCPMAXP and PIPEMAXP is used to specify the maximum packet size for each pipe. DCP and PIPE1 to PIPE5 can be set to any of the maximum pipe sizes defined by the USB specification. For PIPE6 to PIPE9, 64 bytes are the upper limit of the maximum packet size. The maximum packet size should be set before beginning the transfer (PID = BUF).

1. DCP: 64 should be set when using high-speed operation.
2. DCP: Select and set 8, 16, 32, or 64 when using full-speed operation.
3. PIPE1 to PIPE5: 512 should be set when using high-speed bulk transfer.
4. PIPE1 to PIPE5: Select and set 8, 16, 32, or 64 when using full-speed bulk transfer.
5. PIPE1 and PIPE2: Set a value between 1 and 1024 when using high-speed isochronous transfer.
6. PIPE1 and PIPE2: Set a value between 1 and 1023 when using full-speed isochronous transfer.
7. PIPE6 to PIPE9: Set a value between 1 and 64.

The high bandwidth transfers used with interrupt transfers and isochronous transfers are not supported.

(5) Transaction Counter (For PIPE1 to PIPE5 in Reading Direction)

When the specified number of transactions have been completed in the data packet receiving direction, this module recognizes that the transfer has ended. The transaction counter function is available when the pipes assigned to the D0FIFO/D1FIFO port have been set in the direction of reading data from the buffer memory. Two transaction counters are provided: one is the TRNCNT register that specifies the number of transactions to be executed and the other is the current counter that internally counts the number of executed transactions. When the current counter value matches the number of the transactions specified in TRNCNT, reading the buffer memory is enabled. The current counter of the transaction counter function is initialized by the TRCLR bit, so that the transactions can be counted again starting from the beginning. The information read from TRNCNT differs depending on the setting of the TRENB bit.

- TRENB = 0: The specified transaction counter value can be read.
- TRENB = 1: The current counter value indicating the internally counted number of executed transactions can be read.

When operating the TRCLR bit, the following should be noted.

- If the transactions are being counted and PID = BUF, the current counter cannot be cleared.
- If there is any data left in the buffer, the current counter cannot be cleared.

chanhvinh.com

(6) 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:

(a) Response PID settings when the host controller function 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.

Setup transactions for the DCP are set with the SUREQ bit.

(b) Response PID settings when the function controller function 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.

For setup transactions, an ACK response is always returned, regardless of the PID setting, and the USB request is stored in the register.

This module may carry out writing to the PID bits, depending on the results of the transaction.

(a) When the host controller function has been selected and the response PID is set by hardware

- NAK setting: In the following cases, PID = NAK is set and issuing of tokens is automatically stopped:
  - When, during a transfer other than isochronous transfer, three receive errors such as no response, bit stuffing error, or CRC error are returned in succession after token transmission.
  - When, during isochronous transfer, three receive errors such bit stuffing error or CRC error are returned in succession after token transmission.

chanhvinh.com

— When a short packet is received in the data stage of control read transfer while the SHTNAK bit in the DCPCFG register is set to 1.
— If a short packet is received when the SHTNAK bit in PIPECFG has been set to 1 for bulk transfer.
— If the transaction counter ended when the SHTNAK bit has been set to 1 for 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.

(b) When the function controller function has been selected and the response PID is set by hardware

- NAK setting: In the following cases, PID = NAK is set and NAK is always returned in response to transactions:
  — When the SETUP token is received normally (DCP only).
  — If the transaction counter ended or a short packet is received when the SHTNAK bit in PIPECFG has been set to 1 for bulk transfer.
- BUF setting: There is no BUF writing by this module.
- STALL setting: In the following cases, PID = STALL is set and STALL is always returned in response to transactions:
  — When the size of the receive data packet exceeds the maximum packet size.
  — When a control transfer sequence error has been detected (DCP only).

chanhvinh.com

(7) 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 was transmitted can be confirmed with the SQMON bit in DCPCTR and 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 bit in DCPCTR and the SQSET bit in PIPEnCTR can be used to change the data PID sequence bit.

When the function controller function has been selected and control transfer is used, this module automatically sets the sequence bit when a stage transition is made. The bit is set to DATA1 when the setup stage ends. In the status stage, DATA1 is returned without referencing the sequence bit. Therefore, settings are not required. However, when the host controller function 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 function or function controller function is selected.

With pipes for which isochronous transfer has been set, sequence bit operation cannot be carried out using the SQSET bit.

(8) Response PID = NAK Function

This module has a function that disables pipe operation (PID response = NAK) at the timing at which the final data packet of a transaction is received (this module automatically distinguishes this based on reception of a short packet or the transaction counter) by setting the SHTNAK bit in PIPECFG to 1.

When a double buffer is being used for the buffer memory, using this function enables reception of data packets in transfer units. If pipe operation has disabled, the pipe has to be set to the enabled state again (PID response = BUF).

This function can be used only when bulk transfers are used.

chanhvinh.com

(9) Auto Transfer MODE

With the pipes for bulk transfer (PIPE1 to PIPE5), when the ATREPM bit in PIPEnCTR is set to 1, a transition is made to auto response mode. During an OUT transfer (DIR = 0), OUT-NAK mode is entered, and during an IN transfer (DIR = 1), null auto response mode is entered.

(a) OUT-NAK Mode

With the pipes for bulk OUT transfer, NAK is returned in response to an OUT or PING token and an NRDY interrupt is output when the ATREPM bit is set to 1. To make a transition from normal mode to OUT-NAK mode, OUT-NAK mode should be specified in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). After pipe operation has been enabled, OUT-NAK mode becomes valid. However, if an OUT token is received immediately before pipe operation is disabled, the token data is normally received, and an ACK is retuned to the host.

To make a transition from OUT-NAK mode to normal mode, OUT-NAK mode should be canceled in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). In normal mode, reception of OUT data is enabled and an ACK is returned in response to a PING token if the buffer is ready to receive data.

(b) Null Auto Response Mode

With the pipes for bulk IN transfer, zero-length packets are continuously transmitted when the ATREPM bit is set to 1.

To make a transition from normal mode to null auto response mode, null auto response mode should be set in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). After pipe operation has been enabled, null auto response mode becomes valid. Before setting null auto response mode, INBUFM = 0 should be confirmed because the mode can be set only when the buffer is empty. If the INBUFM bit is 1, the buffer should be emptied with the ACLRM bit. While a transition to null auto response mode is being made, data should not be written from the FIFO port.

To make a transition from null auto response mode to normal mode, pipe operation disabled state (response PID = NAK) should be retained for the period of zero-length packet transmission (full-speed: 10 μs, high-speed: 3 μs) before canceling null auto response mode. In normal mode, data can be written from the FIFO port; therefore, packet transmission to the host is enabled by enabling pipe operation (response PID = BUF).

chanhvinh.com

26.4.4 FIFO Buffer Memory

(1) FIFO Buffer Memory Allocation

Figure 26.9 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).

The buffer memory sets independent areas for each pipe. In the memory areas, 64 bytes comprise one block, and the memory areas are set using the first block number of the number of blocks (specified using the BUFNMB and BUFSIZE bits in PIPEBUF).

Independent 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 PIPEnCFG, 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 PIPEnCFG, 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 buffer memory (reading and writing data). A pipe is assigned to the FIFO port by specifying the pipe number using the CURPIPE bit in C/DnFIFOSEL.

The buffer statuses of the various pipes can be confirmed using the BSTS bit in DCPCTR and the INBUFM bit in PIPEnCTR. Also, the access right of the FIFO port can be confirmed using the FRDY bit in CFIFOCTR or DnFIFOCTR.

chanhvinh.com

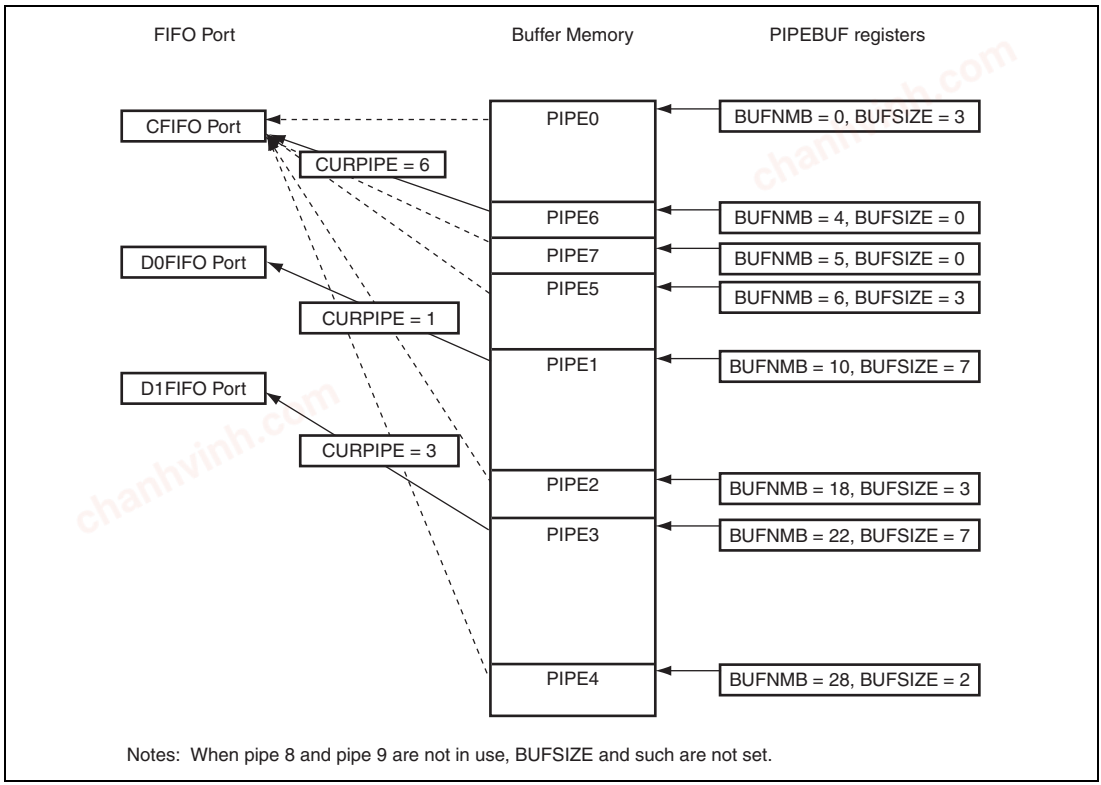


Figure 26.9 Example of a Buffer Memory Map

(a) Buffer Status

Tables 26.18 and 26.19 show the buffer status. The buffer memory status can be confirmed using the BSTS bit in DCPCTR and the INBUFM bit in PIPEnCTR. The access direction for the buffer memory can be specified using either the DIR bit in PIPEnCFG or the ISEL bit in CFIFOSEL (when DCP is selected).

The INBUFM bit is valid for PIPE0 to PIPE5 in the sending direction.

For an IN pipe uses double buffer, the BSTS bit can be used to monitor the buffer memory status of CPU side and the INBUFM bit to monitor the buffer memory status of SIE side. In the case like the BEMP interrupt may not shows the buffer empty status because the CPU (direct memory access controller) writes data slowly, the INBUFM bit can be used to confirm the end of sending.

chanhvinh.com

Table 26.18 Buffer Status Indicated by the BSTS Bit

ISEL or DIR	BSTS	Buffer Memory State
0 (receiving direction)	0	There is no received data, or data is being received. Reading from the FIFO port is inhibited.
0 (receiving direction)	1	There is received data, or a zero-length packet has been received. Reading from the FIFO port is allowed. However, because reading is not possible when a zero-length packet is received, the buffer must be cleared.
1 (transmitting direction)	0	The transmission has not been finished. Writing to the FIFO port is inhibited.
1 (transmitting direction)	1	The transmission has been finished. CPU write is allowed.

Table 26.19 Buffer Status Indicated by the INBUFM Bit

IDIR	INBUFM	Buffer Memory State
0 (receiving direction)	Invalid	Invalid
1 (transmitting direction)	0	The transmission has been finished. There is no waiting data to be transmitted.
1 (transmitting direction)	1	The FIFO port has written data to the buffer. There is data to be transmitted

chanhvinh.com

(b) FIFO Buffer Clearing

Table 26.20 shows the clearing of the FIFO buffer memory by this module. The buffer memory can be cleared using the three bits indicated below.

Table 26.20 List of Buffer Clearing Methods

Bit Name	BCLR	DCLRM	ACLRM
Register	CFIFOCTR DnFIFOCTR	DnFIFOSEL	PIPEnCTR
Function	Clears the buffer memory on the CPU side	In this mode, after the data of the specified pipe has been read, the 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

(c) Buffer Areas

Table 26.21 shows the FIFO buffer memory map of this controller. The buffer memory has special fixed areas to which pipes are assigned in advance, and user areas that can be set by the user.

The buffer for the DCP is a special fixed area that is used both for control read transfers and control write transfers.

The PIPE6 to PIPE9 area is assigned in advance, but the area for pipes that are not being used can be assigned to PIPE1 to PIPE5 as a user area.

The settings should ensure that the various pipes do not overlap. Note that each area is twice as large as the setting value in the double buffer.

Also, the buffer size should not be specified using a value that is less than the maximum packet size.

chanhvinh.com

Table 26.21 Buffer Memory Map

Buffer Memory Number	Buffer Size	Pipe Setting	Note
H'0 to H'3	256 bytes	Fixed area only for the DCP	Single buffer, continuous transfers enabled
H'4	64 bytes	Fixed area for PIPE6	Single buffer
H'5	64 bytes	Fixed area for PIPE7	Single buffer
H'6	64 bytes	Fixed area for PIPE8	Single buffer
H'7	64 bytes	Fixed area for PIPE9	Single buffer
H'8 to H'7F	Up to 7616 bytes	PIPE1 to PIPE5 user area	Double buffer can be set, continuous transfers enabled

(d) Auto Buffer Clear Mode Function

With this module, all of the received data packets are discarded if the ACLRM bit in PIPEnCTR is set to 1. If a normal data packet has been received, the ACK response is returned to the host controller. This function can be set only in the buffer memory reading direction.

Also, if the ACLRM bit is set to 1 and then to 0, the buffer memory of the selected pipe can be cleared regardless of the access direction.

(e) Buffer Memory Specifications (Single/Double Setting)

Either a single or double buffer can be selected for PIPE1 to PIPE5, using the DBLB bit in PIPEnCFG. The double buffer is a function that assigns two memory areas specified with the BUFSIZE bit in PIPEBUF to the same pipe. Figure 26.10 shows an example of buffer memory settings for this module.

chanhvinh.com

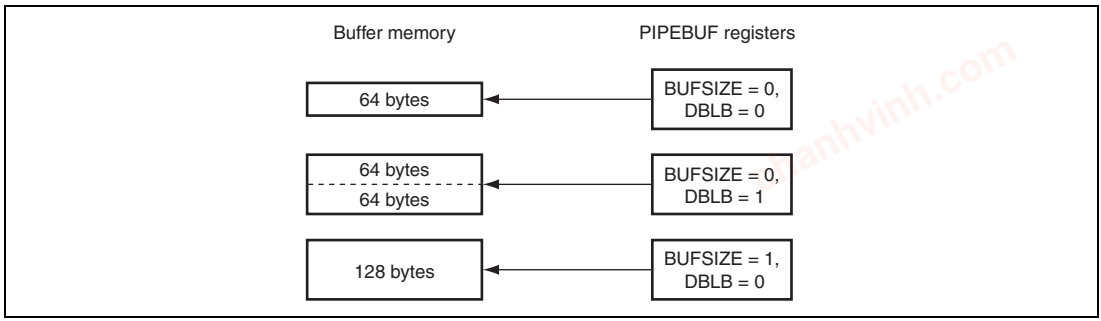


Figure 26.10 Example of Buffer Memory Settings

(f) Buffer Memory Operation (Continuous Transfer Setting)

Either the continuous transfer mode or the non-continuous transfer mode can be selected, using the CNTMD bit in PIPEnCFG. This selection is valid for PIPE1 to PIPE5.

The continuous transfer mode function is a function that sends and receives multiple transactions in succession. When the continuous transfer mode is set, data can be transferred without interrupts being issued to the CPU, up to the buffer sizes assigned for each of the pipes.

In the continuous sending mode, the data being written is divided into packets of the maximum packet size and sent. If the data being sent is less than the buffer size (short packet, or the integer multiple of the maximum packet size is less than the buffer size), BVAL = 1 must be set after the data being sent has been written.

In the continuous reception mode, interrupts are not issued during reception of packets up to the buffer size, until the transaction counter has ended, or a short packet is received.

Table 26.22 describes the relationship between the transfer mode settings by CNTMD bit and the timings at which reading data or transmitting data from the FIFO buffer is enabled.

chanhvinh.com

Table 26.22 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	Method of Determining if Reading or Transmitting Data is Enabled
Non-continuous transfer (CNTMD = 0)	In the receiving direction (DIR = 0), reading data from the FIFO buffer is enabled when: • This module receives one packet.
	• In the transmitting direction (DIR = 1), transmitting data from the FIFO buffer is Data of the maximum packet size is written to the FIFO buffer. or • Data of the short packet size (including 0-byte data) is written to the FIFO buffer and then writes 1 to BVAL.
Continuous transfer (CNTMD = 1)	In the receiving direction (DIR = 0), reading data from the FIFO buffer is enabled when: • 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). • This module receives a short packet other than a zero-length packet. • This module receives a zero-length packet when data is already stored in the FIFO buffer assigned to the selected pipe. or • 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 when: • 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. or • The number of data bytes less than the size of a single FIFO buffer plane (including 0-byte data) assigned to the selected pipe is written to the FIFO buffer and then 1 is written to BVAL. • A number of data bytes less than the size of a single FIFO buffer plane assigned to the selected pipe is written to the FIFO buffer and the DMA transfer end signal is received when the last byte is written.

chanhvinh.com

Figure 26.11 shows an example of buffer memory operation for this module.

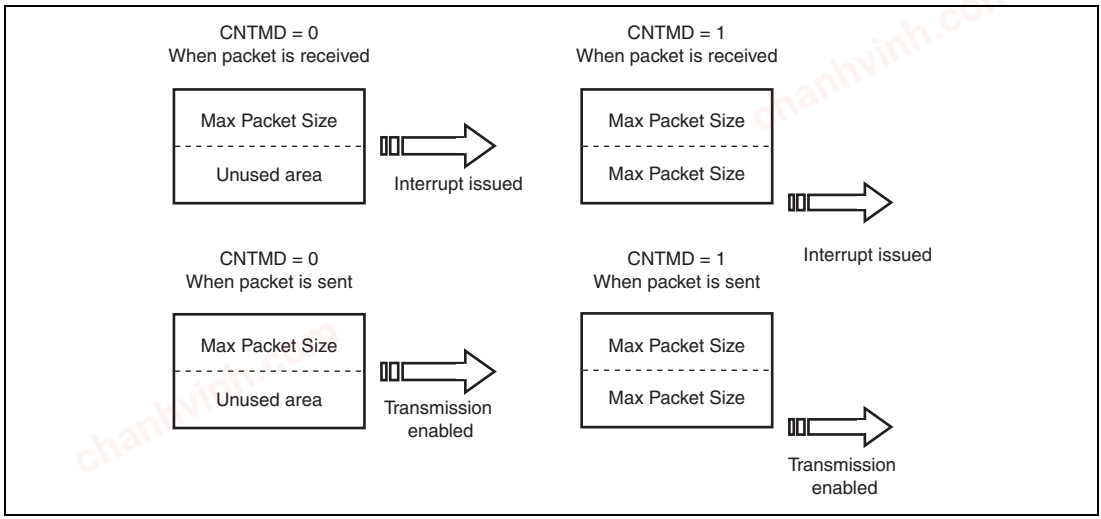


Figure 26.11 Example of Buffer Memory Operation

(2) FIFO Port Functions

Table 26.23 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 enable sending of data before the buffer is full (or before the maximum packet size for non-continuous transfers), the BVAL bit in C/DnFIFOCTR must be set to end the writing. 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 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 is being received (DTLN = 0), so the BCLR bit in the register must be used to release the buffer. The length of the data being received can be confirmed using the DTLN bit in C/DnFIFOCTR.

chanhvinh.com

Table 26.23 FIFO Port Function Settings

Register Name	Bit Name	Function	Note
C/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 DMA transfers	For DnFIFO only
	MBW	FIFO port access bit width	
	BIGEND	Selects FIFO port endian	
	ISEL	FIFO port access direction	
	CURPIPE	Selects the current pipe	For DCP only
C/DnFIFOCTR	BVAL	Ends writing to the buffer memory	
	BCLR	Clears the buffer memory on the CPU side	
	DTLN	Checks the length of received data	

(a) FIFO Port Selection

Table 26.24 shows the pipes that can be selected with the various FIFO ports. The pipe to be accessed is selected using the CURPIPE bit in C/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 modification is being executed by this module.) Then, the FIFO port can be accessed after FRDY = 1 is checked .

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 PIPEnCFG. The ISEL bit determines this only for the DCP.

Table 26.24 FIFO Port Access Categorized by Pipe

Pipe	Access Method	Port that can be Used
DCP	CPU access	CFIFO port register
PIPE1 to PIPE9	CPU access	CFIFO port register
	DMA access	D0FIFO/D1FIFO port register

chanhvinh.com

(b) REW Bit

It is possible to temporarily stop access to the pipe currently being accessed, access a different pipe, and then continue processing using the current pipe once again. The REW bit in C/DnFIFOSEL is used for this.

If a pipe is selected when the REW bit is set to 1 and at the same time the CURPIPE bit in C/DnFIFOSEL is set, the pointer used for reading from and writing to the buffer memory is reset, and reading or writing can be carried out from the first byte. Also, if a pipe is selected with 0 set for the REW bit, data can be read and written in continuation of the previous selection, without the pointer used for reading from and writing to the buffer memory being reset.

To access the FIFO port, FRDY = 1 must be ensured after selecting a pipe.

(c) Accessing FIFO Port for Odd Data

For reading data from the FIFO port, when the number of data bits to be read is smaller than the access width specified by the MBW bits in the FIFO port select registers, read the data with the specified width and discard the unnecessary bits through software.

For writing data to the FIFO port, when the number of data bits to be written is smaller than the access width specified by the MBW bits in the FIFO port select registers, access the registers as shown in the following examples. In the examples, the FIFO port access width is 32 bits (MBW = 10) and 24-bit data is written to the FIFO port.

chanhvinh.com

Example 1 for writing odd data: Writing data with 16-bit width once and then with 8-bit width once

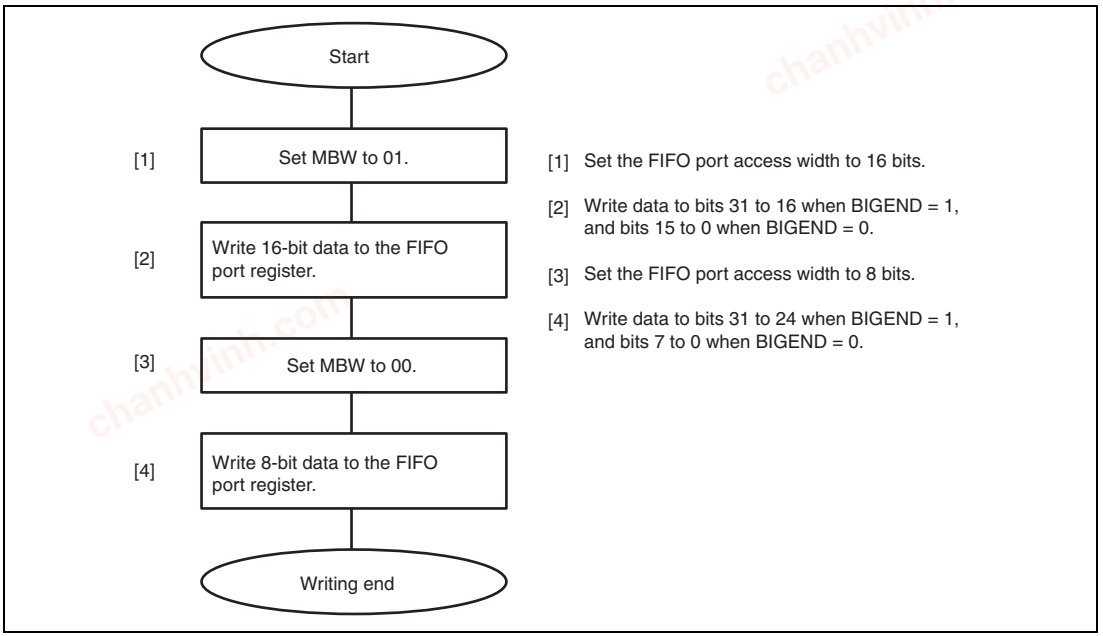


Figure 26.12 Example 1 for Writing Odd Data to FIFO Port

Example 1 for writing odd data 2: Writing data with 8-bit width three times

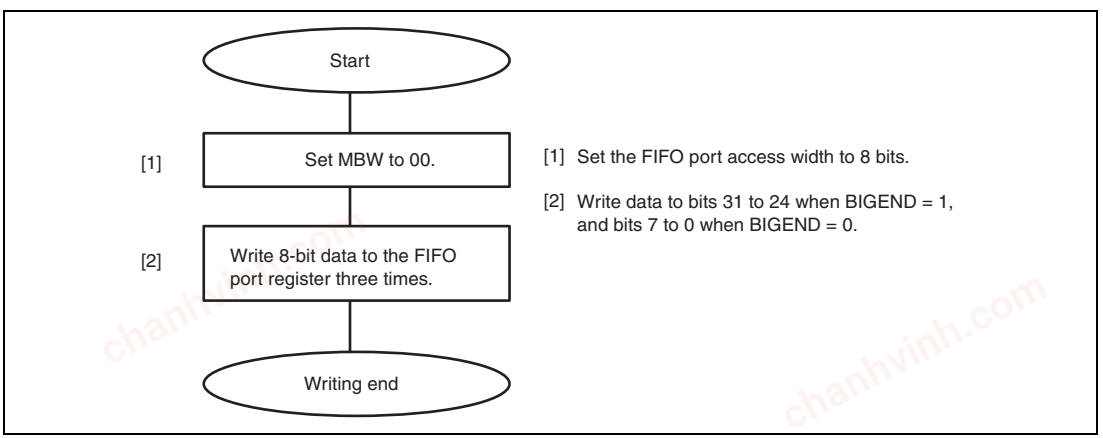


Figure 26.13 Example 2 for Writing Odd Data to FIFO Port

chanhvinh.com

(d) Modifying MBW Bits when the Pipe is in the Receiving Direction

When the specified pipe is in the receiving direction, modify the MBW bits in the FIFO port select registers (CFIFOSEL, D0FIFOSEL, D1FIFOSEL) simultaneously with the CURPIPE bits. When the DCP is currently set (CURPIPE = 000) in the CFIFO port select register, modify the MBW bits simultaneously with the CURPIPE bits or ISEL bit.

To modify only the MBW bits for the currently set pipe, follow the procedure below. Once the buffer memory starts to be read out, however, do not modify the MBW bits until the entire data has been read out.

When the selected CURPIPE is in the writing direction to buffer memory, the port access width can be changed simply by setting the MBW bits.

Once the buffer memory starts to be written to, however, do not modify the port access width from 8 bits to 16 or 32 bits, or from16 bits to 32 bits.

When CURPIPE setting is not DCP (000) for D0FIFO, D1FIFO, or CFIFO

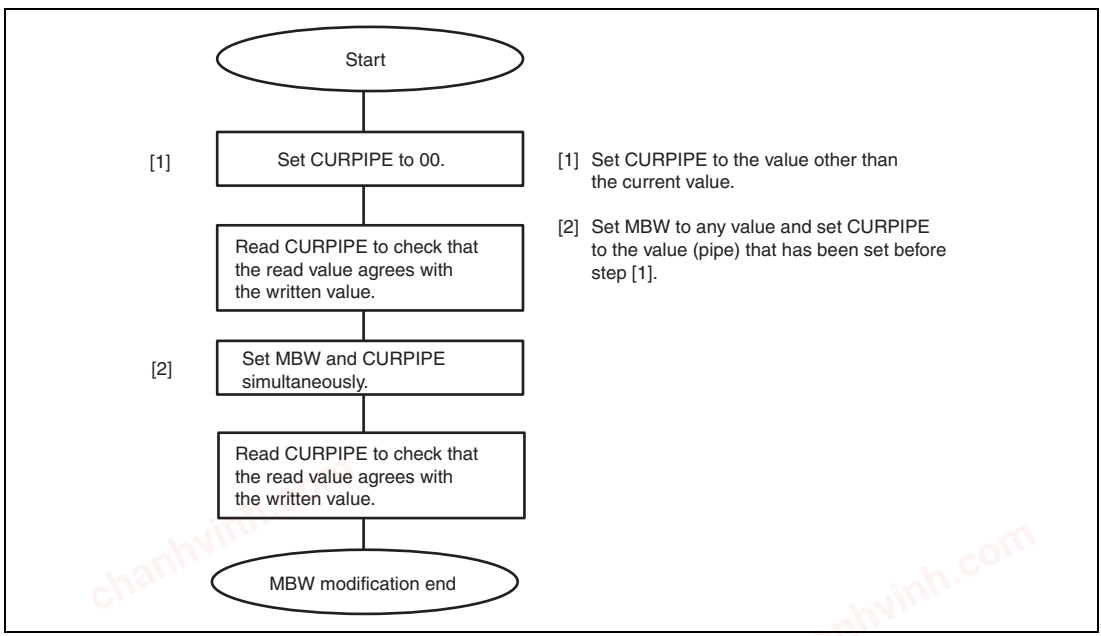


Figure 26.14 MBW Modification Procedure Example when CURPIPE Setting is not DCP (000) for D0FIFO, D1FIFO, or CFIFO

chanhvinh.com

When CURPIPE setting is DCP (000) for CFIFO

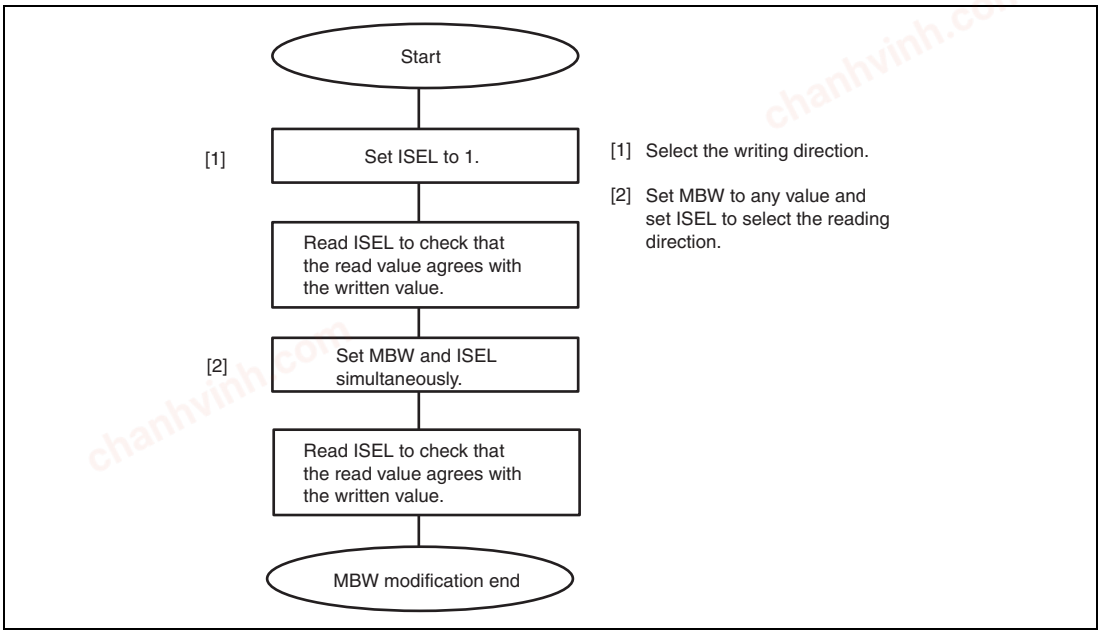


Figure 26.15 MBW Modification Procedure Example When CURPIPE Setting is DCP (000)

(3) DMA Transfers (D0FIFO/D1FIFO Port)

(a) Overview of DMA Transfers

For pipes 1 to 9, the FIFO port can be accessed using the direct memory access controller. When accessing the buffer for the pipe targeted for DMA transfer is enabled, a DMA transfer request is issued.

The unit of transfer to the FIFO port should be selected using the MBW bit in DnFIFOSEL and the pipe targeted for the DMA transfer should be selected using the CURPIPE bit. The selected pipe should not be changed during the DMA transfer.

(b) Auto Recognition of DMA Transfer Completion

With this module, it is possible to complete FIFO data writing through DMA transfer by controlling DMA transfer end signal input. The DMA transfer end signal is output from the direct memory access controller when the controller transfers data through DMA for the times specified by the DMA transfer count register (DMATCR) of the direct memory access controller. When a transfer end signal is sampled, the module enables buffer memory transmission (the same

chanhvinh.com

condition as when BVAL = 1). Whether to sample the DMA transfer end signal can be specified through the TENDE bit in DnFBCFG.

Note that this function cannot be used when the DMA transfer size is set to 16 bytes.

(c) 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 26.25 shows the packet reception and buffer memory clearing processing for each of the various settings. As shown, 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 26.25 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

26.4.5 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 256-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 Function is Selected

(a) Setup Stage

USQREQ, 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. 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 (SIGN1 or SACK bits 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. Also specify the transfer direction using the DIR bit in the DCPCFG register.

For the first data packet of the data stage, the data PID must be transferred as DATA1. Transaction is done by setting the data PID = DATA1 and the PID bit = BUF using the SQSET bit in DCPCTR. Completion of data transfer is detected using the BRDY or BEMP interrupts.

Setting continuous transfer mode allows data transfers over multiple packets. Note that when continuous transfer mode is set for the receiving direction, the BRDY interrupt is not generated until the buffer becomes full or a short packet is received (the integer multiple of the maximum packet size, and less than 256 bytes).

For control write transfers, when the number of data bytes to be sent is the integer multiple of the maximum packet size, a zero-length packet must be controlled to be sent at the end.

chanhvinh.com

(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.

(2) Control Transfers when the Function Controller Function 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 operates in the setup stage is noted below.

(i) When a new USB request is received, this module sets the following registers:

- Set the VALID bit in INTSTS0 to 1.
- Set the PID bit 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 no-data control transfers, and controls the stage transition. For a wrong sequence, the sequence error of the control transfer stage transition interrupt is generated, and the software is notified. For information on the stage control of this module, see figure 26.7.

chanhvinh.com

(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.

A transaction is executed by setting the PID bits in the DCPCTR register to BUF. The BRDY interrupt or the BEMP interrupt can be used to detect the end of data transfer. Use the BRDY interrupt to detect the end of control write transfers and the BEMP interrupt to detect the end of control read transfers.

If the data being transferred is larger than the size of the DCP buffer memory, the data transfer should be carried out using the BRDY interrupt for control write transfers and the BEMP interrupt for control read transfers.

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 bit in DCPCTR set to PID = 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.

(i) For control read transfers:

This module receives a zero-length packet from the USB host and sends an ACK response.

(ii) For control write transfers and no-data control transfers:

This module sends the zero-length packet from the USB host and receives an ACK response.

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.

(i) bmRequestType ≠ H'00

(ii) wIndex ≠ H'00

(ii) wLength ≠ H'00

(iv) wValue > H'7F

(v) DVSQ = 011 (Configured)

For all requests other than the SET_ADDRESS request, the corresponding response is required.

chanhvinh.com

26.4.6 Bulk Transfers (PIPE1 to PIPE5)

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 Function is Selected

This module automatically sends a PING packet in the OUT direction.

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. The module returns to PING packet sending mode when a NAK or NYET is received during an OUT transaction.

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 a NYET/NAK handshake, clearing the sequence toggle bits (SQCLR), and setting the buffer clear bit (ACLRM) in PIPEnCTR.

chanhvinh.com

(2) NYET Handshake Control when the Function Controller Function is Selected

Table 26.26 lists the responses to received tokens during bulk transmission and control transmission. The USB 2.0 host/function module returns a NYET when an OUT token is received while there is only one packet of empty space in the buffer memory, during both bulk transmission and control transmission. However, an ACK is returned instead of a NYET even under the above conditions when a short packet is received.

Table 26.26 List of Responses to Received Tokens

Value Set for PID Bit in DCPCTR	Buffer Memory State	Token	Response	Note
NAK/STALL	—	SETUP	ACK	—
	—	IN/OUT/ PING	NAK/STALL	—
BUF	—	SETUP	ACK	—
	RCV-BRDY1	OUT/PING	ACK	If an OUT token is received, a data packet is received.
	RCV-BRDY2	OUT	NYET	Notifies whether a data packet can be received
	RCV-BRDY2	OUT (Short)	ACK	Notifies whether a data packet can be received
	RCV-BRDY2	PING	ACK	Notifies that a data packet can be received
	RCV-NRDY	OUT/PING	NAK	Notifies that a data packet cannot be received
	TRN-BRDY	IN	DATA0/DATA1	A data packet is transmitted
	TRN-NRDY	IN	NAK	Notifies that a data packet cannot be transmitted

[Legend]

RCV-BRDY1: When an OUT/PING token is received, there is space in the buffer memory for two or more packets.

RCV-BRDY2: 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 sent in the buffer memory.

TRN-NRDY: When an IN token is received, there is no data to be sent in the buffer memory.

chanhvinh.com

26.4.7 Interrupt Transfers (PIPE6 to PIPE9)

When the function controller function 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 function is selected, this module can set the timing of issuing a token using the interval timer. 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 Function is Selected

For interrupt transfers, intervals between transactions are set in the IITV bits in PIPEPERI. This controller issues an interrupt transfer token based on the specified intervals.

(a) Counter Initialization

This controller 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.

chanhvinh.com

(b) 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.

26.4.8 Isochronous Transfers (PIPE1 and PIPE2)

This module has the following functions pertaining to isochronous transfers.

1. Notification of isochronous transfer error information
2. Interval counter (specified by the IITV bit)
3. Isochronous IN transfer data setup control (IDLY function)
4. Isochronous IN transfer buffer flush function (specified by the IFIS bit)

This module does not support the High Bandwidth transfers of isochronous transfers.

When using more than one pipe simultaneously for isochronous transfers, follow the packet constraints provided in section 5.6.3, Isochronous Transfer Packet Size Constraints, in Universal Serial Bus Revision 2.0 Specification.

(1) Error Detection with Isochronous Transfers

This module has a function for detecting the error information noted below, so that when errors occur in isochronous transfers, software can control them. Tables 26.27 and 26.28 show the priority in which errors are confirmed and the interrupts that are generated.

(a) PID errors

- If the PID of the packet being received is illegal

(b) CRC errors and bit stuffing errors

- If an error occurs in the CRC of the packet being received, or the bit stuffing is illegal

(c) Maximum packet size exceeded

- The maximum packet size exceeded the set value.

chanhvinh.com

(d) Overrun and underrun errors

- When host controller function is selected:
  - When using isochronous IN transfers (reception), the IN token was received but the buffer memory is not empty.
  - When using isochronous OUT transfers (transmission), the OUT token was transmitted, but the data was not in the buffer memory.
- When function controller function 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 the buffer memory was not empty.

(e) Interval errors

When function controller function is selected, the following cases are considered as interval errors:

- During an isochronous IN transfer, the token could not be received during the interval frame.
- During an isochronous OUT transfer, the OUT token was received during frames other than the interval frame.

chanhvinh.com

Table 26.27 Error Detection when a Token is Received

Detection Priority	Error	Generated Interrupt and Status
1	PID errors	No interrupts are generated in both cases when the host controller function is selected and the function controller function is selected (ignored as a corrupted packet).
2	CRC error and bit stuffing errors	No interrupts generated in both cases when the host controller function is selected and the function controller function is selected (ignored as a corrupted packet).
3	Overrun and underrun errors	An NRDY interrupt is generated to set the OVRN bit in both cases when host controller function is selected and function controller function is selected. When the host controller function is selected, no tokens are transmitted. When the function controller function is selected, a zero-length packet is transmitted in response to IN token. However, no data packets are received in response to OUT token.
4	Interval errors	An NRDY interrupt is generated when the function controller function is selected. It is not generated when the host controller function is selected.

chanhvinh.com

Table 26.28 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 to set the CRCE bit in both cases when the host controller function is selected and the function controller function is selected.
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 function is selected and the function controller function is selected.

(2) DATA-PID

This module does not support High Bandwidth transfers. When the function controller function is selected, this module operates as follows in response to the received PID.

(a) IN direction

— DATA0: Sent as data packet PID
— DATA1: Not sent
— DATA2: Not sent
— mDATA: Not sent

(b) 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

(c) 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

The isochronous interval can be set using the IITV bits in PIPEPERI. The interval counter enables the functions shown in table 26.29 when the function controller function is selected. When the host controller function is selected, this module generates the token issuance timing. When the host controller function is selected, the interval counter operation is the same as the interrupt transfer operation.

Table 26.29 Functions of the Interval Counter when the Function Controller Function is Selected

Transfer Direction	Function	Conditions for Detection
IN	IN 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 isochronisms can be maintained even if an SOF is damaged. The frame interval that can be set is the 2^{IITV} frame or 2^{IITV} μ frames.

(a) Interval Counter Initialization when the Function Controller Function is Selected

This module initializes the interval counter under the following conditions.

- Power-on reset
  The IITV bit is initialized.
- Buffer memory initialization 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 conditions 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.

chanhvinh.com

The interval counter is not initialized under the conditions noted below.

1. When the PID bit is set to NAK or STALL

   The interval timer does not stop. This module attempts the transactions at the subsequent interval.
2. The USB bus reset or the USB is suspended

   The IITV bit is not initialized. When the SOF has been received, the counter is restarted from the value prior to the reception of the SOF.

(b) Interval Counting and Transfer Control when the Host Controller Function is Selected

This module controls the interval between token issuance operations based on the IITV bit settings. Specifically, this module issues a token for a 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 have been set to BUF.

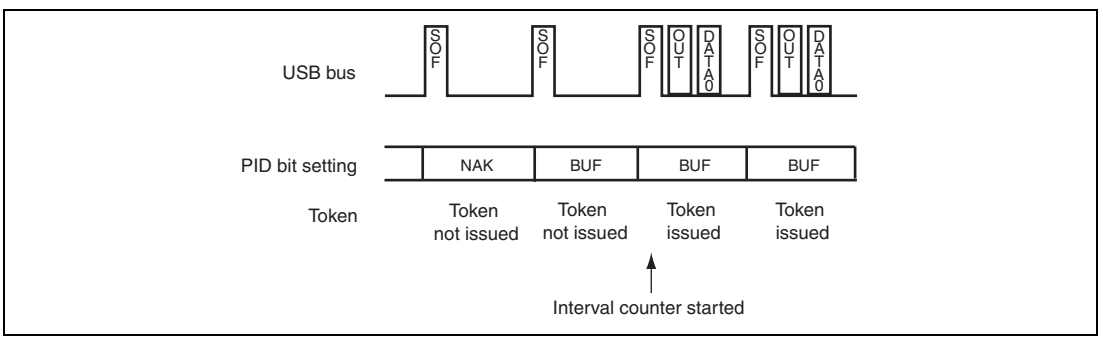


Figure 26.16 Token Issuance when IITV = 0

chanhvinh.com

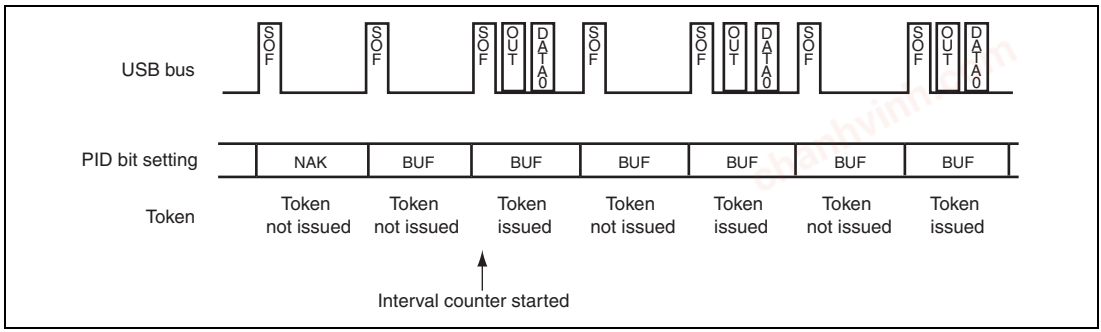


Figure 26.17 Token Issuance when IITV = 1

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.

- 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 data is read from the FIFO buffer too late).
- 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 data is written to the FIFO buffer too late).
  The token issuance interval is reset when a power-on reset is applied or the ACLRM bit is set to 1.

chanhvinh.com

(c) Interval Counting and Transfer Control when the Function Controller Function is Selected

- When the selected pipe is for isochronous OUT transfers

  This module generates the NRDY interrupt when this module fails to receive a data packet within the interval set by the IITV bits in terms of (μ) frames.

  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.

  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 has been set to BUF.

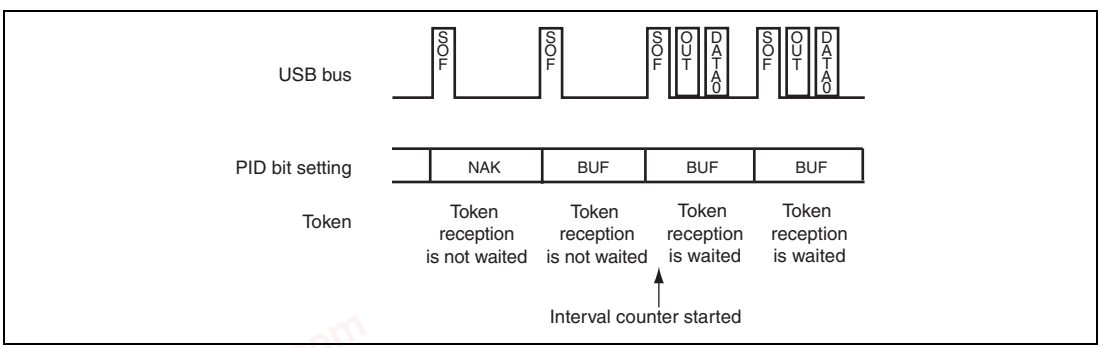


Figure 26.18 Relationship between (μ) Frames and Expected Token Reception when IITV = 0

— 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.

chanhvinh.com

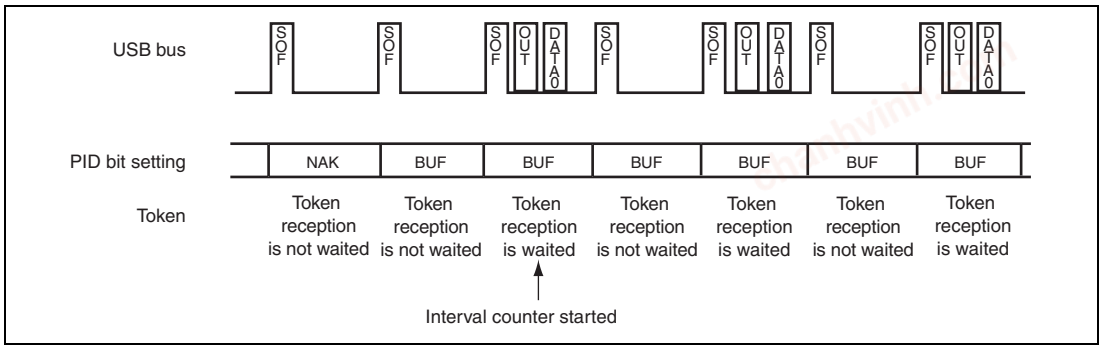


Figure 26.19 Relationship between (μ) Frames and Expected Token Reception when IITV = 1

- 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 by the IITV bits in terms of (μ) frames 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.

  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 interval count clearing condition is any of the following in function controller mode.

  — When a power-on reset is applied.

  — When the ACLRM bit is set to 1.

  — When this module detects a USB bus reset.

chanhvinh.com

(4) Setup of Data to be Transmitted using Isochronous Transfer when the Function Controller Function is Selected

With isochronous data transmission using this module in function controller function, 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, and it makes it possible to designate the frame from which transmission began.

If a double buffer is used for the buffer memory, transmission will be enabled for only one of the two buffers even after the writing of data to both buffers has been completed, that buffer memory being the one to which the data writing was completed first. For this reason, even if multiple IN tokens are received, 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. If the buffer memory is not in the transmission enabled state, however, a zero-length packet is sent and an underrun error occurs.

Figure 26.20 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

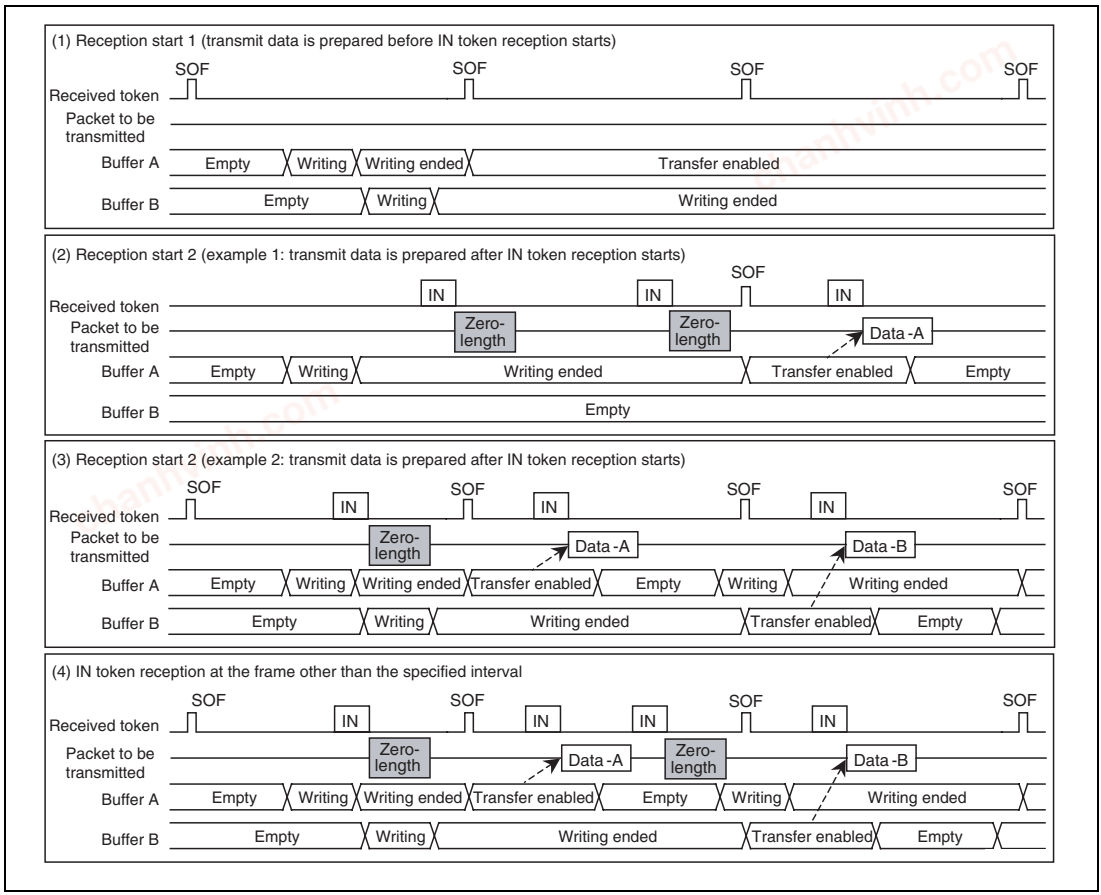


Figure 26.20 Example of Data Setup Function Operation

(5) Isochronous Transfer Transmission Buffer Flush when the Function Controller Function is Selected

If an SOF packet or a μSOF packet is received without receiving an IN token in the interval frame during isochronous data transmission, this module operates as if an IN token had been corrupted, 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, the buffer memory that was cleared is seen as the data having been sent at the same interval frame, and transmission is enabled for the buffer memory that is not discarded with SOF or μSOF packets reception.

chanhvinh.com

The timing at which the operation of the buffer flush function varies depending on the value set for the IITV bit.

- 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 26.21 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 according to the buffer state.

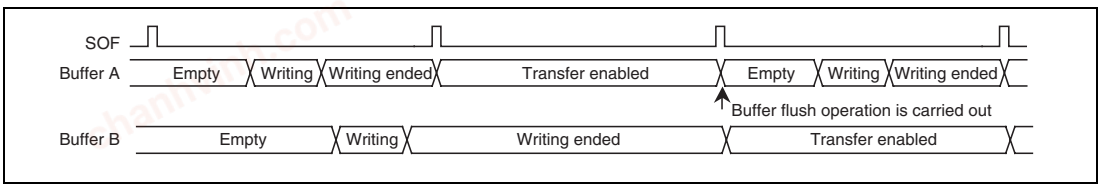


Figure 26.21 Example of Buffer Flush Function Operation

Figure 26.22 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 IN buffer flush function is activated.

If an interval error occurs during an IN transfers, 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.

chanhvinh.com

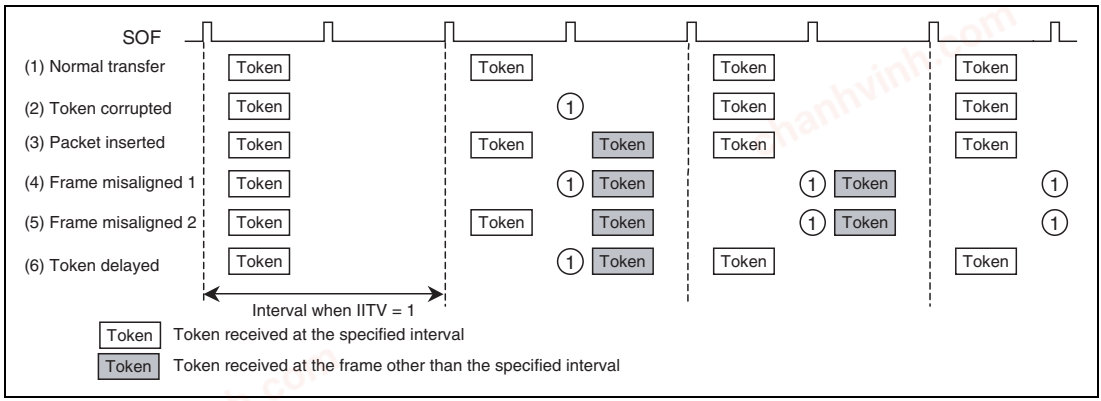


Figure 26.22 Example of an Interval Error Being Generated when IITV = 1

26.4.9 SOF Interpolation Function

When the function controller function is selected and if data could not be received at intervals of 1 ms (when using full-speed operation) or 125 µs (when using high-speed operation) because an SOF packet was corrupted or missing, this module interpolates the SOF. The SOF interpolation operation begins when the USBE and SCKE bits in SYSCFG 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.

- 125 µs/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 detection. These functions also operate normally with SOF interpolation, if the SOF packet was corrupted.

chanhvinh.com

- Refreshing of the frame number and the micro-frame number
- SOFR interrupt and μSOF lock
- Isochronous transfer interval count

If an SOF packet is missing when full-speed operation is being used, the FRNM bit in FRMNUM is not refreshed. If a μSOF packet is missing during high-speed operation, the UFRNM bit in UFRMNUM is refreshed. However, if a μSOF packet for which the UFRNM = 000 is missing, the FRNM bit is not refreshed. In this case, the FRNM bit is not refreshed even if successive μSOF packets other than UFRNM = 000 are received normally.

26.4.10 Pipe Schedule

(1) Conditions for Generating a Transaction

When the host controller function is selected and UACT has been set to 1, this module generates a transaction under the conditions noted in table 26.30.

Table 26.30 Conditions for Generating a Transaction

	Conditions for Generation				
Transaction	**DIR**	**PID**	**IITV0**	**Buffer State**	**SUREQ**
Setup	—*1	—*1	—*1	—*1	1 setting
Control transfer data stage, status stage, bulk transfer	IN	BUF	Invalid	Receive area exists	—*1
	OUT	BUF	Invalid	Send data exists	—*1
Interrupt transfer	IN	BUF	Valid	Receive area exists	—*1
	OUT	BUF	Valid	Send data exists	—*1
Isochronous transfer	IN	BUF	Valid	*2	—*1
	OUT	BUF	Valid	*3	—*1

Notes: 1. Symbols (—) in the table indicate that the condition is one that is unrelated to the generating of tokens. "Valid" indicates that, for interrupt transfers and isochronous transfers, the condition is generated only in transfer frames that are based on the interval counter. "Invalid" indicates that the condition is generated regardless of the interval counter.

2. This indicates that a transaction is generated regardless of whether or not there is a receive area. If there was no receive area, however, the received data is destroyed.

3. This indicates that a transaction is generated regardless of whether or not there is any data to be sent. If there was no data to be sent, however, a zero-length packet is sent.

chanhvinh.com

(2) Transfer Schedule

This section describes the transfer scheduling within a frame of this module. After the module sends an SOF, the transfer is carried out in the sequence described below.

1. Execution of periodic transfers

   A pipe is searched in the order of Pipe 1 → Pipe 2 → Pipe 6 → Pipe 7 → Pipe 8 → Pipe 9, and then, if the pipe is one for which an isochronous or interrupt transfer transaction can be generated, the transaction is generated.
2. Setup transactions for control transfers

   The DCP is checked, and if a setup transaction is possible, it is sent.
3. Execution of bulk and control transfer data stages and status stages

   A pipe is searched in the order of DCP → Pipe 1 → Pipe 2 → Pipe 3 → Pipe 4 → Pipe 5, and then, if the pipe is one for which a bulk or control transfer data stage or a control transfer status stage transaction can be generated, the transaction is generated.
   If a transfer is generated, processing moves to the next pipe transaction regardless of whether the response from the peripheral device is ACK or NAK. Also, if there is time for the transfer to be done within the frame, step 3 is repeated.

(3) USB Communication Enabled

Setting the UACT bit of the DVSTCTR register to 1 initiates sending of an SOF or µSOF, and makes it possible to generate a transaction.

Setting the UACT bit to 0 stops the sending of the SOF or µSOF and initiates a suspend state. If the setting of the UACT bit is changed from 1 to 0, processing stops after the next SOF or µSOF is sent.

chanhvinh.com

26.5 Usage Notes

26.5.1 Procedure for Setting the USB Transceiver

When this module is to be used, start by making settings for the internal USB transceiver.

The method for the settings is described below. Figure 26.23 also gives an example of program code to implement the procedure.

(1) Write 1 to the UACS25 bit in the USB AC characteristics switching register 1 (USBACSWR1).

```
Initialization routine
(1) Set 1 to UACS25.
        MOVI20    #H'FFFFC0C2, R0
        MOV.W     #H'0200, R1
        MOV.W     R1, @R0
```

Figure 26.23 Procedure for Setting the USB Transceiver

26.5.2 Power Supply for USB Transceiver

- Set the voltage level of power supply $USBAV_{CC}$, $USBDV_{CC}$, and $USBUV_{CC}$ to the same as V_{CC}.
- Set the voltage level of power supply $USBAPV_{CC}$ and $USBDPV_{CC}$ to the same as PV_{CC}.
- Set the voltage level of ground $USBAV_{SS}$, $USBDV_{SS}$, $USBUV_{SS}$, $USBAPV_{SS}$, and $USBDPV_{SS}$ to the same as V_{SS}.
- Separate analog power supplies $USBAV_{CC}$, $USBAV_{SS}$, $USBAPV_{CC}$, and $USBAPV_{SS}$ from the digital power supplies.

chanhvinh.com

Section 27 Video Display Controller 3

27.1 Overview

The video display controller 3 provides the following four functions. Note that the video display function and video recording function cannot be used together.

1. Video display function: Reduces the size of the input video, buffers the resultant video data in the memory, and then displays the video on the panel.
2. Video recording function: Stores a specified number of input video fields in on-chip large-capacity RAM or SDRAM.
3. Function for overlaying graphics images (two planes) on the input video.
4. Function for outputting the control signals for the TFT-LCD panel.

27.2 Features

Table 27.1 Features

Item	Function
Operating frequency	Video input clock: 27 MHz Panel clock: 4 MHz to 36 MHz (depends on the panel specifications)
Input video standard	8-bit input conforming to the ITU-R BT.656 standard (27 MHz) 8-bit serial input conforming to the ITU-R BT.601 standard (27 MHz)
Video recording function	Stores video data in the RGB565 format at a rate of 1/2 field (NTSC: 30 fps; PAL: 25 fps).
Video quality adjustment function	Contrast adjustment and brightness adjustment
Video scaling processing	Vertical: ×1/2, ×1/3, or ×1/4 Horizontal: ×2/3, ×1/2, ×1/3, or ×1/4 For the support of PAL, each scaled value can be further multiplied by 6/7.
Graphics images	Two planes (layers 1 and 2) RGB565 progressive format (R = 5 bits, G = 6 bits, and B = 5 bits; 16 bits in total) αRGB4444 progressive format (α = 4 bits, R = 4 bits, G = 4 bits, and B = 4 bits; 16 bits in total)

chanhvinh.com

Item	Function
Graphics functions	α blending window function: Mixes the input video and layers 1 and 2 according to transparency rate α in the specified region (fade-in and fade-out functions are available). Chroma-keying function: Mixes images with applying the specified RGB color according to transparency rate α. Dot α function: Mixes images according to transparency rate α when the target is a graphics image in the αRGB4444 format. For each dot, the priority among the α values of the above functions is as follows: α blending window > chroma-keying > dot α.
Output video size	640 pixels × 480 lines (VGA size) 480 pixels × 240 lines (WQVGA size) 320 pixels × 240 lines (QVGA size) in landscape 240 pixels × 320 lines (QVGA size) in portrait Note: The maximum input video size that can be displayed is 480 pixels × 240 lines for NTSC and 480 pixels × 288 lines for PAL.
Output video format	RGB565 progressive video output (16-bit parallel output)
Sync signal output	Outputs the control signals for the TFT-LCD panel.
Interrupt output	Line interrupt output (can be output on a desired line) VSYNC cycle fluctuation detection signal for BT.601 and BT.656 video input Field write completion signal Overflow/underflow detection signal for the internal buffer

chanhvinh.com

27.3 Input/Output Pins

Table 27.2 Pin Configuration

Symbol	I/O	Pin Name	Function
DV_CLK	Input	Video input clock	BT.601 or BT.656 clock input pin.
DV_VSYNC	Input	VSYNC input	BT.601 VSYNC signal input pin.
DV_HSYNC	Input	HSYNC input	BT.601 HSYNC signal input pin.
DV_DATA7 to DV_DATA0	Input	BT.601 or BT.656 input	BT.601 or BT.656 data signal input pins.
LCD CLK	Output	Panel clock	Panel clock output pin.
LCD_EXTCLK	Input	Panel clock source	Panel clock source input pin.
LCD_VSYNC	Output	Panel VSYNC output	Vertical sync signal output pin for the panel.
LCD_HSYNC	Output	Panel HSYNC output	Horizontal sync signal output pin for the panel.
LCD_DE	Output	Panel data enable output	Data enable signal or data start position pulse signal output pin for the panel.
LCD_DATA15 to LCD_DATA0	Output	Panel data output	Data output pins for the panel. MSB LSB MSB LSB [15:11]: Red [4:0] [10:5]: Green [5:0] [4:0]: Blue [4:0]
LCD_M_DISP	Output	Panel control signal	Alternating signal for the panel.

chanhvinh.com

27.4 Configuration

This module consists of seven functional blocks listed in table 27.3. Figure 27.1 shows the entire block diagram of this module.

Table 27.3 Functional Blocks

Block Name	Overview of Functions
Input timing control block	Selects the timing of the sync signal input with respect to the clock rising or falling edge and selects the sync signal polarity. It also selects the timing of the BT.601 and BT.656 video input signals with respect to the clock rising or falling edge.
Video receiving block	(1) Captures the input video and applies the scaling, contrast, and brightness processing. (2) Converts the YC format into the RGB565 format and stores the data through the IV1-BUS. (3) Applies field skipping processing, and stores the resultant data in the RGB565 format through the IV1-BUS.
Video supplying block	Reads video data through the IV2-BUS.
Graphics block 1	Reads a graphics image (layer 1) from the memory through the IV3-BUS, overlays it on the video sent from the video supplying block, and outputs the result to graphics block 2.
Graphics block 2	Reads a graphics image (layer 2) from the memory through the IV4-BUS, overlays it on the output from graphics block 1, and outputs the result to the panel control block.
Panel control block	Generates the sync signals for output to the panel.
Output timing control block	Controls the timing of the sync signal output with respect to the clock rising or falling edge and controls the sync signal polarity. It also controls the timing of the RGB666 video output signals with respect to the clock rising or falling edge.

chanhvinh.com

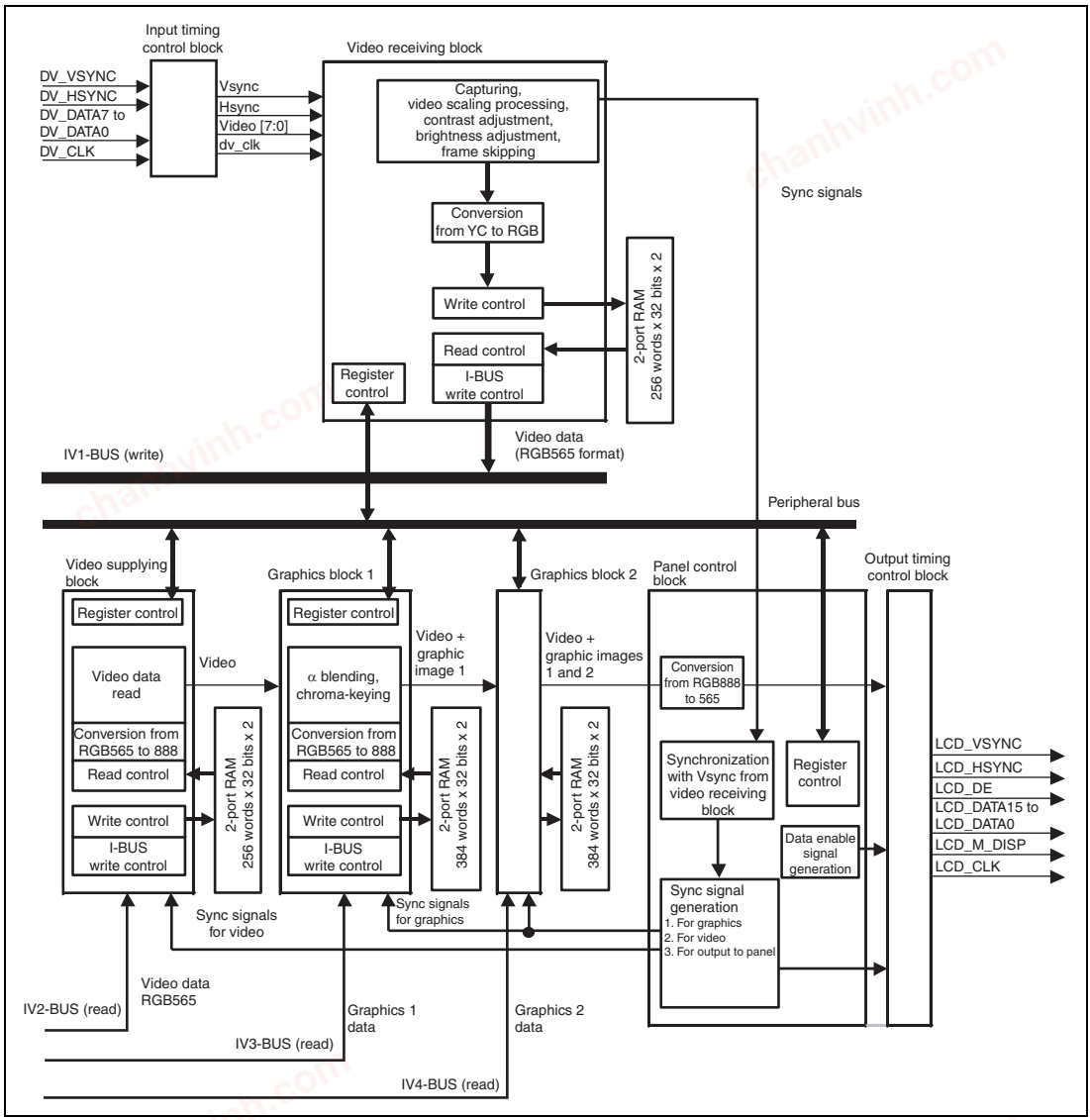


Figure 27.1 Block Diagram

chanhvinh.com

27.5 Input Video Interface

27.5.1 BT.601 Video Input

The DV_VSYNC and DV_HSYNC signals conforming to BT.601 should be input with the timing shown in figure 27.2. The timing for sampling these signal inputs can be selected as either the rising or falling edge of the DV_CLK. Negative polarity is also supported. The field type is determined according to the DV_VSYNC and DV_HSYNC timing.

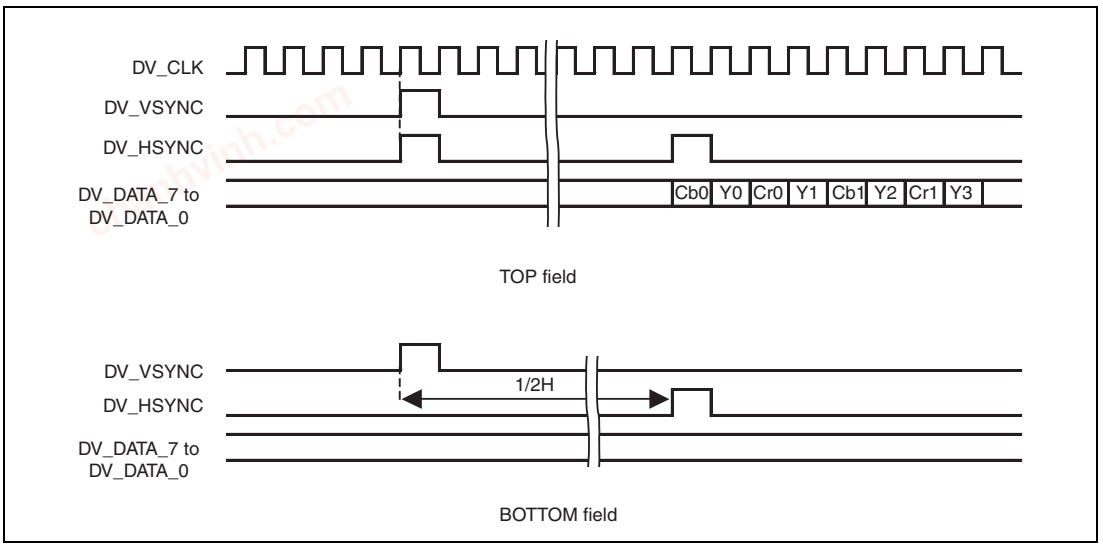


Figure 27.2 BT.601 Input Interface Signals

chanhvinh.com

27.5.2 BT.656 Video Input

This module supports 8-bit video input conforming to the BT.656 standard. From the reference code (EAV or SAV), the Vsync, Hsync, and field information are obtained. Tables 27.4 to 27.6 and figures 27.3 and 27.4 give an overview of the standard.

Table 27.4 Reference Code (EAV/SAV)

Bit Number	7	6	5	4	3	2	1	0
Function	**Fixed**	**F**	**V**	**H**	**P3**	**P2**	**P1**	**P0**
0	1	0	0	0	0	0	0	0
1	1	0	0	1	1	1	0	1
2	1	0	1	0	1	0	1	1
3	1	0	1	1	0	1	1	0
4	1	1	0	0	0	1	1	1
5	1	1	0	1	1	0	1	0
6	1	1	1	0	1	1	0	0

[Legend]

F: 0: Fist field; 1: Second field

V: 1 for the vertical blanking period; otherwise, 0.

H: 0: SAV; 1: EAV

P3 to P0: Protection bits

Table 27.5 525-Line Control in BT.656

Line Number	EAV	SAV	Field Control (F)		Vertical Blanking Control (V)	
4 to 19	B6	AB	0	First field	1	Blanking field
20 to 263	9D	80	0		0	Digital active field
264 to 265	B6	AB	0		1	Blanking field
266 to 282	F1	EC	1	Second field	1	Blanking field
283 to 525	DA	C7	1		0	Digital active field
1 to 3	F1	EC	1		1	Blanking field

chanhvinh.com

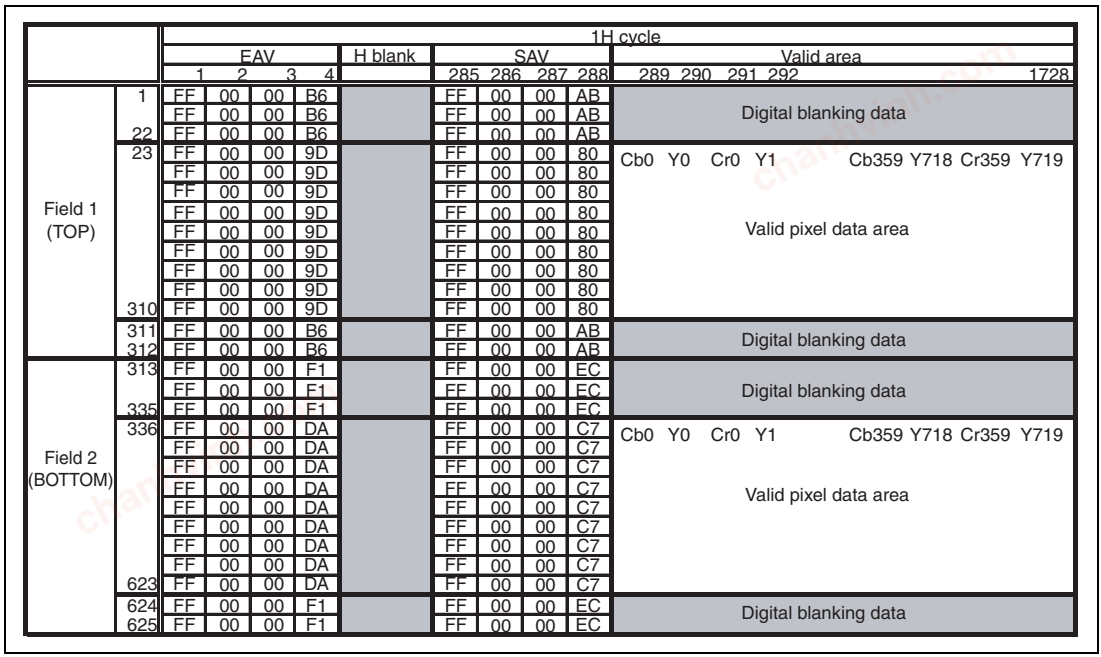
Table 27.6 625-Line Control in BT.656

Line Number	EAV	SAV	Field	Control (F)	Vertical Blanking Control (V)	
1 to 22	B6	AB	0	First field	1	Blanking field
23 to 310	9D	80	0		0	Digital active field
311 to 312	B6	AB	0		1	Blanking field
313 to 335	F1	EC	1	Second field	1	Blanking field
336 to 623	DA	C7	1		0	Digital active field
624 to 625	F1	EC	1		1	Blanking field

		1H cycle																	
		EAV				H blank	SAV				Valid area								
		1	2	3	4		273	274	275	276	277	278	279	280					1716
Field 2	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									
Field 1 (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			Cb359	Y718	Cr359	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									
		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									
	264	FF	00	00	B6		FF	00	00	AB	Digital blanking data								
	265	FF	00	00	B6		FF	00	00	AB									
Field 2 (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			Cb359	Y718	Cr359	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									
		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									

Figure 27.3 BT.656 Timing (525 Lines)

chanhvinh.com



		1H cycle																	
		EAV				H blank	SAV				Valid area								
		1	2	3	4		285	286	287	288	289	290	291	292					1728
Field 1 (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			Cb359	Y718	Cr359	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									
		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									
	311	FF	00	00	B6		FF	00	00	AB	Digital blanking data								
	312	FF	00	00	B6		FF	00	00	AB									
Field 2 (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			Cb359	Y718	Cr359	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									
		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									
	624	FF	00	00	F1		FF	00	00	EC	Digital blanking data								
	625	FF	00	00	F1		FF	00	00	EC									

Figure 27.4 BT.656 Timing (625 Lines)

chanhvinh.com

27.6 Functional Descriptions

27.6.1 Video Display Function

This module provides a function for overlaying the input video and graphics images (two planes) and displaying them on the panel (video display function).

(1) Capturing

The BT.601 or BT.656 input video data is captured from a desired position with a desired size. The position should be specified with respect to the left end of the first line of the TOP field.

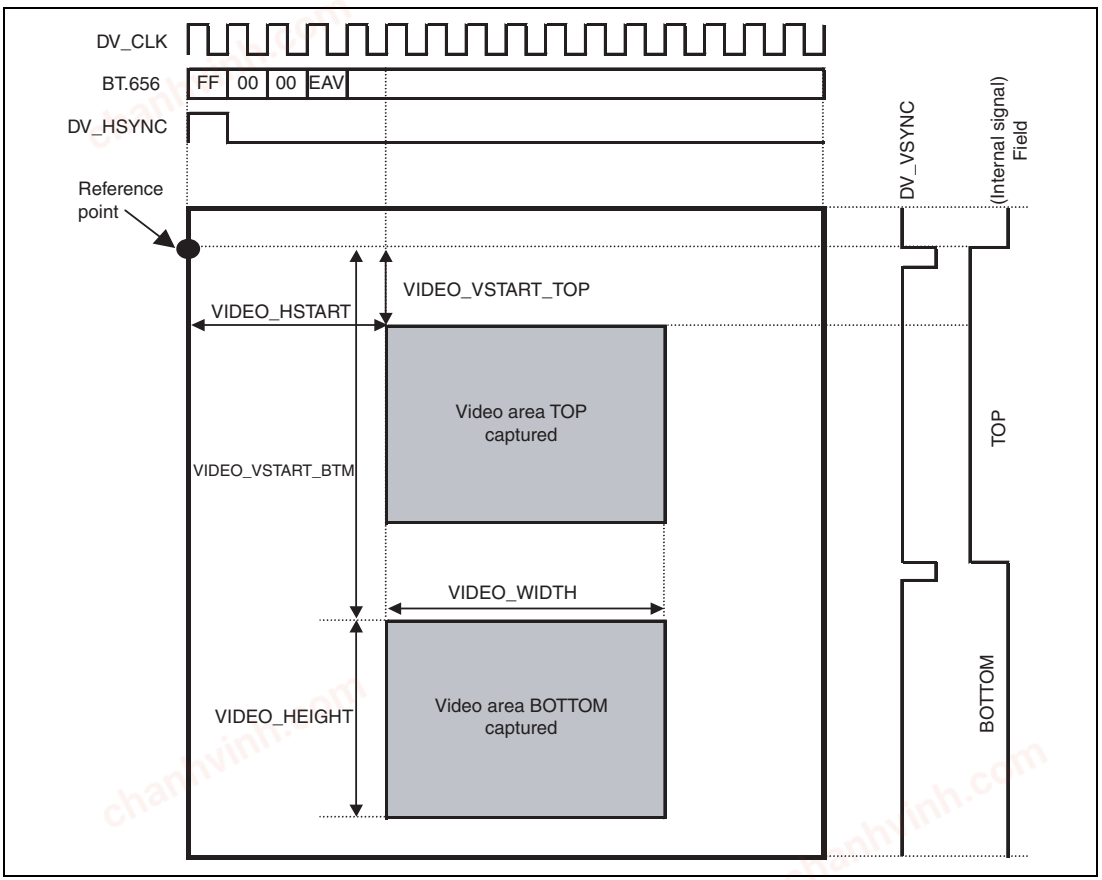


Figure 27.5 Concept of Capturing Function

chanhvinh.com

(2) Video and Graphics Image Processing Flow

The captured video is reduced in size, the contrast and brightness are adjusted, and then the resultant data is stored in the RGB565 format in the memory. The video data stored in the memory and graphics images (two planes) are overlaid and the result is displayed in the panel.

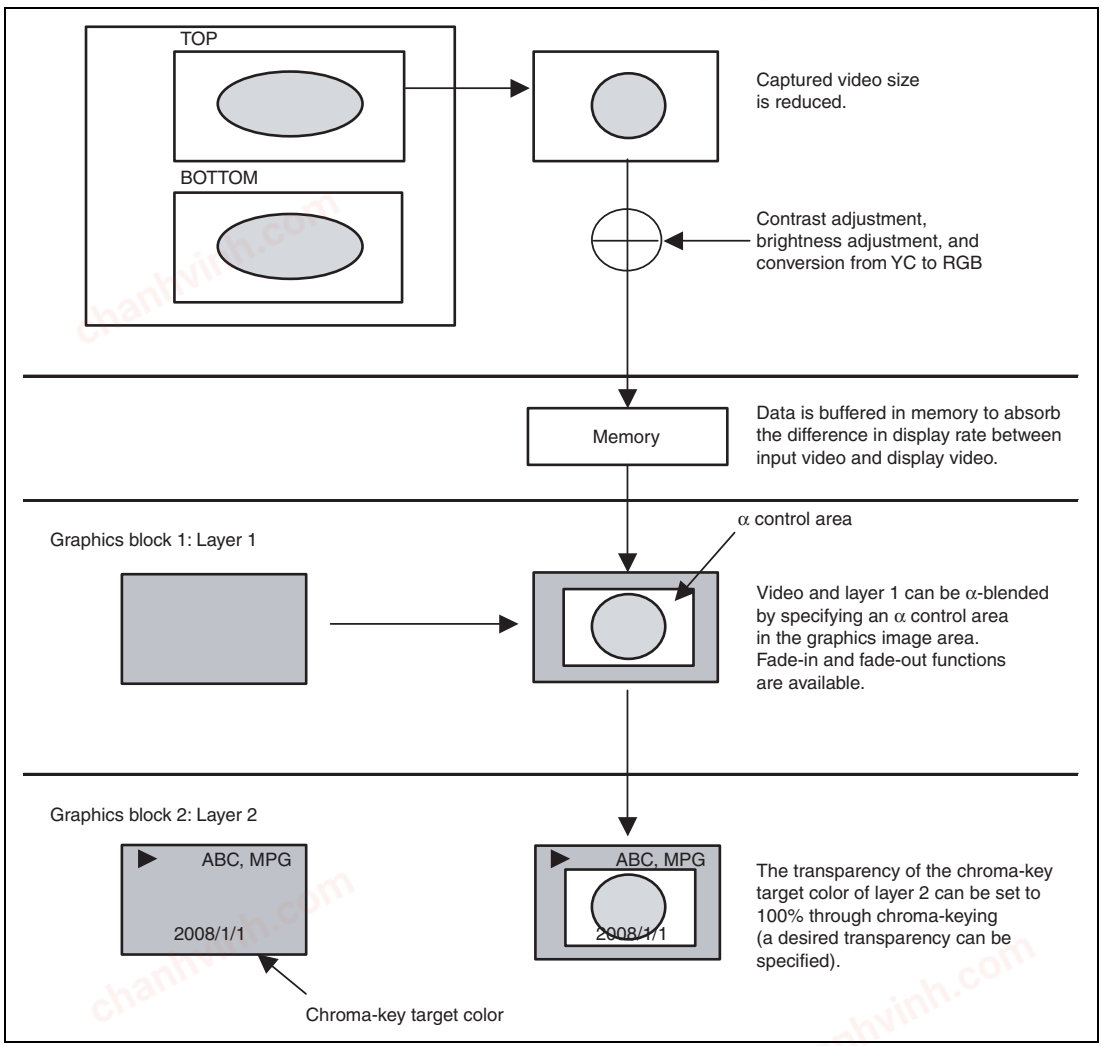


Figure 27.6 Overview of Video and Graphics Image Processing

chanhvinh.com

(3) Conversion from YC to RGB

After contrast and brightness adjustment, the YC422 format is converted to the RGB888 format in accordance with the ITU-R BT601 standard.

$$Y_1 = Y$$
$$Cb_1 = Cb - 128$$
$$Cr_1 = Cr - 128$$
$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.164 & 0.000 & 1.596 \\ 1.164 & -0.391 & -0.813 \\ 1.164 & 2.018 & 0.000 \end{bmatrix} \begin{bmatrix} Y_1 \\ Cb_1 \\ Cr_1 \end{bmatrix}$$

(4) Conversion from RGB888 to RGB565

$$R' = R \times \frac{31}{255}$$
$$G' = G \times \frac{63}{255}$$
$$B' = B \times \frac{31}{255}$$

chanhvinh.com

(5) Relationship between Input Video Vsync and Reference Vsync

The reference Vsync is a signal that is used as the reference when specifying the timing of the Vsync signals for video, graphics images, and output to the panel (see figure 27.7). The timing of the reference Vsync signal can be specified through register settings with respect to the reference point; that is, the left end of the first line of the TOP field.

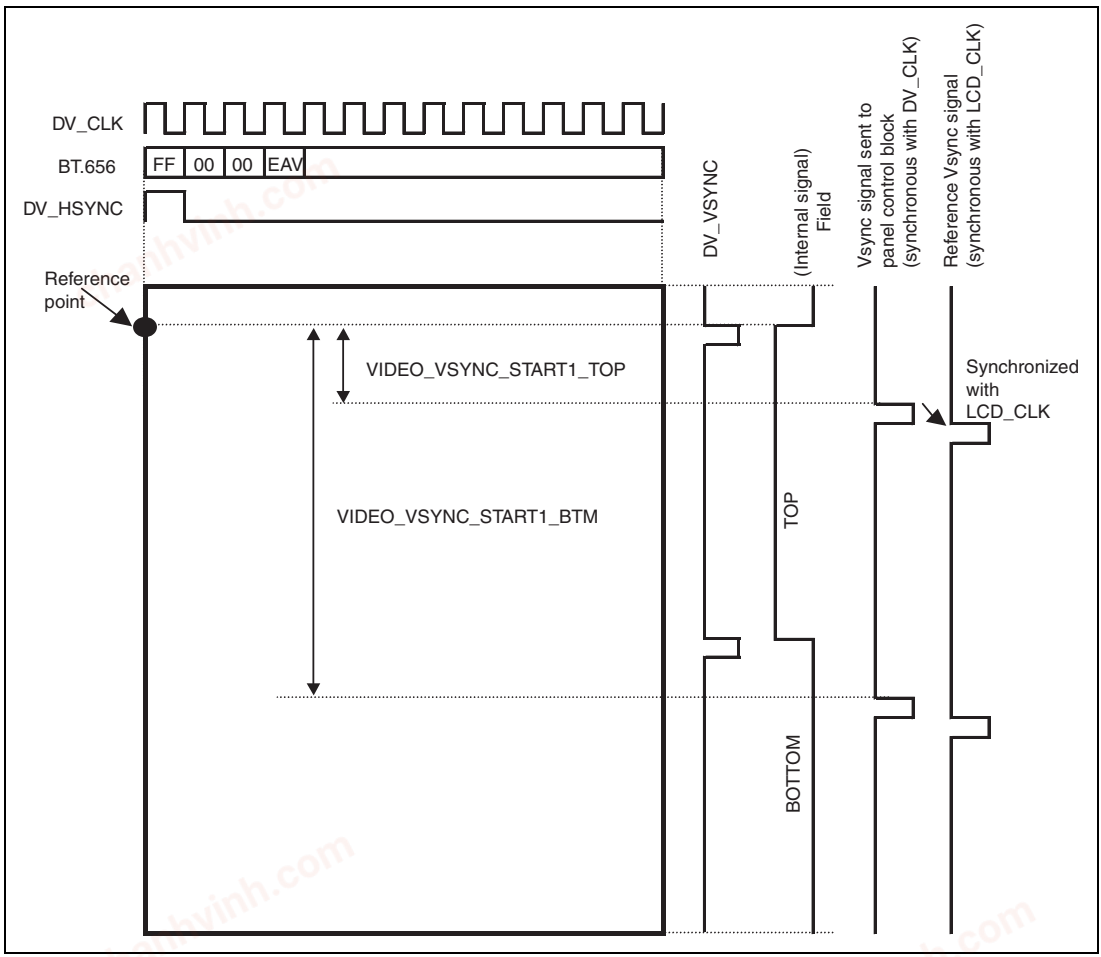


Figure 27.7 Timing of Reference Vsync Generation

chanhvinh.com

(6) Controlling Video and Graphics Image Display Positions

This module provides functions for generating the Vsync and Hsync signals for video, graphics images, and output to the panel with desired timings with respect to the reference Vsync signal. The positions of the video and graphics image displayed on the panel are shown in figure 27.8.

The timing of the Vsync signal for video has a restriction because the period for buffering the input video in the large-capacity on-chip RAM should be taken into account. For details, refer to section 27.8, Usage Notes.

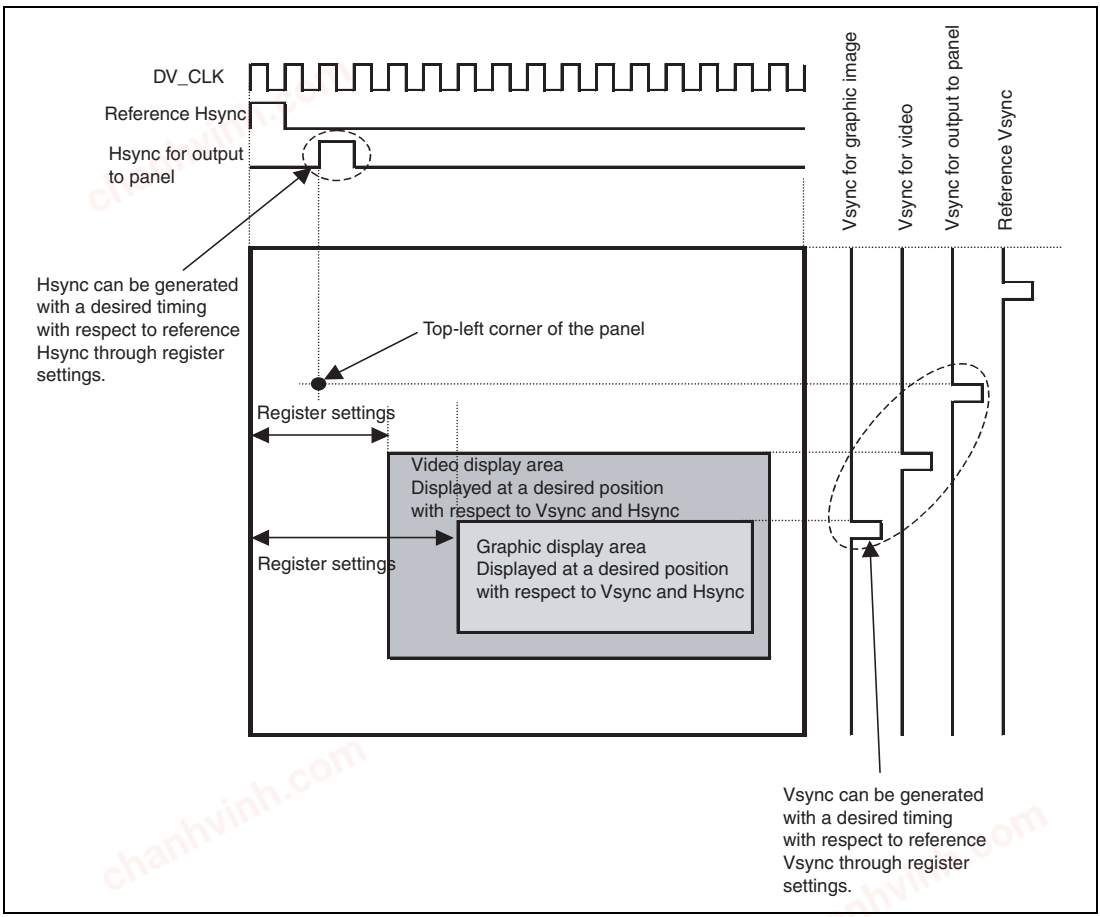


Figure 27.8 Video and Graphics Image Display Positions on Panel

chanhvinh.com

27.6.2 Video Recording Function

This module stores the captured video in the RGB565 format at a 1/2-field rate (NTSC: 30 fps; PAL: 25 fps) (video recording function) after contrast and brightness adjustment. For the capturing function, refer to section 27.6.1 (1), Capturing.

The destination address of recording is calculated according to the register settings of the base address, line offset, field offset, and number of fields to be stored. After data has been written for the specified number of fields, the memory is overwritten by new video from the first address. The current field number being written can be monitored through the status register. After one field of video data is written, an interrupt is sent to the CPU to indicate the completion of writing.

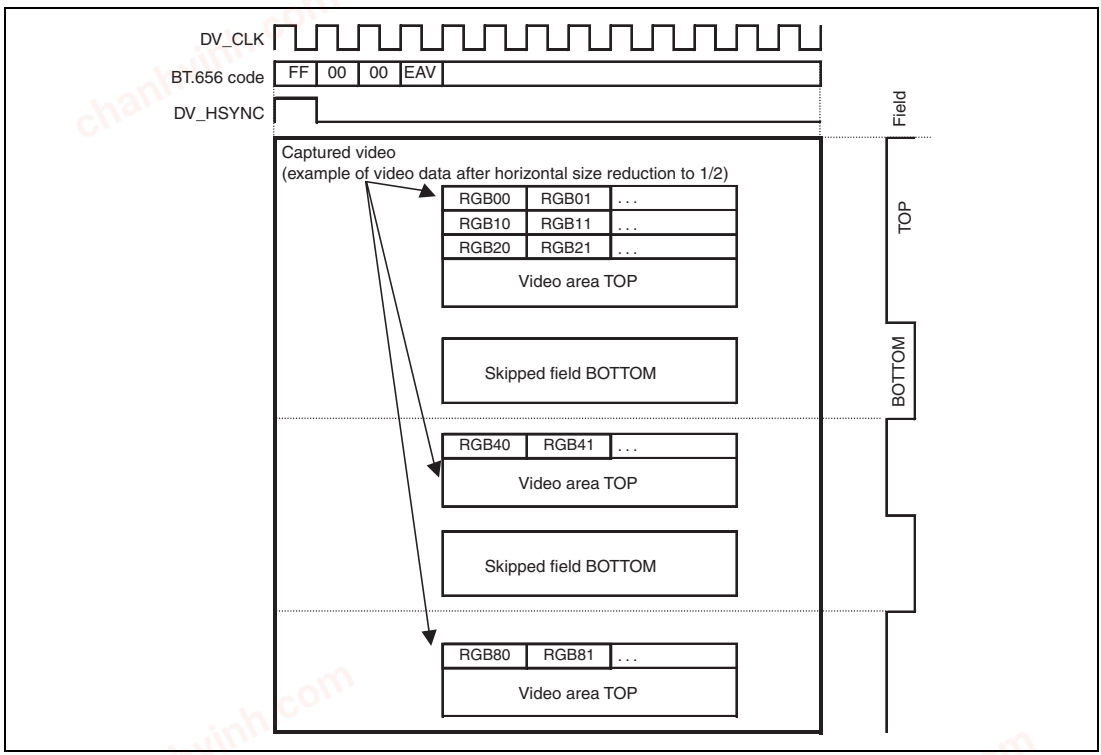


Figure 27.9 Correspondence between Video Data and Addresses

chanhvinh.com

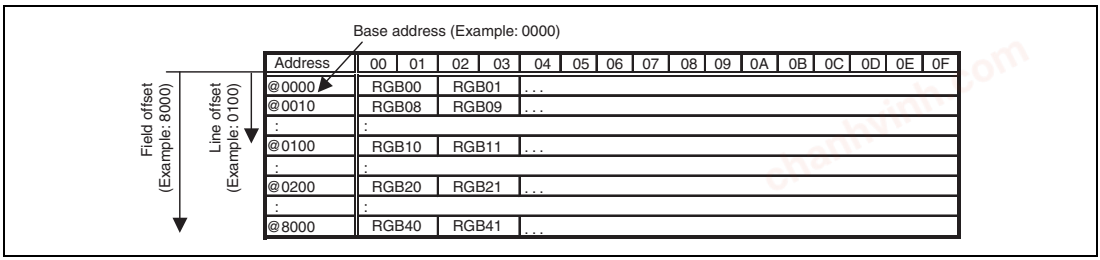


Figure 27.10 Schematic Diagram of Video Data Allocation on On-Chip Large-Capacity RAM or SDRAM

chanhvinh.com

27.6.3 Panel Control Signal Output Function

(1) Sync Signal Generation

Figure 27.11 shows an example of sync signal format that can be generated. The vertical sync signal can be output with a desired pulse width specified in 1-H units and the horizontal sync signal in 1-clock-cycle units.

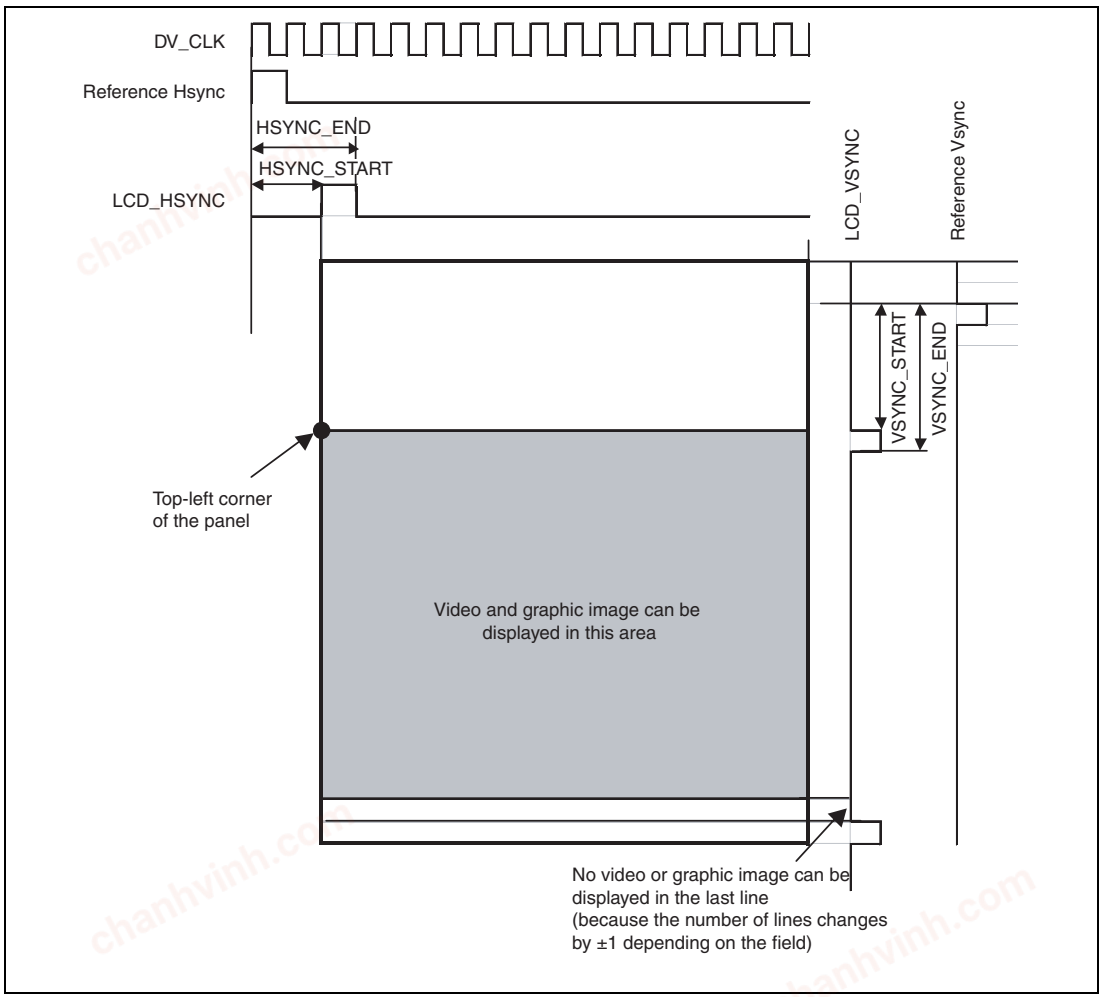


Figure 27.11 Example of Sync Signal Output

chanhvinh.com

(2) Data Enable Signal Generation

The data enable signal (LCD_DE) for output to the panel can be generated as a rectangular area with a desired size. In addition, a signal with one pulse width can be generated one cycle before valid data to indicate the start position of valid data.

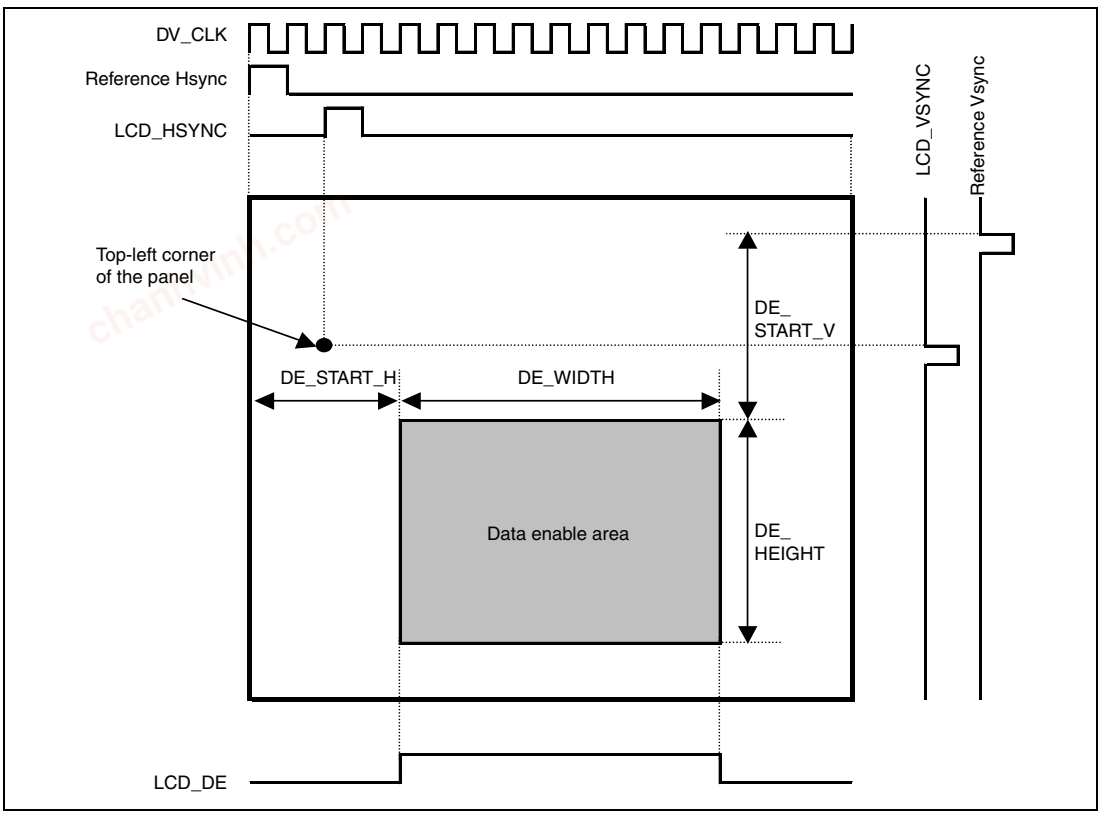


Figure 27.12 Timing of Data Enable Signal

chanhvinh.com

(3) AC Modulation Signal (Alternating Signal)

This output signal is toggled between high and low (H -> L-> H -> ...) every specified number of lines.

The interval is calculated as (set value) + 1 lines.

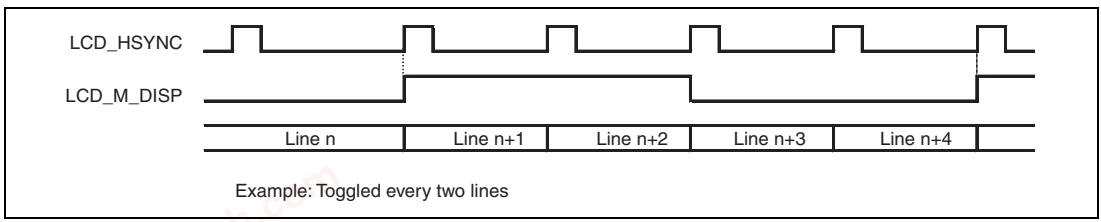


Figure 27.13 LCD_M_DISP Signal Description

(4) Sync Signal Output Timing

The sync signal output timing is shown below.

The vertical sync signal changes in synchronization with the rising edge (the falling edge when the output is invered) of the horizontal sync signal.

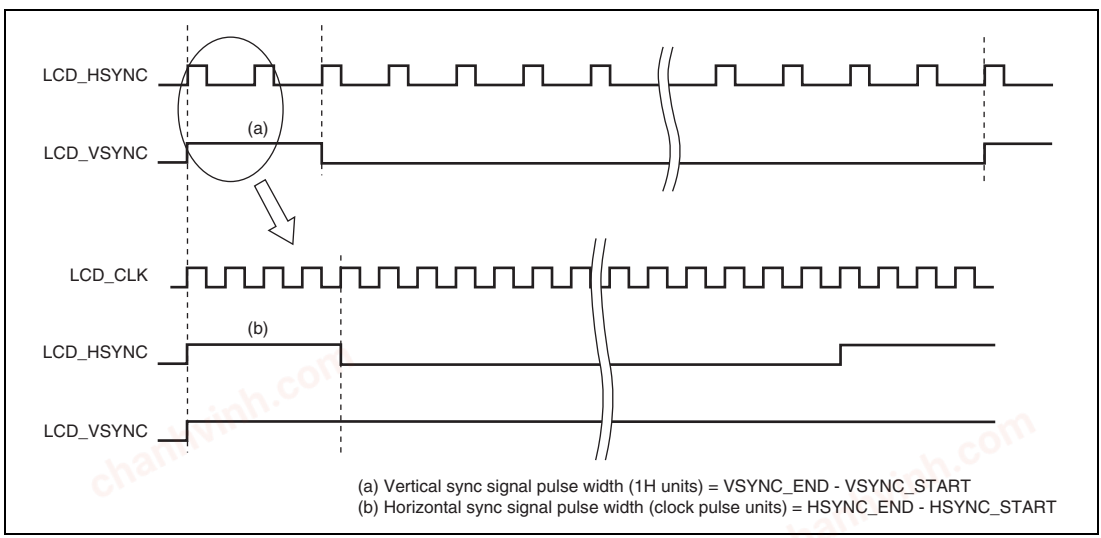


Figure 27.14 Sync Signal Output Timing

chanhvinh.com

27.7 Register Descriptions

Tables 27.7 to 27.10 show the register configuration.

Table 27.7 Register Configuration in Video Receiving and Supplying Blocks

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Video operating mode register	VIDEO_MODE	R/W	H'0000 0000	H'FFFF 2000	32, 16, 8
Video interrupt control register	VIDEO_INT_CNT	R/W	H'0000 0000	H'FFFF 2004	32, 16, 8
Video input timing control register	VIDEO_TIM_CNT	R/W	H'0000 0000	H'FFFF 2008	32, 16, 8
Valid video size register	VIDEO_SIZE	R/W	H'00F0 02D0	H'FFFF 2100	32, 16, 8
Vertical valid video start position register	VIDEO_VSTART	R/W	H'0010 0117	H'FFFF 2104	32, 16, 8
Horizontal valid video start position register	VIDEO_HSTART	R/W	H'0000 0114	H'FFFF 2108	32, 16, 8
Timing control register 1 for vertical sync signal for video	VIDEO_VSYNC_TIM1	R/W	H'0000 0000	H'FFFF 210C	32, 16, 8
Video storing field count register	VIDEO_SAVE_NUM	R/W	H'0000 0000	H'FFFF 2110	32, 16, 8
Video scaling and correction register	VIDEO_IMAGE_CNT	R/W	H'8080 0300	H'FFFF 2114	32, 16, 8
Video base address register	VIDEO_BASEADR	R/W	H'0000 0000	H'FFFF 2118	32, 16, 8
Video line offset register	VIDEO_LINE_OFFSET	R/W	H'0000 0000	H'FFFF 211C	32, 16, 8
Video field offset register	VIDEO_FIELD_OFFSET	R/W	H'0000 0000	H'FFFF 2120	32, 16, 8
Video line buffer count register	VIDEO_LINEBUFF_NUM	R/W	H'0000 0000	H'FFFF 2124	32, 16, 8
Video display and recording size register	VIDEO_DISP_SIZE	R/W	H'00F0 0168	H'FFFF 2128	32, 16, 8
Horizontal video display position register	VIDEO_DISP_HSTART	R/W	H'0000 0000	H'FFFF 212C	32, 16, 8

Note: While operations of the video receiving and supplying blocks are in progress, writing to the registers in video receiving and supplying blocks except the operation enable bits is prohibited.

chanhvinh.com

Table 27.8 Register Configuration in Graphics Block 1

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Graphics block control register	GRCMEN1	R/W	H'0000 0000	H'FFFF 2800	32, 16, 8
Bus control register	GRCBUSCNT1	R/W	H'0000 0000	H'FFFF 2804	32, 16, 8
Graphics block interrupt control register	GRCINTCNT1	R/W	H'0000 0000	H'FFFF 2808	32, 16, 8
Graphics image base address register	GROPSADR1	R/W	H'0000 0000	H'FFFF 2B08	32, 16, 8
Graphics image size register	GROPSWH1	R/W	H'0000 0000	H'FFFF 2B0C	32, 16, 8
Graphics image line offset register	GROPSOFST1	R/W	H'0000 0000	H'FFFF 2B10	32, 16, 8
Graphics image start position register	GROPDPHV1	R/W	H'0000 0000	H'FFFF 2B14	32, 16, 8
α control area size register	GROPEWH1	R/W	H'0000 0000	H'FFFF 2B18	32, 16, 8
α control area start position register	GROPEDPHV1	R/W	H'0000 0000	H'FFFF 2B1C	32, 16, 8
α control register	GROPEDPA1	R/W	H'FF00 0000	H'FFFF 2B20	32, 16, 8
Chroma-key control register	GROPCRKY0_1	R/W	H'0000 0000	H'FFFF 2B24	32, 16, 8
Chroma-key color register	GROPCRKY1_1	R/W	H'0000 0000	H'FFFF 2B28	32, 16, 8
Color register for outside of graphics image area	GROPBASERGB1	R/W	H'0000 0000	H'FFFF 2B2C	32, 16, 8

chanhvinh.com

Table 27.9 Register Configuration in Graphics Block 2

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Graphics block control register	GRCMEN2	R/W	H'0000 0000	H'FFFF 3000	32, 16, 8
Bus control register	GRCBUSCNT2	R/W	H'0000 0000	H'FFFF 3004	32, 16, 8
Graphics block interrupt control register	GRCINTCNT2	R/W	H'0000 0000	H'FFFF 3008	32, 16, 8
Graphics image base address register	GROPSADR2	R/W	H'0000 0000	H'FFFF 3308	32, 16, 8
Graphics image size register	GROPSWH2	R/W	H'0000 0000	H'FFFF 330C	32, 16, 8
Graphics image line offset register	GROPSOFST2	R/W	H'0000 0000	H'FFFF 3310	32, 16, 8
Graphics image start position register	GROPDPHV2	R/W	H'0000 0000	H'FFFF 3314	32, 16, 8
α control area size register	GROPEWH2	R/W	H'0000 0000	H'FFFF 3318	32, 16, 8
α control area start position register	GROPEDPHV2	R/W	H'0000 0000	H'FFFF 331C	32, 16, 8
α control register	GROPEDPA2	R/W	H'FF00 0000	H'FFFF 3320	32, 16, 8
Chroma-key control register	GROPCRKY0_2	R/W	H'0000 0000	H'FFFF 3324	32, 16, 8
Chroma-key color register	GROPCRKY1_2	R/W	H'0000 0000	H'FFFF 3328	32, 16, 8
Color register for outside of graphics image area	GROPBASERGB2	R/W	H'0000 0000	H'FFFF 332C	32, 16, 8

chanhvinh.com

Table 27.10 Register Configuration in Display Control Block

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
SG mode register	SGMODE	R/W	H'0000 0000	H'FFFF 3800	32, 16, 8
Interrupt output control register	SGINTCNT	R/W	H'0000 0000	H'FFFF 3804	32, 16, 8
Sync signal control register	SYNCNT	R/W	H'0000 0000	H'FFFF 3808	32, 16, 8
Panel clock select register	PANEL_CLKSEL	R/W	H'0000 0001	H'FFFF 380C	32, 16, 8
Sync signal size register	SYN_SIZE	R/W	H'020D 035A	H'FFFF 3900	32, 16, 8
Timing control register for vertical sync signal for output to panel	PANEL_VSYNC_TIM	R/W	H'0000 0001	H'FFFF 3904	32, 16, 8
Timing control register for horizontal sync signal for output to panel	PANEL_HSYNC_TIM	R/W	H'0000 000A	H'FFFF 3908	32, 16, 8
Timing control register 2 for vertical sync signal for video	VIDEO_VSYNC_TIM2	R/W	H'0000 0000	H'FFFF 390C	32, 16, 8
Timing control register for vertical sync signal for graphics image	GRA_VSYNC_TIM	R/W	H'0000 0000	H'FFFF 3910	32, 16, 8
AC modulation signal toggle line count	AC_LINE_NUM	R/W	H'0000 000C	H'FFFF 3914	32, 16, 8
DE area size register	DE_SIZE	R/W	H'0000 0000	H'FFFF 3920	32, 16, 8
DE area start position register	DE_START	R/W	H'0000 0000	H'FFFF 3924	32, 16, 8

chanhvinh.com

27.7.1 Video Operating Mode Register (VIDEO_MODE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	RGB565	INV_ CbCr	-	-	-	-	-	-	BURST_ MODE_ DISP	BURST_ MODE_ MAIN	-	-	ENDIAN _DISP	ENDIAN _MAIN
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/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
	-	SEL_ EXSYNC	SEL_ 656601	SEL_ 525625	-	-	-	-	-	-	-	VIDEO_ MODE	-	-	VIDEO_ DISP_ EXE	VIDEO_ MAIN_ EXE
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	R	R	R	R/W	R	R	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	RGB565	0	R/W	Specifies the method of conversion from RGB888 to RGB565. 0: Converted by the operation described in section 27.6.1 (4), Conversion from RGB888 to RGB565. 1: Lower bits are truncated.
28	INV_CbCr	0	R/W	Specifies Cb and Cr inversion. 0: Normal operation 1: Inversion
27 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21	BURST_MODE _DISP	0	R/W	Selects the mode of transfer through the IV2-BUS in the video supplying block. Reading out from the areas except the large-capacity on-chip RAM requires this bit to be set to 0. 0: 16-byte burst transfer 1: 128-byte burst transfer

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
20	BURST_MODE_MAIN	0	R/W	Selects the mode of transfer through the IV1-BUS in the video receiving block. Writing out to the areas except the large-capacity on-chip RAM requires this bit to be set to 0. 0: 16-byte burst transfer 1: 128-byte burst transfer
19, 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17	ENDIAN_DISP	0	R/W	Specifies the endian for the IV2-BUS in the video supplying block. 0: Big endian 1: Little endian
16	ENDIAN_MAIN	0	R/W	Specifies the endian for the IV1-BUS in the video supplying block. 0: Big endian 1: Little endian
15	—	All 0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	SEL_EXSYNC	0	R/W	Enables the external input sync signal. 0: Disabled 1: Enabled
13	SEL_656601	0	R/W	Specifies the format of the input video. 0: BT.656 input (be sure to set SEL_EXSYNC = 0) 1: BT.601 input (be sure to set SEL_EXSYNC = 1)
12	SEL_525625	0	R/W	Specifies the number of lines for the input video. 0: 525 lines (NTSC) 1: 625 lines (PAL)
11 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	VIDEO_MODE	0	R/W	Specifies the operating mode for the video receiving block. 0: Video recording mode (Be sure to set the EX_SYNC_MODE bit in the SGMODE register to 0) 1: Video display mode (Be sure to set the EX_SYNC_MODE bit in the SGMODE register to 1)
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	VIDEO_DISP_EXE	0	R/W	Enables the video supplying block operation. Setting to 1 starts video supply to the graphics blocks. The register setting is updated with the VSYNC timing. 0: Disabled 1: Enabled
0	VIDEO_MAIN_EXE	0	R/W	Enables the video receiving block operation. Setting to 1 starts video storing in the large-capacity on-chip RAM or SDRAM. 0: Disabled*[2] 1: Enabled

Notes: 1. Operation of the video receiving and supplying blocks proceeds after bit VIDEO_DISP_EXE or VIDEO_MAIN_EXE, respectively, is set to 1, and detection of VSYCN. Operation stops when the protocol for the internal bus is ended after the corresponding bit has been cleared to 0.

2. To disable the video receiving block operation, please follow the procedure shown in section 27.10.1, The Procedure of Disabling the Video Receiving Block Operation.

chanhvinh.com

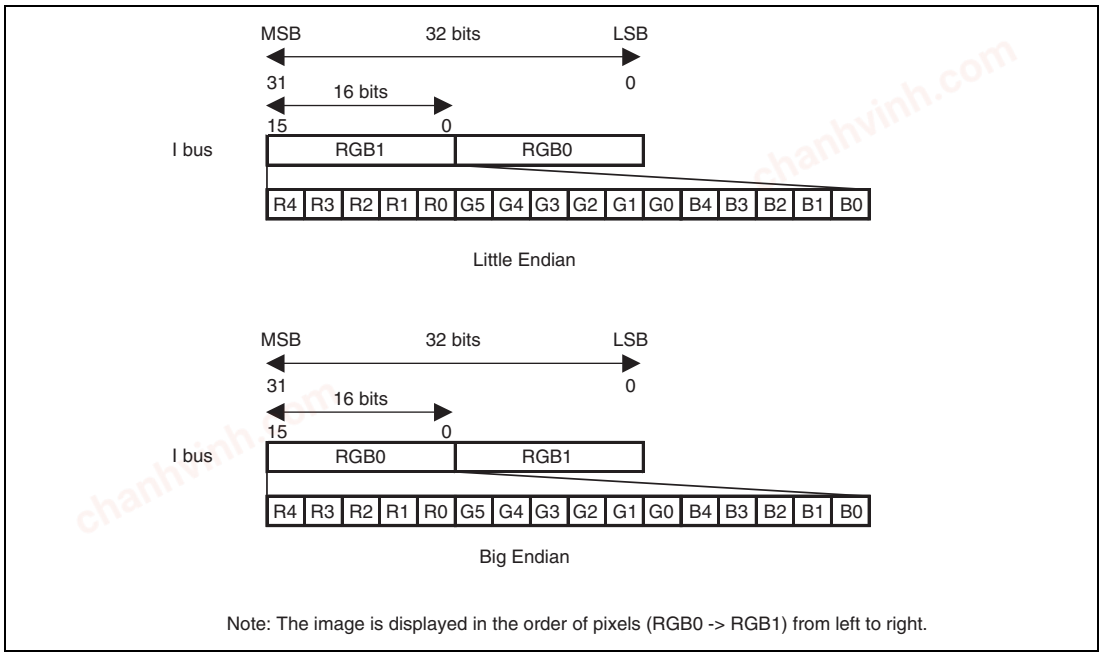


Figure 27.15 Description of Endian

chanhvinh.com

27.7.2 Video Interrupt Control Register (VIDEO_INT_CNT)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	INT_V_EN	-	-	-	INT_F_EN	-	-	-	INT_UF_EN	-	-	-	INT_OF_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/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
	-	-	-	V_PERIOD	-	-	-	F_END	-	-	-	UNDER_FLOW	-	-	-	OVER_FLOW
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*3	R	R	R	R/W*3	R	R	R	R/W*3	R	R	R	R/W*3

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_V_EN	0	R/W	Enables output of interrupts for indicating VSYNC cycle fluctuation detected in the input video. 0: Disabled 1: Enabled
27 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	INT_F_EN	0	R/W	Enables output of write completion interrupts. Set to 0 in video display mode. 0: Disabled 1: Enabled
23 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	INT_UF_EN	0	R/W	Enables output of underflow interrupts 0: Disabled 1: Enabled
19 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	INT_OF_EN	0	R/W	Enables output of overflow interrupts. 0: Disabled 1: Enabled
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	V_PERIOD	0	R/W*[3]	Indicates the VSYNC cycle fluctuation status in the input video. 0: The VSYNC cycles in the input video are constant. 1: The VSYNC cycles in the input video are not constant.
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	F_END	0	R/W*[3]	Indicates the status of writing one field of video data to the memory.*[1] 0: Writing is in progress. 1: Writing has been completed.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	UNDER_FLOW	0	R/W*[3]	Indicates the underflow status of the buffer used to read video data from the memory.*[1] 0: No underflow has occurred. 1: An underflow has occurred.
3 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	OVER_FLOW	0	R/W*3	Indicates the overflow status of the buffer used to write video data to the memory.*1*2 0: No overflow has occurred. 1: An overflow has occurred

Notes: 1. The status bits (bits 12, 8, 4, and 0) always operate, regardless of the settings of operation-enabling bits. After it has been set to 1, a status bit retains this value until it is cleared to 0.

2. The overflow flag is set to 1 in the cases shown below.

(a) (The number of operations of writing to the buffer in the video display controller 3) − (the number of operations of reading from the buffer) > amount of data for two lines

(b) (The number of operations of writing to the buffer in the video display controller 3) < (the number of operations of reading from the buffer)

Case (a) arises when there is a shortage of IV1-BUS bandwidth.

On the other hand, case (b) arises when less than the normal amount of data is written for one line, due to a malfunction of external input.

3. Only 0 can be written.

chanhvinh.com

27.7.3 Video Input Timing Control Register (VIDEO_TIM_CNT)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	FIELD_SKEW[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/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
	-	-	-	-	-	-	-	-	-	-	VSYNC_TYP	HSYNC_TYP	-	VSYNC_TIM	HSYNC_TIM	VIDEO_TIM
Initial value:	0	0	0	0	0	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	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	FIELD_SKEW [9:0]	All 0	R/W	These bits specify the timing of the VSYNC input signal for the BT.601 input. When the VSYNC signal is input within the DV_CLK cycles specified in these bits before or after the HSYNC signal, it is determined as the VSYNC signal for the TOP field.
15 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	VSYNC_TYP	0	R/W	Controls inversion of the DV_VSYNC input signal. 0: Not inverted 1: Input is inverted
4	HSYNC_TYP	0	R/W	Controls inversion of the DV_HSYNC input signal. 0: Not inverted 1: Input is inverted
3	—	All 0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	VSYNC_TIM	0	R/W	Specifies the timing of the DV_VSYNC input signal. 0: Latched at the rising edge of the DV_CLK. 1: Latched at the falling edge of the DV_CLK.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
1	HSYNC_TIM	0	R/W	Specifies the timing of the DV_HSYNC input signal. 0: Latched at the rising edge of the DV_CLK. 1: Latched at the falling edge of the DV_CLK.
0	VIDEO_TIM	0	R/W	Specifies the timing of the DV_DATA input signal. 0: Latched at the rising edge of the DV_CLK. 1: Latched at the falling edge of the DV_CLK.

chanhvinh.com

27.7.4 Valid Video Size Register (VIDEO_SIZE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	VIDEO_HEIGHT[8:0]								
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	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
	-	-	-	-	-	-	VIDEO_WIDTH[9:0]									
Initial value:	0	0	0	0	0	0	1	0	1	1	0	1	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 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24 to 16	VIDEO_ HEIGHT[8:0]	H'0F0	R/W	These bits specify the vertical size of the valid video in number of lines.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	VIDEO_ WIDTH[9:0]	H'2D0	R/W	These bits specify the horizontal size of the valid video in number of pixels. A value from 8 to 800 (pixels) can be specified. The lowest bit should always be set to 0.

chanhvinh.com

27.7.5 Vertical Valid Video Start Position Register (VIDEO_VSTART)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	VIDEO_VSTART_TOP[8:0]								
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/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
	-	-	-	-	-	-	-	VIDEO_VSTART_BTM[8:0]								
Initial value:	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1	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 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24 to 16	VIDEO_VSTART_TOP[8:0]	H'010	R/W	These bits specify in number of lines the vertical start position of the valid video in the TOP field. Setting to H'000 is prohibited.
15 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8 to 0	VIDEO_VSTART_BTM[8:0]	H'117	R/W	These bits specify in number of lines the vertical start position of the valid video in the BOTTOM field.

chanhvinh.com

27.7.6 Horizontal Valid Video Start Position Register (VIDEO_HSTART)

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
	-	-	-	-	-	-	-	VIDEO_HSTART[8:0]								
Initial value:	0	0	0	0	0	0	0	1	0	0	0	1	0	1	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	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 to 0	VIDEO_ HSTART[8:0]	H'114	R/W	These bits specify in number of DV_CLK cycles the horizontal start position of the valid video in the field.

Note: Capture does not occur when the right edge of video has been cut.

chanhvinh.com

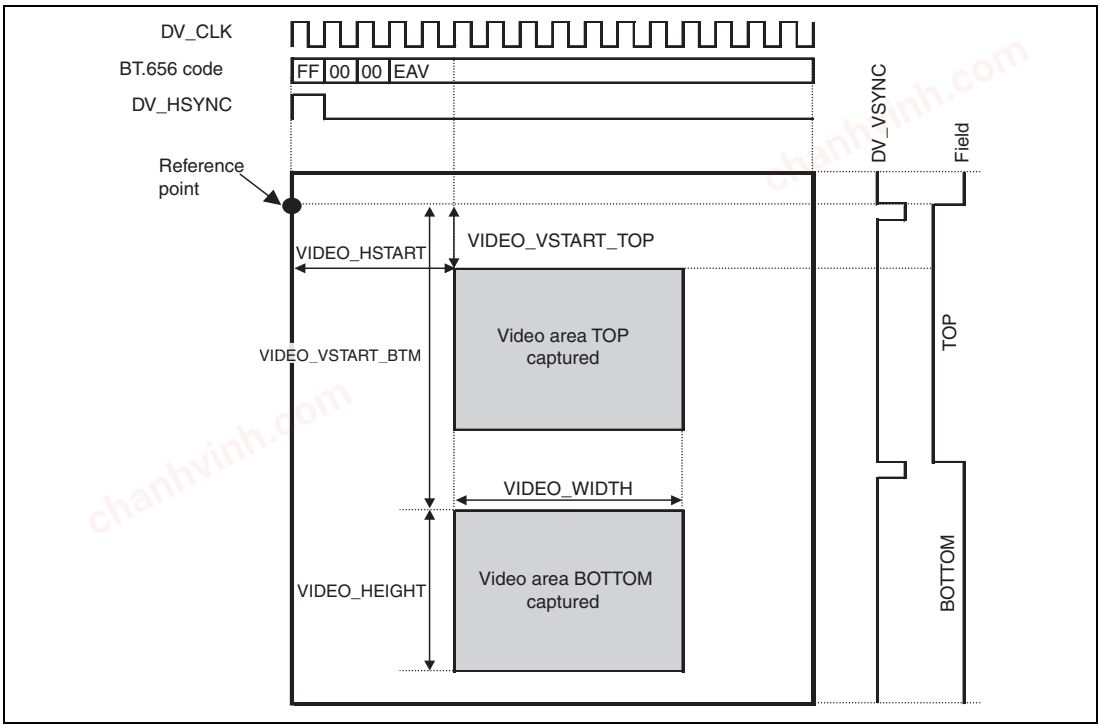


Figure 27.16 Concept of Capturing Function

chanhvinh.com

27.7.7 Timing Control Register 1 for Vertical Sync Signal for Video (VIDEO_VSYNC_TIM1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	VIDEO_VSYNC_START1_TOP[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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	VIDEO_VSYNC_START1_BTM[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 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25 to 16	VIDEO_VSYNC_START1_TOP [9:0]	H'000	R/W	These bits specify the reference Vsync position in the TOP field in number of lines.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	VIDEO_VSYNC_START1_BTM[9:0]	H'000	R/W	These bits specify the reference Vsync position in the BOTTOM field in number of lines.

chanhvinh.com

27.7.8 Video Storing Field Count Register (VIDEO_SAVE_NUM)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	FIELD_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	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	FIELD_SAVE_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 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25 to 16	FIELD_NUM [9:0]	H'000	R	These bits indicate the field number of which recording processing has been completed in the video recording mode. These bits are cleared to H'000 when the VIDEO_MAIN_EXE bit in the VIDEO_MODE register is cleared to 0.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	FIELD_SAVE_ NUM [9:0]	H'000	R/W	These bits specify the number of fields to be stored. Set these bits to H'000 in the video display mode (when the VIDEO_MODE bit in the VIDEO_MODE register is set to 1). H'000: One field is stored. H'001: Two fields are stored. H'03B: 60 fields are stored. H'3FF: 1024 fields are stored.

Note: After data has been written for the specified number of fields, the memory is overwritten by new video data from the first address.

Video data is stored at a 1/2-field rate (NTSC: 30 fps; PAL: 25 fps).

When the VIDEO_MAIN_EXE bit in the VIDEO_MODE register is cleared to 0, the address calculation result is initialized and data is written to the initial field storing area.

chanhvinh.com

27.7.9 Video Scaling and Correction Register (VIDEO_IMAGE_CNT)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Contrast[7:0]								Bright[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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	CLIP_Y	CLIP_C	-	SUB_SCALE_V	SCALE_V[1:0]		-	SUB_SCALE_H	SCALE_H[1:0]	
Initial value:	0	0	0	0	0	0	1	1	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	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	Contrast[7:0]	H'80	R/W	These bits specify the contrast of luminance. Contrast = this value / 128 Y0 = Luminance of captured video input Y1 = Contrast × (Y0 – 16)
23 to 16	Bright[7:0]	H'80	R/W	These bits specify the brightness of luminance. BRIGHT = this value – 128 Y2 = Y1 + BRIGHT
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	CLIP_Y	1	R/W	Enables luminance clipping. 0: Disabled 1: Enabled (235 or a greater value is clipped to 235, and 16 or a smaller value is clipped to 16.)
8	CLIP_C	1	R/W	Enables chrominance clipping. 0: Disabled 1: Enabled (240 or a greater value is clipped to 240, and 16 or a smaller value is clipped to 16.)
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	SUB_SCALE_V	0	R/W	Specifies further scaling of video by multiplying the SCALE_V scaling rate by 6/7. 0: ×1 1: ×6/7
5, 4	SCALE_V[1:0]	00	R/W	These bits specify the vertical scaling rate. Be sure to set to 00 in the video recording mode. 00: ×1/2 01: ×1/3 10: ×1/4 11: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	SUB_SCALE_H	0	R/W	Specifies further scaling of video by multiplying the SCALE_H scaling rate by 6/7. 0: ×1 1: ×6/7
1, 0	SCALE_H[1:0]	00	R/W	These bits specify the horizontal scaling rate. 00: ×1/2 01: ×1/3 10: ×2/3 11: ×1/4

Note: The width of a valid image after horizontal scaling is always an even value.

chanhvinh.com

27.7.10 Video Base Address Register (VIDEO_BASEADR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VIDEO_BASEADR[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
	VIDEO_BASEADR[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	VIDEO_ BASEADR [31:0]	H'00000000	R/W	These bits specify the base address of the destination where video data is to be written to. According to the BURST_MODE_MAIN bit setting in the VIDEO_MODE register, the lower bits should be set as follows. In 16-byte burst transfer: The lower four bits should always be 0000. In 128-byte burst transfer: The lower seven bits should always be 000_0000.

chanhvinh.com

27.7.11 Video Line Offset Register (VIDEO_LINE_OFFSET)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VIDEO_LINE_OFFSET[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
	VIDEO_LINE_OFFSET[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	VIDEO_LINE_OFFSET[31:0]	H'00000000	R/W	These bits specify the line offset. According to the BURST_MODE_MAIN bit setting in the VIDEO_MODE register, the lower bits should be set as follows. In 16-byte burst transfer: The lower four bits should always be 0000. In 128-byte burst transfer: The lower seven bits should always be 000_0000.

chanhvinh.com

27.7.12 Video Field Offset Register (VIDEO_FIELD_OFFSET)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VIDEO_FIELD_OFFSET[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
	VIDEO_FIELD_OFFSET[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	VIDEO_FIELD_OFFSET [31:0]	H'00000000	R/W	These bits specify the field offset. This setting is valid in the video recording mode. The lower four bits should always be 0000.

chanhvinh.com

27.7.13 Video Line Buffer Count Register (VIDEO_LINEBUFF_NUM)

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
	-	-	-	-	-	-	-	VIDEO_LINEBUFF_NUM[8:0]								
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	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 to 0	VIDEO_ LINEBUFF_ NUM[8:0]	H'000	R/W	These bits specify how many lines of buffer area are used in the video display mode. 0: One line of buffer area is used. 1: Two lines of buffer area are used. : 1FF: 512 lines of buffer area are used.

chanhvinh.com

27.7.14 Video Display and Recording Size Register (VIDEO_DISP_SIZE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	VIDEO_DISP_HEIGHT [8:0]								
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	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
	-	-	-	-	-	-	VIDEO_DISP_WIDTH [9:0]									
Initial value:	0	0	0	0	0	0	0	1	0	1	1	0	1	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 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24 to 16	VIDEO_DISP_ HEIGHT [8:0]	H'0F0	R/W	These bits specify in number of lines the vertical size of the video data to be read in the video display mode. In the video recording mode, set to 240 lines for NTSC or 288 lines for PAL.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	VIDEO_DISP_ WIDTH [9:0]	H'168	R/W	These bits specify in number of pixels the horizontal size of the video data to be read in the video display mode.* The lowest bit should always be 0.

Note: * Set the setting values of the VIDEO_DISP_WIDTH field to number of vertical pixels in captured video or less. The underflow flag is set to 1 when the setting values are more than number of vertical pixels.

chanhvinh.com

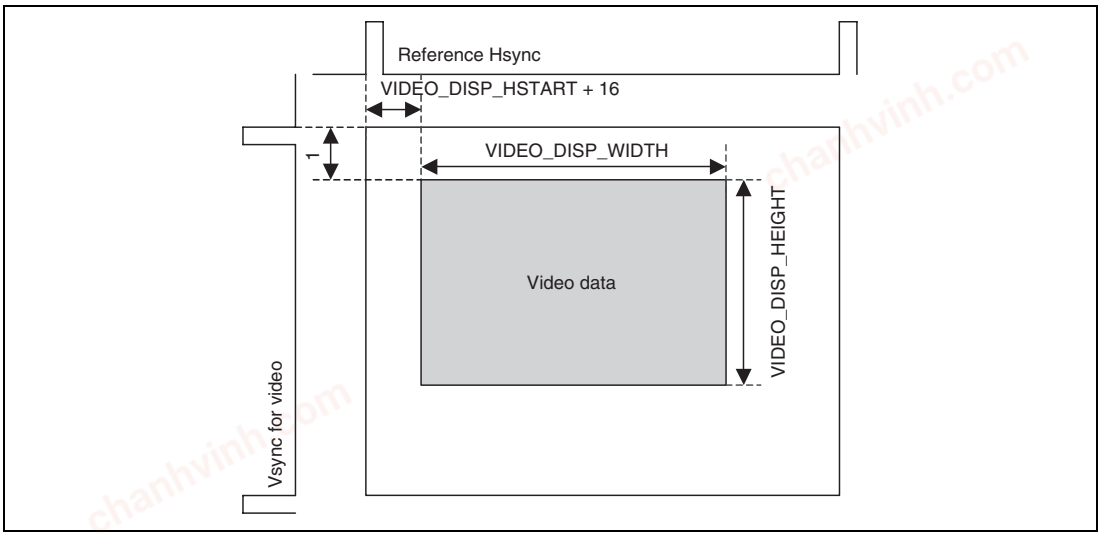


Figure 27.17 Settings for Video Data Reading (from Memory)

27.7.15 Horizontal Video Display Position Register (VIDEO_DISP_HSTART)

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
	-	-	-	-	-	-	VIDEO_DISP_HSTART[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 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	VIDEO_DISP_HSTART [9:0]	H'000	R/W	These bits specify in number of pixels the horizontal start position of the video data to be read in the video display mode.

chanhvinh.com

27.7.16 Graphics Block Control Registers (GRCMEN1 and GRCMEN2)

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

Bit	Bit Name	Initial Value	R/W	Description
31	WE	0	R/W	Enables register value transfer. Writing 1 to this bit transfers the values of registers GRCMEN to GROPEDPHV (except GRCINTCNT) and GROPBASERGB in synchronization with Vsync. After register transfer is competed, this bit is cleared to 0.
30 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	DEN	0	R/W	Enables display of the current layer (graphics image 1 for GRCMEN1 or graphics image 2 for GRCMEN2). 0: Disabled 1: Enabled
0	VEN	0	R/W	Enables display of the lower layer (video input for GRCMEN1 or overlaid graphics image 1 and video input for GRCMEN2). 0: Disabled 1: Enabled

chanhvinh.com

Table 27.11 Functions of Display Enable Bits

DEN	VEN	Operation	Output	Control
0	0	Does not read image data from memory or process lower-layer display.	Outputs the color specified in GROPBASERGB over the entire screen (negates the enable signal output).	
0	1	Does not read image data from memory but processes lower-layer display.	Outputs only the lower layer (outputs the lower layer enable signal).	Displays only the lower layer.
1	0	Reads image data from memory but does not process lower-layer display.	Outputs only the current layer (outputs the current layer enable signal).	Displays only the current layer.
1	1	Reads image data from memory and processes lower-layer display.	Performs the specified processing for the current layer and lower layer and displays them (logically ORs the current layer and lower layer enable signals and outputs the result).	Displays the current and lower layers.

Note: When the α control area (specified by GROPEWH and GROPEDPHV) is larger than the graphics image area (specified by GROPSWH and GROPDPHV), only the lower layer is displayed.

These bits should be set for each layer. When VEN = 0 in the upper layer, the graphics in the lower layer are not output.

chanhvinh.com

27.7.17 Bus Control Registers (GRCBUSCNT1 and GRCBUSCNT2)

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
	-	-	-	-	-	-	-	BURST_ MODE	-	-	-	BUS_ FORMAT	-	-	-	ENDIAN
Initial value:	0	0	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	BURST_ MODE	0	R/W	Selects the mode of transfer through the I bus (GRCBUSCNT1: IV3-BUS; GRCBUSCNT2: IV4-BUS). Reading out from the areas except the large-capacity on-chip RAM requires this bit to be set to 0. 0: 16-byte burst transfer 1: 128-byte burst transfer
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	BUS_ FORMAT	0	R/W	Specifies the data format for the I bus (GRCBUSCNT1: IV3-BUS; GRCBUSCNT2: IV4-BUS). 0: RGB 565 format 1: αRGB 4444 format
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ENDIAN	0	R/W	Specifies the endian for the I bus (GRCBUSCNT1: IV3-BUS; GRCBUSCNT2: IV4-BUS). 0: Big endian 1: Little endian

chanhvinh.com

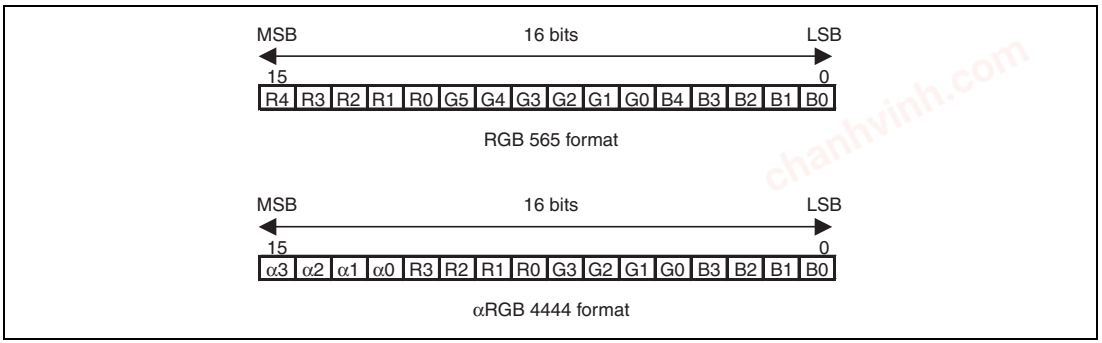


Figure 27.18 Bus Format

Table 27.12 α Value and Blending Ratio

α Value (Decimal)	Current Layer	Lower Layer
15	15/15	0/15
14	14/15	1/15
13	13/15	2/15
:	:	:
2	2/15	13/15
1	1/15	14/15
0	0/15	16/15

chanhvinh.com

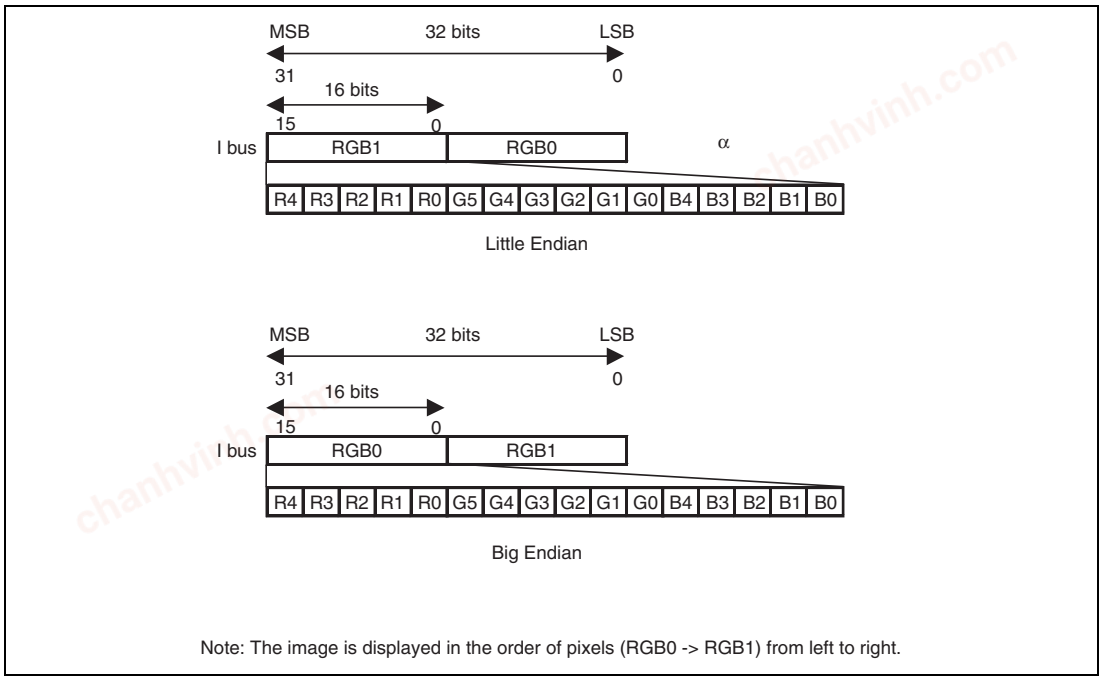


Figure 27.19 Descriptions of Endian

chanhvinh.com

27.7.18 Graphics Block Interrupt Control Registers (GRCINTCNT1 and GRCINTCNT2)

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

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

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	INT_UF_EN	0	R/W	Enables output of underflow interrupts. 0: Disabled 1: Enabled
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	UNDER_FLOW	0	R/W*2	Indicates the underflow status of the buffer used to read graphics images from the memory.*1 0: No underflow has occurred. 1: An underflow has occurred.

Notes: 1. The status bit (bit 0) always operates regardless of the operation enabling bit settings. After being set to 1, the status bit remains at 1 until cleared 0.
2. Only 0 can be written.

chanhvinh.com

27.7.19 Graphics Image Base Address Registers (GROPSADR1 and GROPSADR2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	GROPSADR[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/W	R/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
	GROPSADR[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 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28 to 0	GROPSADR [28:0]	H'00000000	R/W	These bits specify the base address from which a graphics image is to be read. The lowest bit should always be 0.

Note: This module processes 16-bit RGB data; it cannot handle data located beyond a 2-byte alignment boundary.

27.7.20 Graphics Image Size Registers (GROPSWH1 and GROPSWH2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	GROPSH[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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	GROPSW[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 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25 to 16	GROPSH [9:0]	H'000	R/W	These bits specify the height of the graphics image area in number of lines.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	GROPSW [9:0]	H'000	R/W	These bits specify the width of the graphics image area in number of pixels (Max. 640).

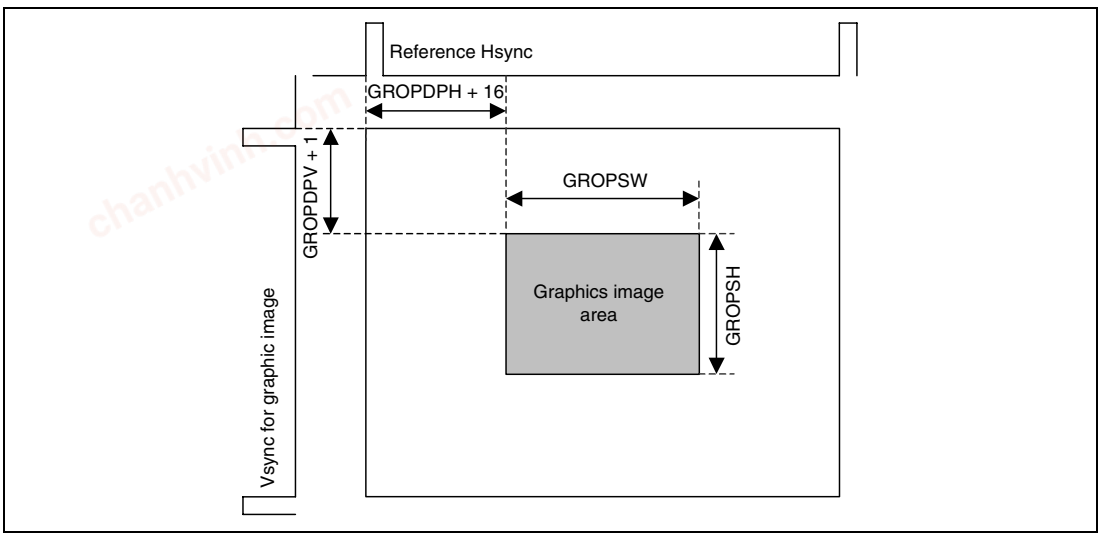


Figure 27.20 Graphics Image Area Settings (Reading from Memory)

A graphics image area should be specified within the following range; otherwise, correct operation is not guaranteed.

(Pixels for 1H) > GROPSW (width) + GROPDPH (horizontal display start position) + (16 pixels)

(Lines for 1 frame) > GROPSH (height) + GROPDPV (vertical display start position) + (1 line)

chanhvinh.com

27.7.21 Graphics Image Line Offset Registers (GROPSOFST1 and GROPSOFST2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	GROPSOFST[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/W	R/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
	GROPSOFST[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 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28 to 0	GROPSOFST [28:0]	H'00000000	R/W	These bits specify the line offset for the graphics image. In 16-byte burst transfer: The lower four bits should always be 0000. In 128-byte burst transfer: The lower seven bits should always be 000_0000.

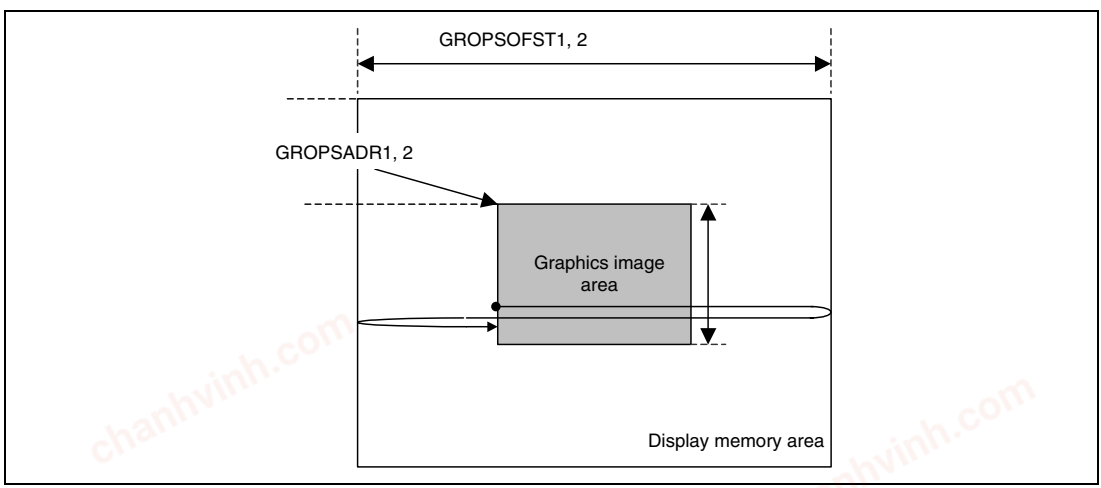


Figure 27.21 Graphics Image Memory Area Settings

The start (left side) address of line n is obtained by adding the base address register value (GROPSADR1 or GROPSADR2) and the line offset (GROPSOFST1 or GROPSOFST2) × n.

chanhvinh.com

27.7.22 Graphics Image Start Position Registers (GROPDPHV1 and GROPDPHV2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	GROPDPV[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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	GROPDPH[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 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25 to 16	GROPDPV [9:0]	H'000	R/W	These bits specify the vertical display start position of the graphics image area in number of lines.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	GROPDPH [9:0]	H'000	R/W	These bits specify the horizontal display start position of the graphics image area in number of pixels.

Note: The display start address is offset as follows (see figure 27.19).
Vertical offset: (GROPDPV value) + 1 line
Horizontal offset: (GROPDPH value) + 16 pixels

chanhvinh.com

27.7.23 α Control Area Size Registers (GROPEWH1 and GROPEWH2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	GROPEH[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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	GROPEW[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 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25 to 16	GROPEH [9:0]	H'000	R/W	These bits specify the height of the α control area in number of lines.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	GROPEW [9:0]	H'000	R/W	These bits specify the width of the α control area in number of pixels.

Note: This register specifies the size of the α control area (rectangle). See figure 27.21.

chanhvinh.com

27.7.24 α Control Area Start Position Registers (GROPEDPHV1 and GROPEDPHV2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	GROPEDPV[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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	GROPEDPH[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 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25 to 16	GROPEDPV [9:0]	H'000	R/W	These bits specify the vertical start position of the α control area in number of lines.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	GROPEDPH [9:0]	H'000	R/W	These bits specify the horizontal start position of the α control area in number of pixels.

chanhvinh.com

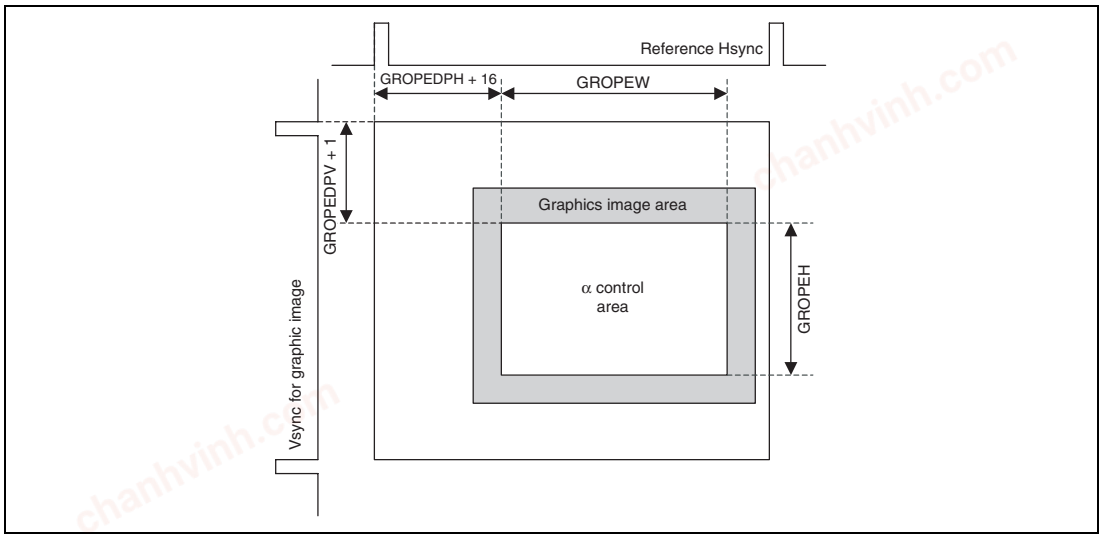


Figure 27.22 α Control Area Settings

27.7.25 α Control Registers (GROPEDPA1 and GROPEDPA2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DEFA[7:0]								ACOEF[7:0]							
Initial value:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	RR/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
	ARATE[7:0]								WE	-	-	AST	-	AMOD[1:0]		AEN
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	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	DEFA[7:0]	H'FF	R/W	These bits specify the initial α value.
23 to 16	ACOEF[7:0]	H'00	R/W	These bits specify a coefficient for α value calculation. This value is added to or subtracted from the DEFA value.
15 to 8	ARATE[7:0]	H'00	R/W	These bits specify the frame rate of α control. (The reference Vsync is used as the unit of counting.) 0: addition and subtraction every frame 1: addition and subtraction every two frames : FF: addition and subtraction every 256 frames

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
7	WE	0	R/W	Enables transfer of the α control register values. Writing 1 to this bit transfers the register values (GROPEDPA register to GROPCRKY1 register) in synchronization with Vsync. After register transfer is competed, this bit is cleared to 0. 0: Disabled 1: Enabled
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	AST	0	R	α Blending Status Flag 0: Addition or subtraction has been completed. 1: Addition or subtraction is in progress
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2, 1	AMOD[1:0]	00	R/W	These bits specify the α processing mode. 00: Initial α value (the value does not change) 01: α value addition 10: α value subtraction 11: Setting prohibited
0	AEN	0	R/W	Enables or disables α control. 0: α control is disabled.* 1: α control is enabled. Note: * When 0 is specified, either the replaced α value after chroma-keying or the dot α function is valid. If neither α function is used, the α control processing becomes the same as when α = 255.

chanhvinh.com

When AEN = 1 and WE = 1, the α value is loaded in the internal circuits in synchronization with Vsync.

If AMOD[1:0] = [0 0], the α value specified in DEFA is applied to the α control area.

If AMOD[1:0] = [0 1], the ACOEF value is added to the DEFA value according to the frame rate (ARATE) and the result is applied to the α control area as the α value. When the α value becomes 255 or larger, processing stops (fade-in).

If AMOD[1:0] = [1 0], the ACOEF value is subtracted from the DEFA value according to the frame rate (ARATE) and the result is applied to the α control area as the α value. When the α value becomes 0 or smaller, processing stops (fade-out).

Table 27.13 α Value and Blending Ratio

α Value (Decimal)	Current Layer	Lower Layer
255	255/255	0/255
254	254/255	1/255
253	253/255	2/255
252	252/255	3/255
:	:	:
2	2/255	253/255
1	1/255	254/255
0	0/255	255/255

chanhvinh.com

27.7.26 Chroma-Key Control Registers (GROPCRKY0_1 and GROPCRKY0_2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CKEN
Initial value:	0	0	0	0	0	0	0	0	0	0	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
	CROMAKR[4:0]					CROMAKG[5:0]						CROMAKB[4:0]				
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 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	CKEN	0	R/W	Enables or disables chroma-key processing. 0: Chroma-key processing is disabled. 1: Chroma-key processing is enabled
15 to 11	CROMAKR [4:0]	00000	R/W	These bits specify chroma-key target color R.
10 to 5	CROMAKG [5:0]	000000	R/W	These bits specify chroma-key target color G.
4 to 0	CROMAKB [4:0]	00000	R/W	These bits specify chroma-key target color B.

Note: When the bus format is αRGB444, only the CROMAKR[3:0], CROMAKG[3:0], and CROMAKB[3:0] bits are valid.

When WE =1 in GROPEDPA, the register setting is loaded in the internal circuits in synchronization with Vsync.

While the chroma-key processing is enabled, if the graphics data values (RGB16 format) of a pixel all match the CROMAKR[4:0], CROMAKG[5:0], and CROMAKB[4:0] settings, the pixel color is replaced with the color (RGB16 format) specified in the chroma-key color register (GROPCRKY1) and α processing specified through the ALPHA[7:0] bits is applied.

Chroma-keying thus enables characters or a cursor to be displayed on the lower layer.

chanhvinh.com

27.7.27 Chroma-Key Color Registers (GROPCRKY1_1 and GROPCRKY1_2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	ALPHA[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
	R[4:0]					G[5:0]						B[4:0]				
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	ALPHA[7:0]	H'00	R/W	These bits specify the α value after replacement.
15 to 11	R[4:0]	00000	R/W	These bits specify the R value after replacement.
10 to 5	G[5:0]	000000	R/W	These bits specify the G value after replacement.
4 to 0	B[4:0]	00000	R/W	These bits specify the B value after replacement.

Note: When the bus format is αRGB444, only the R[3:0], G[3:0], and B[3:0] bits are valid.

Each register specifies a set of color information to replace the color that matches the chroma-key target RGB values.

α calculation is done as follows.

Output R = R (current layer) × α + R (lower layer) × (1 – α)

Output G = G (current layer) × α + G (lower layer) × (1 – α)

Output B = B (current layer) × α + B (lower layer) × (1 – α)

chanhvinh.com

27.7.28 Color Registers for Outside of Graphics Image Area (GROPBASERGB1 and GROPBASERGB2)

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
	BASE_R[4:0]					BASE_G[5:0]						BASE_B[4:0]				
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 11	BASE_R [4:0]	00000	R/W	These bits specify the R value for outside of the graphics image area.
10 to 5	BASE_G [5:0]	000000	R/W	These bits specify the G value for outside of the graphics image area.
4 to 0	BASE_B [4:0]	00000	R/W	These bits specify the B value for outside of the graphics image area.

Note: This setting is valid only when VEN = 0 in GRCMEN.

chanhvinh.com

27.7.29 SG Mode Register (SGMODE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	EX_SYNC_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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RGB565
Initial value:	0	0	0	0	0	0	0	0	0	0	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	EX_SYNC_MODE	0	R/W	Selects the sync signal mode. 0: Free-running mode 1: Synchronized with the Vsync of the video input
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	RGB565	0	R/W	Specifies the method for conversion from RGB888 to RGB565. 0: Calculated as described in section 27.6.1 (4), Conversion from RGB888 to RGB565. 1: Lower bits are truncated.

chanhvinh.com

27.7.30 Interrupt Output Control Register (SGINTCNT)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	INT_LINE_NUM[9:0]										-	-	-	INT_LINE_EN
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/W

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

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 20	INT_LINE_NUM[9:0]	All 0	R/W	These bits specify the line number for which a line interrupt is to be output.
19 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	INT_LINE_EN	0	R/W	Enables output of line interrupts. 0: Disabled 1: Enabled
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	LINE_STATUS	0	R/W*2	Indicates the line interrupt status.*1 0: No interrupt has occurred. 1: An interrupt has occurred.

Notes: 1. The status bit (bit 0) always operates regardless of the operation enabling bit setting. After being set to 1, the status bit remains at 1 until cleared 0.
2. Only 0 can be written.

chanhvinh.com

27.7.31 Sync Signal Control Register (SYNCNT)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	RGB_TIM	-	-	-	-	VSYNC_TIM	HSYNC_TIM	DE_TIM	M_DISP_TIM
Initial value:	0	0	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
	-	-	-	-	-	-	-	-	-	-	-	-	VSYNC_TYPE	HSYNC_TYPE	DE_TYPE	M_DISP_TYPE
Initial value:	0	0	0	0	0	0	0	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 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	RGB_TIM	0	R/W	Specifies the LCD_DATA output timing. 0: Output at the rising edge of the panel clock 1: Output at the falling edge of the panel clock
23 to 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
19	VSYNC_TIM	0	R/W	Specifies the LCD_VSYNC output timing. 0: Output at the rising edge of the panel clock 1: Output at the falling edge of the panel clock
18	HSYNC_TIM	0	R/W	Specifies the LCD_HSYNC output timing. 0: Output at the rising edge of the panel clock 1: Output at the falling edge of the panel clock
17	DE_TIM	0	R/W	Specifies the LCD_DE output timing. 0: Output at the rising edge of the panel clock 1: Output at the falling edge of the panel clock
16	M_DISP_TIM	0	R/W	Specifies the LCD_M_DISP output timing. 0: Output at the rising edge of the panel clock 1: Output at the falling edge of the panel clock

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	VSYNC_TYPE	0	R/W	Controls whether to invert LCD_VSYNC. 0: Not inverted. 1: Output is inverted.
2	HSYNC_TYPE	0	R/W	Controls whether to invert LCD_HSYNC. 0: Not inverted. 1: Output is inverted.
1	DE_TYPE	0	R/W	Controls whether to invert LCD_DE. 0: Not inverted. 1: Output is inverted.
0	M_DISP_TYPE	0	R/W	Controls whether to invert LCD_M_DISP. 0: Not inverted. 1: Output is inverted.

chanhvinh.com

27.7.32 Panel Clock Select Register (PANEL_CLKSEL)

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
	-	-	ICKSEL	ICKEN	-	-	-	-	-	-	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	R	R	R/W	R/W	R/W	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	ICKSEL	0	R/W	Selects the source of the panel clock. 0: External clock is selected (LCD_EXTCLK). 1: Bus clock is selected (Bϕ).
12	ICKEN	0	R/W	Enables or disables the operation of the blocks using the panel clock in this module and output of the panel clock. 0: Operation of the blocks using the panel clock is disabled. 1: Operation of the blocks using the panel clock is enabled. Note: Be sure to clear this bit to 0 before modifying the ICKSEL or DCDR bit.
11 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5 to 0	DCDR[5:0]	000001	R/W	Specifies the division ratio of the input clock frequency. For details, see table 27.14. Note: The settings not shown in table 27.14 are prohibited.

This module can select the bus clock or external clock as the source of the panel clock. It also has a frequency divider providing a division ratio from 1/1 to 1/32.

chanhvinh.com

Table 27.14 Input/Output Clock Frequency and Division Ratio

DCDR[5:0]	Clock Frequency Division Ratio	Input/Output Clock Frequency (MHz)		
		36.000	**48.000*2**	**72.000*2**
000001*1	1/1	36.000	48.000	72.000
000010	1/2	18.000	24.000	36.000
000011	1/3	12.000	16.000	24.000
000100	1/4	9.000	12.000	18.000
000110	1/6	6.000	8.000	12.000
001000	1/8	4.500	6.000	9.000
001100	1/12	3.000	4.000	6.000
010000	1/16	2.250	3.000	4.500
011000	1/24	1.500	2.000	3.000
100000	1/32	1.125	1.500	2.250

Notes: 1. Setting prohibited when the bus clock (Bϕ) is selected for the source of the panel clock.
2. When the bus clock (Bϕ) is selected for the source of the panel clock.

chanhvinh.com

27.7.33 Sync Signal Size Register (SYN_SIZE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	SYN_HEIGHT[9:0]									
Initial value:	0	0	0	0	0	0	1	0	0	0	0	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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	SYN_WIDTH[10:0]										
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/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 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25 to 16	SYN_HEIGHT [9:0]	H'20D	R/W	These bits specify the height including the vertical blanking interval in number of lines. Initial value: H'20D = 525 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	SYN_WIDTH [10:0]	H'35A	R/W	These bits specify the width including the horizontal blanking interval in number of panel clock cycles. Initial value: H'35A = 858 pixels

chanhvinh.com

27.7.34 Vertical Sync Signal Timing Control Register (PANEL_VSYNC_TIM)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	VSYNC_START[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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	VSYNC_END[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 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25 to 16	VSYNC_START [9:0]	H'000	R/W	These bits specify in number of lines the interval between the reference vertical sync signal and the point where the vertical sync signal (VSYNC) for panel is set to 1.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	VSYNC_END [9:0]	H'001	R/W	These bits specify in number of lines the interval between the reference vertical sync signal and the point where the vertical sync signal (VSYNC) for panel is cleared to 0.

Note: Be sure to satisfy VSYNC_START ≠ VSYNC_END; otherwise, correct operation is not guaranteed.

chanhvinh.com

27.7.35 Horizontal Sync Signal Timing Control Register (PANEL_HSYNC_TIM)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	HSYNC_START[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
	-	-	-	-	-	HSYNC_END[10:0]										
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/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	HSYNC_START [10:0]	H'000	R/W	These bits should always be set to H'000. These bits specify in number of panel clock cycles the interval between the reference horizontal sync signal and the point where the horizontal sync signal (HSYNC) for panel is set to 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	HSYNC_END [10:0]	H'00A	R/W	These bits specify in number of panel clock cycles the interval between the reference horizontal sync signal and the point where the horizontal sync signal (HSYNC) for panel is cleared to 0.

Note: Be sure to satisfy HSYNC_START ≠ HSYNC_END; otherwise, correct operation is not guaranteed.

chanhvinh.com

27.7.36 Timing Control Register 2 for Vertical Sync Signal for Video (VIDEO_VSYNC_TIM2)

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
	-	-	-	-	-	-	VIDEO_VSYNC_START2[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 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	VIDEO_VSYNC_START2[9:0]	H'000	R/W	These bits specify in number of lines the interval between the reference vertical sync signal and the point where the vertical sync signal (VSYNC) for video is set to 1.

chanhvinh.com

27.7.37 Timing Control Register for Vertical Sync Signal for Graphics Image (GRA_VSYNC_TIM)

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
	-	-	-	-	-	-	GRA_VSYNC_START[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 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	GRA_VSYNC_START[9:0]	H'000	R/W	These bits specify in number of lines the interval between the reference vertical sync signal and the point where the vertical sync signal (VSYNC) for graphics image is set to 1.

chanhvinh.com

27.7.38 AC Modulation Signal Toggle Line Count Register (AC_LINE_NUM)

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
	-	-	-	-	-	-	-	-	-	-	-	AC_LINE_NUM[4:0]				
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	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 to 0	AC_LINE_NUM [4:0]	H'0C	R/W	These bits specify the AC line number (line where the LCD_M_DISP signal should toggle) in number of lines. Specify the toggle line number – 1. Example: To toggle every 13 lines: AC_LINE_NUM = 13 – 1 = H'0C

Note: When the total number of lines of the panel is even, specify an appropriate value so that the signal toggles at odd-numbered lines.

chanhvinh.com

27.7.39 DE Area Size Register (DE_SIZE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	DE_HEIGHT[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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	DE_WIDTH[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 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25 to 16	DE_HEIGHT [9:0]	H'000	R/W	These bits specify the vertical length (height) of the data enable (DE) signal area in number of 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	DE_WIDTH [10:0]	H'000	R/W	These bits specify the horizontal length (width) of the data enable (DE) signal area in number of panel clock cycles.

chanhvinh.com

27.7.40 DE Area Start Position Register (DE_START)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	DE_START_V[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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	DE_START_H[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 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25 to 16	DE_START_V[9:0]	H'000	R/W	These bits specify in number of lines the vertical interval between the reference vertical sync signal and the start of the data enable (DE) signal output.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	DE_START_H[10:0]	H'000	R/W	These bits specify in number of panel clock cycles the horizontal interval between the reference horizontal sync signal and the start of the data enable (DE) signal output.

Note: Be sure to satisfy SYN_HEIGHT > DE_HEIGHT + DE_START_V; otherwise, correct operation is not guaranteed.

Be sure to satisfy SYN_WIDTH > DE_WIDTH + DE_START_H; otherwise, correct operation is not guaranteed.

chanhvinh.com

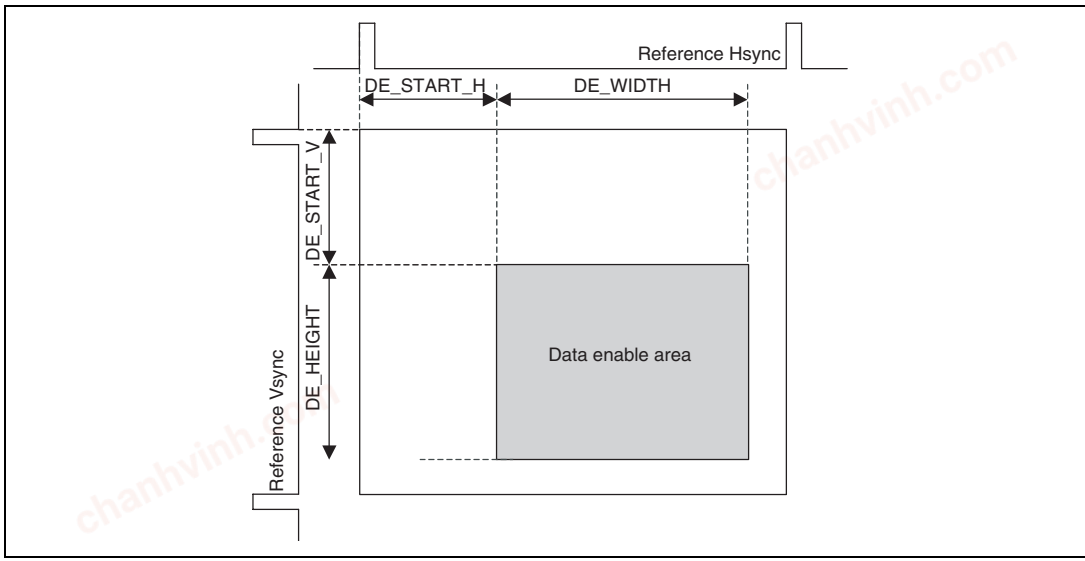


Figure 27.23 DE Area Settings

chanhvinh.com

27.8 Usage Notes

27.8.1 Input Video Format Settings

Table 27.15 shows the register settings for BT.656 input.

Table 27.15 Register Settings for BT.656 Input

Register Name	Setting
VIDEO_MODE[14]	H'0
VIDEO_MODE[13]	H'0
VIDEO_MODE[12]	Select the number of input video lines (NTSC: 525; PAL: 625).
VIDEO_TIM_CNT[25:16]	H'000
VIDEO_TIM_CNT[5]	H'0
VIDEO_TIM_CNT[4]	H'0
VIDEO_TIM_CNT[2]	H'0
VIDEO_TIM_CNT[1]	H'0
VIDEO_TIM_CNT[0]	Select the timing of the input video signal (rising or falling edge).

Table 27.16 shows the register settings for BT.601 input.

Table 27.16 Register Settings for BT.601 Input

Register Name	Setting
VIDEO_MODE[14]	H'1
VIDEO_MODE[13]	H'1
VIDEO_MODE[12]	Select the number of input video lines.
VIDEO_TIM_CNT[25:16]	H'000
VIDEO_TIM_CNT[5]	Change the DV_VSYNC input signal to the active-high signal.
VIDEO_TIM_CNT[4]	Change the DV_HSYNC input signal to the active-high signal.
VIDEO_TIM_CNT[2]	Select the timing of the DV_VSYNC input signal (rising or falling edge).
VIDEO_TIM_CNT[1]	Select the timing of the DV_HSYNC input signal (rising or falling edge).
VIDEO_TIM_CNT[0]	Select the timing of the DV_DATA input signal (rising or falling edge).

chanhvinh.com

27.8.2 How to Use Video Recording Mode

This section describes how to start and stop video recording.

Table 27.17 shows the settings for starting recording, and table 27.18 shows the settings for stopping recording. When recording is restarted after being stopped, the address calculation is initialized and video storing restarts from the base address.

Table 27.17 Register Settings for Starting Video Recording

Register Name	Setting
VIDEO_MODE[20]	H'0
VIDEO_MODE[16]	H'0 (IV1-BUS is a big-endian bus)
VIDEO_MODE[4]	H'0
VIDEO_SIZE[24:16]	For NTSC: H'0F0 For PAL: H'120
VIDEO_SIZE[9:0]	H'2D0
VIDEO_VSTART[24:16]	For BT.656 NTSC: H'010 For BT.656 PAL: H'016 For BT.601: Specify the vertical start position of the valid video in the TOP field.
VIDEO_VSTART[8:0]	For BT.656 NTSC: H'117 For BT.656 PAL: H'14F For BT.601: Specify the vertical start position of the valid video in the BOTTOM field.
VIDEO_HSTART[8:0]	For BT.656 NTSC: H'114 For BT.656 PAL: H'120 For BT.601: Specify the horizontal start position of the valid video.
VIDEO_SAVE_NUM[9:0]	Specify the number of fields to be stored.
VIDEO_IMAGE_CNT[6:4]	H'0
VIDEO_IMAGE_CNT[2:0]	H'0
VIDEO_BASEADR[31:0]	Specify the base address.
VIDEO_LINE_OFFSET[31:0]	Specify the line offset.
VIDEO_FIELD_OFFSET[31:0]	Specify the field offset.

chanhvinh.com

Register Name	Setting
VIDEO_DISP_SIZE[24:16]	For NTSC: H'0F0 For PAL: H'120
VIDEO_DISP_SIZE[9:0]	Specify the horizontal video size to be stored.
SG_MODE[16]	H'0
VIDEO_MODE[0]	H'1 (This setting starts recording. Set this bit at the end of the procedure.)

Table 27.18 Register Settings for Stopping Video Recording

Register Name	Setting
VIDEO_SAVE_NUM[25:16]	These bits hold the field number of which recording processing has been completed.
VIDEO_MODE[0]	H'0 (This setting stops recording. Set this bit at the end of the procedure.)

27.8.3 How to Use Video Display Mode

Table 27.19 Register Settings for Starting Video Display

Register Name	Setting
VIDEO_MODE[21]	H'1
VIDEO_MODE[20]	H'1
VIDEO_MODE[17]	H'0 (IV2-BUS is a big-endian bus)
VIDEO_MODE[16]	H'0 (IV1-BUS is a big-endian bus)
VIDEO_MODE[4]	H'1
VIDEO_INT_CNT[24]	H'0 (This interrupt enable bit is used only for video recording mode.)
VIDEO_SIZE[24:16]	For NTSC: H'0F0 For PAL: H'120
VIDEO_SIZE[9:0]	H'2D0

chanhvinh.com

Register Name	Setting
VIDEO_VSTART[24:16]	For BT.656 NTSC: H'010 For BT.656 PAL: H'016 For BT.601: Specify the vertical start position of the valid video in the TOP field.
VIDEO_VSTART[8:0]	For BT.656 NTSC: H'117 For BT.656 PAL: H'14F For BT.601: Specify the vertical start position of the valid video in the BOTTOM field.
VIDEO_HSTART[8:0]	For BT.656 NTSC: H'114 For BT.656 PAL: H'120 For BT.601: Specify the horizontal start position of the valid video.
VIDEO_VSYNC_TIM1[25:16]	Refer to (1) Video Display Position and Register Settings described below.
VIDEO_VSYNC_TIM1[9:0]	Refer to (1) Video Display Position and Register Settings described below.
VIDEO_SAVE_NUM[9:0]	H'000
VIDEO_IMAGE_CNT[6:4]	Specify the vertical scaling ratio.
VIDEO_IMAGE_CNT[2:0]	Specify the horizontal scaling ratio.
VIDEO_BASEADR[31:0]	Specify the base address.
VIDEO_LINE_OFFSET[31:0]	Specify the line offset.
VIDEO_FIELD_OFFSET[31:0]	H'000
VIDEO_LINEBUFF_NUM[8:0]	Refer to (1) Video Display Position and Register Settings described below.
VIDEO_DISP_SIZE[24:16]	Specify the vertical size of the video to be displayed.
VIDEO_DISP_SIZE [9:0]	Specify the horizontal size of the video to be displayed.
VIDEO_DISP_HSTART	Specify the horizontal position of the video to be displayed.
SG_MODE [16]	H'1
VIDEO_VSYNC_TIM2	Refer to (1) Video Display Position and Register Settings described below.
VIDEO_MODE [1]	H'1
VIDEO_MODE [0]	H'1
GRCMEN2[0], [31]	H'1, H'1
GRCMEN1[0], [31]	H'1, H'1 (This setting starts display. Set this bit at the end of the procedure.)

chanhvinh.com

(1) Video Display Position and Register Settings

As the display block does not have frame memory, specify through registers the vertical start position and number of line buffers for the video according to the panel specifications (HSYNC cycle). For the vertical video position, first calculate the register settings to place the video along the top end, and then calculate the settings to place the video at the center or along the bottom end.

(a) Register settings for vertical position

Step 1: Calculate the settings to place the video along the top end.

Table 27.20 List of Parameters (registers are shaded in grey)

		Register Name or Value	Unit
(1)	T (Hsync_in): HSYNC cycle of the input video	0.064	ms
(2)	Vertical size of the valid video	VIDEO_SIZE[24:16]	Line
(3)	T (Hsync_out): HSYNC cycle for the panel	Depends on the panel specifications	ms
(4)	Vertical size of the video to be displayed	VIDEO_DISP_SIZE[24:16]	Line
(5)	Line buffer margin	6 or a greater value	Line
(6)	Vertical start position of the valid video in the TOP field	VIDEO_VSTART[24:16]	Line
(7)	Vertical start position of the valid video in the BOTTOM field	VIDEO_VSTART[8:0]	Line
(8)	Number of lines between the reference Vsync and the displayable area	Depends on the panel specifications	Line

chanhvinh.com

Table 27.21 Calculations of Register Settings (registers are shaded in grey)

		Register Name or Value	Unit
(9)	VIDEO_VSYNC_TIM2[9:0]	(8) – 1	Line
(10)	Difference in time between video input and output to panel	(1) × (2) – (3) × (4)	Line
(11)	Conversion of number of buffers	When (8) is negative: (10) / 3 When (8) is positive: (10) / 1	Line
(12)	VIDEO_LINEBUFF_NUM[9:0]	(11) + (5)	Line
(13)	VIDEO_VSYNC_TIM1[25:16]*	When (8) is negative: (6) – (8) + (5) / 2 When (8) is positive: (6) + (11) – (8) + (5) / 2	Line
(14)	VIDEO_VSYNC_TIM1[9:0]*	When (8) is negative: (7) – (8) + (5) / 2 When (8) is positive: (7) + (11) – (8) + (5) / 5	Line

Note: * When (8) is negative, the data rate of the input video is larger than that of the video output to the panel. In this case, after video data begins to be stored in the line buffer, start reading from the buffer while only one to several lines of data are stored, so that the buffer does not overflow.

When (8) is positive, the data rate of the input video is smaller than that of the video output to the panel. In this case, start reading from the line buffer after the buffer has stored enough input data, so that the buffer does not underflow (the line buffer does not become empty).

Note that when the line buffer is large enough (for example, when the buffer can store one field of data), the above restrictions are loosened.

Step 2: To place the video at the center or along the bottom end, calculate the new settings from the values obtained in step 1.

Note: Video cannot be displayed in the last line of the panel.

Table 27.22 Calculation of Register Settings (registers are shaded in grey)

		Calculation Expression	Unit
(15)	VIDEO_VSYNC_TIM2[9:0]	(9) + start line to display	Line
(16)	Conversion to the number of input video lines	(3) × start line to display / (1)	Line
(17)	VIDEO_VSYNC_TIM1[25:16]	(13) – (16)	Line
(18)	VIDEO_VSYNC_TIM1[9:0]	(14) – (16)	Line

chanhvinh.com

(b) Register settings for horizontal position

Specify the horizontal position in the VIDEO_DISP_HSTART with respect to the Hsync. The offset value is + 16.

27.8.4 How to Use Graphics Display

As graphics blocks 1 and 2 can be used in the same way, this section describes the register settings for graphics display using the register names in graphics block 1.

Table 27.23 Register Settings for Starting Graphics Display

Register Name	Setting
GRA_VSYNC_TIM	Specify the line number for displaying graphics counted from the reference VSYNC position.
GRCBUSCNT1[0]	H'0 (IV3-BUS is a big-endian bus)
GRCBUSCNT1[4]	Select the data format for IV3-BUS.
GRCBUSCNT1[8]	Select the mode of transfer through IV3-BUS
GROPSADR1	Specify the read address.
GROPSWH1	Specify the graphics image size.
GROPSOFST1	Specify the line offset for the graphics image.
GROPDPHV1	Specify the graphics image start position.
GROPBASERGB1	Specify the color for outside of the graphics image area.
GRCMEN1[0]	Set to H'1 when displaying the lower layer.
GRCMEN1[1], [31]	H'1, H'1 (This setting starts displaying graphics images. Set these bits at the end of the procedure.)

chanhvinh.com

Table 27.24 Register Settings for α Control Area Display

Register Name	Setting
GROPEWH1	Specify the size of the α control area.
GROPEDPHV1	Specify the α control area start position
GROPEDPA1[31:24]	Specify the initial α value
GROPEDPA1[23:16]	Specify the value to be added or subtracted for fade-in or fade-out mode.
GROPEDPA1[15:8]	Specify the frame rate for fade-in or fade-out mode
GROPEDPA1[2:1]	Specify the fade-in or fade-out mode.
GROPEDPA1[0]	H'1 (This setting enables display of the α control area.)
GROPEDPA1[7]	H'1 (Set this bit at the end of the procedure.)

Table 27.25 Register Settings for Chroma-Keying

Register Name	Setting
GROPCRKY0_1[15:0]	Specify the chroma-key target RGB.
GROPCRKY1_1[23:16]	Specify the α value after replacement.
GROPCRKY1_1[15:0]	Specify the RGB values after replacement.
GROPCRKY0_11[16]	H'1 (This setting enables chroma-keying.)
GROPEDPA1[7]	H'1 (Set this bit at the end of the procedure.)

chanhvinh.com

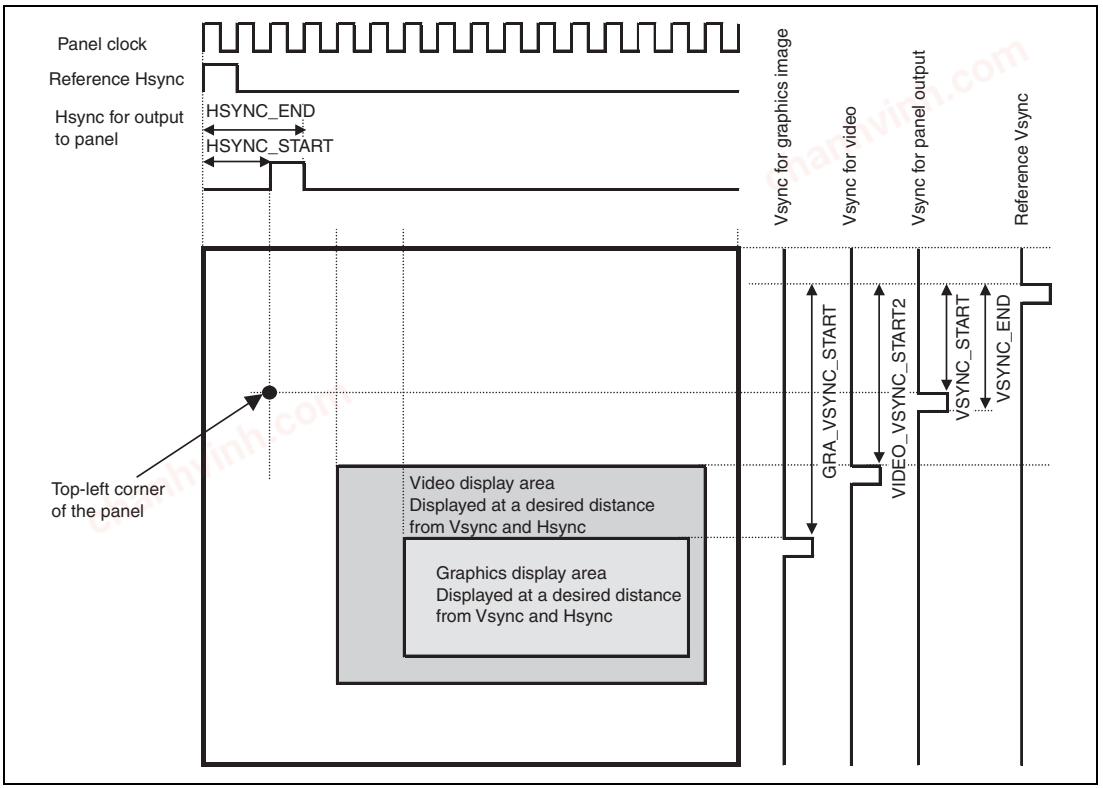


Figure 27.24 Relationship between Video and Graphics Display Positions and Reference Vsync

chanhvinh.com

27.8.5 How to Use Control Signal Output to Panel

The control signals described in section 27.6.3, Panel Control Signal Output Function, can be generated through the register settings shown in table 27.26.

Table 27.26 Registers for Control Signal Output to Panel

Register Name	Setting
SYNCNT	Specify the polarity and timing (rising or falling edge) of the output signals.
PANEL_CLKSEL	Make the LCD_CLK settings.
SYN_SIZE	Specify the panel size.
PANEL_VSYNC_TIM	Specify the timing of LCD_VSYNC.
PANEL_HSYNC_TIM	Specify the timing of LCD_HSYNC.
DE_SIZE	Specify the LCD_DE size.
DE_START	Specify the LCD_DE position.
AC_LINE_NUM	Specify the LCD_M_DISP toggle position.

chanhvinh.com

27.9 Interrupt Requests

This module issues four types of interrupt request: VSYNC cycle fluctuation detection, buffer error, field write completion, and line interrupt requests.

Table 27.27 is a list of interrupt requests.

Table 27.27 List of Interrupt Requests

Interrupt Request	Symbol	Interrupt Request Condition
VSYNC cycle fluctuation detection	VIVSYNCJ	(VIDEO_INT_CNT.V_PERIOD=1)&(VIDEO_INT_CNT.INT_V_EN=1)
Buffer error	VBUFERR	(VIDEO_INT_CNT.UNDER_FLOW=1)&(VIDEO_INT_CNT.INT_UF_EN=1)| (VIDEO_INT_CNT.OVER_FLOW=1)&(VIDEO_INT_CNT.INT_OF_EN=1)| (GRCINTCNT1.UNDER_FLOW=1)&(GRCINTCNT1.INT_UF_EN=1)| (GRCINTCNT2.UNDER_FLOW=1)&(GRCINTCNT2.INT_UF_EN=1)
Field write completion	VIFIELDE	(VIDEO_INT_CNT.F_END=1)&(VIDEO_INT_CNT.INT_F_EN=1)
Line interrupt	VOLINE	(SGINTCNT.LINE_STATUS=1)&(SGINTCNT.INT_LINE_EN=1)

When a condition shown in table 27.27 is evaluated to 1, this module issues an interrupt request.

27.10 Usage Note

27.10.1 The Procedure of Disabling the Video Receiving Block Operation

When disabling the Video receiving block operation, please follow the procedure given below.

1. Unselect the DV_CLK function of PF8 mode in the control register (PFCR2) of general purpose I/O ports. (select the PF8 function)
2. Clear the VIDEO_MAIN_EXE bit in VIDEO_MODE register to 0.
3. Halt the clock to video display controller 3 by the standby control register (STBCR7) of power-down modes.
4. Select the DV_CLK function of PF8 mode in the control register (PFCR2) of general purpose I/O ports.
5. Supply the clock to video display controller 3 by the standby control register (STBCR7) of power-down modes.

chanhvinh.com

Section 28 Sampling Rate Converter

The sampling rate converter converts the sampling rate for data produced by decoders such as WMA, MP3, or AAC.

28.1 Features

Table 28.1 Feature

Item	Channel 0	Channel 1
Data format	16-bit stereo/16-bit monaural	16-bit monaural
Input sampling rate	Selectable from 8 kHz, 11.025 kHz, 12 kHz, 16 kHz, 22.05 kHz, 24 kHz, 32 kHz, 44.1 kHz, and 48 kHz	44.1 kHz
Output sampling rate	Selectable from 44.1 kHz and 48 kHz	Selectable from 8 kHz and 16 kHz

- Processing capacity: A sample output interval is a maximum of 8 μs and 14 μs for channels 0 and 1, respectively. (Pϕ = 36 MHz)
- SNR: 80 db or higher
- Five interrupt sources: Input data FIFO empty, output data FIFO full, output data FIFO overwrite, output data FIFO underflow, and conversion end
- Two DMA transfer sources: Input data FIFO empty and output data FIFO full
- Module standby mode

  Power consumption can be reduced by stopping clock supply to this module when not used.

chanhvinh.com

Figure 28.1 shows a block diagram.

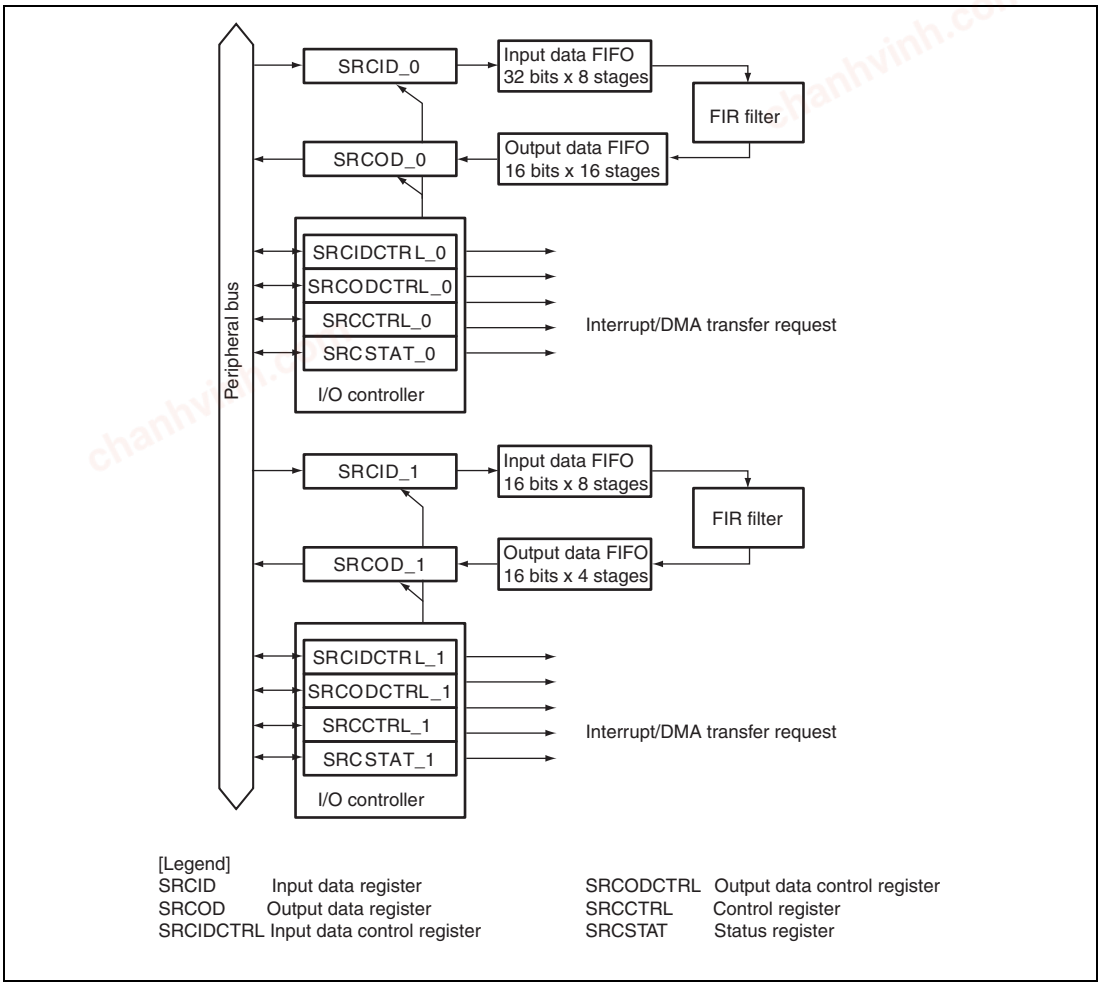


Figure 28.1 Block Diagram

chanhvinh.com

28.2 Register Descriptions

Table 28.2 shows the register configuration.

Table 28.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Input data register_0	SRCID_0	R/W	H'00000000	H'FFFF7000	16, 32
Output data register_0	SRCOD_0	R	H'00000000	H'FFFF7004	16, 32
Input data control register_0	SRCIDCTRL_0	R/W	H'0000	H'FFFF7008	16
Output data control register_0	SRCODCTRL_0	R/W	H'0000	H'FFFF700A	16
Control register_0	SRCCTRL_0	R/W	H'0000	H'FFFF700C	16
Status register_0	SRCSTAT_0	R/(W)*	H'0002	H'FFFF700E	16
Input data register_1	SRCID_1	R/W	H'00000000	H'FFFF7800	16, 32
Output data register_1	SRCOD_1	R	H'00000000	H'FFFF7804	16, 32
Input data control register_1	SRCIDCTRL_1	R/W	H'0000	H'FFFF7808	16
Output data control register_1	SRCODCTRL_1	R/W	H'0000	H'FFFF780A	16
Control register_1	SRCCTRL_1	R/W	H'0000	H'FFFF780C	16
Status register_1	SRCSTAT_1	R/(W)*	H'0002	H'FFFF780E	16

Note: * Bits 15 to 6 and 4 are read-only. Only 0 can be written to bits 5 and 3 after having read as 1.

chanhvinh.com

28.2.1 Input Data Register (SRCID)

SRCID is a readable/writable register that is used to input the data before sampling rate conversion. All the bits are read as 0. The data input to SRCID is stored in the 8-stage input data FIFO. When the number of data units in the input data FIFO is 8, writing to SRCID has no effect.

(1) Input Data Register_0 (SRCID_0)

SRCID_0 is a 32-bit register. For stereo data, bits 31 to 16 are for Lch data, and bits 15 to 0 are for Rch data. For monaural data, data in bits 31 to 16 is valid, and data in bits 15 to 0 is 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/W	R/W	R/W	R/W	R/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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

(2) Input Data Register_1 (SRCID_1)

SRCID_1 is a 16-bit register. Monaural data in bits 15 to 0 is valid.

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

The data subject to sampling rate conversion is aligned differently depending on the IED bit setting in SRCIDCTRL. Tables 28.3 and 28.4 show the relationship between the IED bit setting and data alignment.

Table 28.3 Alignment of Data before Sampling Rate Conversion (Channel 0)

IED	Lch[15:8]	Lch[7:0]	Rch[15:8]	Lch[7:0]
0	SRCID[31:24]	SRCID[23:16]	SRCID[15:8]	SRCID[7:0]
1	SRCID[23:16]	SRCID[31:24]	SRCID[7:0]	SRCID[15:8]

chanhvinh.com

Table 28.4 Alignment of Data before Sampling Rate Conversion (Channel 1)

IED	Upper Byte	Lower Byte
0	SRCID[15:8]	SRCID[7:0]
1	SRCID[7:0]	SRCID[15:8]

28.2.2 Output Data Register (SRCOD)

SRCOD is a 32-bit read-only register used to output the data after sampling rate conversion. The data in the output data FIFO is read through SRCOD. When the number of data in the output data FIFO is zero after the start of conversion, the value previously read is read again.

(1) Output Data Register_0 (SRCOD_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
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

(2) Output Data Register_1 (SRCOD_1)

The conversion result is stored in bits 31 to 16. Bits 15 to 0 are always read as 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
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

chanhvinh.com

The data in SRCOD is aligned differently depending on the OCH and OED bit setting in SRCODCTRL. Tables 28.5 and 28.6 show the correspondence between the OCH and OED bit setting and data alignment in SRCOD.

Table 28.5 Alignment of Data in SRCOD_0

OCH	OED	SRCOD_0[31:24]	SRCOD_0[23:16]	SRCOD_0[15:8]	SRCOD_0[7:0]
0	0	Lch[15:8]	Lch[7:0]	Rch[15:8]*2	Rch[7:0]*2
	1	Lch[7:0]	Lch[15:8]	Rch[7:0]*2	Rch[15:8]*2
1*1	0	Rch[15:8]	Rch[7:0]	Lch[15:8]	Lch[7:0]
	1	Rch[7:0]	Rch[15:8]	Lch[7:0]	Lch[15:8]

Notes: 1. When processing monaural data, do not set the bit to 1.
2. When processing monaural data, the data in these bits is invalid.

Table 28.6 Alignment of Data in SRCOD_1

IED	SRCOD_1[31:24]	SRCOD_1[23:16]
0	Upper byte	Lower byte
1	Lower byte	Upper byte

chanhvinh.com

28.2.3 Input Data Control Register (SRCIDCTRL)

SRCIDCTRL is a 16-bit readable/writable register that specifies the endian format of input data, enables/disables the interrupt requests, and specifies the triggering number of data units.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	IED	IEN	—	—	—	—	—	—	IFTRG[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	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	IED	0	R/W	Input Data Endian Specifies the endian format of the input data. 0: Big endian 1: Little endian
8	IEN	0	R/W	Input Data FIFO Empty Interrupt Enable Enables/disables the input data FIFO empty interrupt request to be issued when the number of data units in the input FIFO becomes equal to or smaller than the triggering number specified by the IFTRG1 and IFTRG0 bits, thus resulting in the IINT bit in the status register (SRCSTAT) being set to 1. 0: Input data FIFO empty interrupt is disabled. 1: Input data FIFO empty interrupt is enabled.
7 to 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	IFTRG[1:0]	00	R/W	Input FIFO Data Triggering Number Specifies the condition in terms of the number on which the IINT bit in the status register (SRCSTAT) is set to 1. When the number of data units in the input FIFO becomes equal to or smaller than the triggering number listed below, the IINT bit is set to 1. 00: 0 01: 2 10: 4 11: 6

chanhvinh.com

28.2.4 Output Data Control Register (SRCODCTRL)

SRCODCTRL is a 16-bit readable/writable register that specifies whether to exchange the channels for the output data, specifies the endian format of output data, enables/disables the interrupt requests, and specifies the triggering number of data units.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	OCH	OED	OEN	—	—	—	—	—	—	OFTRG[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/W	R/W	R/W	R	R	R	R	R	R	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	OCH	0	R/W	Output Data Channel Exchange Specifies whether to exchange the channels for the output data register (SRCOD). When processing monaural data, do not set this bit to 1. 0: Does not exchange the channels (the same order as data input) 1: Exchanges the channels (the opposite order from data input) Note: For channel 1, this bit is reserved and always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
9	OED	0	R/W	Output Data Endian Specifies the endian format of the output data. 0: Big endian 1: Little endian
8	OEN	0	R/W	Output Data FIFO Full Interrupt Enable Enables/disables the output data FIFO full interrupt request to be issued when the number of data units in the output FIFO becomes equal to or greater than the number specified by the OFTRG1 and OFTRG0 bits, thus resulting in the OINT bit in the status register (SRCSTAT) being set to 1. 0: Output data FIFO full interrupt is disabled. 1: Output data FIFO full interrupt is enabled.
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	OFTRG[1:0]	00	R/W	Output FIFO Data Trigger Number Specifies the condition in terms of the number on which the OINT bit in the status register (SRCSTAT) is set to 1. When the number of data units in the output FIFO becomes equal to or greater than the number listed below, the OINT bit is set to 1. For channel 0: 00: 1 01: 4 10: 8 11: 12 For channel 1: 00: 1 01: 1 10: 2 11: 3

chanhvinh.com

28.2.5 Control Register (SRCCTRL)

SRCCTRL is a 16-bit readable/writable register that enables/disables the module operation, enables/disables the interrupt requests, and specifies flush processing, clear processing of the internal work memory, and the input and output sampling rates.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	CEEN	SRCEN	UDEN	OVEN	FL	CL	IFS[3:0]				—	—	—	OFS
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/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	CEEN	0	R/W	Conversion End Interrupt Enable Enables/disables the conversion end interrupt to be generated when the CEF bit in SRCSTAT is set to 1 after flush processing is completed and all the output data is read. 0: Disables conversion end interrupt requests. 1: Enables conversion end interrupt requests.
12	SRCEN	0	R/W	Module Enable Enables/disables this module operation. Writing 1 while SRCEN = 0 clears the internal work memory. 0: Disables this module operation. 1: Enables this module operation. Note: When SRCEN = 1, do not change the settings of the following bits.

Register	Bit	Bit Name
SRCIDCTRL	9	IED
SRCODCTRL	10, 9	OCH, OED
SRCCTRL	7 to 4, 0	IFS[3:0], OFS

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
11	UDEN	0	R/W	Output Data FIFO Underflow Interrupt Enable Enables/disables the output data FIFO underflow interrupt to be generated when output data FIFO is read and the UDF bit in SRCSTAT is set to 1 while the number of data units in the output data FIFO is zero. 0: Disables output data FIFO underflow interrupt requests. 1: Enables output data FIFO underflow interrupt requests.
10	OVEN	0	R/W	Output Data FIFO Overwrite Interrupt Enable Enables/disables the output data FIFO overwrite interrupt request to be issued when the conversion for the next data has been completed while the number of data units in the output FIFO is eight, thus setting the OVF bit in the status register (SRCSTAT) to 1. When OVEN = 1: Conversion processing is stopped until the OVF bit is cleared by the CPU accessing to SRCSTAT when the output data FIFO overwrite interrupt is generated. At this time, conversion result writing to the output data FIFO is also stopped. OVEN = 0: The OVF bit is automatically cleared when the output data FIFO has space, and conversion processing can be continued. 0: Output data FIFO overwrite interrupt is disabled. 1: Output data FIFO overwrite interrupt is enabled.
9	FL	0	R/W	Internal Work Memory Flush Writing 1 to this bit starts converting the sampling rate of all the data in the input FIFO, input buffer memory, and intermediate memory (i.e., flush processing). This bit is always read as 0. When SRCEN = 0, writing 1 to this bit does not trigger flush processing. In addition, when 1 is written to the FL bit while the number of data units in the input buffer memory is less than the values shown in tables 28.9 and 28.10, valid output data cannot be received. Thus the internal work memory is cleared without triggering the flush processing.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
8	CL	0	R/W	Internal Work Memory Clear Writing 1 to this bit clears the input FIFO, output FIFO, input buffer memory, intermediate memory, and accumulator. This bit is always read as 0. Even when SRCEN = 0, writing 1 to this bit clears the processing.
7 to 4	IFS[3:0]	All 0	R/W	Input Sampling Rate Specifies the input sampling rate. 0000: 8.0 kHz 0001: 11.025 kHz 0010: 12.0 kHz 0011: Setting prohibited 0100: 16.0 kHz 0101: 22.05 kHz 0110: 24.0 kHz 0111: Setting prohibited 1000: 32.0 kHz 1001: 44.1 kHz 1010: 48.0 kHz 1011: Setting prohibited 1100: Setting prohibited 1101: Setting prohibited 1110: Setting prohibited 1111: Setting prohibited Note: For channel 1, these bits are reserved and always read as 0. The write value should always be 0.
3 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	OFS	0	R/W	Output Sampling Rate Specifies the output sampling rate. For channel 0: 0: 44.1 kHz 1: 48.0 kHz For channel 1: 0: 8.0 kHz 1: 16.0 kHz Note: When channels 0 and 1 are used at the same time, only the following settings are allowed.

	Channel 0	Channel 1
IFS[3:0]	OFS	OFS
0000	0	0
0100	0	1

chanhvinh.com

After flush processing has been completed, the number of output data units obtained as a result of conversion can be calculated by using the following formula.

$$\text{Number of output data units} = \left\lfloor \text{Number of input data units} \times \frac{\text{Output sampling rate}}{\text{Input sampling rate}} \right\rfloor + 1$$

- Channel 0

$$n = \begin{cases} 6: \text{When IFS[3:0] = 0000} \\ 4: \text{When IFS[3:0] = (0001,0010)} \\ 3: \text{When IFS[3:0] = (0100,1000)} \\ 2: \text{When IFS[3:0] = (0101,0110)} \\ 1: \text{When IFS[3:0] = (1001,1010)} \end{cases}$$

- Channel 1

$$n = 1$$

Conversion processing is not started and thus output data is not obtained until the specified number of data units are input. The minimum number of input data units necessary for obtaining the first output data depends on the IFS and OFS bit settings. Tables 28.7 to 28.10 show the relation between the settings of the IFS and OFS bits and the number of input data required.

Table 28.7 Relation between Sampling Rate Settings and Number of Initial Input Data Units Required (Channel 0)

OFS Setting (Output Sampling Rate [kHz])	IFS Setting (Input Sampling Rate [kHz])								
	0000 (8.0)	0001 (11.025)	0010 (12.0)	0100 (16.0)	0101 (22.05)	0110 (24.0)	1000 (32.0)	1001 (44.1)	1010 (48.0)
0 (44.1)	38	40	40	43	48	48	43	—	63
1 (48.0)	38	40	40	43	48	48	43	32	—

Table 28.8 Relation between Output Sampling Rate Settings and Number of Initial Input Data Units Required (Channel 1)

OFS Setting (Output Sampling Rate [kHz])	Number of Initial Input Data Units
0 (8.0)	63
1 (16.0)	63

chanhvinh.com

Table 28.9 Relation between Sampling Rate Settings and Number of Input Data Units Required for Flush Processing(Channel 0)

OFS Setting (Output Sampling Rate [kHz])	IFS Setting (Input Sampling Rate [kHz])								
	0000 (8.0)	0001 (11.025)	0010 (12.0)	0100 (16.0)	0101 (22.05)	0110 (24.0)	1000 (32.0)	1001 (44.1)	1010 (48.0)
0 (44.1)	27	24	24	22	16	16	22	—	1
1 (48.0)	27	24	24	22	16	16	22	32	—

Table 28.10 Relation between Output Sampling Rate Settings and Number of Input Data Units Required for Flush Processing (Channel 1)

OFS Setting (Output Sampling Rate [kHz])	Number of Input Data Units Required for Flush Processing
0 (8.0)	1
1 (16.0)	1

chanhvinh.com

28.2.6 Status Register (SRCSTAT)

SRCSTAT is a 16-bit readable/writable register that indicates the number of data units in the input and output data FIFOs, whether the various interrupt sources have been generated or not, and the flush processing status.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OFDN[4:0]					IFDN[3:0]				—	CEF	FLF	UDF	OVF	IINT	OINT
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(W)*1	R	R(W)*1	R/(W)*1	R/(W)*1	R/(W)*1

Note: *1 Only 0 can be written after having read as 1.

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	OFDN[4:0]	All 0	R	Output FIFO Data Count Indicates the number of data units in the output FIFO. Note: For channel 1, bits 15 and 14 are reserved and always read as 0. The write value should always be 0.
10 to 7	IFDN[3:0]	All 0	R	Input FIFO Data Count Indicates the number of data units in the input FIFO.
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5	CEF	0	R/(W)*1	Conversion End Flag Indicates that all the output data is read after flush processing is completed. [Clearing conditions] • When 0 has been written to the CEF bit after reading CEF = 1. • When 1 has been written to the CL bit in SRCCTRL. • When 1 has been written to the SRCEN bit in SRCCTRL while SRCEN is 0. [Setting condition] • When the number of data units in the output data FIFO is zero on completion of flush processing.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
4	FLF	0	R	Flush Processing Status Flag Indicates whether flush processing is in progress or not. [Clearing conditions] • When flush processing has been completed. • When 1 has been written to the CL bit in SRCCTRL. • When 1 has been written to the SRCEN bit in SRCCTRL while SRCEN is 0. [Setting condition] • When 1 has been written to the FL bit in SRCCTRL.
3	UDF	0	R/(W)*[1]	Output FIFO Underflow Interrupt Request Flag Indicates that the output data FIFO is read when the number of data units in the output data FIFO is zero. [Clearing conditions] • When 0 has been written to the UDF bit after reading UVF = 1. • When 1 has been written to the CL bit in SRCCTRL. • When 1 has been written to the SRCEN bit in SRCCTRL while SRCEN is 0. [Setting condition] • When the output data FIFO is read while the number of data units in the output FIFO is zero.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
2	OVF	0	R/(W)*[1]	Output Data FIFO Overwrite Interrupt Request Flag Indicates that the sampling rate conversion for the next data has been completed when the output data FIFO is full. The conversion is stopped until the OVF flag is cleared. [Clearing conditions] • When 0 has been written to the OVF bit after reading OVF = 1 while the OVEN bit in SRCCTRL is 1. • When the number of data units in the output FIFO decreases after reading SRCOD while the OVEN bit in SRCCTRL is 0. • When 1 has been written to the CL bit in SRCCTRL. • When 1 has been written to the SRCEN bit in SRCCTRL while SRCEN is 0. [Setting condition] • When the sampling rate conversion for the next data has been completed when the output FIFO is full.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
1	IINT	1	R/(W)*[1]	Input Data FIFO Empty Interrupt Request Flag Indicates that the number of data units in the input FIFO has become equal to or smaller than the triggering number specified by the IFTRG1 and IFTRG0 bits in the input data control register (SRCIDCTRL). [Clearing conditions] • When 0 has been written to the IINT bit after reading IINT = 1. • When the number of data units in the input FIFO has exceeded the specified triggering number due to DMA transfer to the input FIFO. [Setting conditions] • When the number of data units in the input FIFO has become equal to or smaller than the specified triggering number. • When 1 has been written to the CL bit in SRCCTRL. • When 1 has been written to the SRCEN bit in SRCCTRL while SRCEN is 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
0	OINT	0	R/(W)*[1]	Output Data FIFO Full Interrupt Request Flag Indicates that the number of data units in the output FIFO has become equal to or greater than the triggering number specified by the OFTRG[1:0] bits in the output data control register (SRCODCTRL). [Clearing conditions] • When 0 has been written to the OINT bit after reading OINT = 1. • When the number of data units in the FIFO has become less than the specified triggering number due to DMA transfer to the output FIFO. • When 1 has been written to the CL bit in SRCCTRL.*[2] • When 1 has been written to the SRCEN bit in SRCCTRL while SRCEN is 0.*[2] [Setting condition] • When the number of data units in the output FIFO has become equal to or greater than the specified triggering number.

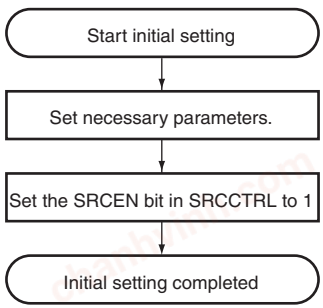
Notes: 1. Only 0 can be written after having read as 1.
2. The setting is valid only in channel 0. In channel 1, the setting is invalid.

chanhvinh.com

28.3 Operation

28.3.1 Initial Setting

Figure 28.2 shows a sample flowchart for initial setting.



Register	Bit	Items to be Set
SRCCTRL	CEEN	Enabling/disabling of the CEF interrupt
	UDEN	Enabling/disabling of the UDF interrupt
	OVEN	Enabling/disabling of the OVF interrupt
	IFS[3:0]	Input sampling rate
	OFS	Output sampling rate
SRCIDCTRL	IED	Input data endian
	IEN	Enabling/disabling of the IDE interrupt
	IFTRG[1:0]	Input data FIFO triggering number
SRCODCTRL	OCH	Exchanging of output data channels
	OED	Output data endian
	OEN	Enabling/disabling of the ODF interrupt
	OFTRG[1:0]	Output data FIFO triggering number

Figure 28.2 Sample Flowchart for Initial Setting

chanhvinh.com

28.3.2 Data Input

Figure 28.3 is a sample flowchart for data input.

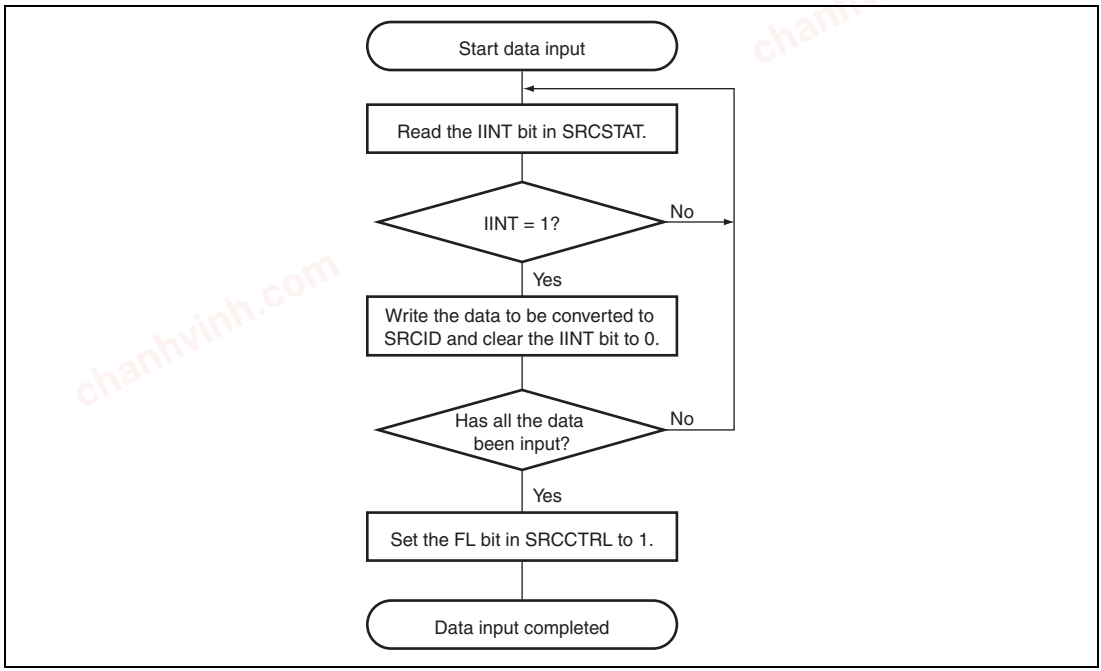


Figure 28.3 Sample Flowchart for Data Input

(1) When Interrupts are Issued to CPU

1. Set the IEN bit in SRCIDCTRL to 1.
2. When the IINT bit in SRCSTAT is set to 1, the IDE interrupt request is issued. In the interrupt processing routine, read the IINT bit and confirm that it is 1, write data to SRCID, and write 0 to the IINT bit. Then return from the interrupt processing routine.
3. Repeat step 2 until all the data has been input, and write 1 to the FL bit in SRCCTRL.

chanhvinh.com

(2) When Interrupts are Used to Activate Direct Memory Access Controller

1. Assign IDEI of this module to one channel of the direct memory access controller.
2. Set the IEN bit in SRCIDCTRL to 1.
3. When the IINT bit in SRCSTAT is set to 1, the IDE interrupt request is issued thus activating the direct memory access controller. When the direct memory access controller has written data to the SRCID thus resulting in the number of data units in the input data FIFO exceeding that of the triggering number specified by the IFTRG1 and IFTRG 0 bits in SRCIDCTRL, the IINT bit is cleared to 0.
4. Repeat step 3 until all the data has been input, and write 1 to the FL bit in SRCCTRL.

(3) When Serial Sound Interface Interrupts are Used for Activating Direct Memory Access Controller to Transfer Input Data from Serial Sound Interface

1. Assign the serial sound interface to one channel of the direct memory access controller as a DMA transfer request source. Set SSIFRDR of the serial sound interface as a transfer source and SRCID of the sampling rate converter as a transfer destination, and set the serial source interface to enable reception operation.
2. When the RDF bit in SSIFSR is set to 1, the serial sound interface interrupt request is issued thus activating the direct memory access controller. The direct memory access controller then reads data from SSIFRDR and writes the data to SRCID.
3. Repeat step 2 until all the data has been input, and write 1 to the FL bit in SRCCTRL.

chanhvinh.com

28.3.3 Data Output

Figure 28.4 is a sample flowchart for data output.

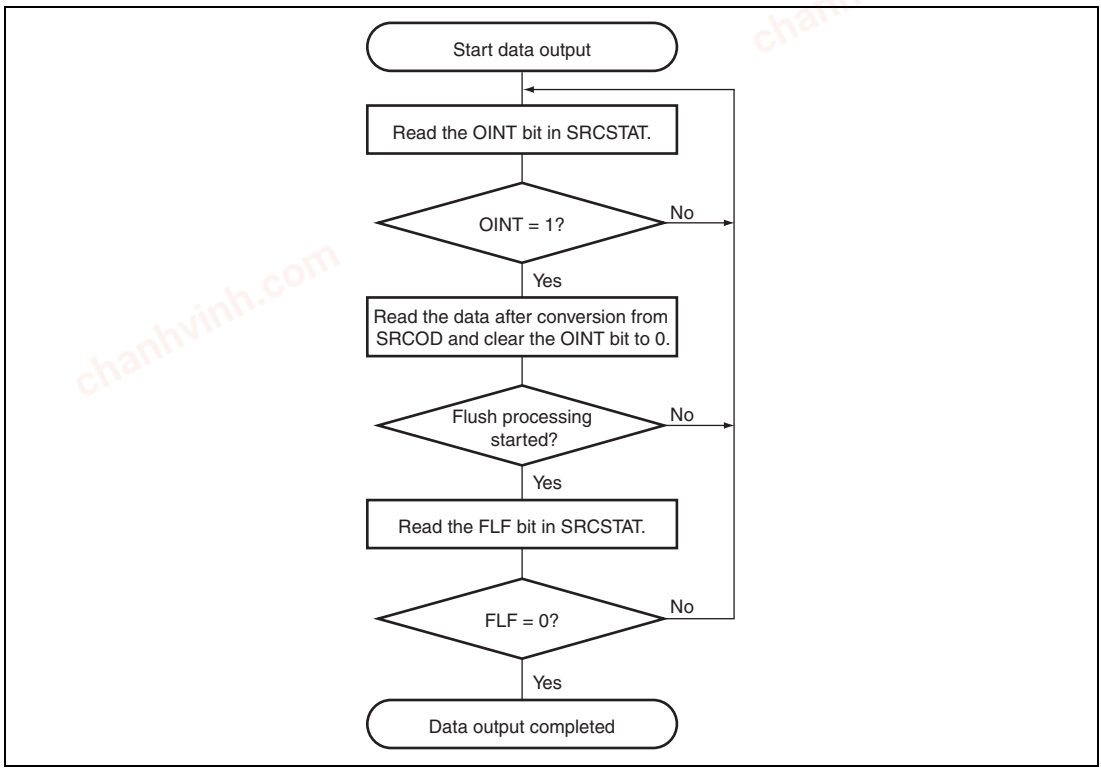


Figure 28.4 Sample Flowchart for Data Output

chanhvinh.com

(1) When Interrupts are Issued to CPU

1. Set the OEN bit in SRCODCTRL to 1.
2. When the OINT bit in SRCSTAT is set to 1, the ODF interrupt request is issued. In the interrupt processing routine, read the OINT bit and confirm that it is 1, read data from SRCOD, and write 0 to the OINT bit. Then return from the interrupt processing routine.
3. After flush processing starts, repeat step 2 until the CEF bit in SRCSTAT is read as 1.

(2) When Interrupts are Used to Activate Direct Memory Access Controller

1. Assign ODFI of this module to one channel of the direct memory access controller.
2. Set the OEN bit in SRCODCTRL to 1.
3. When the OINT bit in SRCSTAT is set to 1, the ODF interrupt request is issued thus activating the direct memory access controller. When the direct memory access controller has read data from SRCOD thus resulting in the number of data units in the output data FIFO being less than the triggering number specified by the OFTRG1 and OFTRG0 bits in SRCODCTRL, the OINT bit is cleared to 0.
4. After flush processing starts, repeat step 3 until the FLF bit in SRCSTAT is read as 0.

(3) When Serial Sound Interface Interrupts are Used for Activating Direct Memory Access Controller to Transfer Output Data to Serial Sound Interface

1. Set the OVEN bit in SRCCTRL to 0 to disable the OVF interrupt request generation.
2. Assign the serial sound interface to one channel of the direct memory access controller as a DMA transfer request source. Set SRCID of the sampling rate converter as a transfer source and SSIFTDR of the serial sound interface as a transfer destination, and set the serial source interface to enable transmission operation.
3. When the TDE bit in SSIFSR is set to 1, the serial sound interface issues an interrupt request thus activating the direct memory access controller. The direct memory access controller then reads data from SRCOD and writes the data to SSIFTDR.
4. After flush processing starts, repeat step 3 until the CEF bit in SRCSTAT is read as 1.

chanhvinh.com

28.4 Interrupts

This module has five interrupt sources: input data FIFO empty (IDEI), output data FIFO full (ODFI), output data FIFO overwrite (OVF), output data FIFO underflow (UDF), and conversion end (CEF). Table 28.11 summarizes the interrupts.

Table 28.11 Interrupt Requests and Generation Conditions

Interrupt Request	Abbreviation	Interrupt Condition	Direct Memory Access Controller Activation
Input data FIFO empty	IDEI	IINT = 1, IEN = 1, and SRCEN = 1	Possible
Output data FIFO full	ODFI	OINT = 1, OEN = 1, and SRCEN = 1	Possible
Output data FIFO overwrite	OVF	OVF = 1, OVEN = 1, and SRCEN = 1	Not possible
Output data FIFO underflow	UDF	UDF = 1, UDEN = 1, and SRCEN = 1	Not possible
Conversion end	CEF	CEF = 1, CEEN = 1, and SRCEN = 1	Not possible

When the interrupt condition is satisfied, the CPU executes the interrupt exception handling routine. The interrupt source flags should be cleared in the routine.

The IDEI and ODFI interrupts can activate the direct memory access controller when the direct memory access controller is set to allow this. If the direct memory access controller is activated, the interrupts from this module are not sent to the CPU. When the direct memory access controller has written data to SRCID resulting in the number of data units in the input data FIFO exceeding that of the specified triggering number, the IINT bit is cleared to 0. Similarly, when the direct memory access controller has read data from SRCOD resulting in the number of data units in the output data FIFO being less than the specified triggering number, the OINT bit is cleared to 0.

chanhvinh.com

28.5 Usage Notes

28.5.1 Notes on Accessing Registers

After the following write access to SRCCTRL, three cycles of the peripheral clock (Pϕ) elapse before the corresponding bit in SRCSTAT is updated.

- Before the FLF bit in SRCSTAT is set after 1 is written to the FL bit in SRCCTRL
- Before each bit in SRCSTAT is initialized after 1 is written to the CL bit in SRCCTRL
- Before each bit in SRCSTAT is initialized after 1 is written to the SRCEN bit in SRCCTRL while the SRCEN bit is 0

On the other hand, as the CPU executes any subsequent instruction without waiting for the completion of the register writing, an instruction that immediately follows that used to write to SRCCTRL cannot accurately detect the updated state of SRCSTAT. To check the updated SRCSTAT state, dummy-read SRCCTRL or SRCSTAT after the instruction used to write to SRCCTRL.

28.5.2 Notes on Flush Processing

When 1 is written to the FL bit in SRCCTRL, this module continues conversion processing by adding 0-data to the input data end point. Flush processing, therefore, should be performed when the audio data end point is input and there is no subsequent data.

To perform conversion again after flush processing, clear the internal work memory in either of the following ways.

- Write 1 to the CL bit in SRCCTRL.
- Write 0 and then 1 to the SRCEN bit in SRCCTRL.

chanhvinh.com

28.5.3 Notes on Using Two Channels at the Same Time

When channel 0 and channel 1 are used at the same time, only the settings shown in table 28.12 can be set.

Table 28.12 Settable Combinations for Using Two Channels at the Same Time

	Channel 0		Channel 1	
IFS[3:0]	**CFS**	**Conversion Processing**	**OFS**	**Conversion Processing**
0000	0	8.0 kHz to 44.1 kHz	0	44.1 kHz to 8.0 kHz
0001	0	16.0 kHz to 44.1 kHz	1	44.1 kHz to 16.0 kHz

chanhvinh.com

chanhvinh.com
chanhvinh.com
chanhvinh.com
chanhvinh.com

chanhvinh.com

Section 29 SD Host Interface

Renesas Technology Corporation is only able to provide information contained in this section to parties with which we have concluded a nondisclosure agreement. Please contact one of our sales representatives for details.

chanhvinh.com

chanhvinh.com

chanhvinh.com
chanhvinh.com
chanhvinh.com
chanhvinh.com

chanhvinh.com

Section 30 Decompression Unit

Renesas Technology Corporation is only able to provide information contained in this section to parties with which we have concluded a nondisclosure agreement. Please contact one of our sales representatives for details.

chanhvinh.com

chanhvinh.com

chanhvinh.com
chanhvinh.com
chanhvinh.com
chanhvinh.com

chanhvinh.com

Section 31 On-Chip RAM

This LSI has an on-chip high-speed RAM, which achieves fast access, an on-chip large-capacity RAM for display area and work area (32 Kbytes for 1-Mbyte version and 320 Kbytes for 640-Kbyte version 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 on-chip high-speed RAM and 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 5 for 1-Mbyte and page 0 to 2 for 640-Kbyte versions of 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.

31.1 Features

- Page
  - The on-chip high-speed RAM consists of four pages. The size of one page is 16 Kbytes.
  - The size of each page is determined depending on the display area sizes (QVGA, WQVGA, and VGA sizes).
  - The on-chip data retention RAM consists of two pages for 1-Mbyte and four pages for 640-Kbyte versions. For 1-Mbyte version, the size of each page is 16-Kbytes. For 640-Kbyte version, page 0 has 16-Kbytes, page 1 has 16-Kbytes, page 2 has 128-Kbytes, and page 3 has 160-Kbytes.
- Memory map

  The on-chip RAM is located in the address spaces shown in tables 31.1 to 31.5.

Table 31.1 Address Spaces of On-Chip High-Speed RAM

Page	Address
Page 0	H'FFF80000 to H'FFF83FFF
Page 1	H'FFF84000 to H'FFF87FFF
Page 2	H'FFF88000 to H'FFF8BFFF
Page 3	H'FFF8C000 to H'FFF8FFFF

chanhvinh.com

Table 31.2 Address Spaces of On-Chip Large-Capacity RAM (1-Mbyte Version)

Page	Cache-enabled Address	Cache-disabled Address
Page 0 (160 Kbytes)	H'1C000000 to H'1C027FFF	H'3C000000 to H'3C027FFF
Page 1 (80 Kbytes)	H'1C028000 to H'1C03BFFF	H'3C028000 to H'3C03BFFF
Page 2 (80 Kbytes)	H'1C03C000 to H'1C04FFFF	H'3C03C000 to H'3C04FFFF
Page 3 (160 Kbytes)	H'1C050000 to H'1C077FFF	H'3C050000 to H'3C077FFF
Page 4 (240 Kbytes)	H'1C078000 to H'1C0B3FFF	H'3C078000 to H'3C0B3FFF
Page 5 (304 Kbytes)	H'1C0B4000 to H'1C0FFFFF	H'3C0B4000 to H'3C0FFFFF

Table 31.3 Address Spaces of On-Chip Data Retention RAM (1-Mbyte Version)

Page	Cache-enabled Address	Cache-disabled Address
Page 0 (16 Kbytes)	H'1C0F8000 to H'1C0FBFFF	H'3C0F8000 to H'3C0FBFFF
Page 1 (16 Kbytes)	H'1C0FC000 to H'1C0FFFFF	H'3C0FC000 to H'3C0FFFFF

Table 31.4 Address Spaces of On-Chip Large-Capacity RAM (640-Kbyte Version)

Page	Cache-enabled Address	Cache-disabled Address
Page 0 (160 Kbytes)	H'1C000000 to H'1C027FFF	H'3C000000 to H'3C027FFF
Page 1 (80 Kbytes)	H'1C028000 to H'1C03BFFF	H'3C028000 to H'3C03BFFF
Page 2 (80 Kbytes)	H'1C03C000 to H'1C04FFFF	H'3C03C000 to H'3C04FFFF
Page 3 (160 Kbytes)	H'1C050000 to H'1C077FFF	H'3C050000 to H'3C077FFF
Page 4 (160 Kbytes)	H'1C078000 to H'1C09FFFF	H'3C078000 to H'3C09FFFF

Table 31.5 Address Spaces of On-Chip Data Retention RAM (640-Kbyte Version)

Page	Cache-enabled Address	Cache-disabled Address
Page 0 (16 Kbytes)	H'1C000000 to H'1C003FFF	H'3C000000 to H'3C003FFF
Page 1 (16 Kbytes)	H'1C004000 to H'1C007FFF	H'3C004000 to H'3C007FFF
Page 2 (128 Kbytes)	H'1C008000 to H'1C027FFF	H'3C008000 to H'3C027FFF
Page 3 (160 Kbytes)	H'1C028000 to H'1C04FFFF	H'3C028000 to H'3C04FFFF

chanhvinh.com

- Ports

  Each page of the on-chip high-speed RAM has two independent read and write ports and is connected to the internal DMA bus (ID bus), CPU instruction fetch bus (F bus), and CPU memory access bus (M bus). (Note that the F bus is connected only to the read ports.)

  The F bus and M bus are used for access by the CPU, and the ID bus is used for access by the direct memory access controller.

  Each page of the on-chip large-capacity RAM has one read and write port and is connected to the internal CPU bus (IC bus), internal DMA bus (ID bus) and internal graphic buses 1 to 4 (IV1 to IV4). For 1-Mbyte version, pages 0 and 1 of the on-chip RAM for data retention are included in page 5 of the on-chip large-capacity RAM. Accordingly, pages 0 and 1 of the on-chip RAM for data retention are shared with the read and write port of page 5 of the on-chip large-capacity RAM. On the other hand, since the on-chip RAM for data retention is included in page 0 to 2 for 640-Kbyte version, it is shared with the read and write port of the same pages.

- Priority

  When the same page of the on-chip high-speed RAM is accessed from different buses simultaneously, the access is processed according to the priority. The priority is ID bus > M bus > F bus.

  When the same page of the on-chip large-capacity RAM is accessed from different buses simultaneously, the access is processed according to the priority. The priority is IV1 bus > IV2 bus > IV3 bus > IV4 bus > IC bus (when the IC bus does not have the bus mastership in the preceding bus cycle) > ID bus > IC bus (when the IC bus has the bus mastership in the preceding bus cycle)

- Number of access cycles

  On-chip high-speed RAM: the number of cycles for access to read or write from buses F and I is one cycle of Iϕ. Number of cycles for access from the ID bus depend on the ratio of the CPU clock (Iϕ) to the bus clock (Bϕ). Table 31.6 indicates number of cycles for access from the ID bus.

chanhvinh.com

Table 31.6 Number of Cycles for Access to On-Chip High-Speed RAM from the ID Bus

Read/Write	Ratio of Iϕ and Bϕ	Number of Access (Bϕ) Cycles
Read	1:1	3
	2:1	2
	3:1	2
	4:1	2
	6:1	1
	8:1	1
Write	1:1	2
	2:1	2
	3:1	2
	4:1	2
	6:1	1
	8:1	1

Note: For the settable ratios of Iϕ to Bϕ, see section 5, Clock Pulse Generator.

On-chip large-capacity RAM: the number of cycles for access to read or write from any bus is one cycle of Bϕ.

chanhvinh.com

31.2 Usage Notes

31.2.1 Page Conflict

When the same page of the on-chip high-speed RAM or the on-chip large-capacity RAM is accessed from different buses 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 bus.

31.2.2 RAME and RAMWE Bits

Before disabling memory operation or write access to the on-chip high-speed RAM through the RAME or RAMWE bit, be sure to read from any address and then write to the same address in each page; otherwise, the last written data in each page may not be actually written to the RAM.

```
// For page 0
  MOV.L  #H'FFF80000,R0
  MOV.L  @R0,R1
  MOV.L  R1,@R0

// For page 1
  MOV.L  #H'FFF84000,R0
  MOV.L  @R0,R1
  MOV.L  R1,@R0

// For page 2
  MOV.L  #H'FFF88000,R0
  MOV.L  @R0,R1
  MOV.L  R1,@R0

// For page 3
  MOV.L  #H'FFF8C000,R0
  MOV.L  @R0,R1
  MOV.L  R1,@R0
```

Figure 31.1 Examples of Read/Write

chanhvinh.com

31.2.3 Data Retention

Data in the on-chip high-speed RAM and 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 High-Speed RAM

Data are retained on a power-on reset by disabling the setting of either the RAME or RAMWE bit.

Data are not retained when the setting of the RAME and RAMWE bits are both enabled.

(b) 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.

(c) 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 High-Speed RAM and 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

Section 32 General Purpose I/O Ports

This LSI has 10 general purpose I/O ports: A, B, C, D, E, F, G, H, J, and K.

All port pins are multiplexed with other peripheral module pin functions.

Each port is provided with registers for selecting the pin functions and those I/O directions of multiplex pins, data registers for storing the pin data and port registers for reading the states of the pins.

32.1 Features

- By setting the control registers, multiplexed pin functions can be selectable.
- When the general I/O function or TIOC I/O function of multi-function timer pulse unit 2 is specified, the I/O direction can be selected by I/O register settings.

Table 32.1 Number of General Purpose I/O Pins

Port	SH7262	SH7264
A	4 I/O pins	
B	22 I/O pins	
C	9 I/O pins	11 I/O pins
D	16 I/O pins	
E	6 input pins with open-drain outputs	
F	13 I/O pins	
G	21 I/O pins	25 I/O pins
H	4 input pins	8 input pins
J	4 I/O pins	12 I/O pins
K	No pin	12 I/O pins
Total	99 pins (89 I/O pins, 6 input pins with open-drain outputs, and 4 input pins)	129 pins (115 I/O pins, 6 input pins with open-drain outputs, and 8 input pins)

chanhvinh.com

Tables 32.2 to 32.11 show the multiplex pins of this LSI. The registers and pin functions in the shaded cells are available only in the SH7264 Group. Furthermore, pin functions marked with * can be used in 640-Kbyte version only.

Table 32.2 Multiplexed Pins (Port A)

	$\overline{\text{RES}}$ Pin input	
	H	**L**
Port	**Function 1**	**Function 2**
A	PA3	MD_CLK0
	PA2	MD_CLK1
	PA1	MD_BOOT0
	PA0	MD_BOOT1

Note: The function 2 of port A is enabled in the state of $\overline{\text{RES}}$ = L and always general I/O functions in the state of $\overline{\text{RES}}$ = H.

chanhvinh.com

Table 32.3 Multiplexed Pins (Port B)

	Setting Mode Bit (PBnMD[1:0])		
	00	01	10
Setting Register	Function 1	Function 2	Function 3
PBCR5	PB22	A22	$\overline{CS4}$
	PB21	A21	SPDIF_OUT*
	PB20	A20	SPDIF_IN*
PBCR4	PB19	A19	TIOC4D
	PB18	A18	TIOC4C
	PB17	A17	TIOC4B
	PB16	A16	TIOC4A
PBCR3	PB15	A15	TIOC3D
	PB14	A14	TIOC3C
	PB13	A13	TIOC3B
	PB12	A12	TIOC3A
PBCR2	PB11	A11	TIOC2B
	PB10	A10	TIOC2A
	PB9	A9	TIOC1B
	PB8	A8	TIOC1A
PBCR1	PB7	A7	TIOC0D
	PB6	A6	TIOC0C
	PB5	A5	TIOC0B
	PB4	A4	TIOC0A
PBCR0	PB3	A3	—
	PB2	A2	—
	PB1	A1	—

chanhvinh.com

Table 32.4 Multiplexed Pins (Port C)

	Setting Mode Bit (PCnMD[1:0])			
	00	**01**	**10**	**11**
Setting Register	**Function 1**	**Function 2**	**Function 3**	**Function 4**
PCCR2	PC10	TIOC2B	—	—
	PC9	TIOC2A	—	—
	PC8	$\overline{CS3}$	TIOC4D	IRQ7
PCCR1	PC7	CKE	TIOC4C	IRQ6
	PC6	$\overline{CAS}$	TIOC4B	IRQ5
	PC5	$\overline{RAS}$	TIOC4A	IRQ4
	PC4	$\overline{WE1}$/DQMU/$\overline{WE}$	—	—
PCCR0	PC3	$\overline{WE0}$/DQML	—	—
	PC2	RD/$\overline{WR}$	—	—
	PC1	$\overline{RD}$	—	—
	PC0	$\overline{CS0}$	—	—

Table 32.5 Multiplexed Pins (Port D)

	Setting Mode Bit (PDnMD[1:0])		
	00	**01**	**10**
Setting Register	**Function 1**	**Function 2**	**Function 3**
PDCR3	PD15	D15/NAF7	PWM2H
	PD14	D14/NAF6	PWM2G
	PD13	D13/NAF5	PWM2F
	PD12	D12/NAF4	PWM2E
PDCR2	PD11	D11/NAF3	PWM2D
	PD10	D10/NAF2	PWM2C
	PD9	D9/NAF1	PWM2B
	PD8	D8/NAF0	PWM2A
PDCR1	PD7	D7/$\overline{FWE}$	PWM1H
	PD6	D6/FALE	PWM1G
	PD5	D5/FCLE	PWM1F
	PD4	D4/$\overline{FRE}$	PWM1E

chanhvinh.com

	Setting Mode Bit (PBnMD[1:0])		
	00	01	10
Setting Register	Function 1	Function 2	Function 3
PDCR0	PD3	D3	PWM1D
	PD2	D2	PWM1C
	PD1	D1	PWM1B
	PD0	D0	PWM1A

Note: The function 2 of bus state controller and /or the function of NAND flush memory controller change automatically. (See section 9, Bus State Controller.).

Table 32.6 Multiplexed Pins (Port E)

	Setting Mode Bit (PEnMD[2:0])					
	000	001	010	011	100	101
Setting Register	Function 1	Function 2	Function 3	Function 4	Function 5	Function 6
PECR1	PE5	SDA2	—	DV_HSYNC	—	—
	PE4	SCL2	—	DV_VSYNC	—	—
PECR0	PE3	SDA1	—	IRQ3	—	—
	PE2	SCL1	—	IRQ2	—	—
	PE1	SDA0	$\overline{\text{IOIS16}}$	IRQ1	TCLKA	$\overline{\text{ADTRG}}$
	PE0	SCL0	AUDIO_CLK	IRQ0	—	—

chanhvinh.com

Table 32.7 Multiplexed Pins (Port F)

	Setting Mode Bit (PFnMD[2:0])						
	000	**001**	**010**	**011**	**100**	**101**	**110**
Setting Register	**Function 1**	**Function 2**	**Function 3**	**Function 4**	**Function 5**	**Function 6**	**Function 7**
PFCR3	PF12	$\overline{BS}$	AUDIO_XOUT*	MISO0	TIOC3D	SPDIF_OUT	—
PFCR2	PF11	A25	SSIDATA3	MOSI0	TIOC3C	SPDIF_IN	—
	PF10	A24	SSIWS3	SSL00	TIOC3B	$\overline{FCE}$	—
	PF9	A23	SSISCK3	RSPCK0	TIOC3A	FRB	—
	PF8	$\overline{CE2B}$	SSIDATA3	DV_CLK	—	—	SD_CD*
PFCR1	PF7	$\overline{CE2A}$	SSIWS3	DV_DATA7	TCLKD	—	SD_WP*
	PF6	$\overline{CS6}/\overline{CE1B}$	SSISCK3	DV_DATA6	TCLKB	—	SD_D1*
	PF5	$\overline{CS5}/\overline{CE1A}$	SSIDATA2	DV_DATA5	TCLKC	—	SD_D0*
	PF4	$\overline{ICIOWR}/\overline{AH}$	SSIWS2	DV_DATA4	TxD3	—	SD_CLK*
PFCR0	PF3	$\overline{ICIORD}$	SSISCK2	DV_DATA3	RxD3	—	SD_CMD*
	PF2	$\overline{BACK}$	SSIDATA1	DV_DATA2	TxD2	DACK0	SD_D3*
	PF1	$\overline{BREQ}$	SSIWS1	DV_DATA1	RxD2	DREQ0	SD_D2*
	PF0	$\overline{WAIT}$	SSISCK1	DV_DATA0	SCK2	TEND0	—

chanhvinh.com

Table 32.8 Multiplexed Pins (Port G)

Setting Register	Setting Mode Bit (PGnMD[2:0])					
	000	001	010	011	100	101
	Function 1	Function 2	Function 3	Function 4	Function 5	Function 6
PGCR7	PG0	LCD_DATA0	SD_D2	PINT0	$\overline{\text{WDTOVF}}$	—
PGCR6	PG24	MISO1	TIOC0D	—	—	—
PGCR5	PG23	MOSI1	TIOC0C	—	—	—
	PG22	SSL10	TIOC0B	—	—	—
	PG21	RSPCK1	TIOC0A	—	—	—
	PG20	LCD_EXTCLK	—	MISO1	TxD7	—
PGCR4	PG19	LCD_CLK	TIOC2B	MOSI1	RxD7	—
	PG18	LCD_DE	TIOC2A	SSL10	TxD6	—
	PG17	LCD_HSYNC	TIOC1B	RSPCK1	RxD6	—
	PG16	LCD_VSYNC	TIOC1A	TxD3	$\overline{\text{CTS1}}$	—
PGCR3	PG15	LCD_DATA15	TIOC0D	RxD3	$\overline{\text{RTS1}}$	—
	PG14	LCD_DATA14	TIOC0C	—	SCK1	—
	PG13	LCD_DATA13	TIOC0B	—	TxD1	—
	PG12	LCD_DATA12	TIOC0A	—	RxD1	—
PGCR2	PG11	LCD_DATA11	SSITxD0	IRQ3	TxD5	SIOFTxD
	PG10	LCD_DATA10	SSIRxD0	IRQ2	RxD5	SIOFRxD
	PG9	LCD_DATA9	SSIWS0	—	TxD4	SIOFSYNC
	PG8	LCD_DATA8	SSISCK0	—	RxD4	SIOFSCK
PGCR1	PG7	LCD_DATA7	SD_CD	PINT7	IRQ7*	—
	PG6	LCD_DATA6	SD_WP	PINT6	IRQ6*	—
	PG5	LCD_DATA5	SD_D1	PINT5	IRQ5*	—
	PG4	LCD_DATA4	SD_D0	PINT4	IRQ4*	—
PGCR0	PG3	LCD_DATA3	SD_CLK	PINT3	—	—
	PG2	LCD_DATA2	SD_CMD	PINT2	—	—
	PG1	LCD_DATA1	SD_D3	PINT1	—	—

chanhvinh.com

Table 32.9 Multiplexed Pins (Port H)

	Setting Mode Bit (PHnMD)	
	0	1
Setting Register	Function 1	Function 2
PHCR1	PH7	AN7
	PH6	AN6
	PH5	AN5
	PH4	AN4
PHCR0	PH3	AN3
	PH2	AN2
	PH1	AN1
	PH0	AN0

Table 32.10 Multiplexed Pins (Port J)

	Setting Mode Bit (PJnMD[2:0])					
	000	001	010	011	100	101
Setting Register	Function 1	Function 2	Function 3	Function 4	Function 5	Function 6
PJCR2	PJ11	PWM2H	DACK1	—	—	—
	PJ10	PWM2G	DREQ1	—	—	—
	PJ9	PWM2F	TEND1	AUDIO_XOUT*	—	—
	PJ8	PWM2E	$\overline{\text{RTS3}}$	—	—	—
PJCR1	PJ7	TIOC1B	$\overline{\text{CTS3}}$	—	—	—
	PJ6	TIOC1A	SCK3	—	—	—
	PJ5	IERxD	TxD3	—	—	—
	PJ4	IETxD	RxD3	—	—	—
PJCR0	PJ3	CRx1	CRx0/CRx1	IRQ1	—	—
	PJ2	CTx1	CTx0&CTx1	$\overline{\text{CS2}}$	SCK0	LCD_M_DISP
	PJ1	CRx0	IERxD	IRQ0	RxD0	—
	PJ0	CTx0	IETxD	$\overline{\text{CS1}}$	TxD0	A0

chanhvinh.com

Table 32.11 Multiplexed Pins (Port K: Available Only in the SH7264 Group)

	Setting Mode Bit (PKnMD[1:0])		
	00	**01**	**10**
Setting Register	**Function 1**	**Function 2**	**Function 3**
PKCR2	PK11	PWM2D	SSITxD0
	PK10	PWM2C	SSIRxD0
	PK9	PWM2B	SSIWS0
	PK8	PWM2A	SSISCK0
PKCR1	PK7	PWM1H	SD_CD
	PK6	PWM1G	SD_WP
	PK5	PWM1F	SD_D1
	PK4	PWM1E	SD_D0
PKCR0	PK3	PWM1D	SD_CLK
	PK2	PWM1C	SD_CMD
	PK1	PWM1B	SD_D3
	PK0	PWM1A	SD_D2

chanhvinh.com

32.2 Register Descriptions

Table 32.12 lists the register configuration.

Table 32.12 Register Configuration

Port	Register Name	Abbreviation	R/W	Intial Value	Address	Access Size
A	Port A I/O register 0	PAIOR0	R/W	H'0000	H'FFFE3812	8, 16
	Port A data register 1	PADR1	R/W	H'0000	H'FFFE3814	8, 16*[2], 32*[2]
	Port A data register 0	PADR0	R/W	H'0000	H'FFFE3816	8, 16*[2]
	Port A port register 0	PAPR0	R	H'xxxx	H'FFFE381A	8, 16
B	Port B control register 5	PBCR5	R/W	H'0000/H'0001*[1]	H'FFFE3824	8, 16, 32
	Port B control register 4	PBCR4	R/W	H'0000/H'1111*[1]	H'FFFE3826	8, 16
	Port B control register 3	PBCR3	R/W	H'0000/H'1111*[1]	H'FFFE3828	8, 16, 32
	Port B control register 2	PBCR2	R/W	H'0000/H'1111*[1]	H'FFFE382A	8, 16
	Port B control register 1	PBCR1	R/W	H'0000/H'1111*[1]	H'FFFE382C	8, 16, 32
	Port B control register 0	PBCR0	R/W	H'0000/H'1110*[1]	H'FFFE382E	8, 16
	Port B I/O register 1	PBIOR1	R/W	H'0000	H'FFFE3830	8, 16, 32
	Port B I/O register 0	PBIOR0	R/W	H'0000	H'FFFE3832	8, 16
	Port B data register 1	PBDR1	R/W	H'0000	H'FFFE3834	8, 16, 32
	Port B data register 0	PBDR0	R/W	H'0000	H'FFFE3836	8, 16
	Port B port register 1	PBPR1	R	H'xxxx	H'FFFE3838	8, 16, 32
	Port B port register 0	PBPR0	R	H'xxxx	H'FFFE383A	8, 16
C	Port C control register 2	PCCR2	R/W	H'0000	H'FFFE384A	8, 16
	Port C control register 1	PCCR1	R/W	H'0000	H'FFFE384C	8, 16, 32
	Port C control register 0	PCCR0	R/W	H'0000/H'0011*[1]	H'FFFE384E	8, 16
	Port C I/O register 0	PCIOR0	R/W	H'0000	H'FFFE3852	8, 16
	Port C data register 0	PCDR0	R/W	H'0000	H'FFFE3856	8, 16
	Port C port register 0	PCPR0	R	H'xxxx	H'FFFE385A	8, 16

chanhvinh.com

Port	Register Name	Abbreviation	R/W	Intial Value	Address	AccessSize
D	Port D control register 3	PDCR3	R/W	H'0000/H'1111*[1]	H'FFFE3868	8, 16, 32
	Port D control register 2	PDCR2	R/W	H'0000/H'1111*[1]	H'FFFE386A	8, 16
	Port D control register 1	PDCR1	R/W	H'0000/H'1111*[1]	H'FFFE386C	8, 16, 32
	Port D control register 0	PDCR0	R/W	H'0000/H'1111*[1]	H'FFFE386E	8, 16
	Port D I/O register 0	PDIOR0	R/W	H'0000	H'FFFE3872	8, 16
	Port D data register 0	PDDR0	R/W	H'0000	H'FFFE3876	8, 16
	Port D port register 0	PDPR0	R	H'xxxx	H'FFFE387A	8, 16
E	Port E control register 1	PECR1	R/W	H'0000	H'FFFE388C	8, 16, 32
	Port E control register 0	PECR0	R/W	H'0000	H'FFFE388E	8, 16
	Port E I/O register 0	PEIOR0	R/W	H'0000	H'FFFE3892	8, 16
	Port E data register 0	PEDR0	R/W	H'0000	H'FFFE3896	8, 16
	Port E port register 0	PEPR0	R	H'xxxx	H'FFFE389A	8, 16
F	Port F control register 3	PFCR3	R/W	H'0000	H'FFFE38A8	8, 16, 32
	Port F control register 2	PFCR2	R/W	H'0000	H'FFFE38AA	8, 16
	Port F control register 1	PFCR1	R/W	H'0000	H'FFFE38AC	8, 16, 32
	Port F control register 0	PFCR0	R/W	H'0000	H'FFFE38AE	8, 16
	Port F I/O register 0	PFIOR0	R/W	H'0000	H'FFFE38B2	8, 16
	Port F data register 0	PFDR0	R/W	H'0000	H'FFFE38B6	8, 16
	Port F port register 0	PFPR0	R	H'xxxx	H'FFFE38BA	8, 16
G	Port G control register 7	PGCR7	R/W	H'0000	H'FFFE38C0	8*[3], 16, 32
	Port G control register 6	PGCR6	R/W	H'0000	H'FFFE38C2	8, 16
	Port G control register 5	PGCR5	R/W	H'0000	H'FFFE38C4	8, 16, 32
	Port G control register 4	PGCR4	R/W	H'0000	H'FFFE38C6	8, 16
	Port G control register 3	PGCR3	R/W	H'0000	H'FFFE38C8	8, 16, 32
	Port G control register 2	PGCR2	R/W	H'0000	H'FFFE38CA	8, 16
	Port G control register 1	PGCR1	R/W	H'0000	H'FFFE38CC	8, 16, 32
	Port G control register 0	PGCR0	R/W	H'0000	H'FFFE38CE	8, 16
	Port G I/O register 1	PGIOR1	R/W	H'0000	H'FFFE38D0	8, 16, 32
	Port G I/O register 0	PGIOR0	R/W	H'0000	H'FFFE38D2	8, 16
	Port G data register 1	PGDR1	R/W	H'0000	H'FFFE38D4	8, 16, 32
	Port G data register 0	PGDR0	R/W	H'0000	H'FFFE38D6	8, 16

chanhvinh.com

Port	Register Name	Abbreviation	R/W	Intial Value	Address	AccessSize
G	Port G port register 1	PGPR1	R	H'xxxx	H'FFFE38D8	8, 16, 32
	Port G port register 0	PGPR0	R	H'xxxx	H'FFFE38DA	8, 16
H	Port H control register 1	PHCR1	R/W	H'0000	H'FFFE38EC	8, 16, 32
	Port H control register 0	PHCR0	R/W	H'0000	H'FFFE38EE	8, 16
	Port H port register 0	PHPR0	R	H'xxxx	H'FFFE38FA	8, 16
J	Port J control register 2	PJCR2	R/W	H'0000	H'FFFE390A	8, 16
	Port J control register 1	PJCR1	R/W	H'0000	H'FFFE390C	8, 16, 32
	Port J control register 0	PJCR0	R/W	H'0000	H'FFFE390E	8, 16
	Port J I/O register 0	PJIOR0	R/W	H'0000	H'FFFE3912	8, 16
	Port J data register 0	PJDR0	R/W	H'0000	H'FFFE3916	8, 16
	Port J port register 0	PJPR0	R	H'xxxx	H'FFFE391A	8, 16
K	Port K control register 2	PKCR2	R/W	H'0000	H'FFFE392A	8, 16
	Port K control register 1	PKCR1	R/W	H'0000	H'FFFE392C	8, 16, 32
	Port K control register 0	PKCR0	R/W	H'0000	H'FFFE392E	8, 16
	Port K I/O register 0	PKIOR0	R/W	H'0000	H'FFFE3932	8, 16
	Port K data register 0	PKDR0	R/W	H'0000	H'FFFE3936	8, 16
	Port K port register 0	PKPR0	R	H'xxxx	H'FFFE393A	8, 16

Notes: 1. The initial value depends on the boot mode of the LSI.
2. In 16- or 32-bit access, the register can be read but cannot be written to.
3. In 8-bit access, the register can be read but cannot be written to.

chanhvinh.com

32.2.1 Port A I/O Register 0 (PAIOR0)

PAIOR0 is a 16-bit readable/writable register that is set the pins on port A as inputs or outputs. The PA3IOR0 to PA0IOR bits correspond to the PA3 PA0 pins, respectively. If a bit in PAIOR0 is set to 1, the corresponding pin on port A functions as output. If it is cleared to 0, the corresponding pin function as input. The direction (input or output) should only be modified by writing once for each pin.

Bits 15 to 4 in PAIOR0 are reserved. These bits are always read as 0. The write value should always be 0.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	PA3 IOR	PA2 IOR	PA1 IOR	PA0 IOR
Initial value:	0	0	0	0	0	0	0	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

32.2.2 Port A Data Registers 1, 0 (PADR1, PADR0)

PBDR1 and PBDR0 are 16-bit readable/writable registers that store port A data. The PA3DR to PA0DR bits correspond to the PA3 to PA0 pins, respectively.

When a pin function is general output, if a value is written to PADR1 or PADR0, that value is output from the pin, and if PADR1 or PADR0 is read, the register value is returned regardless of the pin state.

When a pin function is general input, if PADR1 or PADR0 is read, the pin state, not the register value, is returned directly. If a value is written to PADR1 or PADR0, although that value is written into PADR1 or PADR0, it does not affect the pin state. Table 32.13 summarizes PADR1 and PADR0 read/write operation.

(1) Port A Data Register 1 (PADR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	PA3 DR	-	-	-	-	-	-	-	PA2 DR
Initial value:	0	0	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
15 to 9	—	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
8	PA3DR	0	R/W	See table 32.13
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PA2DR	0	R/W	See table 32.13

(2) Port A Data Register 0 (PADR0)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	PA1DR	-	-	-	-	-	-	-	PA0DR
Initial value:	0	0	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
15 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PA1DR	0	R/W	See table 32.13
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PA0DR	0	R/W	See table 32.13

Table 32.13 Port A Data Registers 1, 0 (PADR1, PADR0) Read/Write Operation

- Bits 8 and 0 in PADR1 and Bits 8 and 0 in PADR0

PAIOR0	Pin Function	Read Operation	Write Operation
0	General input	Pin state	Can write to PADR1 and 0, but does not affect the pin state
1	General output	PADR1, 0 value	Value written is output from the pin

chanhvinh.com

32.2.3 Port A Port Register 0 (PAPR0)

PAPR0 is a 16-bit read-only register, in which the PA3PR to PA0PR bits correspond to the PA3 to PA0 pins, respectively. PAPR0 always returns the states of the pins.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	PA3PR	PA2PR	PA1PR	PA0PR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	PA3	PA2	PA1	PA0
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	Reserved These bits are always read as 0. The write value should always be 0.
3	PA3PR	Pin state	R	The pin state is returned. These bits cannot be modified.
2	PA2PR	Pin state	R	
1	PA1PR	Pin state	R	
0	PA0PR	Pin state	R	

32.2.4 Port B Control Registers 0 to 5 (PBCR0 to PBCR5)

PBCR0 to PBCR5 are 16-bit readable/writable registers that are used to select the functions of the multiplexed pins on port B.

(1) Port B Control Register 5 (PBCR5)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PB22MD[1:0]		-	-	PB21MD[1:0]		-	-	PB20MD[1:0]	
Initial value:	0	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	R	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	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
9, 8	PB22MD[1:0]	00	R/W	PB22 Mode Select the function of the PB22. 00: PB22 10: $\overline{CS4}$ 01: A22 11: Setting prohibited
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	PB21MD[1:0]	00	R/W	PB21 Mode Select the function of the PB21. 00: PB21 10: SPDIF_OUT (640-Kbyte version only) 01: A21 11: Setting prohibited Note: For 1-Mbyte version, bit 5 is reserved and always read as 0. The write value should always be 0.
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PB20MD[1:0]	00/01	R/W	PB20 Mode Select the function of the PB20. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PB20 (initial value) 01: A20 (initial value) / 01: A20 10: Setting prohibited / 10: SPDIF_IN (640-Kbyte version only) 11: Setting prohibited / 11: Setting prohibited Note: For 1-Mbyte version, bit 1 is reserved and always read as 0. The write value should always be 0.

chanhvinh.com

(2) Port B Control Register 4 (PBCR4)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PB19MD[1:0]		-	-	PB18MD[1:0]		-	-	PB17MD[1:0]		-	-	PB16MD[1:0]	
Initial value:	0	0	0	0/1	0	0	0	0/1	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
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	PB19MD[1:0]	00/01	R/W	PB19 Mode Select the function of the PB19. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PB19 (initial value) 01: A19 (initial value) / 01: A19 10: Setting prohibited / 10: TIOC4D 11: Setting prohibited / 11: Setting prohibited
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PB18MD[1:0]	00/01	R/W	PB18 Mode Select the function of the PB18. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PB18 (initial value) 01: A18 (initial value) / 01: A18 10: Setting prohibited / 10: TIOC4C 11: Setting prohibited / 11: Setting prohibited
7, 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, 4	PB17MD[1:0]	00/01	R/W	PB17 Mode Select the function of the PB17. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PB17 (initial value) 01: A17 (initial value) / 01: A17 10: Setting prohibited / 10: TIOC4B 11: Setting prohibited / 11: Setting prohibited
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PB16MD[1:0]	00/01	R/W	PB16 Mode Select the function of the PB16. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PB16 (initial value) 01: A16 (initial value) / 01: A16 10: Setting prohibited / 10: TIOC4A 11: Setting prohibited / 11: Setting prohibited

chanhvinh.com

(3) Port B Control Register 3 (PBCR3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PB15MD[1:0]		-	-	PB14MD[1:0]		-	-	PB13MD[1:0]		-	-	PB12MD[1:0]	
Initial value:	0	0	0	0/1	0	0	0	0/1	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
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	PB15MD[1:0]	00/01	R/W	PB15 Mode Select the function of the PB15. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PB15 (initial value) 01: A15 (initial value) / 01: A15 10: Setting prohibited / 10: TIOC3D 11: Setting prohibited / 11: Setting prohibited
11, 10	—	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
9, 8	PB14MD[1:0]	00/01	R/W	PB14 Mode Select the function of the PB14. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PB14 (initial value) 01: A14 (initial value) / 01: A14 (initial value) 10: Setting prohibited / 10: TIOC3C 11: Setting prohibited / 11: Setting prohibited
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	PB13MD[1:0]	00/01	R/W	PB13 Mode Select the function of the PB13. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PB13 (initial value) 01: A13 (initial value) / 01: A13 (initial value) 10: Setting prohibited / 10: TIOC3B 11: Setting prohibited / 11: Setting prohibited
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PB12MD[1:0]	00/01	R/W	PB12 Mode Select the function of the PB12. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PB12 (initial value) 01: A12 (initial value) / 01: A12 (initial value) 10: Setting prohibited / 10: TIOC3A 11: Setting prohibited / 11: Setting prohibited

chanhvinh.com

(4) Port B Control Register 2 (PBCR2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PB11MD[1:0]		-	-	PB10MD[1:0]		-	-	PB9MD[1:0]		-	-	PB8MD[1:0]	
Initial value:	0	0	0	0/1	0	0	0	0/1	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
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	PB11MD[1:0]	00/01	R/W	PB11 Mode Select the function of the PB11. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PB11 (initial value) 01: A11 (initial value) / 01: A11 (initial value) 10: Setting prohibited / 10: TIOC2B 11: Setting prohibited / 11: Setting prohibited
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PB10MD[1:0]	00/01	R/W	PB10 Mode Select the function of the PB10. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PB10 (initial value) 01: A10 (initial value) / 01: A10 (initial value) 10: Setting prohibited / 10: TIOC2A 11: Setting prohibited / 11: Setting prohibited
7, 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, 4	PB9MD[1:0]	00/01	R/W	PB9 Mode Select the function of the PB9. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PB9 (initial value) 01: A9 (initial value) / 01: A9 (initial value) 10: Setting prohibited / 10: TIOC1B 11: Setting prohibited / 11: Setting prohibited
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PB8MD[1:0]	00/01	R/W	PB8 Mode Select the function of the PB8. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PB8 (initial value) 01: A8 (initial value) / 01: A8 (initial value) 10: Setting prohibited / 10: TIOC1A 11: Setting prohibited / 11: Setting prohibited

(5) Port B Control Register 1 (PBCR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PB7MD[1:0]		-	-	PB6MD[1:0]		-	-	PB5MD[1:0]		-	-	PB4MD[1:0]	
Initial value:	0	0	0	0/1	0	0	0	0/1	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
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	PB7MD[1:0]	00/01	R/W	PB7 Mode Select the function of the PB7. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PB7 (initial mode) 01: A7 (initial value) / 01: A7 (initial value) 10: Setting prohibited / 10: TIOC0D 11: Setting prohibited / 11: Setting prohibited
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PB6MD[1:0]	00/01	R/W	PB6 Mode Select the function of the PB6. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PB6 (initial value) 01: A6 (initial value) / 01: A6 (initial value) 10: Setting prohibited / 10: TIOC0C 11: Setting prohibited / 11: Setting prohibited
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	PB5MD[1:0]	00/01	R/W	PB5 Mode Select the function of the PB5. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PB5 (initial value) 01: A5 (initial value) / 01: A5 (initial value) 10: Setting prohibited / 10: TIOC0B 11: Setting prohibited / 11: Setting prohibited
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	PB4MD[1:0]	00/01	R/W	PB4 Mode Select the function of the PB4. Boot mode 0 — Boot mode 1 to 3 00: Setting prohibited — 00: PB4 (initial value) 01: A4 (initial value) — 01: A4 (initial value) 10: Setting prohibited — 10: TIOC0A 11: Setting prohibited — 11: Setting prohibited

(6) Port B Control Register 0 (PBCR0)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PB3MD0	-	-	-	PB2MD0	-	-	-	PB1MD0	-	-	-	-
Initial value:	0	0	0	0/1	0	0	0	0/1	0	0	0	0/1	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

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	PB3MD0	0/1	R/W	PB3 Mode Select the function of the PB3. Boot mode 0 — Boot mode 1 to 3 0: Setting prohibited — 0: PB3 (initial value) 1: A3 (initial value) — 1: A3
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PB2MD0	0/1	R/W	PB2 Mode Select the function of the PB2. Boot mode 0 — Boot mode 1 to 3 0: Setting prohibited — 0: PB2 (initial value) 1: A2 (initial value) — 1: A2

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	PB1MD0	0/1	R/W	PB1 Mode Select the function of the PB1. Boot mode 0: 0: Setting prohibited; 1: A1 (initial value) Boot mode 1 to 3: 0: PB1 (initial value); 1: A1
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

32.2.5 Port B I/O Registers 0, 1 (PBIOR0, PBIOR1)

PBIOR0 and PBIOR1 are 16-bit readable/writable registers that are used to set the pins on port B as inputs or outputs. The PB22IOR to PB1IOR bits correspond to the PB22 to PB1 pins, respectively. PBIOR1 and PBIOR0 are enabled when the port B pins are functioning as general-purpose I/O (PB22 to PB1) or TIOC I/O of multi-function timer pulse unit 2. In other states, they are disabled. If a bit in PBIOR1 or PBIOR0 is set to 1, the corresponding pin on port B functions as output pin. If it is cleared to 0, the corresponding pin functions as an input pin.

Bits 15 to 7 in PBIOR1 and bit 0 in PBIOR0 are reserved. These bits are always read as 0. The write value should always be 0.

(1) Port B I/O Register 1 (PBIOR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	PB22IOR	PB21IOR	PB20IOR	PB19IOR	PB18IOR	PB17IOR	PB16IOR
Initial value:	0	0	0	0	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

(2) Port B I/O Register 0 (PBIOR0)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PB15IOR	PB14IOR	PB13IOR	PB12IOR	PB11IOR	PB10IOR	PB9IOR	PB8IOR	PB7IOR	PB6IOR	PB5IOR	PB4IOR	PB3IOR	PB2IOR	PB1IOR	-
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

chanhvinh.com

32.2.6 Port B Data Registers 0, 1 (PBDR0, PBDR1)

PBDR0 and PBDR1 are 16-bit readable/writable registers that store port B data. The PB22DR to PB1DR bits correspond to the PB22 to PB1 pins, respectively.

When a pin function is general output, if a value is written to PBDR1 or PBDR0, the value is output directly from the pin, and if PBDR is read, the register value is returned directly regardless of the pin state.

When a pin function is general input, if PBDR1 or PBDR0 is read, the pin state, not the register value, is returned directly. If a value is written to PBDR1 or PBDR0, although that value is written into PBDR1 or PBDR0, it does not affect the pin state. Table 32.14 summarizes PBDR1/PBDR0 read/write operation.

(1) Port B Data Register 1 (PBDR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	PB22DR	PB21DR	PB20DR	PB19DR	PB18DR	PB17DR	PB16DR
Initial value:	0	0	0	0	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
15 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6	PB22DR	0	R/W	See table 32.14
5	PB21DR	0	R/W	
4	PB20DR	0	R/W	
3	PB19DR	0	R/W	
2	PB18DR	0	R/W	
1	PB17DR	0	R/W	
0	PB16DR	0	R/W	

chanhvinh.com

(2) Port B Data Register 0 (PBDR0)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PB15 DR	PB14 DR	PB13 DR	PB12 DR	PB11 DR	PB10 DR	PB9 DR	PB8 DR	PB7 DR	PB6 DR	PB5 DR	PB4 DR	PB3 DR	PB2 DR	PB1 DR	-
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

Bit	Bit Name	Initial Value	R/W	Description
15	PB15DR	0	R/W	See table 32.14
14	PB14DR	0	R/W	
13	PB13DR	0	R/W	
12	PB12DR	0	R/W	
11	PB11DR	0	R/W	
10	PB10DR	0	R/W	
9	PB9DR	0	R/W	
8	PB8DR	0	R/W	
7	PB7DR	0	R/W	
6	PB6DR	0	R/W	
5	PB5DR	0	R/W	
4	PB4DR	0	R/W	
3	PB3DR	0	R/W	
2	PB2DR	0	R/W	
1	PB1DR	0	R/W	
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

chanhvinh.com

Table 32.14 Port B Data Registers 1, 0 (PBDR1, PBDR0) Read/Write Operation

- Bits 6 to 0 of PBDR1 and Bits 15 to 1 of PBDR0

PBIOR1, 0	Pin Function	Read Operation	Write Operation
0	General input	Pin state	Can write to PBDR0/PBDR1, but it has no effect on the pin state.
	Other than general input	Pin state	Can write to PBDR0/PBDR1, but it has no effect on the pin state.
1	General output	PBDR0/PBDR1 value	Value written is output to the pin
	Other than general output	PBDR0/PBDR1 value	Can write to PBDR0/PBDR1, but it has no effect on the pin state.

32.2.7 Port B Port Registers 0, 1 (PBPR0, PBPR1)

PBPR (PBPR0, PBPR1) is 16-bit read-only register, in which the PB22PR to PB1PR bits correspond to the PB22 to PB1 pins, respectively. PBPR always returns the states of the pins regardless of the PBCR5 to PBCR0 settings.

(1) Port B Port Register 1 (PBPR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	PB22PR	PB21PR	PB20PR	PB19PR	PB18PR	PB17PR	PB16PR
Initial value:	0	0	0	0	0	0	0	0	0	PB22	PB21	PB20	PB19	PB18	PB17	PB16
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	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6	PB22PR	Pin state	R	The pin state is returned. These bits cannot be modified.
5	PB21PR	Pin state	R	
4	PB20PR	Pin state	R	
3	PB19PR	Pin state	R	
2	PB18PR	Pin state	R	
1	PB17PR	Pin state	R	
0	PB16PR	Pin state	R	

chanhvinh.com

(2) Port B Port Register 0 (PBPR0)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PB15PR	PB14PR	PB13PR	PB12PR	PB11PR	PB10PR	PB9PR	PB8PR	PB7PR	PB6PR	PB5PR	PB4PR	PB3PR	PB2PR	PB1PR	-
Initial value:	PB15	PB14	PB13	PB12	PB11	PB10	PB9	PB8	PB7	PB6	PB5	PB4	PB3	PB2	PB1	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	PB15PR	Pin state	R	The pin state is returned. These bits cannot be modified.
14	PB14PR	Pin state	R	
13	PB13PR	Pin state	R	
12	PB12PR	Pin state	R	
11	PB11PR	Pin state	R	
10	PB10PR	Pin state	R	
9	PB9PR	Pin state	R	
8	PB8PR	Pin state	R	
7	PB7PR	Pin state	R	
6	PB6PR	Pin state	R	
5	PB5PR	Pin state	R	
4	PB4PR	Pin state	R	
3	PB3PR	Pin state	R	
2	PB2PR	Pin state	R	
1	PB1PR	Pin state	R	
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

chanhvinh.com

32.2.8 Port C Control Registers 0 to 2 (PCCR0 to PCCR2)

PCCR0 to PCCR2 are 16-bit readable/writable registers that are used to select the functions of the multiplexed pins on port C.

(1) Port C Control Register 2 (PCCR2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	PC10MD0	-	-	-	PC9MD0	-	-	PC8MD[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
15 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PC10MD0	0	R/W	PC10 Mode Select the function of the PC10. 0: PC10 1: TIOC2B Note: This bit is reserved in the SH7262 Group. It is always read as 0. The write value should always be 0.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PC9MD0	0	R/W	PC9 Mode Select the function of the PC9. 0: PC9 1: TIOC2A Note: This bit is reserved in the SH7262 Group. It is always read as 0. The write value should always be 0.
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	PC8MD[1:0]	00	R/W	PC8 Mode Select the function of the PC8 00: PC8 10:TIOC4D 01: $\overline{CS3}$ 11: IRQ7

(2) Port C Control Register 1 (PCCR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PC7MD[1:0]		-	-	PC6MD[1:0]		-	-	PC5MD[1:0]		-	-	-	PC4MD0
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	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	PC7MD[1:0]	00	R/W	PC7 Mode Select the function of the PC7 00: PC7 10: TIOC4C 01: CKE 11: IRQ6
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PC6MD[1:0]	00	R/W	PC6 Mode Select the function of the PC6 00: PC6 10: TIOC4B 01: $\overline{CAS}$ 11: IRQ5
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	PC5MD[1:0]	00	R/W	PC5 Mode Select the function of the PC5 00: PC5 10: TIOC4A 01: $\overline{RAS}$ 11: IRQ4

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PC4MD0	0	R/W	PC4 Mode Select the function of the PC4 0: PC4 1: $\overline{WE1}$/DQMU/$\overline{WE}$

(3) Port C Control Register 0 (PCCR0)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PC3MD0	-	-	-	PC2MD0	-	-	-	PC1MD0	-	-	-	PC0MD0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0/1	0	0	0	0/1
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
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	PC3MD0	0	R/W	PC3 Mode Select the function of the PC3 0: PC3 1: $\overline{WE0}$/DQML
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PC2MD0	0	R/W	PC2 Mode Select the function of the PC2 0: PC2 1: RD/$\overline{WR}$
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	PC1MD0	0/1	R/W	PC1 Mode Select the function of the PC1 Boot mode 0 — Boot mode 1 to 3 0: Setting prohibited — 0: PC1 (initial value) 1: $\overline{RD}$ (initial value) — 1: $\overline{RD}$
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PC0MD0	0/1	R/W	PC0 Mode Select the function of the PC0 Boot mode 0 — Boot mode 1 to 3 0: Setting prohibited — 0: PC0 (initial value) 1: $\overline{CS0}$ (initial value) — 1: $\overline{CS0}$

32.2.9 Port C I/O Register 0 (PCIOR0)

PCIOR0 is a 16-bit readable/writable register that is used to set the pins on port C as inputs or outputs. The PC10IOR to PC0IOR bits correspond to the PC10 to PC0 pins, respectively. PCIOR0 is enabled when the port C pins are functioning as general-purpose I/O (PC10 to PC0) or TIOC I/O of multi-function timer pulse unit 2. In other states, PCIOR0 is disabled. If a bit in PCIOR0 is set to 1, the corresponding pin on port C functions as an output pin. If it is cleared to 0, the corresponding pin functions as an input pin.

Bits 15 to 11 in PCIOR0, and bits 10 and 9 in PCIOR0 in the SH7262 Group are reserved. These bits are always read as 0. The write value should always be 0.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	PC10 IOR	PC9 IOR	PC8 IOR	PC7 IOR	PC6 IOR	PC5 IOR	PC4 IOR	PC3 IOR	PC2 IOR	PC1 IOR	PC0 IOR
Initial value:	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

chanhvinh.com

32.2.10 Port C Data Register 0 (PCDR0)

PCDR0 is a 16-bit readable/writable register that stores port C data. The PC10DR to PC0DR bits correspond to the PC10 to PC0 pins, respectively.

When a pin function is general output, if a value is written to PCDR0, that value is output directly from the pin, and if PCDR0 is read, the register value is returned directly regardless of the pin state.

When a pin function is general input, if PCDR0 is read, the pin state, not the register value, is returned directly. If a value is written to PCDR0, although that value is written into PCDR0, it does not affect the pin state. Table 32.15 summarizes PCDR0 read/write operation.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	PC10DR	PC9DR	PC8DR	PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR
Initial value:	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	PC10DR	0	R/W	See table 32.15
9	PC9DR	0	R/W	Note: Bits 10 and 9 are reserved in the SH7262 Group. These bits are always read as 0. The write value should always be 0.
8	PC8DR	0	R/W	
7	PC7DR	0	R/W	
6	PC6DR	0	R/W	
5	PC5DR	0	R/W	
4	PC4DR	0	R/W	
3	PC3DR	0	R/W	
2	PC2DR	0	R/W	
1	PC1DR	0	R/W	
0	PC0DR	0	R/W	

chanhvinh.com

Table 32.15 Port C Data Register 0 (PCDR0) Read/Write Operation

- Bits 10 to 0 of PCDR0

PCIOR0	Pin Function	Read Operation	Write Operation
0	General input	Pin state	Can write to PCDR0, but it has no effect on the pin state.
	Other than general input	Pin state	Can write to PCDR0, but it has no effect on the pin state.
1	General output	PCDR0 value	Value written is output from pin
	Other than general output	PCDR0 value	Can write to PCDR0, but it has no effect on the pin state

chanhvinh.com

32.2.11 Port C Port Register 0 (PCPR0)

PCPR0 is a 16-bit read-only register, in which the PC10PR to PC0PR bits correspond to the PC10 to PC0 pins, respectively. PCPR0 always returns the states of the pins regardless of the PCCR0 to PCCR2 settings.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	PC10PR	PC9PR	PC8PR	PC7PR	PC6PR	PC5PR	PC4PR	PC3PR	PC2PR	PC1PR	PC0PR
Initial value:	0	0	0	0	0	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
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	PC10PR	Pin state	R	The pin state is returned. These bits cannot be modified. Note: Bits 10 and 9 are reserved in the SH7262 Group. These bits are always read as 0. The write value should always be 0.
9	PC9PR	Pin state	R	
8	PC8PR	Pin state	R	
7	PC7PR	Pin state	R	
6	PC6PR	Pin state	R	
5	PC5PR	Pin state	R	
4	PC4PR	Pin state	R	
3	PC3PR	Pin state	R	
2	PC2PR	Pin state	R	
1	PC1PR	Pin state	R	
0	PC0PR	Pin state	R	

chanhvinh.com

32.2.12 Port D Control Register 0 to 3 (PDCR0 to PDCR3)

PDCR0 to PDCR3 are 16-bit readable/writable registers that are used to select the functions of the multiplexed pins on port D.

(1) Port D Control Register 3 (PDCR3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PD15MD[1:0]		-	-	PD14MD[1:0]		-	-	PD13MD[1:0]		-	-	PD12MD[1:0]	
Initial value:	0	0	0	0/1	0	0	0	0/1	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
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	PD15MD[1:0]	00/01	R/W	PD15 Mode Select the function of the PD15. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PD15 (initial value) 01: D15/NAF7 (initial value) / 01: D15/NAF7 10: Setting prohibited / 10: PWM2H 11: Setting prohibited / 11: Setting prohibited
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PD14MD[1:0]	00/01	R/W	PD14 Mode Select the function of the PD14. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PD14 (initial value) 01: D14/NAF6 (initial value) / 01: D14/NAF6 10: Setting prohibited / 10: PWM2G 11: Setting prohibited / 11: Setting prohibited
7, 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, 4	PD13MD[1:0]	00/01	R/W	PD13 Mode Select the function of the PD13. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PD13 (initial value) 01: D13/NAF5 (initial value) / 01: D13/NAF5 10: Setting prohibited / 10: PWM2F 11: Setting prohibited / 11: Setting prohibited
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PD12MD[1:0]	00/01	R/W	PD12 Mode Select the function of the PD12. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PD12 (initial value) 01: D12/NAF4 (initial value) / 01: D12/NAF4 10: Setting prohibited / 10: PWM2E 11: Setting prohibited / 11: Setting prohibited

(2) Port D Control Register 2 (PDCR2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PD11MD[1:0]		-	-	PD10MD[1:0]		-	-	PD9MD[1:0]		-	-	PD8MD[1:0]	
Initial value:	0	0	0	0/1	0	0	0	0/1	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
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	PD11MD[1:0]	00/01	R/W	PD11 Mode Select the function of the PD11. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PD11 (initial value) 01: D11/NAF3 (initial value) / 01: D11/NAF3 10: Setting prohibited / 10: PWM2D 11: Setting prohibited / 11: Setting prohibited
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PD10MD[1:0]	00/01	R/W	PD10 Mode Select the function of the PD10. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PD10 (initial value) 01: D10/NAF2 (initial value) / 01: D10/NAF2 10: Setting prohibited / 10: PWM2C 11: Setting prohibited / 11: Setting prohibited
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	PD9MD[1:0]	00/01	R/W	PD9 Mode Select the function of the PD9. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PD9 (initial value) 01: D9/NAF1 (initial value) / 01: D10/NAF1 10: Setting prohibited / 10: PWM2B 11: Setting prohibited / 11: Setting prohibited
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	PD8MD[1:0]	00/01	R/W	PD8 Mode Select the function of the PD8. Boot mode 0 — Boot mode 1 to 3 00: Setting prohibited — 00: PD8 (initial value) 01: D8/NAF0 (initial value) — 01: D8/NAF0 10: Setting prohibited — 10: PWM2A 11: Setting prohibited — 11: Setting prohibited

(3) Port D Control Register 1 (PDCR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PD7MD[1:0]		-	-	PD6MD[1:0]		-	-	PD5MD[1:0]		-	-	PD4MD[1:0]	
Initial value:	0	0	0	0/1	0	0	0	0/1	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
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	PD7MD[1:0]	00/01	R/W	PD7 Mode Select the function of the PD7. Boot mode 0 — Boot mode 1 to 3 00: Setting prohibited — 00: PD7 (initial value) 01: D7/$\overline{\text{FWE}}$ (initial value) — 01: D7/$\overline{\text{FWE}}$ 10: Setting prohibited — 10: PWM1H 11: Setting prohibited — 11: Setting prohibited
11, 10	—	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
9, 8	PD6MD[1:0]	00/01	R/W	PD6 Mode Select the function of the PD6. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PD6 (initial value) 01: D6/FALE (initial value) / 01: D6/FALE 10: Setting prohibited / 10: PWM1G 11: Setting prohibited / 11: Setting prohibited
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	PD5MD[1:0]	00/01	R/W	PD5 Mode Select the function of the PD5. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PD5 (initial value) 01: D5/FCLE (initial value) / 01: D5/FCLE 10: Setting prohibited / 10: PWM1F 11: Setting prohibited / 11: Setting prohibited
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PD4MD[1:0]	00/01	R/W	PD4 Mode Select the function of the PD4. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PD4 (initial value) 01: D4/$\overline{\text{FRE}}$ (initial value) / 01: D4/$\overline{\text{FRE}}$ 10: Setting prohibited / 10: PWM1E 11: Setting prohibited / 11: Setting prohibited

chanhvinh.com

(4) Port D Control Register 0 (PDCR0)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PD3MD[1:0]		-	-	PD2MD[1:0]		-	-	PD1MD[1:0]		-	-	PD0MD[1:0]	
Initial value:	0	0	0	0/1	0	0	0	0/1	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
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	PD3MD[1:0]	00/01	R/W	PD3 Mode Select the function of the PD3. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PD3 (initial value) 01: D3 (initial value) / 01: D3 10: Setting prohibited / 10: PWM1D 11: Setting prohibited / 11: Setting prohibited
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PD2MD[1:0]	00/01	R/W	PD2 Mode Select the function of the PD2. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PD2 (initial value) 01: D2 (initial value) / 01: D2 10: Setting prohibited / 10: PWM1C 11: Setting prohibited / 11: Setting prohibited
7, 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, 4	PD1MD[1:0]	00/01	R/W	PD1 Mode Select the function of the PD1. Boot mode 0 — Boot mode 1 to 3 00: Setting prohibited — 00: PD1 (initial value) 01: D1 (initial value) — 01: D1 10: Setting prohibited — 10: PWM1B 11: Setting prohibited — 11: Setting prohibited
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PD0MD[1:0]	00/01	R/W	PD0 Mode Select the function of the PD0.R Boot mode 0 — Boot mode 1 to 3 00: Setting prohibited — 00: PD0 (initial value) 01: D0 (initial value) — 01: D0 10: Setting prohibited — 10: PWM1A 11: Setting prohibited — 11: Setting prohibited

32.2.13 Port D I/O Register 0 (PDIOR0)

PDIOR0 is a 16-bit readable/writable register that is used to set the pins on port D as inputs or outputs. The PD15IOR to PD0IOR bits correspond to the PD15 to PD0 pins, respectively. The setting of PDIOR0 is valid for the pins for which general I/O function is selected and has no effect on the pins for which other function is selected. If a bit in PDIOR0 is set to 1, the corresponding pin on port D functions as an output. If it is cleared to 0, the corresponding pin functions as an input.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD15 IOR	PD14 IOR	PD13 IOR	PD12 IOR	PD11 IOR	PD10 IOR	PD9 IOR	PD8 IOR	PD7 IOR	PD6 IOR	PD5 IOR	PD4 IOR	PD3 IOR	PD2 IOR	PD1 IOR	PD0 IOR
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

chanhvinh.com

32.2.14 Port D Port Registers 0 (PDDR0)

PDDR0 is a 16-bit readable/writable register that stores port D data. The PD15DR to PD0DR bits correspond to the PD15 to PD0 pins, respectively.

When a pin function is general output, if a value is written to PDDR0, that value is output directly from the pin, and if PDDR0 is read, the register value is returned directly regardless of the pin state.

When a pin function is general input, if PDDR0 is read, the pin state, not the register value, is returned directly. If a value is written to PDDR0, although that value is written into PDDR0, it does not affect the pin state. Table 32.16 summarizes PDDR0 read/write operation.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD15 DR	PD14 DR	PD13 DR	PD12 DR	PD11 DR	PD10 DR	PD9 DR	PD8 DR	PD7 DR	PD6 DR	PD5 DR	PD4 DR	PD3 DR	PD2 DR	PD1 DR	PD0 DR
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	PD15DR	0	R/W	See table 32.16
14	PD14DR	0	R/W	
13	PD13DR	0	R/W	
12	PD12DR	0	R/W	
11	PD11DR	0	R/W	
10	PD10DR	0	R/W	
9	PD9DR	0	R/W	
8	PD8DR	0	R/W	
7	PD7DR	0	R/W	
6	PD6DR	0	R/W	
5	PD5DR	0	R/W	
4	PD4DR	0	R/W	
3	PD3DR	0	R/W	
2	PD2DR	0	R/W	
1	PD1DR	0	R/W	
0	PD0DR	0	R/W	

chanhvinh.com

Table 32.16 Port D Data Register 0 (PDDR0) Read/Write Operation

- Bits 15 to 0 of PDDR0

PCIOR0	Pin Function	Read Operation	Write Operation
0	General input	Pin state	Can write to PDDR0, but it has no effect on the pin state.
	Other than general input	Pin state	Can write to PDDR0, but it has no effect on the pin state.
1	General output	PDDR0 value	Value written is output from pin
	Other than general output	PDDR0 value	Can write to PDDR0, but it has no effect on the pin state

chanhvinh.com

32.2.15 Port D Port Registers 0 (PDPR0)

PDPR0 is a 16-bit read-only register, in which the PD15PR to PD0PR bits correspond to the PD15 to PD0 pins, respectively. PDPR0 always returns the states of the pins regardless of the PDCR0 to PDCR3 settings.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD15PR	PD14PR	PD13PR	PD12PR	PD11PR	PD10PR	PD9PR	PD8PR	PD7PR	PD6PR	PD5PR	PD4PR	PD3PR	PD2PR	PD1PR	PD0PR
Initial value:	PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
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	PD15PR	Pin state	R	The pin state is returned. These bits cannot be modified.
14	PD14PR	Pin state	R	
13	PD13PR	Pin state	R	
12	PD12PR	Pin state	R	
11	PD11PR	Pin state	R	
10	PD10PR	Pin state	R	
9	PD9PR	Pin state	R	
8	PD8PR	Pin state	R	
7	PD7PR	Pin state	R	
6	PD6PR	Pin state	R	
5	PD5PR	Pin state	R	
4	PD4PR	Pin state	R	
3	PD3PR	Pin state	R	
2	PD2PR	Pin state	R	
1	PD1PR	Pin state	R	
0	PD0PR	Pin state	R	

chanhvinh.com

32.2.16 Port E Control Registers 0, 1 (PECR0, PECR1)

PECR1 and PECR0 are 16-bit readable/writable registers that are used to select the functions of the multiplexed pins on port E.

(1) Port E Control Register 1 (PECR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PE5MD[1:0]		-	-	PE4MD[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	Bit Name	Initial Value	R/W	Description
15 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	PE5MD[1:0]	00	R/W	PE5 Mode Select the function of the PE5. 00: PE5 10: Setting prohibited 01: SDA2 11: DV_HSYNC
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PE4MD[1:0]	00	R/W	PE4 Mode Select the function of the PE4. 00: PE4 10: Setting prohibited 01: SCL2 11: DV_VSYNC

chanhvinh.com

(2) Port E Control Register 0 (PECR0)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PE3MD[1:0]		-	-	PE2MD[1:0]		-	PE1MD[2:0]			-	-	PE0MD[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/W	R/W	R/W	R	R	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	PE3MD[1:0]	00	R/W	PE3 Mode Select the function of the PE3. 00: PE3 10: Setting prohibited 01: SDA1 11: IRQ3
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PE2MD[1:0]	00	R/W	PE2 Mode Select the function of the PE2. 00: PE2 10: Setting prohibited 01: SCL1 11: IRQ2
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	PE1MD[2:0]	000	R/W	PE1 Mode Select the function of the PE1. 000: PE1 100: TCLKA 001: SDA0 101: $\overline{\text{ADTRG}}$ 010: $\overline{\text{IOIS16}}$ 110: Setting prohibited 011: IRQ1 111: Setting prohibited
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	PE0MD[1:0]	00	R/W	PE0 Mode Select the function of the PE0. 00: PE0 10: AUDIO_CLK 01: SCL0 11: IRQ0

32.2.17 Port E I/O Register 0 (PEIOR0)

PEIOR0 is a 16-bit readable/writable register that is used to set the pins on port F as inputs or outputs. The PE5IOR to PE0IOR bits correspond to the PE5 to PE0 pins respectively. PEIOR0 is enabled when the port E pins are functioning as general-purpose inputs/outputs (PE5 to PE0). In other states, it is disabled. If a bit in PEIOR0 is set to 1, the corresponding pin on port E functions as an output pin. If it is cleared to 0, the corresponding pin functions as an input pin.

Bits 15 to 6 in PEIOR0 are reserved. This bit is always read as 0. The write value should always be 0.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PE5 IOR	PE4 IOR	PE3 IOR	PE2 IOR	PE1 IOR	PE0 IOR
Initial value:	0	0	0	0	0	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

32.2.18 Port E Data Register 0 (PEDR0)

PEDR0 is a 16-bit readable/writable register that stores port E data. The PE5DR to PE0DR bits correspond to the PE5 to PE0 pins, respectively.

6 pins on Port E are open-drain outputs. When a pin function is general output, if 0 is written to PEDR0, 0 is output from the pin and if 1 is written to, the pin will be in the high-impedance state. If PEDR0 is read, the register value is returned directly regardless of the pin state.

When a pin function is general input, if PEDR0 is read, the pin state, not the register value, is returned directly. If a value is written to PEDR0, although that value is written into PEDR0, it does not affect the pin state. Table 32.17 summarizes PEDR0 read/write operation.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PE5 DR	PE4 DR	PE3 DR	PE2 DR	PE1 DR	PE0 DR
Initial value:	0	0	0	0	0	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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
15 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	PE5DR	0	R/W	See table 32.17
4	PE4DR	0	R/W	
3	PE3DR	0	R/W	
2	PE2DR	0	R/W	
1	PE1DR	0	R/W	
0	PE0DR	0	R/W	

Table 32.17 Port E Data Register 0 (PEDR0) Read/Write Operation

- Bits 5 to 0 of PEDR0

PEIOR0	Pin Operation	Read Operation	Write Operation
0	General input	Pin state	Can write to PEDR0, but it has no effect on the pin state.
	Other than general input	Pin state	Can write to PEDR0, but it has no effect on the pin state
1	General output	PEDR0 value	When PexDR=0, 0 outputs from the pin. When PexDR=1, the pin is in the high-impedance state.
	Other than general output	PEDR0 value	Can write to PEDR0, but it has no effect on the pin state

chanhvinh.com

32.2.19 Port E Port Register 0 (PEPR0)

PEPR0 is a 16-bit read-only register, in which the PE5PR to PE0PR bits correspond to the PE5 to PE0 pins, respectively. PEPR0 always returns the states of the pins regardless of the PECR0 and PECR1 settings.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PE5PR	PE4PR	PE3PR	PE2PR	PE1PR	PE0PR
Initial value:	0	0	0	0	0	0	0	0	0	0	PE5	PE4	PE3	PE2	PE1	PE0
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 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	PE5PR	Pin state	R	The pin state is returned. These bits cannot be modified.
4	PE4PR	Pin state	R	
3	PE3PR	Pin state	R	
2	PE2PR	Pin state	R	
1	PE1PR	Pin state	R	
0	PE0PR	Pin state	R	

chanhvinh.com

32.2.20 Port F Control Register 0 to 3 (PFCR0 to PFCR3)

PFCR0 to PFCR3 are 16-bit readable/writable registers that are used to select the functions of the multiplexed pins on port F.

(1) Port F Control Register 3 (PFCR3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	PF12MD[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
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	PF12MD[2:0]	000	R/W	PE12 Mode Select the function of the PE12. 000: PE12 001: $\overline{BS}$ 010: AUDIO_XOUT (640-Kbyte version only) 011: MISO0 100: TIOC3D 101: SPDIF_OUT 110: Setting prohibited 111: Setting prohibited

(2) Port F Control Register 2 (PFCR2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PF11MD[2:0]			-	PF10MD[2:0]			-	PF9MD[2:0]			-	PF8MD[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	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
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	PF11MD[2:0]	000	R/W	PF11 Mode Select the function of the PF11. 000: PF11 100: TIOC3C 001: A25 101: SPDIF_IN 010: SSIDATA3 110: Setting prohibited 011: MOSI0 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PF10MD[2:0]	000	R/W	PF10 Mode Select the function of the PF10. 000: PF10 100: TIOC3B 001: A24 101: $\overline{\text{FCE}}$ 010: SSIWS3 110: Setting prohibited 011: SSL00 111: Setting prohibited
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	PF9MD[2:0]	000	R/W	PF9 Mode Select the function of the PF9. 000: PF9 100: TIOC3A 001: A23 101: FRB 010: SSISCK3 110: Setting prohibited 011: RSPCK0 111: Setting prohibited
3	—	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
2 to 0	PF8MD[2:0]	000	R/W	PF8 Mode Select the function of the PF8. 000: PF8 100: Setting prohibited 001: $\overline{\text{CE2B}}$ 101: Setting prohibited 010: SSIDATA3 110: SD_CD (640-Kbyte version only) 011: DV_CLK 111: Setting prohibited Note: For 1-Mbyte version, bit 2 is reserved and always read as 0. The write value should always be 0.

(3) Port F Control Register 1 (PFCR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PF7MD[2:0]			-	PF6MD[2:0]			-	PF5MD[2:0]			-	PF4MD[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	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
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PF7MD[2:0]	000	R/W	PF7 Mode Select the function of the PF7. 000: PF7 100: TCLKD 001: $\overline{\text{CE2A}}$ 101: Setting prohibited 010: SSIWS3 110: SD_WP (640-Kbyte version only) 011: DV_DATA7 111: Setting prohibited
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
10 to 8	PF6MD[2:0]	000	R/W	PF6 Mode Select the function of the PF6. 000: PF6 001: $\overline{CS6}$/$\overline{CE1B}$ 010: SSISCK3 011: DV_DATA6 100: TCLKB 101: Setting prohibited 110: SD_D1 (640-Kbyte version only) 111: Setting prohibited
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	PF5MD[2:0]	000	R/W	PF5 Mode Select the function of the PF5. 000: PF5 001: $\overline{CS5}$/$\overline{CE1A}$ 010: SSIDATA2 011: DV_DATA5 100: TCLKC 101: Setting prohibited 110: SD_D0 (640-Kbyte version only) 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PF4MD[2:0]	000	R/W	PF4 Mode Select the function of the PF4. 000: PF4 001: $\overline{ICIOWR}$/$\overline{AH}$ 010: SSIWS2 011: DV_DATA4 100: TxD3 101: Setting prohibited 110: SD_CLK (640-Kbyte version only) 111: Setting prohibited

chanhvinh.com

(4) Port F Control Register 0 (PFCR0)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PF3MD[2:0]			-	PF2MD[2:0]			-	PF1MD[2:0]			-	PF0MD[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	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
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PF3MD[2:0]	000	R/W	PF3 Mode Select the function of the PF3. 000: PF3 001: $\overline{\text{ICIORD}}$ 010: SSISCK2 011: DV_DATA3 100: RxD3 101: Setting prohibited 110: SD_CMD (640-Kbyte version only) 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PF2MD[2:0]	000	R/W	PF2 Mode Select the function of the PF2. 000: PF2 001: $\overline{\text{BACK}}$ 010: SSIDATA1 011: DV_DATA2 100: TxD2 101: DACK0 110: SD_D3 (640-Kbyte version only) 111: Setting prohibited
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	PF1MD[2:0]	000	R/W	PF1 Mode Select the function of the PF1. 000: PF1 100: RxD2 001: $\overline{\text{BREQ}}$ 101: DREQ0 010: SSIWS1 110: SD_D2 011: DV_DATA1 (640-Kbyte version only) 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PF0MD[2:0]	000	R/W	PF0 Mode Select the function of the PF0. 000: PF0 100: SCK2 001: $\overline{\text{WAIT}}$ 101: TEND0 010: SSISCK1 110: Setting prohibited 011: DV_DATA0 111: Setting prohibited

chanhvinh.com

32.2.21 Port F I/O Register 0 (PFIOR0)

PFIOR0 is a 16-bit readable/writable register that is used to set the pins on port F as inputs or outputs. The PF12IOR to PF0IOR bits correspond to the PF12 to PF0 pins, respectively. PFIOR0 is enabled when the port F pins are functioning as general-purpose I/O (PF12 to PF0) or TIOC I/O of multi-function timer pulse unit 2. In other states, they are disabled. If a bit in PFIOR0 is set to 1, the corresponding pin on port F functions as an output. If it is cleared to 0, the corresponding pin functions as an input.

Bits 15 to 13 in PFIOR0 are reserved. This bit is always read as 0. The write value should always be 0.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PF12 IOR	PF11 IOR	PF10 IOR	PF9 IOR	PF8 IOR	PF7 IOR	PF6 IOR	PF5 IOR	PF4 IOR	PF3 IOR	PF2 IOR	PF1 IOR	PF0 IOR
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

32.2.22 Port F Data Register 0 (PFDR0)

PFDR0 is a 16-bit readable/writable register that stores port F data. The PF12DR to PF0DR bits correspond to the PF12 to PF pins respectively.

When a pin function is general output, if a value is written to PFDR0, that value is output directly from the pin, and if PEDR0 is read, the register value is returned directly regardless of the pin state.

When a pin function is general input, if PFDR0 is read, the pin state, not the register value, is returned directly. If a value is written to PFDR0, although that value is written into PFDR0, it does not affect the pin state. Table 32.18 summarizes PFDR0 read/write operation.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PF12 DR	PF11 DR	PF10 DR	PF9 DR	PF8 DR	PF7 DR	PF6 DR	PF5 DR	PF4 DR	PF3 DR	PF2 DR	PF1 DR	PF0 DR
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

chanhvinh.com

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	PF12DR	0	R/W	See table 32.18
11	PF11DR	0	R/W	
10	PF10DR	0	R/W	
9	PF9DR	0	R/W	
8	PF8DR	0	R/W	
7	PF7DR	0	R/W	
6	PF6DR	0	R/W	
5	PF5DR	0	R/W	
4	PF4DR	0	R/W	
3	PF3DR	0	R/W	
2	PF2DR	0	R/W	
1	PF1DR	0	R/W	
0	PF0DR	0	R/W	

Table 32.18 Port F Data Register 0 (PFDR0) Read/Write Operation

- Bits 12 to 0 of PFDR0

PFIOR0	Pin Operation	Read Operation	Write Operation
0	General input	Pin state	Can write to PFDR0, but it has no effect on the pin state
	Other than general input	Pin state	Can write to PFDR0, but it has no effect on the pin state
1	General output	PFDR0 value	Value written is output from pin
	Other than general output	PFDR0 value	Can write to PFDR0, but it has no effect on the pin state

chanhvinh.com

32.2.23 Port F Port Register 0 (PFPR0)

PFPR0 is a 16-bit read-only register, in which PF12PR to PF0PR bits correspond to the PF12 to PF0 pins, respectively. PFPR0 always returns the states of the pins regardless of the PFCR0 to PFCR3 settings.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PF12PR	PF11PR	PF10PR	PF9PR	PF8PR	PF7PR	PF6PR	PF5PR	PF4PR	PF3PR	PF2PR	PF1PR	PF0PR
Initial value:	0	0	0	PF12	PF11	PF10	PF9	PF8	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
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	PF12PR	Pin state	R	The pin state is returned. These bits cannot be modified.
11	PF11PR	Pin state	R	
10	PF10PR	Pin state	R	
9	PF9PR	Pin state	R	
8	PF8PR	Pin state	R	
7	PF7PR	Pin state	R	
6	PF6PR	Pin state	R	
5	PF5PR	Pin state	R	
4	PF4PR	Pin state	R	
3	PF3PR	Pin state	R	
2	PF2PR	Pin state	R	
1	PF1PR	Pin state	R	
0	PF0PR	Pin state	R	

chanhvinh.com

32.2.24 Port G Control Register 0 to 7 (PGCR0 to PGCR7)

PGCR0 to PGCR7 are 16-bit readable/writable registers that are used to select the functions of the multiplexed pins on port G.

(1) Port G Control Register 7 (PGCR7)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	PG0MD[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

Note: To write to PGCR7, write by 16-bit or 32-bit access such that the write value for bits 15 to 8 is H'5A. In 8-bit access, the register cannot be written to.

Bit	Bit Name	Initial Value	R/W	Description
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should be H'5A for bits 15 to 8 and 0 for bits 7 to 3.
2 to 0	PG0MD[2:0]	000*	R/W	PG0 Mode Select the function of the PG0. 000: PG0 100: $\overline{\text{WDTOVF}}$ 001: LCD_DATA0 101: Setting prohibited 010: SD_D2 110: Setting prohibited 011: PINT0 111: Setting prohibited

Note: * Not initialized by a reset triggered by watchdog timer overflow.

(2) Port G Control Register 6 (PGCR6: Available Only in the SH7264 Group)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PG24MD[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
15 to 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	PG24MD[1:0]	00	R/W	PG24 Mode Select the function of the PG24. 00: PG24 10: TIOC0D 01: MISO1 11: Setting prohibited

(3) Port G Control Register 5 (PGCR5)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PG23MD[1:0]		-	-	PG22MD[1:0]		-	-	PG21MD[1:0]		-	PG20MD[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/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	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	PG23MD[1:0]	00	R/W	PG23 Mode Select the function of the PG23. 00: PG23 10: TIOC0C 01: MOSI1 11: Setting prohibited Note: These bits are reserved in the SH7262 Group. They are always read as 0. 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	PG22MD[1:0]	00	R/W	PG22 Mode Select the function of the PG22. 00: PG22 10: TIOC0B 01: SSL10 11: Setting prohibited Note: These bits are reserved in the SH7262 Group. They are always read as 0. The write value should always be 0.
7, 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, 4	PG21MD[1:0]	00	R/W	PG21 Mode Select the function of the PG21. 00: PG21 10: TIOC0A 01: RSPCK1 11: Setting prohibited Note: These bits are reserved in the SH7262 Group. They are always read as 0. 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 to 0	PG20MD[2:0]	000	R/W	PG20 Mode Select the function of the PG20. 000: PG20 100: TxD7 001: LCD_EXTCLK 101: Setting prohibited 010: Setting prohibited 110: Setting prohibited 011: MISO0 111: Setting prohibited

(4) Port G Control Register 4 (PGCR4)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PG19MD[2:0]			-	PG18MD[2:0]			-	PG17MD[2:0]			-	PG16MD[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	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
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PG19MD[2:0]	000	R/W	PG19 Mode Select the function of the PG19. 000: PG19 100: RxD7 001: LCD_CLK 101: Setting prohibited 010: TIOC2B 110: Setting prohibited 011: MOSI1 111: Setting prohibited

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PG18MD[2:0]	000	R/W	PG18 Mode Select the function of the PG18. 000: PG18 100: TxD6 001: LCD_DE 101: Setting prohibited 010: TIOC2A 110: Setting prohibited 011: SSL10 111: Setting prohibited
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	PG17MD[2:0]	000	R/W	PG17 Mode Select the function of the PG17. 000: PG17 100: RxD6 001: LCD_HSYNC 101: Setting prohibited 010: TIOC1B 110: Setting prohibited 011: RSPCK1 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PG16MD[2:0]	000	R/W	PG16 Mode Select the function of the PG16. 000: PG16 100: $\overline{\text{CTS1}}$ 001: LCD_VSYNC 101: Setting prohibited 010: TIOC1A 110: Setting prohibited 011: TxD3 111: Setting prohibited

chanhvinh.com

(5) Port G Control Register 3 (PGCR3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PG15MD[2:0]			-	PG14MD[2:0]			-	PG13MD[2:0]			-	PG12MD[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	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
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PG15MD[2:0]	000	R/W	PG15 Mode Select the function of the PG15. 000: PG15 100: $\overline{\text{RTS1}}$ 001: LCD_DATA15 101: Setting prohibited 010: TIOC0D 110: Setting prohibited 011: RxD3 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PG14MD[2:0]	000	R/W	PG14 Mode Select the function of the PG14. 000: PG14 100: SCK1 001: LCD_DATA14 101: Setting prohibited 010: TIOC0C 110: Setting prohibited 011: Setting prohibited 111: Setting prohibited
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	PG13MD[2:0]	000	R/W	PG13 Mode Select the function of the PG13. 000: PG13 100: TxD1 001: LCD_DATA13 101: Setting prohibited 010: TIOC0B 110: Setting prohibited 011: Setting prohibited 111: Setting prohibited

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PG12MD[2:0]	000	R/W	PG12 Mode Select the function of the PG12. 000: PG12 100: RxD1 001: LCD_DATA12 101: Setting prohibited 010: TIOC0A 110: Setting prohibited 011: Setting prohibited 111: Setting prohibited

(6) Port G Control Register 2 (PGCR2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PG11MD[2:0]			-	PG10MD[2:0]			-	PG9MD[2:0]			-	PG8MD[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	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
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PG11MD[2:0]	000	R/W	PG11 Mode Select the function of the PG11. 000: PG11 100: TxD5 001: LCD_DATA11 101: SIOFTxD 010: SSITxD0 110: Setting prohibited 011: IRQ3 111: Setting prohibited
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
10 to 8	PG10MD[2:0]	000	R/W	PG10 Mode Select the function of the PG10. 000: PG10 100: RxD5 001: LCD_DATA10 101: SIOFRxD 010: SSIRxD0 110: Setting prohibited 011: IRQ2 111: Setting prohibited
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	PG9MD[2:0]	000	R/W	PG9 Mode Select the function of the PG9. 000: PG9 100: TxD4 001: LCD_DATA9 101: SIOFSYNC 010: SSIWS0 110: Setting prohibited 011: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PG8MD[2:0]	000	R/W	PG8 Mode Select the function of the PG8. 000: PG8 100: RxD4 001: LCD_DATA8 101: SIOFSCK 010: SSISCK0 110: Setting prohibited 011: Setting prohibited 111: Setting prohibited

chanhvinh.com

(7) Port G Control Register 1 (PGCR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PG7MD[2:0]			-	PG6MD[2:0]			-	PG5MD[2:0]			-	PG4MD[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	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
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PG7MD[2:0]	000	R/W	PG7 Mode Select the function of the PG7. 000: PG7 001: LCD_DATA7 010: SD_CD 011: PINT7 100: IRQ7 (640-Kbyte version only) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited Note: For 1-Mbyte version, bit 14 is reserved and always read as 0. The write value should always be 0.
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PG6MD[2:0]	000	R/W	PG6 Mode Select the function of the PG6. 000: PG6 001: LCD_DATA6 010: SD_WP 011: PINT6 100: IRQ6 (640-Kbyte version only) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited Note: For 1-Mbyte version, bit 10 is reserved and always read as 0. The write value should always be 0.
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	PG5MD[2:0]	000	R/W	PG5 Mode Select the function of the PG5. 000: PG5 001: LCD_DATA5 010: SD_D1 011: PINT5 100: IRQ5 (640-Kbyte version only) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited Note: For 1-Mbyte version, bit 6 is reserved and always read as 0. 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 to 0	PG4MD[2:0]	000	R/W	PG4 Mode Select the function of the PG4. 000: PG4 001: LCD_DATA4 010: SD_D0 011: PINT4 100: IRQ4 (640-Kbyte version only) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited Note: For 1-Mbyte version, bit 2 is reserved and always read as 0. The write value should always be 0.

chanhvinh.com

(8) Port G Control Register 0 (PGCR0)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PG3MD[1:0]		-	-	PG2MD[1:0]		-	-	PG1MD[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	R

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	PG3MD[1:0]	00	R/W	PG3 Mode Select the function of the PG3. 00: PG3 10: SD_CLK 01: LCD_DATA3 11: PINT3
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PG2MD[1:0]	00	R/W	PG2 Mode Select the function of the PG2. 00: PG2 10: SD_CMD 01: LCD_DATA2 11: PINT2
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	PG1MD[1:0]	00	R/W	PG1 Mode Select the function of the PG1. 00: PG1 10: SD_D3 01: LCD_DATA1 11: PINT1
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

32.2.25 Port G I/O Registers 0, 1 (PGIOR0, PGIOR1)

PGIOR1 and PGIOR0 are 16-bit readable/writable registers that are used to set the pins on port G as inputs or outputs. The PG24IOR to PG0IOR bits correspond to the PG24 to PG0, respectively. PGIOR1 and PGIOR0 are enabled when the port G pins are functioning as general-purpose I/O (PG24 to PG0) or TIOC I/O of multi-function timer pulse unit 2. In other states, they are disabled. If bits in PGIOR1 and PGIOR0 are set to 1, corresponding pins on port G functions as outputs. If they are cleared to 0, the corresponding pins function as inputs.

Bits15 to 9 in PGIOR1, and bits 8 to 5 in PGIOR1 in the SH7262 Group are reserved. These bits are always read as 0. The write values should always be 0.

(1) Port G IO Register 1 (PGIOR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	PG24 IOR	PG23 IOR	PG22 IOR	PG21 IOR	PG20 IOR	PG19 IOR	PG18 IOR	PG17 IOR	PG16 IOR
Initial value:	0	0	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

(2) Port G IO Register 0 (PGIOR0)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PG15 IOR	PG14 IOR	PG13 IOR	PG12 IOR	PG11 IOR	PG10 IOR	PG9 IOR	PG8 IOR	PG7 IOR	PG6 IOR	PG5 IOR	PG4 IOR	PG3 IOR	PG2 IOR	PG1 IOR	PG0 IOR
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

32.2.26 Port G Data Register 0, 1 (PGDR0, PGDR1)

PGDR1 and PGDR0 are 16-bit readable/writable registers that store port G data. The PG24DR to PG0DR bits correspond to the PG24 to PGDR0 pins, respectively.

When a pin function is general output, if a value is written to PGDR1 or PGDR0, that value is output from the pin, and if PGDR1 or PGDR0 is read, the register value is returned directly regardless of the pin state.

When a pin function is general input, if PGDR1 or PGDR0 is read, the pin state, not the register value, is returned directly. If a value is written to PGDR1 or PGDR0, although that value is written into PGDR1 or PGDR0, it does not affect the pin state. Table 32.19 summarizes PGDR1/PGDR0 read/write operation.

chanhvinh.com

(1) Port G Data Register 1 (PGDR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	PG24 DR	PG23 DR	PG22 DR	PG21 DR	PG20 DR	PG19 DR	PG18 DR	PG17 DR	PG16 DR
Initial value:	0	0	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	Bit Name	Initial Value	R/W	Description
15 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PG24DR	0	R/W	See table 32.19
7	PG23DR	0	R/W	Note: Bits 8 to 5 are reserved in the SH7262 Group. These bits are always read as 0. The write value should always be 0.
6	PG22DR	0	R/W	
5	PG21DR	0	R/W	
4	PG20DR	0	R/W	
3	PG19DR	0	R/W	
2	PG18DR	0	R/W	
1	PG17DR	0	R/W	
0	PG16DR	0	R/W	

chanhvinh.com

(2) Port G Data Register 0 (PGDR0)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PG15 DR	PG14 DR	PG13 DR	PG12 DR	PG11 DR	PG10 DR	PG9 DR	PG8 DR	PG7 DR	PG6 DR	PG5 DR	PG4 DR	PG3 DR	PG2 DR	PG1 DR	PG0 DR
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	PG15DR	0	R/W	See table 32.19
14	PG14DR	0	R/W	
13	PG13DR	0	R/W	
12	PG12DR	0	R/W	
11	PG11DR	0	R/W	
10	PG10DR	0	R/W	
9	PG9DR	0	R/W	
8	PG8DR	0	R/W	
7	PG7DR	0	R/W	
6	PG6DR	0	R/W	
5	PG5DR	0	R/W	
4	PG4DR	0	R/W	
3	PG3DR	0	R/W	
2	PG2DR	0	R/W	
1	PG1DR	0	R/W	
0	PG0DR	0	R/W	

Table 32.19 Port G Data Registers 1, 0 (PGDR1, PGDR0) Read/Write Operation

- Bits 8 to 0 of PGDR1 and Bits 15 to 0 of PGDR0

PGIOR1, 0	Pin Function	Read Operation	Write Operation
0	General input	Pin state	Can write to PGDR0/PGDR1, but it has no effect on the pin state
	Other than general input	Pin state	Can write to PGDR0/PGDR1, but it has no effect on the pin state
1	General output	PGDR0/PGDR1 value	Value written is output to pin
	Other than general output	PGDR0/PGDR1 value	Can write to PGDR0/PGDR1, but it has no effect on the pin state

chanhvinh.com

32.2.27 Port G Port Register 0, 1 (PGPR0, PGPR1)

PGPR1 and PGPR0 are 16-bit read-only registers, in which the PG24PR to PG0PR bits correspond to the PG24 to PG0 pins, respectively. PGPR1 and PGPR0 always return the states of the pins regardless of the PGCR7 to PGCR0 settings.

(1) Port G Port Register 1 (PGPR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	PG24PR	PG23PR	PG22PR	PG21PR	PG20PR	PG19PR	PG18PR	PG17PR	PG16PR
Initial value:	0	0	0	0	0	0	0	PG24	PG23	PG22	PG21	PG20	PG19	PG18	PG17	PG16
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 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PG24PR	Pin state	R	The pin state is returned. These bits cannot be modified. Note: Bits 8 to 5 are reserved in the SH7262 Group. These bits are always read as 0. The write value should always be 0.
7	PG23PR	Pin state	R	
6	PG22PR	Pin state	R	
5	PG21PR	Pin state	R	
4	PG20PR	Pin state	R	
3	PG19PR	Pin state	R	
2	PG18PR	Pin state	R	
1	PG17PR	Pin state	R	
0	PG16PR	Pin state	R	

chanhvinh.com

(2) Port G Port Register 0 (PGPR0)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PG15 PR	PG14 PR	PG13 PR	PG12 PR	PG11 PR	PG10 PR	PG9 PR	PG8 PR	PG7 PR	PG6 PR	PG5 PR	PG4 PR	PG3 PR	PG2 PR	PG1 PR	PG0 PR
Initial value:	PG15	PG14	PG13	PG12	PG11	PG10	PG9	PG8	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0
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	PG15PR	Pin state	R	The pin state is returned. These bits cannot be modified.
14	PG14PR	Pin state	R	
13	PG13PR	Pin state	R	
12	PG12PR	Pin state	R	
11	PG11PR	Pin state	R	
10	PG10PR	Pin state	R	
9	PG9PR	Pin state	R	
8	PG8PR	Pin state	R	
7	PG7PR	Pin state	R	
6	PG6PR	Pin state	R	
5	PG5PR	Pin state	R	
4	PG4PR	Pin state	R	
3	PG3PR	Pin state	R	
2	PG2PR	Pin state	R	
1	PG1PR	Pin state	R	
0	PG0PR	Pin state	R	

chanhvinh.com

32.2.28 Port H Control Register 0, 1 (PHCR0, PHCR1)

PHCR1 and PHCR0 are 16-bit readable/writable registers that are used to select the function of the multiplexed pins on port H.

(1) Port H Control Register 1 (PHCR1: Available Only in the SH7264 Group)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PH7MD0	-	-	-	PH6MD0	-	-	-	PH5MD0	-	-	-	PH4MD0
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
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	PH7MD0	0	R/W	PH7 Mode Select the function of the PH7. 0: PH7 1: AN7
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PH6MD0	0	R/W	PH6 Mode Select the function of the PH6. 0: PH6 1: AN6
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PH5MD0	0	R/W	PH5 Mode Select the function of the PH5. 0: PH5 1: AN5
3 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	PH4MD0	0	R/W	PH4 Mode Select the function of the PH4. 0: PH4 1: AN4

(2) Port H Control Register 0 (PHCR0)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PH3 MD0	-	-	-	PH2 MD0	-	-	-	PH1 MD0	-	-	-	PH0 MD0
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
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	PH3MD0	0	R/W	PH3 Mode Select the function of the PH3. 0: PH3 1: AN3
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PH2MD0	0	R/W	PH2 Mode Select the function of the PH2. 0: PH2 1: AN2
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	PH1MD0	0	R/W	PH1 Mode Select the function of the PH1. 0: PH1 1: AN1
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PH0MD0	0	R/W	PH0 Mode Select the function of the PH0. 0: PH0 1: AN0

chanhvinh.com

32.2.29 Port H Port Register 0 (PHPR0)

PHPR0 is a 16-bit read-only register, in which the PH7PR to PH0PR bits correspond to the PH7 to PH0 pins, respectively. PHPR0 always returns the states of the pins when the general input function is selected. This register is read as 1 during operation of the A/D converter.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PH7PR	PH6PR	PH5PR	PH4PR	PH3PR	PH2PR	PH1PR	PH0PR
Initial value:	0	0	0	0	0	0	0	0	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0
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	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	PH7PR	Pin state	R	The pin state is returned. These bits cannot be modified. Note: Bits 7 to 4 are reserved in the SH7262 Group. These bits are always read as 1.
6	PH6PR	Pin state	R	
5	PH5PR	Pin state	R	
4	PH4PR	Pin state	R	
3	PH3PR	Pin state	R	
2	PH2PR	Pin state	R	
1	PH1PR	Pin state	R	
0	PH0PR	Pin state	R	

chanhvinh.com

32.2.30 Port J Control Register 0 to 2 (PJCR0 to PJCR2)

PJCR2 to PJCR0 are 16-bit readable/writable registers that are used to select the functions of the multiplexed pins on port J.

(1) Port J Control Register 2 (PJCR2: Available Only in the SH7264 Group)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PJ11MD[1:0]		-	-	PJ10MD[1:0]		-	-	PJ9MD[1:0]		-	-	PJ8MD[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	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	PJ11MD[1:0]	00	R/W	PJ11 Mode Select the function of the PJ11. 00: PJ11 10: DACK1 01: PWM2H 11: Setting prohibited
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PJ10MD[1:0]	00	R/W	PJ10 Mode Select the function of the PJ10. 00: PJ10 10: DREQ1 01: PWM2G 11: Setting prohibited
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	PJ9MD[1:0]	00	R/W	PJ9 Mode Select the function of the PJ9. 00: PJ9 10: TEND1 01: PWM2F 11: AUDIO_XOUT (640-Kbyte version only)
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	PJ8MD[1:0]	00	R/W	PJ8 Mode Select the function of the PJ8. 00: PJ8 10: $\overline{\text{RTS3}}$ 01: PWM2E 11: Setting prohibited

(2) Port J Control Register 1 (PJCR1: Available Only in the SH7264 Group)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PJ7MD[1:0]		-	-	PJ6MD[1:0]		-	-	PJ5MD[1:0]		-	-	PJ4MD[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	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	PJ7MD[1:0]	00	R/W	PJ7 Mode Select the function of the PJ7. 00: PJ7 10: $\overline{\text{CTS3}}$ 01: TIOC1B 11: Setting prohibited
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PJ6MD[1:0]	00	R/W	PJ6 Mode Select the function of the PJ6. 00: PJ6 10: SCK3 01: TIOC1A 11: Setting prohibited
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	PJ5MD[1:0]	00	R/W	PJ5 Mode Select the function of the PJ5. 00: PJ5 10: TxD3 01: IERxD 11: Setting prohibited

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PJ4MD[1:0]	00	R/W	PJ4 Mode Select the function of the PJ4. 00: PJ4 10: RxD3 01: IETxD 11: Setting prohibited

(3) Port J Control Register 0 (PJCR0)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PJ3MD[1:0]		-	PJ2MD[2:0]			-	PJ1MD[2:0]			-	PJ0MD[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/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
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	PJ3MD[1:0]	00	R/W	PJ3 Mode Select the function of the PJ3. 00: PJ3 10: CRx0/CRx1 01: CRx1 11: IRQ1
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PJ2MD[2:0]	000	R/W	PJ2 Mode Select the function of the PJ2. 000: PG2 100: SCK0 001: CTx1 101: LCD_M_DISP 010: CRx0&CRx1 110: Setting prohibited 011: $\overline{CS2}$ 111: Setting prohibited
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	PJ1MD[2:0]	000	R/W	PJ1 Mode Select the function of the PJ1. 000: PJ1 100: RxD0 001: CRx0 101: Setting prohibited 010: IERxD 110: Setting prohibited 011: $\overline{IRQ0}$ 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PJ0MD[2:0]	000	R/W	PJ0 Mode Select the function of the PJ0. 000: PJ0 100: TxD0 001: CTx0 101: A0 010: IERxD 110: Setting prohibited 011: $\overline{CS1}$ 111: Setting prohibited

32.2.31 Port J I/O register 0 (PJIOR0)

PJIOR0 is a 16-bit readable/writable register that is used to set the pins on port J as inputs or outputs. The PJ11IIOR to PJ0IOR bits correspond to the PJ11 to PJ0 pins respectively. The setting of PJIOR0 is valid for the pins for which general I/O (PJ11 to PJ0) function or TIOC I/O function of multi-function timer pulse unit 2 is selected and has no effect on the pins for which other function is selected. If a bit in PJIOR0 is set to 1, the corresponding pin on port J functions as an output pin. If it is cleared to 0, the corresponding pin functions as an input pin.

Bits 15 to 12 in PJIOR0, and bits 11 to 4 in PJIOR0 in the SH7262 Group are reserved. These bits are always read as 0. The write value should always be 0.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	PJ11 IOR	PJ10 IOR	PJ9 IOR	PJ8 IOR	PJ7 IOR	PJ6 IOR	PJ5 IOR	PJ4 IOR	PJ3 IOR	PJ2 IOR	PJ1 IOR	PJ0 IOR
Initial 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

chanhvinh.com

32.2.32 Port J Data Register 0 (PJDR0)

PJDR0 is a 16-bit readable/writable register that stores port J data. The PJ11DR to PJ0DR bits correspond to the PJ11 to PJ0 pins, respectively.

When a pin function is general output, if a value is written to PJDR0, that value is output from the pin, and if PJDR0 is read, the register value is returned directly regardless of the pin state.

When a pin function is general input, if PJDR0 is read, the pin state, not the register value, is returned directly. If a value is written to PJDR0, although that value is written into PJDR0, it does not affect the pin state. Table 32.20 summarizes PJDR0 read/write operation.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	PJ11DR	PJ10DR	PJ9DR	PJ8DR	PJ7DR	PJ6DR	PJ5DR	PJ4DR	PJ3DR	PJ2DR	PJ1DR	PJ0DR
Initial 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
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	PJ11DR	0	R/W	See table 32.20
10	PJ10DR	0	R/W	Note: Bits 11 to 4 are reserved in the SH7262 Group. These bits are always read as 0. The write value should always be 0.
9	PJ9DR	0	R/W	
8	PJ8DR	0	R/W	
7	PJ7DR	0	R/W	
6	PJ6DR	0	R/W	
5	PJ5DR	0	R/W	
4	PJ4DR	0	R/W	
3	PJ3DR	0	R/W	
2	PJ2DR	0	R/W	
1	PJ1DR	0	R/W	
0	PJ0DR	0	R/W	

chanhvinh.com

Table 32.20 Port J Data Registers 0 (PJDR0) Read/Write Operation

- Bits 11 to 0 of PJDR0

PJIOR0	Pin Function	Read Operation	Write Operation
0	General input	Pin state	Can write to PJDR0, but it has no effect on the pin state.
	Other than general input	Pin state	Can write to PJDR0, but it has no effect on the pin state.
1	General output	PJDR0 value	Value written is output from pin
	Other than general output	PJDR0 value	Can write to PJDR0, but it has no effect on the pin state

32.2.33 Port J Port Register 0 (PJPR0)

PJPR0 is a 16-bit read-only register, in which the PJ11PR to PJ0PR bits correspond to the PJ11 to PJ0 pins, respectively. PJPR0 always returns the states of the pins regardless of the PJCR0 to PGCR2 settings.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	PJ11PR	PJ10PR	PJ9PR	PJ8PR	PJ7PR	PJ6PR	PJ5PR	PJ4PR	PJ3PR	PJ2PR	PJ1PR	PJ0PR
Initial value:	0	0	0	0	PJ11	PJ10	PJ9	PJ8	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0
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 12	—	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
11	PJ11PR	Pin state	R	The pin state is returned. These bits cannot be modified. Note: Bits 11 to 4 are reserved in the SH7262 Group. These bits are always read as 0. The write value should always be 0.
10	PJ10PR	Pin state	R	
9	PJ9PR	Pin state	R	
8	PJ8PR	Pin state	R	
7	PJ7PR	Pin state	R	
6	PJ6PR	Pin state	R	
5	PJ5PR	Pin state	R	
4	PJ4PR	Pin state	R	
3	PJ3PR	Pin state	R	
2	PJ2PR	Pin state	R	
1	PJ1PR	Pin state	R	
0	PJ0PR	Pin state	R	

32.2.34 Port K Control Register 0 to 2 (PKCR0 to PKCR2: Available Only in the SH7264 Group)

PKCR0 to PKCR2 are 16-bit readable/writable registers that are used to select the function of the multiplexed pins on port K.

(1) Port K Control Register 2 (PKCR2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PK11MD[1:0]		-	-	PK10MD[1:0]		-	-	PK9MD[1:0]		-	-	PK8MD[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	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	PK11MD[1:0]	00	R/W	PK11 Mode Select the function of the PK11. 00: PK11 10: SSITxD0 01: PWM2D 11: Setting prohibited

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PK10MD[1:0]	00	R/W	PK10 Mode Select the function of the PK10. 00: PK10 10: SSIRxD0 01: PWM2C 11: Setting prohibited
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	PK9MD[1:0]	00	R/W	PK9 Mode Select the function of the PK9. 00: PK9 10: SSIWS0 01: PWM2B 11: Setting prohibited
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PK8MD[1:0]	00	R/W	PK8 Mode Select the function of the PK8. 00: PK8 10: SSISCK0 01: PWM2A 11: Setting prohibited

chanhvinh.com

(2) Port K Control Register 1 (PKCR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PK7MD[1:0]		-	-	PK6MD[1:0]		-	-	PK5MD[1:0]		-	-	PK4MD[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	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	PK7MD[1:0]	00	R/W	PK7 Mode Select the function of the PK7. 00: PK7 10: SD_CD 01: PWM1H 11: Setting prohibited
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PK6MD[1:0]	00	R/W	PK6 Mode Select the function of the PK6. 00: PK6 10: SD_WP 01: PWM1G 11: Setting prohibited
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	PK5MD[1:0]	00	R/W	PK5 Mode Select the function of the PK5. 00: PK5 10: SD_D1 01: PWM1F 11: Setting prohibited
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PK4MD[1:0]	00	R/W	PK4 Mode Select the function of the PK4. 00: PK4 10: SD_D0 01: PWM1E 11: Setting prohibited

chanhvinh.com

(3) Port K Control Register 0 (PKCR0)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PK3MD[1:0]		-	-	PK2MD[1:0]		-	-	PK1MD[1:0]		-	-	PK0MD[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	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	PK3MD[1:0]	00	R/W	PK3 Mode Select the function of the PK3. 00: PK3 10: SD_CLK 01: PWM1D 11: Setting prohibited
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PK2MD[1:0]	00	R/W	PK2 Mode Select the function of the PK2. 00: PK2 10: SD_CMD 01: PWM1C 11: Setting prohibited
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	PK1MD[1:0]	00	R/W	PK1 Mode Select the function of the PK1. 00: PK1 10: SD_D3 01: PWM1B 11: Setting prohibited
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PK0MD[1:0]	00	R/W	PK0 Mode Select the function of the PK0. 00: PK0 10: SD_D2 01: PWM1A 11: Setting prohibited

chanhvinh.com

32.2.35 Port K I/O Register 0 (PKIOR0: Available Only in the SH7264 Group)

PKIOR0 is a 16-bit readable/writable register that is used to set the pins on port K as inputs or outputs. The PK11IOR to PK0IOR bits correspond to the PK11 to PK0 pins respectively. The setting of PKIOR0 is valid for the pins for which general I/O function is selected and has no effect on the pins for which other function is selected. If a bit in PKIOR0 is set to 1, the corresponding pin on port K functions as an output pin. If it is cleared to 0, the corresponding pin functions as an input pin.

Bits 15 to 12 in PKIOR0 are reserved. These bits are always read as 0. The write value should always be 0.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	PK11 IOR	PK10 IOR	PK9 IOR	PK8 IOR	PK7 IOR	PK6 IOR	PK5 IOR	PK4 IOR	PK3 IOR	PK2 IOR	PK1 IOR	PK0 IOR
Initial 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

32.2.36 Port K Data Register 0 (PKDR0: Available Only in the SH7264 Group)

PKDR0 is a 16-bit readable/writable register that stores port K data. The PK11DR to PK0DR bits correspond to the PK11 to PK0 pins, respectively.

When a pin function is general output, if a value is written to PKDR0, that value is output from the pin, and if PKDR0 is read, the register value is returned directly regardless of the pin state.

When a pin function is general input, if PKDR0 is read, the pin state, not the register value, is returned directly. If a value is written to PKDR0, although that value is written into PKDR0, it does not affect the pin state. Table 32.21 summarizes /PKDR0 read/write operation.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	PK11 DR	PK10 DR	PK9 DR	PK8 DR	PK7 DR	PK6 DR	PK5 DR	PK4 DR	PK3 DR	PK2 DR	PK1 DR	PK0 DR
Initial 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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	PK11DR	0	R/W	See table 32.21
10	PK10DR	0	R/W	
9	PK9DR	0	R/W	
8	PK8DR	0	R/W	
7	PK7DR	0	R/W	
6	PK6DR	0	R/W	
5	PK5DR	0	R/W	
4	PK4DR	0	R/W	
3	PK3DR	0	R/W	
2	PK2DR	0	R/W	
1	PK1DR	0	R/W	
0	PK0DR	0	R/W	

Table 32.21 Port K Data Register 0 (PKDR0) Read/Write Operation

- Bits 11 to 0 of PKDR0

PKIOR0	Pin Function	Read Operation	Write Operation
0	General input	Pin state	Can write to PKDR0, but it has no effect on the pin state.
	Other than general input	Pin state	Can write to PKDR0, but it has no effect on the pin state.
1	General output	PKDR0 value	Value written is output from pin
	Other than general output	PKDR0 value	Can write to PKDR0, but it has no effect on the pin state

chanhvinh.com

32.2.37 Port K Port Register 0 (PKPR0: Available Only in the SH7264 Group)

PKPR0 is a 16-bit read-only register, in which the PK11PR to PK0PR bits correspond to the PK11 to PK0 pins, respectively. PKPR0 always returns the states of the pins regardless of the PKCR0 to PKCR2 settings.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	PK11PR	PK10PR	PK9PR	PK8PR	PK7PR	PK6PR	PK5PR	PK4PR	PK3PR	PK2PR	PK1PR	PK0PR
Initial value:	0	0	0	0	PK11	PK10	PK9	PK8	PK7	PK6	PK5	PK4	PK3	PK2	PK1	PK0
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 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	PK11PR	Pin state	R	The pin state is returned. These bits cannot be modified.
10	PK10PR	Pin state	R	
9	PK9PR	Pin state	R	
8	PK8PR	Pin state	R	
7	PK7PR	Pin state	R	
6	PK6PR	Pin state	R	
5	PK5PR	Pin state	R	
4	PK4PR	Pin state	R	
3	PK3PR	Pin state	R	
2	PK2PR	Pin state	R	
1	PK1PR	Pin state	R	
0	PK0PR	Pin state	R	

32.3 Usage Notes

32.3.1 Notes on Output Function from PE0 to PE5

About each pair of ports, that is PE0 and PE1, PE2 and PE3, PE4 and PE5, when one port is set to other than general purpose I/O function (PE0 to PE5) and I2C Bus Interface 3 function (SCL0 to SCL2, SDA0 to SDA2), the other port can't be used for open-drain output. In that case, the other port can be used for general purpose or other input function.

When one port is used for open-drain output, please set the other port to general purpose I/O function.

chanhvinh.com

Section 33 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.

33.1 Features

33.1.1 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 33.1 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 33.1 States of Power-Down Modes

		State*1									
Power-Down Mode	Transition Conditions	Clock Pulse Genera-tor	CPU	CPU Register	High-Speed On-Chip RAM Cash Memory	Large-Capacity On-Chip RAM (for Data Retention)	On-Chip Peripheral Modules	Realtime Clock	Power supply	External Memory	Canceling Procedure
Sleep mode	Execute SLEEP instruction with STBY bit in STBCR cleared to 0	Running	Halted	Held	Running	Running	Running	Running*2	Running	Auto-refresh	• Interrupt • Manual reset • Power-on reset • DMA address error
Software standby mode	Execute SLEEP instruction with STBY bit in STBCR set to 1 and DEEP bit to 0	Halted	Halted	Held	Halted (contents are held *5*6)	Halted (contents are held *5*7)	Halted	Running*2	Running	Self-refresh	• NMI interrupt • IRQ interrupt • Power-on reset
Deep standby mode	Execute SLEEP instruction with STBY and DEEP bits in STBCR set to 1	Halted	Halted	Halted	Halted (contents are not held)	Halted (contents in on-chip data-retention RAM are held*3)	Halted	Running*2	Halted	Self-refresh	• NMI interrupt*4 • Power-on reset*4 • Realtime clock alarm interrupt*4 • Change on the pins for canceling*4

chanhvinh.com

		State*[1]									
Power-Down Mode	**Transition Conditions**	**Clock Pulse Genera-tor**	**CPU**	**CPU Register**	**High-Speed On-Chip RAM Cash Memory**	**Large-Capacity On-Chip RAM (for Data Retention)**	**On-Chip Peripheral Modules**	**Realtime Clock**	**Power supply**	**External Memory**	**Canceling Procedure**
Module standby mode	Set the MSTP bits in STBCR2 to STBCR8 to 1	Running	Running	Held	Running	Running	Specified module halted	Halted	Running	Auto-refresh	• Clear MSTP bit to 0 • Power-on reset (only for the user debugging interface and direct memory access controller)

Notes:
1. The pin state is retained or set to high impedance. For details, see appendix A, Pin States.
2. The realtime clock operates when the START bit in the RCR2 register is set to 1. For details, see section 14, 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.
3. 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. When the deep standby is canceled by a power-on reset, the retained contents are initialized. RRAMKP3 and RRAMKP2 can be used only for 640-Kbyte version.
4. Deep standby mode can be canceled by an interrupt (NMI or realtime clock alarm interrupt), a power-on reset, or change on the pins for canceling (PC8 to PC5,PG10 to PG11, PJ3, and PJ1). Even when deep standby mode is canceled by a source other than a reset, power-on reset exception handling is executed instead of interrupt exception handling. PG10 to PG11 operates as pins for canceling only in 640-Kbyte version.
5. When software standby mode is canceled by a power-on reset, the retained contents are initialized.
6. By setting the RAME bit in SYSCR1 or RAMWE bit in SYSCR2 to disable accesses, contents in the high-speed on-chip RAM can be retained even when software standby mode is canceled by a power-on reset.
7. By setting the VRAME bit in SYSCR3 or VRAMWE bit in SYSCR4 to disable accesses, contents in the large-capacity on-chip RAM (including on-chip data-retention RAM) can be retained even when software standby mode is canceled by a power-on reset.

chanhvinh.com

33.2 Register Descriptions

Table 33.2 shows the register configuration.

Table 33.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Standby control register 1	STBCR	R/W	H'00	H'FFFE0014	8
Standby control register 2	STBCR2	R/W	H'00	H'FFFE0018	8
Standby control register 3	STBCR3	R/W	H'7E	H'FFFE0408	8
Standby control register 4	STBCR4	R/W	H'FF	H'FFFE040C	8
Standby control register 5	STBCR5	R/W	H'FF	H'FFFE0410	8
Standby control register 6	STBCR6	R/W	H'FF	H'FFFE0414	8
Standby control register 7	STBCR7	R/W	H'FF	H'FFFE0418	8
Standby control register 8	STBCR8	R/W	H'FF	H'FFFE041C	8
Software reset control register	SWRSTCR	R/W	H'00	H'FFFE0430	8
System control register 1	SYSCR1	R/W	H'FF	H'FFFE0400	8
System control register 2	SYSCR2	R/W	H'FF	H'FFFE0404	8
System control register 3	SYSCR3	R/W	H'FF	H'FFFE0420	8
System control register 4	SYSCR4	R/W	H'FF	H'FFFE0424	8
System control register 5*	SYSCR5	R/W	H'00	H'FFFE0428	8
On-chip data-retention RAM area setting register	RRAMKP	R/W	H'00	H'FFFE6800	8
Deep standby control register	DSCTR	R/W	H'00	H'FFFE6802	8
Deep standby cancel source select register	DSSSR	R/W	H'0000	H'FFFE6804	16
Deep standby cancel edge select register	DSESR	R/W	H'0000	H'FFFE6806	16
Deep standby cancel source flag register	DSFR	R/W	H'0000	H'FFFE6808	16
XTAL crystal oscillator gain control register	XTALCTR	R/W	H'00	H'FFFE6810	8

Note: * Can be used for 640-Kbyte version.

chanhvinh.com

33.2.1 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 33.4, 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 SLEEP instruction puts chip into sleep mode. 10: Executing SLEEP instruction puts chip into software standby mode. 11: Executing SLEEP instruction puts chip into deep standby mode.
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

33.2.2 Standby Control Register 2 (STBCR2)

STBCR2 is an 8-bit readable/writable register that controls the operation of modules.

Note: When writing to this register, see section 33.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	MSTP10	-	MSTP8	MSTP7	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP10	0	R/W	Module Stop 10 When the MSTP10 bit is set to 1, the clock supply to the user debugging interface is halted. 0: The user debugging interface runs. 1: Clock supply to the user debugging interface halted.
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5	MSTP8	0	R/W	Module Stop 8 When the MSTP8 bit is set to 1, the clock supply to the direct memory access controller is halted. 0: The direct memory access controller runs. 1: Clock supply to the direct memory access controller halted.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
4	MSTP7	0	R/W	Module Stop 7 When the MSTP7 bit is set to 1, the clock supply to the FPU is halted. After setting the MSTP7 bit to 1, the MSTP7 bit cannot be cleared by writing 0. This means that, after the clock supply to the FPU is halted by setting the MSTP7 bit to 1, the supply cannot be restarted by clearing the MSTP7 bit to 0. To restart the clock supply to the FPU after it was halted, reset the LSI by a power-on reset. 0: The FPU runs. 1: Clock supply to the FPU is halted.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

33.2.3 Standby Control Register 3 (STBCR3)

STBCR3 is an 8-bit readable/writable register that controls the operation of modules.

Note: When writing to this register, see section 33.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	HIZ	MSTP 36	MSTP 35	MSTP 34	MSTP 33	MSTP 32	-	MSTP 30
Initial value:	0	1	1	1	1	1	1	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

chanhvinh.com

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 appendix A, Pin States. This bit must not be set while the TME bit in WTSCR of the watchdog timer is 1. To set the output pin to high-impedance, set the HIZ bit to 1 only while the TME bit is 0. 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	MSTP36	1	R/W	Module Stop 36 When the MSTP36 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.
5	MSTP35	1	R/W	Module Stop 35 When the MSTP35 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.
4	MSTP34	1	R/W	Module Stop 34 When the MSTP34 bit is set to 1, the clock supply to the SD host interface 0 is halted. 0: The SD host interface 0 runs. 1: Clock supply to the SD host interface 0 is halted.
3	MSTP33	1	R/W	Module Stop 33 When the MSTP33 bit is set to 1, the clock supply to the SD host interface 1 is halted. 0: The SD host interface 1 runs. 1: Clock supply to the SD host interface 1 is halted.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
2	MSTP32	1	R/W	Module Stop 32 When the MSTP32 bit is set to 1, the clock supply to the AD converter is halted. 0: The AD converter runs. 1: Clock supply to the AD converter is halted.
1	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
0	MSTP30	0	R/W	Module Stop 30 When the MSTP30 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 bit RCKSEL in the control register 5 (RCR5) to 0. After the settings above, set bit MSTP30 to 1.

33.2.4 Standby Control Register 4 (STBCR4)

STBCR4 is an 8-bit readable/writable register that controls the operation of modules.

Note: When writing to this register, see section 33.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	MSTP47	MSTP46	MSTP45	MSTP44	MSTP43	MSTP42	MSTP41	MSTP40
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

chanhvinh.com

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 unit with FIFO is halted. 0: Channel 0 of the serial communication unit with FIFO runs. 1: Clock supply to channel 0 of the serial communication unit 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 unit with FIFO is halted. 0: Channel 1 of the serial communication unit with FIFO runs. 1: Clock supply to channel 1 of the serial communication unit 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 unit with FIFO is halted. 0: Channel 2 of the serial communication unit with FIFO runs. 1: Clock supply to channel 2 of the serial communication unit 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 unit with FIFO is halted. 0: Channel 3 of the serial communication unit with FIFO runs. 1: Clock supply to channel 3 of the serial communication unit with FIFO is halted.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
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 unit with FIFO is halted. 0: Channel 4 of the serial communication unit with FIFO runs. 1: Clock supply to channel 4 of the serial communication unit with FIFO is halted.
2	MSTP42	1	R/W	Module Stop 42 When the MSTP42 bit is set to 1, the clock supply to channel 5 of the serial communication unit with FIFO is halted. 0: Channel 5 of the serial communication unit with FIFO runs. 1: Clock supply to channel 5 of the serial communication unit with FIFO is halted.
1	MSTP41	1	R/W	Module Stop 41 When the MSTP41 bit is set to 1, the clock supply to channel 6 of the serial communication unit with FIFO is halted. 0: Channel 6 of the serial communication unit with FIFO runs. 1: Clock supply to channel 6 of the serial communication unit with FIFO is halted.
0	MSTP40	1	R/W	Module Stop 40 When the MSTP40 bit is set to 1, the clock supply to channel 7 of the serial communication unit with FIFO is halted. 0: Channel 7 of the serial communication unit with FIFO runs. 1: Clock supply to channel 7 of the serial communication unit with FIFO is halted.

chanhvinh.com

33.2.5 Standby Control Register 5 (STBCR5)

STBCR5 is an 8-bit readable/writable register that controls the operation of modules.

Note: When writing to this register, see section 33.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	MSTP57	MSTP56	MSTP55	-	MSTP53	MSTP52	MSTP51	MSTP50
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R	R/W	R/W	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 I^2C bus interface 3 is halted. 0: Channel 0 of the I^2C bus interface 3 runs. 1: Clock supply to channel 0 of the I^2C bus interface 3 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 I^2C bus interface 3 is halted. 0: Channel 1 of the I^2C bus interface 3 runs. 1: Clock supply to channel 1 of the I^2C bus interface 3 is halted.
5	MSTP55	1	R/W	Module Stop 55 When the MSTP55 bit is set to 1, the clock supply to channel 2 of the I^2C bus interface 3 is halted. 0: Channel 2 of the I^2C bus interface 3 runs. 1: Clock supply to channel 2 of the I^2C bus interface 3 is halted.
4	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
3	MSTP53	1	R/W	Module Stop 53 When the MSTP53 bit is set to 1, the clock supply to channel 0 of the controller area network is halted. 0: Channel 0 of the controller area network runs. 1: Clock supply to channel 0 of the controller area network is halted.
2	MSTP52	1	R/W	Module Stop 52 When the MSTP52 bit is set to 1, the clock supply to channel 1 of the controller area network is halted. 0: Channel 1 of the controller area network runs. 1: Clock supply to channel 1 of the controller area network is halted.
1	MSTP51	1	R/W	Module Stop 51 When the MSTP51 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.
0	MSTP50	1	R/W	Module Stop 50 When the MSTP50 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.

chanhvinh.com

33.2.6 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 33.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	MSTP 67	MSTP 66	MSTP 65	MSTP 64	MSTP 63	MSTP 62	MSTP 61	MSTP 60
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

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 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.
6	MSTP66	1	R/W	Module Stop 66 When the MSTP66 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.
5	MSTP65	1	R/W	Module Stop 65 When the MSTP65 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.
4	MSTP64	1	R/W	Module Stop 64 When the MSTP64 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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
3	MSTP63	1	R/W	Module Stop 63 When the MSTP63 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.
2	MSTP62	1	R/W	Module Stop 62 When the MSTP62 bit is set to 1, the clock supply to channel 0 of the sampling rate converter is halted. 0: Channel 0 of the sampling rate converter runs. 1: Clock supply to channel 0 of the sampling rate converter is halted.
1	MSTP61	1	R/W	Module Stop 61 When the MSTP61 bit is set to 1, the clock supply to channel 1 of the sampling rate converter is halted. 0: Channel 1 of the sampling rate converter runs. 1: Clock supply to channel 1 of the sampling rate converter C is halted.
0	MSTP60	1	R/W	Module Stop 60 When the MSTP60 bit is set to 1, the clock supply to the USB 2.0 host/function module is halted. 0: The USB 2.0 host/function module runs. 1: Clock supply to the USB 2.0 host/function module is halted.

chanhvinh.com

33.2.7 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 33.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	MSTP77	MSTP76	-	MSTP74	-	MSTP72	-	MSTP70
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R	R/W	R	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP77	1	R/W	Module Stop 77 When the MSTP77 bit is set to 1, the clock supply to the serial I/O with FIFO is halted. 0: The serial I/O with FIFO runs. 1: Clock supply to the serial I/O with FIFO is halted.
6	MSTP76	1	R/W	Module Stop 76 When the MSTP76 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.
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 When the MSTP74 bit is set to 1, the clock supply to the video display controller 3 is halted. 0: The video display controller 3 runs. 1: Clock supply to the video display controller 3 is halted.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
3	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
2	MSTP72	1	R/W	Module Stop 72 When the MSTP72 bit is set to 1, the clock supply to the compare match timer is halted. 0: The compare match timer runs. 1: Clock supply to the compare match timer is halted.
1	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
0	MSTP70	1	R/W	Module Stop 70 When the MSTP70 bit is set to 1, the clock supply to the NAND flash memory controller is halted. 0: The NAND flash memory controller runs. 1: Clock supply to the NAND flash memory controller is halted.

chanhvinh.com

33.2.8 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 33.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	MSTP87	-	-	-	-	-	-	MSTP80
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP87	1	R/W	Module Stop 87 When the MSTP87 bit is set to 1, the clock supply to the motor control PWM timer is halted. 0: The motor control PWM timer runs. 1: Clock supply to the motor control PWM timer is halted.
6 to 1	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
0	MSTP80	1	R/W	Module Stop 80 When the MSTP80 bit is set to 1, the clock supply to the decompression unit is halted. 0: The decompression unit runs. 1: Clock supply to the decompression unit is halted.

chanhvinh.com

33.2.9 Software Reset Control Register (SWRSTCR)

SWRSTCR is an 8-bit readable/writable register that controls a software reset for the serial sound interface and IEBus™ controller and the operation of the crystal resonator for audio.

Note: When writing to this register, see section 33.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	AXTALE	-	-	IEBSRST	SSIF3SRST	SSIF2SRST	SSIF1SRST	SSIF0SRST
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R/W	R/W	R/W	R/W	R/W

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, 5	—	All 0	R	Reserved This bit is always read as 0. The write value should always be 0.
4	IEBSRST	0	R/W	IEBus™ Controller Software Reset Controls the IEBus™ controller reset with software. 0: The IEBus™ controller reset is canceled. 1: The IEBus™ controller is reset.
3	SSIF3SRST	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.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
2	SSIF2SRST	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.
1	SSIF1SRST	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.
0	SSIF0SRST	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.

chanhvinh.com

33.2.10 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 high-speed on-chip RAM.

When an RAMEn (n = 0 to 3) bit is set to 1, access to page n is enabled. When an RAMEn 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 an RAMEn bit is 1.

Note that when clearing the RAMEn bit to 0, be sure to execute an instruction to read from or write to the same arbitrary address in each page before setting the RAMEn bit. If such an instruction is not executed, the data last written to page n may not be written to the high-speed on-chip RAM.

SYSCR1 should be set with a program located in an area other than the high-speed 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 33.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	RAME3	RAME2	RAME1	RAME0
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	RAME3	1	R/W	RAM Enable 3 (corresponding area: page 3* in high-speed on-chip RAM) 0: Access to page 3 is disabled. 1: Access to page 3 is enabled.
2	RAME2	1	R/W	RAM Enable 2 (corresponding area: page 2* in high-speed on-chip RAM) 0: Access to page 2 is disabled. 1: Access to page 2 is enabled.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
1	RAME1	1	R/W	RAM Enable 1 (corresponding area: page 1* in high-speed on-chip RAM) 0: Access to page 1 is disabled. 1: Access to page 1 is enabled.
0	RAME0	1	R/W	RAM Enable 0 (corresponding area: page 0* in high-speed on-chip RAM) 0: Access to page 0 is disabled. 1: Access to page 0 is enabled.

Note: * For addresses in each page, see section 31, On-Chip RAM.

33.2.11 System Control Register 2 (SYSCR2)

SYSCR2 is an 8-bit readable/writable register that enables or disables writing to a specified page in the high-speed on-chip RAM.

When an RAMEn (n = 0 to 3) bit is set to 1, writing to page n is enabled. When an RAMEn bit is cleared to 0,writing to page n is ignored. The initial value of an RAMEn bit is 1.

Note that when clearing the RAMWEn bit to 0, be sure to execute an instruction to read from or write to the same arbitrary address in each page before setting the RAMWEn bit. If such an instruction is not executed, the data last written to page n may not be written to the high-speed on-chip RAM.

SYSCR2 should be set with a program located in an area other than the high-speed 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 33.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	RAM WE3	RAM WE2	RAM WE1	RAM WE0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

chanhvinh.com

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	RAMWE3	1	R/W	RAM Write Enable 3 (corresponding area: page 3* in high-speed on-chip RAM) 0: Writing to page 3 is disabled. 1 Writing to page 3 is enabled.
2	RAMWE2	1	R/W	RAM Write Enable 2 (corresponding area: page 2* in high-speed on-chip RAM) 0: Writing to page 2 is disabled. 1: Writing to page 2 is enabled.
1	RAMWE1	1	R/W	RAM Write Enable 1 (corresponding area: page 1* in high-speed on-chip RAM) 0: Writing to page 1 is disabled. 1: Writing to page 1 is enabled.
0	RAMWE0	1	R/W	RAM Write Enable 0 (corresponding area: page 0* in high-speed on-chip RAM) 0: Writing to page 0 is disabled. 1: Writing to page 0 is enabled.

Note: * For addresses in each page, see section 31, On-Chip RAM.

33.2.12 System Control Register 3 (SYSCR3)

SYSCR3 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 5) bit 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.

SYSCR3 should be set with a program located in an area other than the large-capacity on-chip RAM. Furthermore, an instruction to read SYSCR3 should be located immediately after the instruction to write to SYSCR3. If not, normal access is not guaranteed.

Note: When writing to this register, see section 33.4, Usage Notes.

chanhvinh.com

Bit:	7	6	5	4	3	2	1	0
	-	-	VRAME5	VRAME4	VRAME3	VRAME2	VRAME1	VRAME0
Initial value:	1	1	1	1	1	1	1	1
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	VRAME5	1	R/W	RAM Enable 5 (corresponding area: page 5* in large-capacity on-chip RAM) 0: Access to page 5 is disabled. 1: Access to page 5 is enabled. Note: This bit is reserved in 640-Kbyte version and read as 1. The write value should always be 0.
4	VRAME4	1	R/W	RAM Enable 4 (corresponding area: page 4* 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* 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* 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* 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* in large-capacity on-chip RAM) 0: Access to page 0 is disabled. 1: Access to page 0 is enabled.

Note: * For addresses in each page, see section 31, On-Chip RAM.

chanhvinh.com

33.2.13 System Control Register 4 (SYSCR4)

SYSCR4 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 5) bit 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.

SYSCR4 should be set with a program located in an area other than the large-capacity on-chip RAM. Furthermore, an instruction to read SYSCR4 should be located immediately after the instruction to write to SYSCR4. If not, normal access is not guaranteed.

Note: When writing to this register, see section 33.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	-	-	VRAM WE5	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/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	VRAMWE5	1	R/W	RAM Write Enable 5 (corresponding area: page 5* in large-capacity on-chip RAM) 0: Writing to page 5 is disabled. 1: Writing to page 5 is enabled. Note: This bit is reserved in 640-Kbyte version and read as 1. The write value should always be 0.
4	VRAMWE4	1	R/W	RAM Write Enable 4 (corresponding area: page 4* 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* in large-capacity on-chip RAM) 0: Writing to page 3 is disabled. 1: Writing to page 3 is enabled.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
2	VRAMWE2	1	R/W	RAM Write Enable 2 (corresponding area: page 2* 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* 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* in large-capacity on-chip RAM) 0: Writing to page 0 is disabled. 1: Writing to page 0 is enabled.

Note: * For addresses in each page, see section 31, On-Chip RAM.

33.2.14 System Control Register 5 (SYSCR5)

SYSCR5 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 SYSCR5 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.

SYSCR5 should be set with a program located in an area other than the on-chip data-retention RAM.

SYSCR5 can be used only in 640-Kbyte version.

Note: When writing to this register, see section 33.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	RRAM WE3	RRAM WE2	RRAM WE1	RRAM WE0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

chanhvinh.com

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*3*4 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.

Notes:
1. For addresses in each page, see section 31, On-Chip RAM.
2. When the VRAME0 bit in SYSCR3 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 SYSCR4 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.
3. When the VRAME1 bit in SYSCR3 is cleared to 0 (access to page 1 in large-capacity on-chip RAM is invalid), the first half (page 1 in large-capacity on-chip RAM) of the on-chip data-retention RAM cannot be accessed (read and written), regardless of the setting of this bit.
   When the VRAME1 bit in SYSCR4 is cleared to 0 (writing to page 1 in large-capacity on-chip RAM is invalid), the first half of the on-chip data-retention RAM cannot be written, regardless of the setting of this bit.
4. When the VRAME2 bit in SYSCR3 is cleared to 0 (access to page 2 in large-capacity on-chip RAM is invalid), the second half (page 2 in large-capacity on-chip RAM) of the on-chip data-retention RAM cannot be accessed (read and written), regardless of the setting of this bit.
   When the VRAMWE2 bit in SYSCR4 is cleared to 0 (writing to page 2 in large-capacity on-chip RAM is invalid), the second half of the on-chip data-retention RAM cannot be written, regardless of the setting of this bit.

chanhvinh.com

33.2.15 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 33.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	RRAM KP3	RRAM KP2	RRAM KP1	RRAM KP0
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* 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: For 1-Mbyte version, this bit is reserved and always read as 0. The write value should always be 0.
2	RRAMKP2	0	R/W	On-Chip Data-Retention RAM Storage Area 2 (corresponding area: page 2* 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: For 1-Mbyte version, this bit is reserved and always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
1	RRAMKP1	0	R/W	On-Chip Data-Retention RAM Storage Area 1 (corresponding area: page 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* 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: * For addresses in each page, see section 31, On-Chip RAM.

chanhvinh.com

33.2.16 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 33.4, 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.
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. [1-Mbyte version] Program counter (PC): H'1C0F8000 Stack pointer (SP): H'1C0F8004 [640-Kbyte version] Program counter (PC): H'1C000000 Stack pointer (SP): H'1C000004
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

chanhvinh.com

33.2.17 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 (PC8 to PC5, PG11, PG10, PJ3, and PJ1) 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 33.4, Usage Notes.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	PG11	PG10	-	-	RTCAR	PC8	PC7	PC6	PC5	PJ3	PJ1
Initial value:	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/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	PG11	0	R/W	Cancel by Change on PG11 0: Deep standby mode is not canceled by change on the PG11 pin. 1: Deep standby mode is canceled by change on the PG11 pin. Note: For 1-Mbyte version, this bit is reserved and always read as 0. The write value should always be 0.
9	PG10	0	R/W	Cancel by Change on PG10 0: Deep standby mode is not canceled by change on the PG10 pin. 1: Deep standby mode is canceled by change on the PG10 pin. Note: For 1-Mbyte version, this bit is reserved and always read as 0. The write value should always be 0.
8, 7	—	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
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	PC8	0	R/W	Cancel by Change on PC8 0: Deep standby mode is not canceled by change on the PC8 pin. 1: Deep standby mode is canceled by change on the PC8 pin.
4	PC7	0	R/W	Cancel by Change on PC7 0: Deep standby mode is not canceled by change on the PC7 pin. 1: Deep standby mode is canceled by change on the PC7 pin.
3	PC6	0	R/W	Cancel by Change on PC6 0: Deep standby mode is not canceled by change on the PC6 pin. 1: Deep standby mode is canceled by change on the PC6 pin.
2	PC5	0	R/W	Cancel by Change on PC5 0: Deep standby mode is not canceled by change on the PC5 pin. 1: Deep standby mode is canceled by change on the PC5 pin.
1	PJ3	0	R/W	Cancel by Change on PJ3 0: Deep standby mode is not canceled by change on the PJ3 pin. 1: Deep standby mode is canceled by change on the PJ3 pin.
0	PJ1	0	R/W	Cancel by Change on PJ1 0: Deep standby mode is not canceled by change on the PJ1 pin. 1: Deep standby mode is canceled by change on the PJ1 pin.

chanhvinh.com

33.2.18 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 33.4, Usage Notes.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	PG11E	PG10E	NMIE	-	-	PC8E	PC7E	PC6E	PC5E	PJ3E	PJ1E
Initial value:	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/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	PG11E	0	R/W	PG11 Edge Detection 0: Falling edge of PG11 is detected. 1: Rising edge of PG11 is detected. Note: For 1-Mbyte version, this bit is reserved and always read as 0. The write value should always be 0.
9	PG10E	0	R/W	PG10 Edge Detection 0: Falling edge of PG10 is detected. 1: Rising edge of PG10 is detected. Note: For 1-Mbyte version, this bit is reserved and always read as 0. The write value should always be 0.
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	PC8E	0	R/W	PC8 Edge Detection 0: Falling edge of PC8 is detected. 1: Rising edge of PC8 is detected.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
4	PC7E	0	R/W	PC7 Edge Detection 0: Falling edge of PC7 is detected. 1: Rising edge of PC7 is detected.
3	PC6E	0	R/W	PC6 Edge Detection 0: Falling edge of PC6 is detected. 1: Rising edge of PC6 is detected.
2	PC5E	0	R/W	PC5 Edge Detection 0: Falling edge of PC5 is detected. 1: Rising edge of PC5 is detected.
1	PJ3E	0	R/W	PJ3 Edge Detection 0: Falling edge of PJ3 is detected. 1: Rising edge of PJ3 is detected.
0	PJ1E	0	R/W	PJ1 Edge Detection 0: Falling edge of PJ1 is detected. 1: Rising edge of PJ1 is detected.

chanhvinh.com

33.2.19 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, realtime clock alarm interrupt, or change on the pins for canceling) 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 33.4, Usage Notes.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IO KEEP	-	-	-	-	PG11F	PG10F	NMIF	-	RTC ARF	PC8F	PC7F	PC6F	PC5F	PJ3F	PJ1F
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/(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)*	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 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10	PG11F	0	R/(W)*	PG11 Flag 0: No change on the PG11 pin 1: Change on the PG11 pin Note: For 1-Mbyte version, this bit is reserved and always read as 0. The write value should always be 0.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
9	PG10F	0	R/(W)*	PG10 Flag 0: No change on the PG10 pin 1: Change on the PG10 pin Note: For 1-Mbyte version, this bit is reserved and always read as 0. The write value should always be 0.
8	NMIF	0	R/(W)*	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)*	RTCAR Flag 0: No realtime clock alarm interrupt generated 1: Realtime clock alarm Interrupt generated
5	PC8F	0	R/(W)*	PC8 Flag 0: No change on the PC8 pin 1: Change on the PC8 pin
4	PC7F	0	R/(W)*	PC7 Flag 0: No change on the PC7 pin 1: Change on the PC7 pin
3	PC6F	0	R/(W)*	PC6 Flag 0: No change on the PC6 pin 1: Change on the PC6 pin
2	PC5F	0	R/(W)*	PC5 Flag 0: No change on the PC5 pin 1: Change on the PC5 pin
1	PJ3F	0	R/(W)*	PJ3 Flag 0: No change on the PJ3 pin 1: Change on the PJ3 pin
0	PJ1F	0	R/(W)*	PJ1 Flag 0: No change on the PJ1 pin 1: Change on the PJ1 pin

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

chanhvinh.com

33.2.20 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 XTAL input, XTALCTR 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 XTAL input, XTALCTR is initialized to H'00 when software standby or deep standby mode is entered. XTALCTR is also 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 33.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	GAIN
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	GAIN	0	R/W	XTAL Crystal Oscillator Gain Select 0: Large gain 1: Small gain

chanhvinh.com

33.3 Operation

33.3.1 Sleep Mode

(1) Transition to Sleep Mode

Executing the SLEEP instruction when the STBY bit in STBCR1 is 0 causes a transition from the program execution state to sleep mode. Although the CPU halts immediately after executing the SLEEP instruction, the contents of its internal registers remain unchanged. The on-chip peripheral modules continue to run in sleep mode. The clock output from the CKIO pin is continued.

(2) Canceling Sleep Mode

Sleep mode is canceled by an interrupt (NMI, IRQ, and on-chip peripheral module), a DMA address error, or a reset (manual reset or 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 equal to or lower than the interrupt mask level that is set in the status register (SR) of the CPU, 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.
- Canceling by a DMA address error

  When a DMA address error occurs, sleep mode is canceled and DMA address error exception handling is executed.
- Canceling by a reset

  Sleep mode is canceled by a power-on reset or a manual reset.

chanhvinh.com

33.3.2 Software Standby Mode

(1) Transition to Software Standby Mode

The LSI switches from a program execution state to software standby mode by executing the SLEEP 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 36.3, Register States in Each Operating Mode.

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 SLEEP instruction after reading STBCR1 to have the values written to STBCR1 by the CPU to be definitely reflected in the SLEEP instruction.

The procedure for switching to software standby mode is as follows:

1. Clear the TME bit in the timer control register of the watchdog timer (WTCSR) to 0 to stop the watchdog time.
2. Set the timer counter of the watchdog timer (WTCNT) to 0 and the CKS[2:0] bits in WTCSR to appropriate values to secure the specified oscillation settling time.
3. After setting the STBY and DEEP bits in STBCR1 to 1 and 0 respectively, read STBCR. Then, execute a SLEEP 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 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 equal to or lower than the interrupt mask level specified in the status register (SR) of the CPU, the interrupt request is not accepted and software standby mode is not canceled.

  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. When software standby mode is canceled by the falling edge of the NMI pin, the NMI pin should be high when software standby mode is entered (when the clock pulse stops) and should be low when software standby mode is canceled (when the clock is initiated after the oscillation settling). When software standby mode is canceled by the rising edge of the NMI pin, the NMI pin should be low when software standby mode is entered (when the clock pulse stops) and should be high when software standby mode is canceled (when the clock is initiated after the oscillation settling) (This is the same with the IRQ pin.)

- 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.

chanhvinh.com

(3) Note on Release from Software Standby Mode

Release from software standby mode is triggered by interrupts (NMI and IRQ) or resets (manual reset and power-on reset). If, however, a SLEEP 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 are not generated before execution of the SLEEP 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, bits 13 and 12 in FRQCR should be modified.

chanhvinh.com

33.3.3 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 33.1.

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 SLEEP 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.

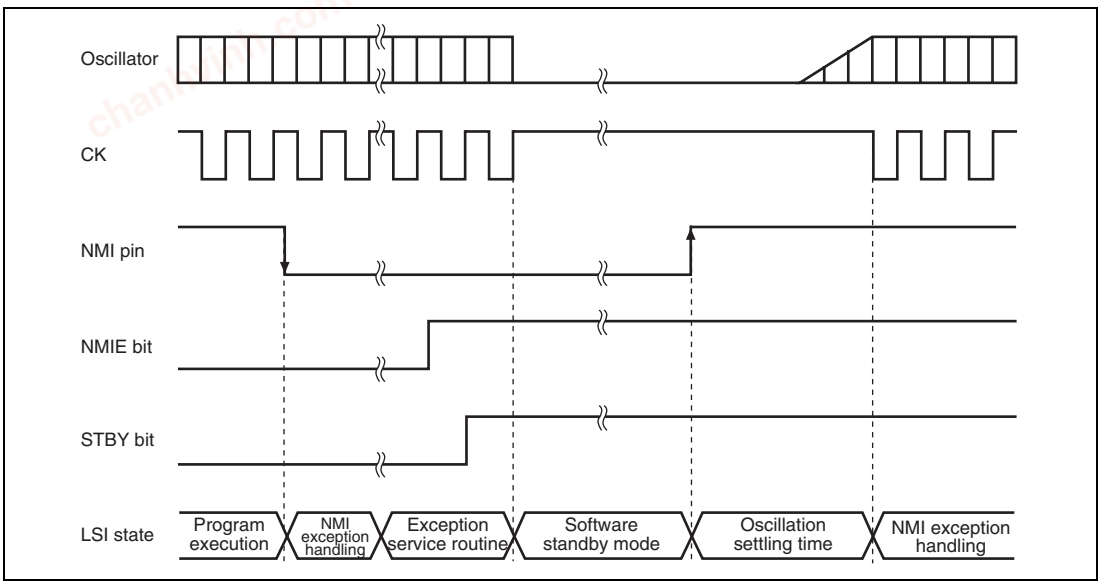


Figure 33.1 NMI Timing in Software Standby Mode (Application Example)

chanhvinh.com

33.3.4 Deep Standby Mode

(1) Transition to Deep Standby Mode

The LSI switches from a program execution state to deep standby mode by executing the SLEEP 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 DSFR, and then executes processing for the next instruction. However, it actually takes one or more cycles to write. Therefore, execute a SLEEP instruction after reading DSFR to reflect the values written to DSFR by the CPU in the SLEEP instruction without fail.

The procedure for switching to deep standby mode is as follows. Figure 33.2 also shows its flowchart.

1. 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.
2. 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.
3. 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.
4. 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.
5. Set the STBY and DEEP bits in STBCR1 to 1.
6. Read out the DSFR register after clearing the flag in the DSFR register. Then execute the SLEEP instruction.

chanhvinh.com

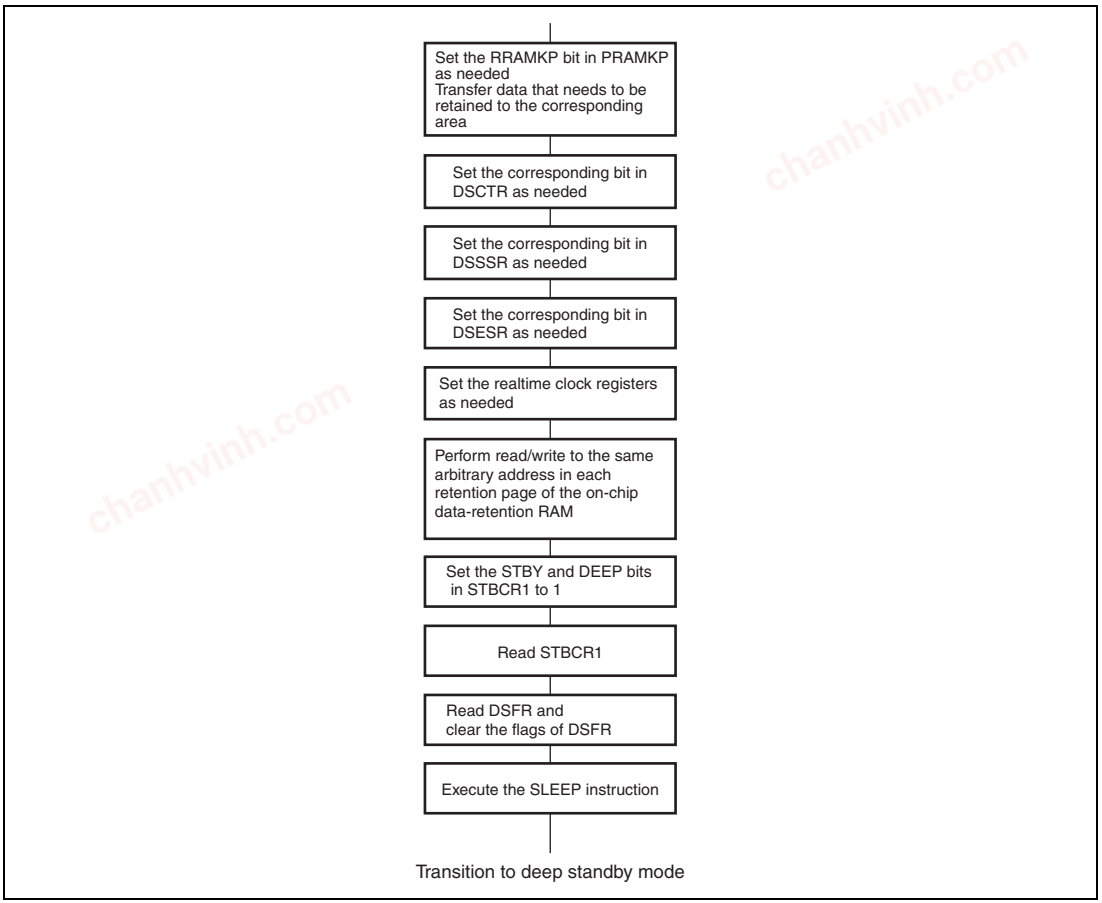


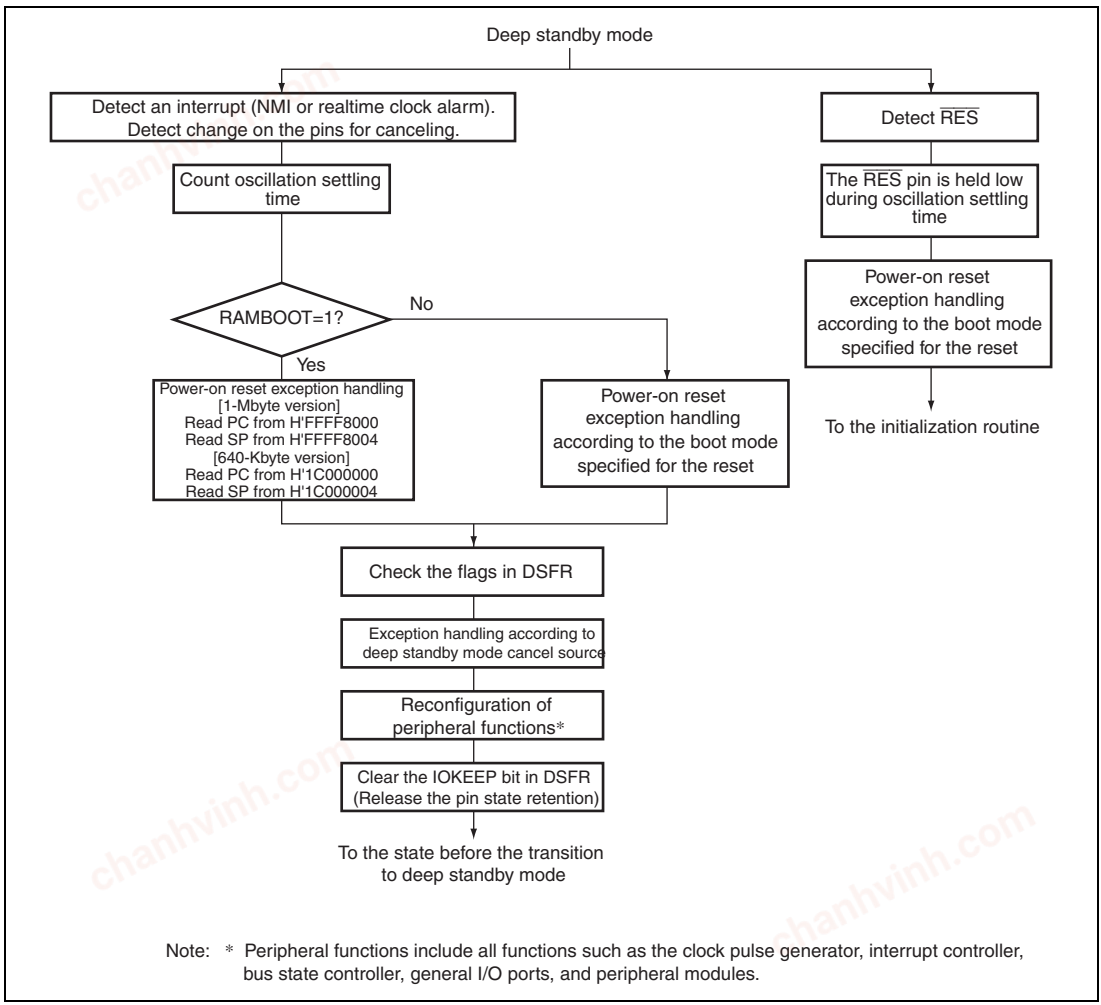
Figure 33.2 Flowchart of Transition to Deep Standby Mode

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 interrupt priority level or the status register (SR) setting in 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 33.3 shows the flowchart of canceling deep standby mode.



Note: * Peripheral functions include all functions such as the clock pulse generator, interrupt controller, bus state controller, general I/O ports, and peripheral modules.

Figure 33.3 Flowchart of Canceling Deep Standby Mode

chanhvinh.com

- 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 14.4.4, 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 an interrupt and until deep standby mode is canceled. When deep standby mode is canceled by the falling edge of the NMI pin, the NMI pin should be high when deep standby mode is entered (when the clock pulse stops) and should be low when deep standby mode is canceled (when the clock is initiated after the oscillation settling). When deep standby mode is canceled by the rising edge of the NMI pin, the NMI pin should be low when deep standby mode is entered (when the clock pulse stops) and should be high when deep standby mode is canceled (when the clock is initiated after the oscillation settling). (This is the same with the pins for canceling.)

  Also, the NMI and all the pins which are selected by DSSSR to cancel deep standby mode should be the following pin levels when the CPU enters deep standby, regardless whether they will really cancel deep standby mode or not.

  — The pins that are set as rising edge should be low when the CPU enters deep standby mode.

  — The pins that are set as falling edge should be high when the CPU enters 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.

chanhvinh.com

(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 33.3 shows the pin states after cancellation of deep standby mode according to the setting of each bit. Table 33.4 lists the external memory control pins.

Table 33.3 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 33.4 External Memory Control Pins in Different Modes

Boot Mode 0 (CS0 Area)	Boot Mode 2 (NAND Flash Memory)	Boot Mode 1, 3 (Serial Flash Memory)
A[20:1] D[15:0] $\overline{CS0}$, $\overline{RD}$, CKIO	NAF[7:0] $\overline{FRE}$, FCLE, FALE, $\overline{FEW}$, $\overline{FCE}$, FRB	RSPCK0, SSL00, MOSI0, MISO0

chanhvinh.com

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.

(4) Notes on Transition to Deep Standby Mode

After deep standby mode is specified, interrupts other than that selected with the deep standby cancel source select register are masked. That is, only an interrupt selected with the deep standby cancel source select register will be accepted. If, however, multiple canceling sources have been specified and multiple canceling sources are input, multiple cancel source flags will be set.

In addition, if a SLEEP instruction to initiate the transition to deep standby mode coincides with the NMI, an IRQ interrupt, or a manual reset, acceptance of the interrupt may initiate the cancellation of deep standby mode.

chanhvinh.com

33.3.5 Module Standby Function

(1) Transition to Module Standby Function

Setting the standby control register MSTP bits to 1 halts the supply of clocks to the corresponding on-chip peripheral modules. This function can be used to reduce the power consumption in the program execution state and sleep mode. Disable a module before placing it in the module standby mode. In addition, do not access the module's registers while it is in the module standby state.

For details on the states of registers, see section 36.3, Register States in Each Operating Mode.

(2) Canceling Module Standby Function

The module standby function can be canceled by clearing each MSTP bit to 0, or by a power-on reset (only possible for the realtime clock, user debugging interface, and direct memory access controller). When taking a module out of the module standby state by clearing the corresponding MSTP bit to 0, read the MSTP bit to confirm that it has been cleared to 0.

33.3.6 Adjustment of XTAL Crystal Oscillator Gain

The gain of the crystal oscillator can be adjusted using the GAIN bit in XTALCTR. To modify the gain, PLL settling time is needed. The settling time is counted using the on-chip watchdog timer.

1. The large gain is selected in the initial state.
2. 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.)
3. Set the GAIN bit to the desired value.
4. 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 36.3, Register States in Each Operating Mode.
5. 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

33.4 Usage Notes

33.4.1 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.

33.4.2 Usage Notes when the Realtime Clock is not Used

When the realtime clock is not used, set the MSTP30 bit in STBCR3 to 1 after setting the bits in registers of the realtime clock shown below.

- 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.

chanhvinh.com

Section 34 User Debugging Interface

This LSI incorporates a user debugging interface for emulator support.

34.1 Features

The user debugging interface is a serial input/output interface that supports JTAG-standard, IEEE Std.1149.1.

This module incorporates a TAP controller for controlling the user debugging interface interrupt function. When the emulation enable command is input, emulation commands become available. When the $\overline{\text{TRST}}$ pin is asserted, emulation commands are disabled.

In ASE mode, emulation commands are available. For connection with the emulator, see the manual for the emulator.

Figure 34.1 shows a block diagram.

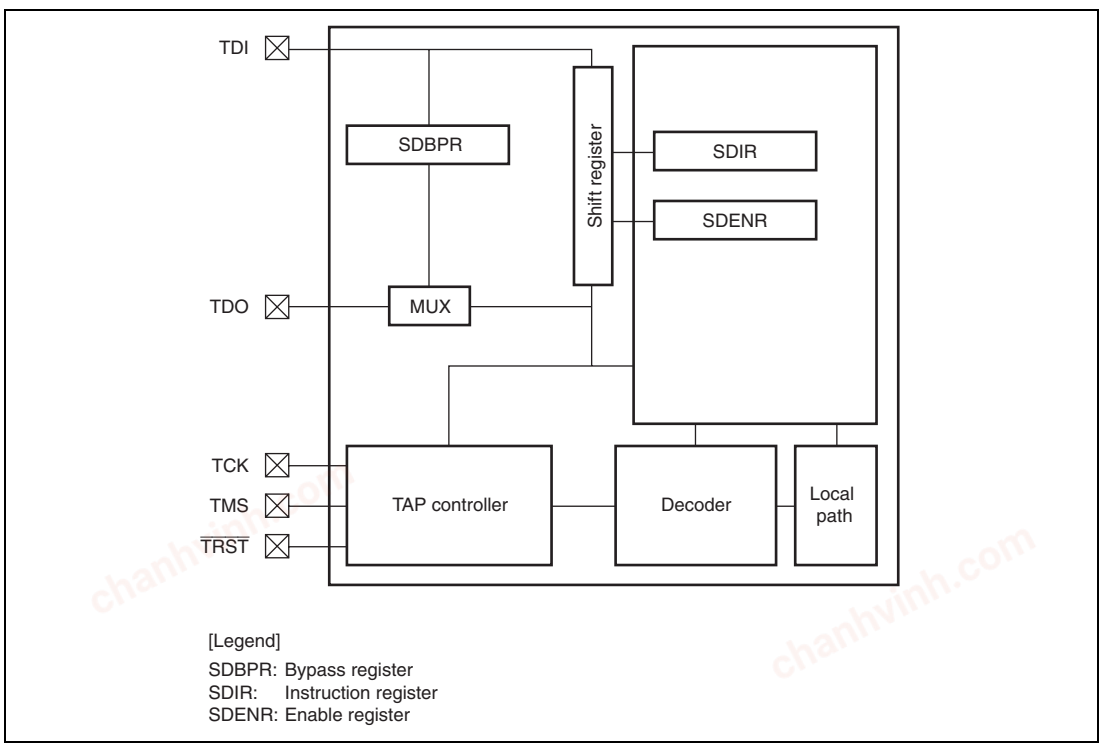


Figure 34.1 Block Diagram

chanhvinh.com

34.2 Input/Output Pins

Table 34.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Serial data input/output clock pin	TCK	Input	Data is serially supplied to this module from the data input pin (TDI), and output from the data output pin (TDO), in synchronization with this clock.
Mode select input pin	TMS	Input	The state of the TAP control circuit is determined by changing this signal in synchronization with TCK. The protocol complies with the JTAG standard (IEEE Std.1149.1).
Reset input pin	$\overline{\text{TRST}}$	Input	Input is accepted asynchronously with respect to TCK, and when low, this module is reset. $\overline{\text{TRST}}$ must be low for a period when power is turned on regardless of using the function. See section 34.4.2, Reset Configuration, for more information.
Serial data input pin	TDI	Input	Data is transferred to this module by changing this signal in synchronization with TCK.
Serial data output pin	TDO	Output	Data is read from this module by reading this pin in synchronization with TCK. The initial value of the data output timing is the TCK falling edge, but this initial value can be changed to the TCK rising edge by inputting the TDO transition timing switching command to SDIR. See section 34.4.3, TDO Output Timing, for more information.
ASE mode select pin	$\overline{\text{ASEMD}}$*	Input	If a low level is input at the $\overline{\text{ASEMD}}$ pin while the $\overline{\text{RES}}$ pin is asserted, ASE mode is entered; if a high level is input, product chip mode is entered. In ASE mode, dedicated emulator function can be used. The input level at the $\overline{\text{ASEMD}}$ pin should be held for at least one cycle after $\overline{\text{RES}}$ negation.

Note: * When the emulator is not in use, fix this pin to the high level.

chanhvinh.com

34.3 Description of the Emulation Command Registers

To use emulation commands, enter the emulation enable command in SDENR. The following registers are provided for emulation.

Table 34.2 Register Configuration for Emulation

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Bypass register	SDBPR	—	—	—	—
Instruction register	SDIR	R	H'EFFD	H'FFFE2000	16
Enable register	SDENR	—	H'4	—	—

34.3.1 Bypass Register (SDBPR)

SDBPR is a 1-bit register that cannot be accessed by the CPU. When SDENR or SDIR is set to BYPASS mode, SDBPR is connected between pins TDI and TDO pins. The initial value is undefined.

34.3.2 Instruction Register (SDIR)

SDIR is a 16-bit read-only register. To use this register, the emulation enable command should be set in SDENR. This register is initialized by $\overline{\text{TRST}}$ assertion or in the TAP test-logic-reset state. This module can write to this register regardless of the CPU mode. When a reserved command is set in this register, the operation is not guaranteed. The initial value is H'EFFD.

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

Note: * The initial value of TI[7:0] is a reserved value, but replace it with a non-reserved value when setting a command.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	TI[7:0]	11101111*	R	Test Instruction The instruction of this module is transferred to SDIR as a serial input from TDI. For commands, see table 34.3.

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 1	R	Reserved These bits are always read as 1.
1	—	0	R	Reserved These bits are always read as 0.
0	—	1	R	Reserved These bits are always read as 1.

Table 34.3 Emulation Commands

Bits 15 to 8								
TI7	TI6	TI5	TI4	TI3	TI2	TI1	TI0	Description
0	1	1	0	—	—	—	—	User debugging interface reset negation
0	1	1	1	—	—	—	—	User debugging interface reset assertion
1	0	0	1	1	1	0	0	TDO transition timing switch
1	0	1	1	—	—	—	—	User debugging interface interrupt
1	1	1	1	—	—	—	—	BYPASS
Other than the above								Reserved

chanhvinh.com

34.3.3 Enable Register (SDENR)

SDENR is a 4-bit register and initialized by $\overline{\text{TRST}}$ assertion or in the TAP test-logic-reset state. This register cannot be accessed by the CPU. After the emulation enable command is entered, the serial input from TDI is transferred to SDIR.

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	EE[3:0]	0100	—	Emulation Enable The serial input from TDI is transferred to SDENR. For commands, see table 34.4.

Table 34.4 Emulation Enable Command

Bits 3 to 0				
TI3	**TI2**	**TI1**	**TI0**	**Description**
0	0	1	1	Emulation enable command
0	1	0	0	Emulation is disabled (initial value)
1	1	1	1	BYPASS
Other than the above				Reserved

chanhvinh.com

34.4 Operation

34.4.1 TAP Controller

Figure 34.2 shows the internal states of the TAP controller. This state machine conforms to the state transitions defined by JTAG.

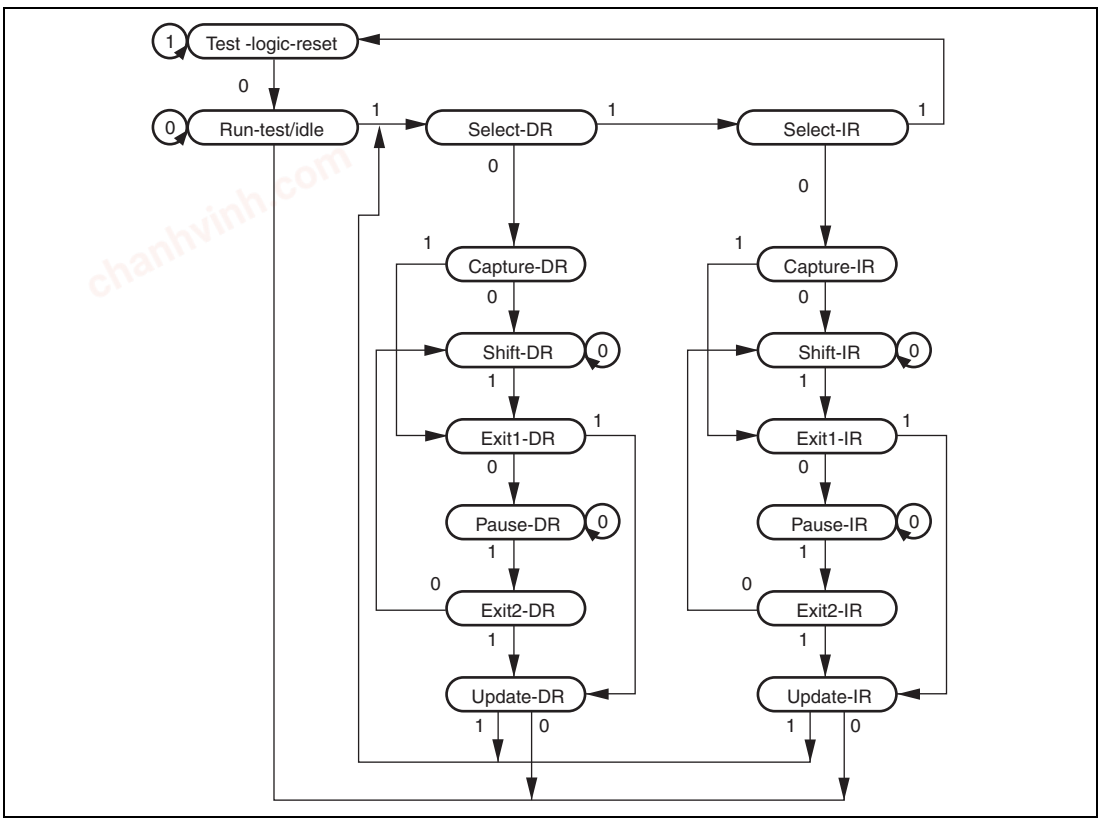


Figure 34.2 TAP Controller State Transitions

Note: The transition condition is the TMS value at the rising edge of TCK. The TDI value is sampled at the rising edge of TCK; shifting occurs at the falling edge of TCK. For details on transition timing of the TDO value, see section 34.4.3, TDO Output Timing. The TDO is at high impedance, except with shift-DR and shift-IR states. During the change to $\overline{\text{TRST}}$ = 0, there is a transition to test-logic-reset asynchronously with TCK.

chanhvinh.com

34.4.2 Reset Configuration

Table 34.5 Reset Configuration

$\overline{\text{ASEMD}}$*1	$\overline{\text{RES}}$	$\overline{\text{TRST}}$	Chip State
H	L	L	Power-on reset and the reset of this module
		H	Power-on reset
	H	L	Reset this module only
		H	Normal operation
L	L	L	Reset hold*2
		H	Power-on reset
	H	L	Reset this module only
		H	Normal operation

Notes: 1. Performs product chip mode and ASE mode settings
$\overline{\text{ASEMD}}$ = H, normal mode
$\overline{\text{ASEMD}}$ = L, ASE mode
2. In ASE mode, reset hold is entered if the $\overline{\text{TRST}}$ pin is driven low while the $\overline{\text{RES}}$ pin is negated. In this state, the CPU does not start up. When $\overline{\text{TRST}}$ is driven high, the operation of this module is enabled, but the CPU does not start up. The reset hold state is cancelled by a power-on reset.

34.4.3 TDO Output Timing

When the emulation command is enabled, a transition on the TDO pin is output on the falling edge of TCK with the initial value. However, setting a TDO transition timing switching command in SDIR via the pin and passing the Update-IR state synchronizes the TDO transition with the rising edge of TCK.

To synchronize the transition of TDO with the falling edge of TCK after setting the TDO transition timing switching command, the $\overline{\text{TRST}}$ pin must be asserted simultaneously with the power-on reset. In the case of power-on reset by the $\overline{\text{RES}}$ pin, the sync reset is still in operation for a certain period in the LSI even after the $\overline{\text{RES}}$ pin is negated. Thus, if the $\overline{\text{TRST}}$ pin is asserted immediately after the negation of the $\overline{\text{RES}}$ pin, the TDO transition timing switching command is cleared, resulting in TDO transitions synchronized with the falling edges of TCK. To prevent this, make sure to allow a period of 20 tcyc or longer between the signal transitions of the $\overline{\text{RES}}$ and $\overline{\text{TRST}}$ pins.

chanhvinh.com

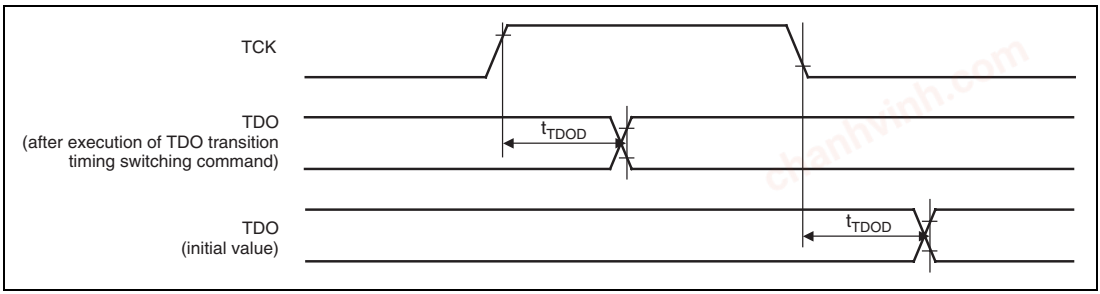


Figure 34.3 User Debugging Interface Data Transfer Timing

34.4.4 User Debugging Interface Reset

A user debugging interface reset occurs when a user debugging interface reset assert command is set in SDIR. A user debugging interface reset is of the same kind as a power-on reset. A user debugging interface reset is cleared by setting a user debugging interface reset negate command. The required time between the user debugging interface reset assert command and user debugging interface reset negate command is the same as time for keeping the $\overline{\text{RES}}$ pin low to apply a power-on reset.

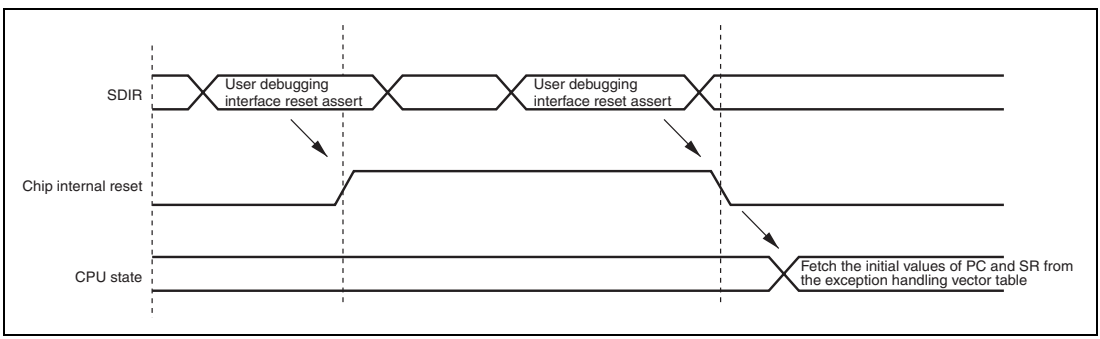


Figure 34.4 User debugging interface Reset

34.4.5 User Debugging Interface Interrupt

The user debugging interface interrupt function generates an interrupt by setting a command from the user debugging interface into SDIR. A user debugging interface interrupt is a general exception/interrupt operation, resulting in fetching the exception service routine start address from the exception handling vector table, jumping to that address, and starting program execution from that address. This interrupt request has a fixed priority level of 15.

User debugging interface interrupts are accepted in sleep mode, but not in software standby mode.

chanhvinh.com

34.5 Usage Notes

1. Once a command of this module has been set, it will not be modified until another command is not set again. If the same command is to be set continuously, the command must be set after a command (BYPASS mode, etc.) that does not affect chip operations is once set.
2. In software standby mode and in this module's standby state, none of the functions of this module can be used. To retain the TAP status before and after standby mode, keep TCK high before entering standby mode.
3. Regardless of whether this module is used, make sure to keep the $\overline{\text{TRST}}$ pin low to initialize this module at power-on or in recovery from deep standby by the $\overline{\text{RES}}$ pin assertion.
4. If the TRST pin is asserted immediately after the setting of the TDO transition timing switching command and the negation of the RES pin, the TDO transition timing switching command is cleared. To avoid this case, make sure to put 20 tcyc or longer between the signal transition timing of the $\overline{\text{RES}}$ and $\overline{\text{TRST}}$ pins. For details, see section 34.4.3, TDO Output Timing.
5. When starting the TAP controller after the negation of the $\overline{\text{TRST}}$ pin, make sure to allow 200 ns or longer after the negation.
6. Please keep TMS pin high for 200 ns from $\overline{\text{TRST}}$ pin negation.

chanhvinh.com

chanhvinh.com
chanhvinh.com
chanhvinh.com
chanhvinh.com

chanhvinh.com

Section 35 Motor Control PWM Timer

This LSI has two channels of on-chip motor control PWM (pulse width modulator) timer with a maximum capability of eight pulse outputs for each channel.

35.1 Features

- Maximum of 16 pulse outputs
  - Two 10-bit PWM channels, each with eight outputs.
  - 10-bit counter (PWCNT) and cycle register (PWCYR).
  - Duty and output polarity can be set for each output.
- Automatic data transfer in every cycle
  - Each of four duty registers (PWDTR) is provided with buffer registers (PWBFR), with data transferred automatically every cycle.
- Duty settings selectable
  - A duty cycle of 0% to 100% can be selected by means of a duty register setting.
- Counting clock selectable
  - There is a choice of five counting clocks (Pϕ, Pϕ/2, Pϕ/4, Pϕ/8, Pϕ/16).
- High-speed access via internal 16-bit bus
- Two interrupt sources
  - An interrupt can be requested independently for each channel by a cycle register compare match.
- Automatic transfer of register data
  - Block transfer and one-word data transfer are available by activating the direct memory access controller.
- Module stop mode can be set

chanhvinh.com

Figure 35.1 shows a block diagram of the motor control PWM timer.

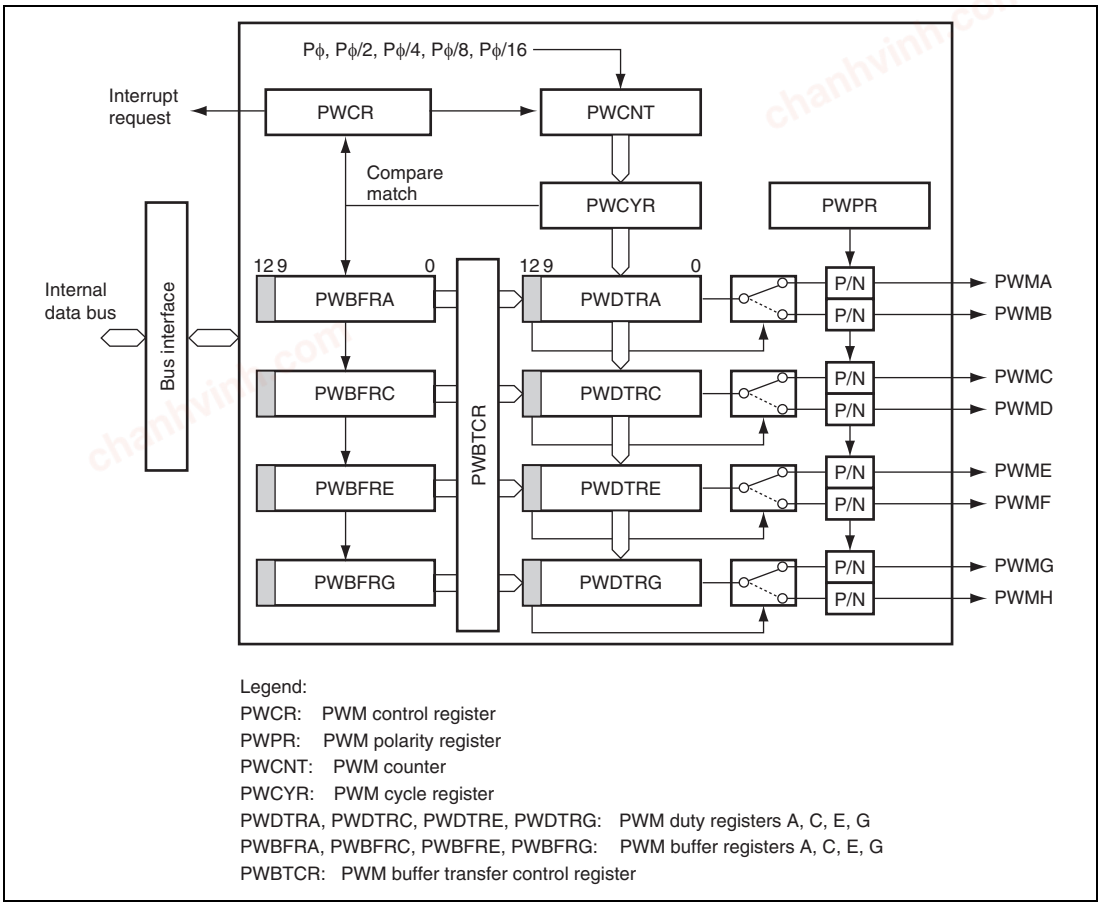


Figure 35.1 Block Diagram of Motor Control PWM Timer

chanhvinh.com

35.2 Input/Output Pins

Table 35.1 shows the pin configuration of this module.

Table 35.1 Pin Configuration

Channel	Name	Abbrev.	I/O	Function
1	PWM output pin 1A	PWM1A	Output	Channel 1A PWM output
	PWM output pin 1B	PWM1B	Output	Channel 1B PWM output
	PWM output pin 1C	PWM1C	Output	Channel 1C PWM output
	PWM output pin 1D	PWM1D	Output	Channel 1D PWM output
	PWM output pin 1E	PWM1E	Output	Channel 1E PWM output
	PWM output pin 1F	PWM1F	Output	Channel 1F PWM output
	PWM output pin 1G	PWM1G	Output	Channel 1G PWM output
	PWM output pin 1H	PWM1H	Output	Channel 1H PWM output
2	PWM output pin 2A	PWM2A	Output	Channel 2A PWM output
	PWM output pin 2B	PWM2B	Output	Channel 2B PWM output
	PWM output pin 2C	PWM2C	Output	Channel 2C PWM output
	PWM output pin 2D	PWM2D	Output	Channel 2D PWM output
	PWM output pin 2E	PWM2E	Output	Channel 2E PWM output
	PWM output pin 2F	PWM2F	Output	Channel 2F PWM output
	PWM output pin 2G	PWM2G	Output	Channel 2G PWM output
	PWM output pin 2H	PWM2H	Output	Channel 2H PWM output

chanhvinh.com

35.3 Register Descriptions

This module has the following registers for each channel as listed in table 35.2.

Table 35.2 Register Description

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
PWM control register_1	PWCR_1	R/W	H'C0	H'FFFEF4E0	8, 16
PWM polarity register_1	PWPR_1	R/W	H'00	H'FFFEF4E4	8, 16
PWM cycle register_1	PWCYR_1	R/W	H'FFFF	H'FFFEF4E6	16
PWM buffer register_1A	PWBFR_1A	R/W	H'EC00	H'FFFEF4E8	16
PWM buffer register_1C	PWBFR_1C	R/W	H'EC00	H'FFFEF4EA	16
PWM buffer register_1E	PWBFR_1E	R/W	H'EC00	H'FFFEF4EC	16
PWM buffer register_1G	PWBFR_1G	R/W	H'EC00	H'FFFEF4EE	16
PWM control register_2	PWCR_2	R/W	H'C0	H'FFFEF4F0	8, 16
PWM polarity register_2	PWPR_2	R/W	H'00	H'FFFEF4F4	8, 16
PWM cycle register_2	PWCYR_2	R/W	H'FFFF	H'FFFEF4F6	16
PWM buffer register_2A	PWBFR_2A	R/W	H'EC00	H'FFFEF4F8	16
PWM buffer register_2C	PWBFR_2C	R/W	H'EC00	H'FFFEF4FA	16
PWM buffer register_2E	PWBFR_2E	R/W	H'EC00	H'FFFEF4FC	16
PWM buffer register_2G	PWBFR_2G	R/W	H'EC00	H'FFFEF4FE	16
PWM buffer transfer control register	PWBTCR	R/W	H'00	H'FFFEF406	8, 16

chanhvinh.com

35.3.1 PWM Control Register_n (PWCR_n) (n = 1, 2)

PWCR_n performs interrupt control, starting/stopping of the counter, and counter clock selection. It also contains a flag that indicates a compare match with the cycle register.

Bit	7	6	5	4	3	2	1	0
Bit Name	—	—	IE	CMF	CST	CKS2	CKS1	CKS0
Initial Value	1	1	0	0	0	0	0	0
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 1	—	Reserved These bits are always read as 1 and cannot be modified.
5	IE	0	R/W	Interrupt Enable Enables or disables an interrupt request in the event of a compare match with PWCYR_n of the corresponding channel. 0: Interrupt disabled 1: Interrupt enabled
4	CMF	0	R/(W)*	Compare Match Flag Indicates the occurrence of a compare match with PWCYR_n of the corresponding channel. [Setting condition] When PWCNT_n = PWCYR_n – 1 [Clearing conditions] • When 0 is written to CMF after reading CMF = 1 • When the direct memory access controller is activated by a compare match interrupt, and the DTA bit in DMDR of the direct memory access controller is 1 (When the CPU is used to clear this flag by writing 0 while the corresponding interrupt is enabled, be sure to read the flag after writing 0 to it.)

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
3	CST	0	R/W	Counter Start Selects starting or stopping of PWCNT_n of the corresponding channel. 0: PWCNT_n is stopped 1: PWCNT_n is started
2	CKS2	0	R/W	Clock Select
1	CKS1	0	R/W	These bits select the operating clock for PWCNT_n of the corresponding channel.
0	CKS0	0	R/W	000: Counts on Pϕ/1 001: Counts on Pϕ/2 010: Counts on Pϕ/4 011: Counts on Pϕ/8 1XX: Counts on Pϕ/16

[Legend]
X: Don't care
Note: * Only 0 can be written, to clear the flag.

35.3.2 PWM Polarity Register_n (PWPR_n) (n = 1, 2)

PWPR_n selects the PWM output polarity.

Bit	7	6	5	4	3	2	1	0
Bit Name	OPSnH	OPSnG	OPSnF	OPSnE	OPSnD	OPSnC	OPSnB	OPSnA
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

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
7	OPSnH	0	R/W	Output Polarity Select
6	OPSnG	0	R/W	Each of these bits selects the PWM output polarity.
5	OPSnF	0	R/W	0: PWM direct output
4	OPSnE	0	R/W	1: PWM inverse output
3	OPSnD	0	R/W	
2	OPSnC	0	R/W	
1	OPSnB	0	R/W	
0	OPSnA	0	R/W	

(n = 1, 2)

35.3.3 PWM Counter_n (PWCNT_n) (n = 1, 2)

PWCNT_n is a 10-bit up-counter incremented by the input clock. The input clock is selected by clock select bits CKS2 to CKS0 in PWCR_n.

PWCNT_n can not be directly accessed by the CPU. PWCNT_n is initialized to H'FC00, when the CST bit in PWCRn is cleared to 0.

35.3.4 PWM Cycle Register_n (PWCYR_n) (n = 1, 2)

PWCYR_n is a 16-bit readable/writable register that sets the PWM conversion cycle.

Bit:	15	14	13	12	11	10	9	8
	PWCY15	PWCY14	PWCY13	PWCY12	PWCY11	PWCY10	PWCY9	PWCY8
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

Bit:	7	6	5	4	3	2	1	0
	PWCY7	PWCY6	PWCY5	PWCY4	PWCY3	PWCY2	PWCY1	PWCY0
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 a PWCYR_n compare match occurs, PWCNT_n is cleared and data is transferred from the buffer register (PWBFR_n) to the duty register (PWDTR_n).

PWCYR_n should be written to only while PWCNT_n is stopped. A value of H'FC00 must not be set to PWCYR_n.

chanhvinh.com

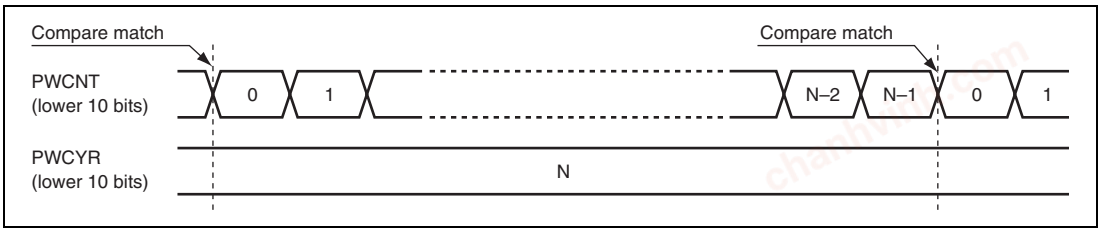


Figure 35.2 Cycle Register Compare Match

35.3.5 PWM Duty Registers_nA, nC, nE, nG (PWDTR_nA, PWDTR_nC, PWDTR_nE, PWDTR_nG) (n = 1, 2)

There are four PWDTR_n registers (PWDTR_nA, PWDTR_nC, PWDTR_nE, and PWDTR_nG). The PWDTR_nA is used for outputs PWMnA and PWMnB, PWDTR_nC for outputs PWMnC and PWMnD, PWDTR_nE for outputs PWMnE and PWMnF, and PWDTR_nG for outputs PWMnG and PWMnH.

PWDTR_n can not be directly accessed by the CPU. When a PWCYR_n compare match occurs, data is transferred from the buffer register (PWBFR_n) to the duty register (PWDTR_n). PWDTR_n is initialized to H'00 when the CST bit is cleared to 0.

Bit	15	14	13	12	11	10	9	8
Bit Name	—	—	—	OTS	—	—	DT9	DT8
Initial Value	—	—	—	0	—	—	0	0
R/W	—	—	—	—	—	—	—	—

Bit	7	6	5	4	3	2	1	0
Bit Name	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
Initial Value	0	0	0	0	0	0	0	0
R/W	—	—	—	—	—	—	—	—

chanhvinh.com

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	—	—	Reserved
12	OTS	0	—	Output Terminal Select Selects the pin used for PWM output. Unselected pins output a low level (or a high level when the corresponding bit in PWPR_n is set to 1). For details, see table 35.3.
11, 10	—	—	—	Reserved
9	DT9	0	—	Duty These bits specify the PWM output duty. A high level (or a low level when the corresponding bit in PWPR_n is set to 1) is output from the time PWCNT_n is cleared by a PWCYR_n compare match until a PWDTR_n compare match occurs. When all of the bits are 0, there is no high-level (or low-level when the corresponding bit in PWPR_n is set to 1) output period.
8	DT8	0	—	
7	DT7	0	—	
6	DT6	0	—	
5	DT5	0	—	
4	DT4	0	—	
3	DT3	0	—	
2	DT2	0	—	
1	DT1	0	—	
0	DT0	0	—	

Table 35.3 Output Selection by OTS Bit

	Bit 12	
Register	OTS	Description
PWDTR_1A/ PWDTR_2A	0	PWMnA output selected
	1	PWMnB output selected
PWDTR_1C/ PWDTR_2C	0	PWMnC output selected
	1	PWMnD output selected
PWDTR_1E/ PWDTR_2E	0	PWMnE output selected
	1	PWMnF output selected
PWDTR_1G/ PWDTR_2G	0	PWMnG output selected
	1	PWMnH output selected

chanhvinh.com

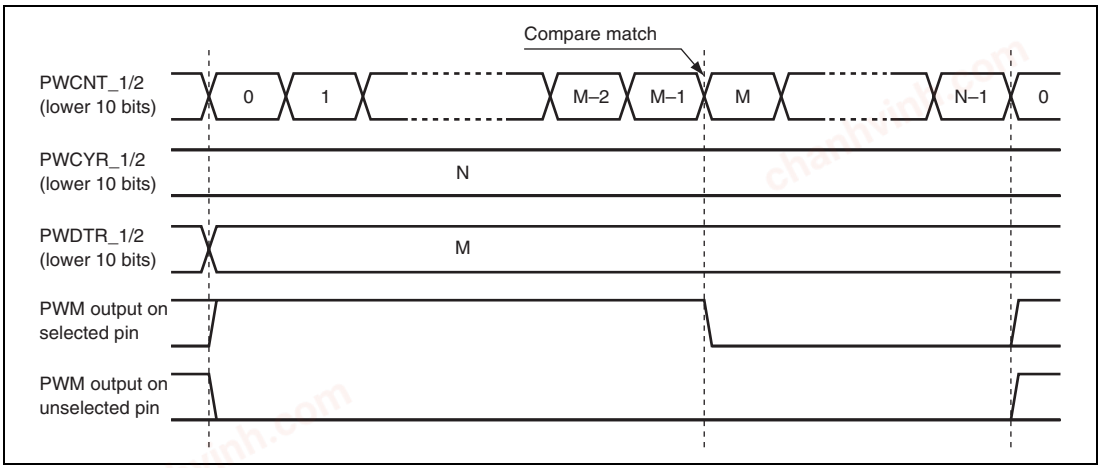


Figure 35.3 Duty Register Compare Match (OPS = 0 in PWPR_n)

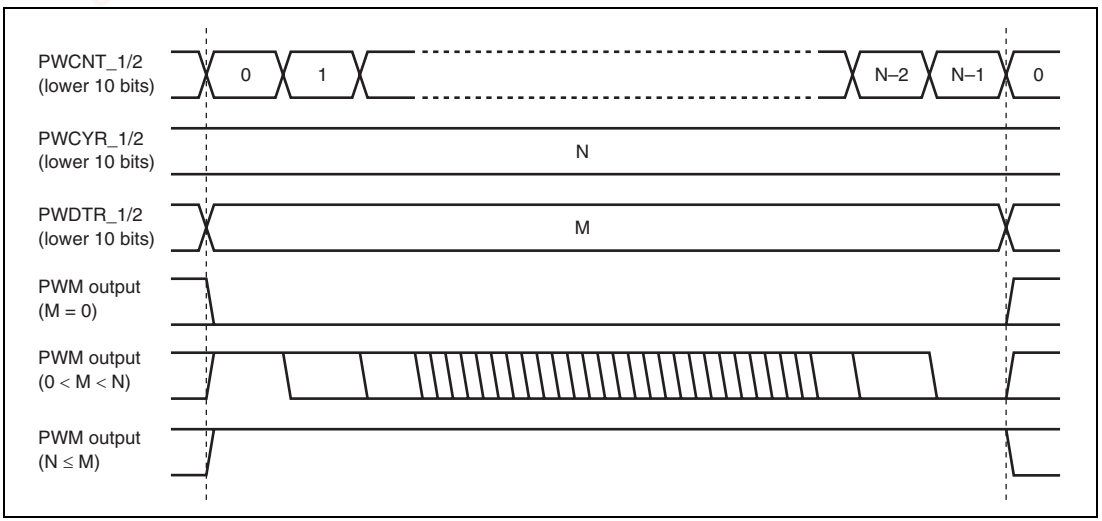


Figure 35.4 Differences in PWM Output According to Duty Register Set Value (OPS = 0 in PWPR_n)

chanhvinh.com

35.3.6 PWM Buffer Registers_nA, nC, nE, nG (PWBFR_nA, PWBFR_nC, PWBFR_nE, PWBFR_nG) (n = 1, 2)

There are four PWBFR_n registers (PWBFR_nA, PWBFR_nC, PWBFR_nE, and PWBFR_nG). When a PWCYR_n compare match occurs, data is transferred from the buffer register (PWBFR_n) to the duty register (PWDTR_n).

Bit:	15	14	13	12	11	10	9	8
	—	—	—	OTS	—	—	DT9	DT8
Initial Value:	1	1	1	0	1	1	0	0
R/W:	R	R	R	R/W	R	R	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
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
15 to 13	—	All 1	R	Reserved These bits are always read as 1 and cannot be modified.
12	OTS	0	R/W	Output Terminal Select Holds the data to be sent to bit 12 in PWDTR_n.
11, 10	—	All 1	R	Reserved These bits are always read as 1 and cannot be modified.
9	DT9	0	R/W	Duty These bits hold the data to be sent to bits 9 to 0 in PWDTR_n.
8	DT8	0	R/W	
7	DT7	0	R/W	
6	DT6	0	R/W	
5	DT5	0	R/W	
4	DT4	0	R/W	
3	DT3	0	R/W	
2	DT2	0	R/W	
1	DT1	0	R/W	
0	DT0	0	R/W	

chanhvinh.com

35.3.7 PWM Buffer Transfer Control Register (PWBTCR)

PWBTCR enables or disables the data transfer from buffer register to duty register with the compare match of PWM counter and PWM cycle register.

Bit	7	6	5	4	3	2	1	0
Bit Name	BTC2G	BTC2E	BTC2C	BTC2A	BTC1G	BTC1E	BTC1C	BTC1A
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	BTC2G	0	R/W	0: Data transfer from PWBFR_n to PWDTR_n is enabled with PWCNT_n and PWCYR_n compare match 1: Data transfer from PWBFR_n to PWDTR_n is disabled with PWCNT_n and PWCYR_n compare match
6	BTC2E	0	R/W	
5	BTC2C	0	R/W	
4	BTC2A	0	R/W	
3	BTC1G	0	R/W	
2	BTC1E	0	R/W	
1	BTC1C	0	R/W	
0	BTC1A	0	R/W	

chanhvinh.com

35.4 Bus Master Interface

35.4.1 16-Bit Data Registers

PWCYR_n and PWBFR_n are 16-bit registers. These registers are linked to the bus master by a 16-bit data bus, and can be read or written in 16-bit units. They cannot be read or written by 8-bit access; 16-bit access must always be used.

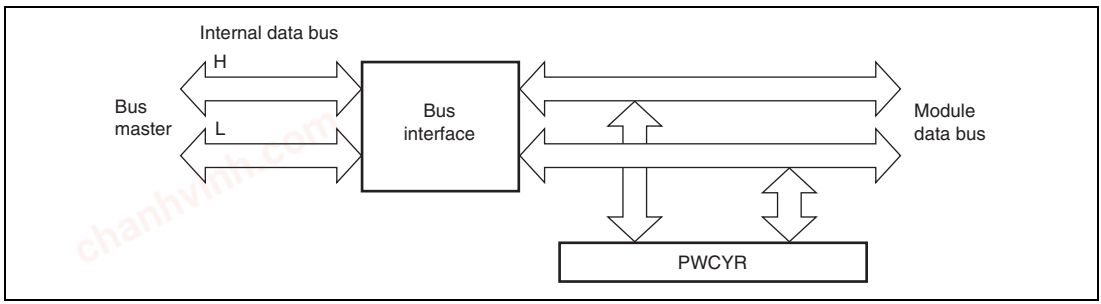


Figure 35.5 16-Bit Register Access Operation (Bus Master ↔ PWCYR_n (16 Bits))

35.4.2 8-Bit Data Registers

PWCR_n, PWPR_n, and PWBTCR are 8-bit registers that can be read and written to in 8-bit units. These registers are linked to the bus master by a 16-bit data bus, and can be read or written by 16-bit access; in this case, the lower eight bits are read as H'FF.

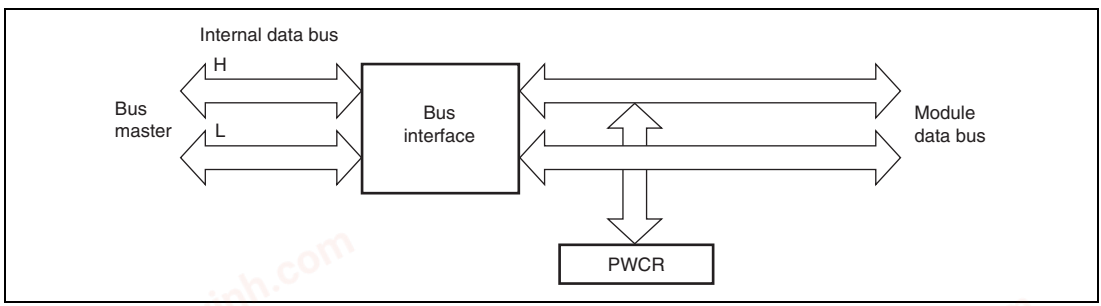


Figure 35.6 8-Bit Register Access Operation (Bus Master ↔ PWCR_n (Upper Eight Bits))

chanhvinh.com

35.5 Operation

35.5.1 PWM Operation

PWM waveforms are output from pins PWM1A to PWM1H and PWM2A to PWM2H as shown in figure 35.7.

(1) Initial Settings

Set the PWM output polarity in PWPR_n; select the clock to be input to PWCNT_n with the CKS2 to CKS0 bits in PWCRn; set the PWM conversion cycle in PWCYR_n; and set the first frame of data in PWBFR_nA, PWBFR_nC, PWBFR_nE, and PWBFR_nG.

(2) Activation

When the CST bit in PWCR_n is set to 1, PWCNT_n starts counting up. On compare match between PWCNT_n and PWCYR_n, data is transferred from the buffer register to the duty register and the CMF bit in PWCR_n is set to 1. At the same time, if the IE bit in PWCR_n has been set to 1, an interrupt can be requested or the direct memory access controller can be activated.

(3) Waveform Output

The PWM outputs selected by the OTS bits in PWDTR_nA, PWDTR_nC, PWDTR_nE, and PWDTR_nG go high when a compare match occurs between PWCNT_n and PWCYR_n. The PWM outputs not selected by the OTS bit are low. When a compare match occurs between PWCNT_n and PWDTR_nA, PWDTR_nC, PWDTR_nE, or PWDTR_nG, the corresponding PWM output goes low. If the corresponding bit in PWPR_n is set to 1, the output is inverted.

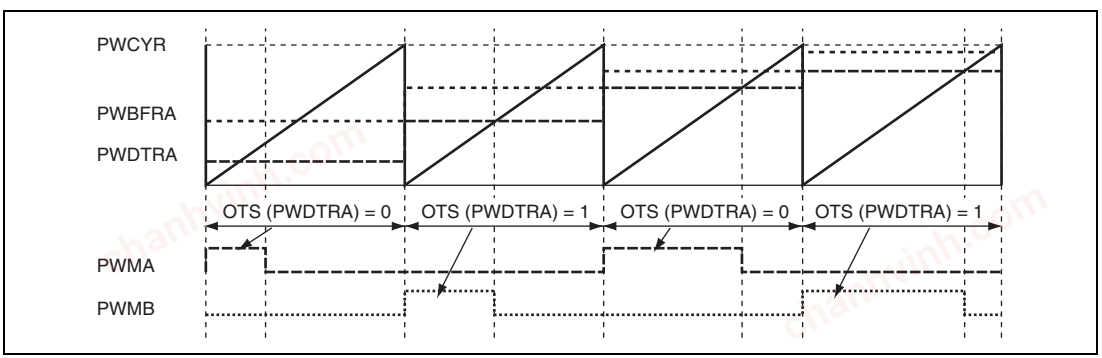


Figure 35.7 PWM Operation

chanhvinh.com

(4) Next Frame

When a compare match occurs between PWCNT_n and PWCYR_n, data is transferred from the buffer register to the duty register. PWCNT_n is reset and starts counting up from H'000. The CMF bit in PWCR_n is set, and if the IE bit in PWCR_n has been set, an interrupt can be requested or the direct memory access controller can be activated.

(5) Stopping

When the CST bit in PWCR_n is cleared to 0, PWCNT_n is reset and stops. All PWM outputs go low (or high if the corresponding bit in PWPR_n is set to 1).

35.5.2 Buffer Transfer Control

Setting a corresponding bit in the PWM buffer transfer control register disables a buffer transfer on compare match. This prevents the output from changing when compare match occurs while the buffer register is being changed. A buffer transfer on compare match is resumed after cleaning the bit.

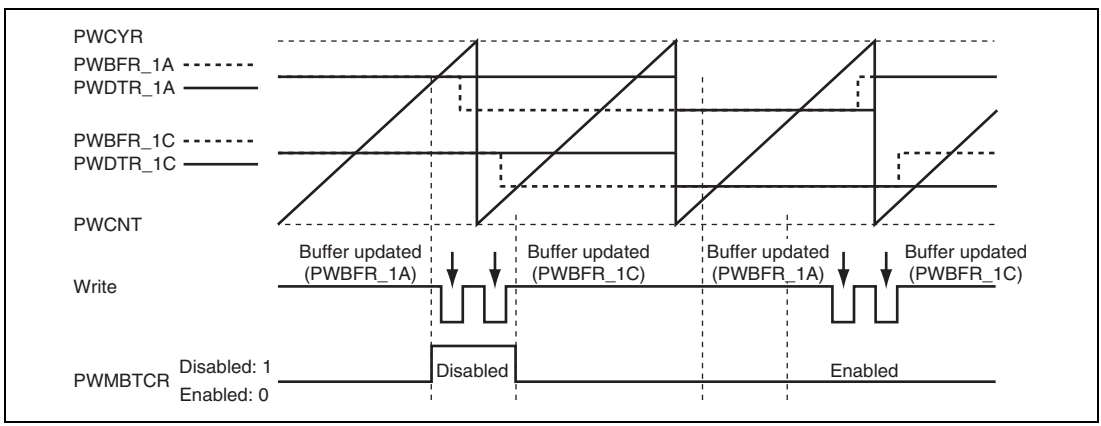


Figure 35.8 Disabling Buffer Transfer

chanhvinh.com

35.6 Usage Note

35.6.1 Conflict between Buffer Register Write and Compare Match

If a PWBFR_n write is performed in the state immediately after a cycle register compare match, the buffer register and duty register are both modified. PWM output changed by the cycle register compare match is not changed by modification of the duty register due to conflict. This may result in unanticipated duty output.

Buffer register modification must be completed before automatic transfer by the direct memory access controller, exception handling due to a compare match interrupt, or the occurrence of a cycle register compare match on detection of the rise of CMF (compare match flag) in PWCR_n.

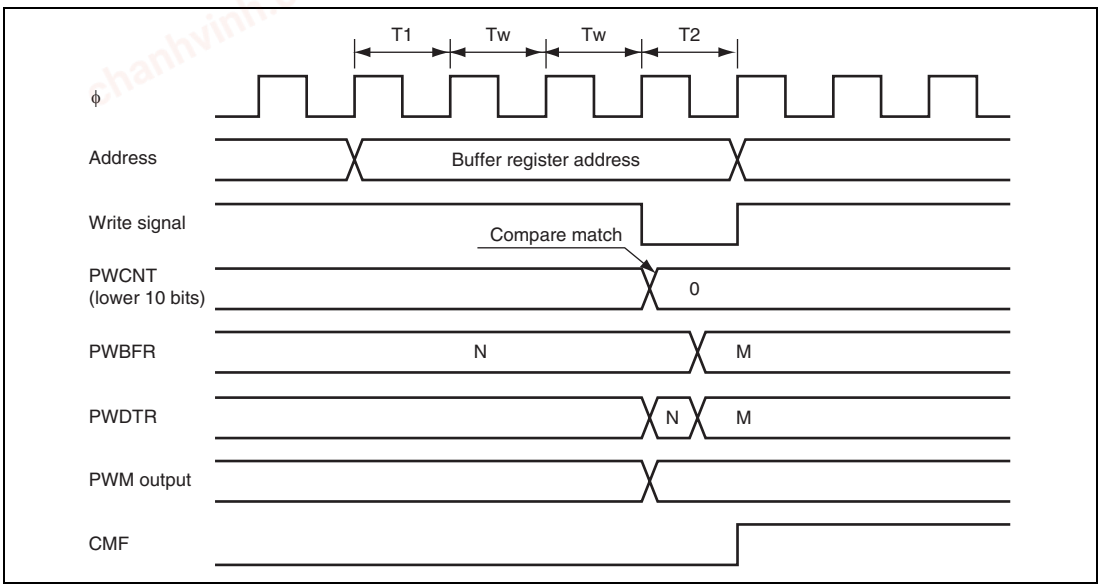


Figure 35.9 Conflict between Buffer Register Write and Compare Match

chanhvinh.com

Section 36 List of Registers

This section gives information on the on-chip I/O registers of this LSI in the following structures.

1. Register Addresses (by functional module, in order of the corresponding section numbers)

- Registers are described by functional module, in order of the corresponding section numbers.
- Access to reserved addresses which are not described in this register address list is prohibited.
- When registers consist of 16 or 32 bits, the addresses of the MSBs are given when big endian mode is selected.

An asterisk (*) in the column "Access Size" indicates that the unit of access in reading differs from that in writing for the given register. For details, see the register descriptions in the relevant section.

2. Register Bits

- Bit configurations of the registers are described in the same order as the Register Addresses (by functional module, in order of the corresponding section numbers).
- Reserved bits are indicated by "—" in the bit name.
- No entry in the bit-name column indicates that the whole register is allocated as a counter or for holding data.

3. Register States in Each Operating Mode

- Register states are described in the same order as the Register Addresses (by functional module, in order of the corresponding section numbers).
- For the initial state of each bit, refer to the description of the register in the corresponding section.
- The register states described are for the basic operating modes. If there is a specific reset for an on-chip peripheral module, refer to the section on that on-chip peripheral module.

4. Notes when Writing to the On-Chip Peripheral Modules

- To access an on-chip module register, two or more peripheral module clock (Pϕ) 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 SLEEP instruction must be performed after setting the STBY bit in the STBCR register to 1. However a dummy read of the STBCR register is required before executing the SLEEP instruction. If a dummy read is omitted, the CPU executes the SLEEP instruction before the STBY bit is set to 1, thus the system enters sleep mode not software standby mode. A dummy read of the STBCR 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

36.1 Register Addresses (by functional module, in order of the corresponding section numbers)

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
Clock pulse generator	Frequency control register	FRQCR	16	H'FFFE0010	16
Interrupt control register	Interrupt control register 0	ICR0	16	H'FFFE0800	16, 32
	Interrupt control register 1	ICR1	16	H'FFFE0802	16, 32
	Interrupt control register 2	ICR2	16	H'FFFE0804	16, 32
	IRQ interrupt request register	IRQRR	16	H'FFFE0806	16, 32
	PINT interrupt enable register	PINTER	16	H'FFFE0808	16, 32
	PINT interrupt request register	PIRR	16	H'FFFE080A	16, 32
	Bank control register	IBCR	16	H'FFFE080C	16, 32
	Bank number register	IBNR	16	H'FFFE080E	16, 32
	Interrupt priority register 01	IPR01	16	H'FFFE0818	16, 32
	Interrupt priority register 02	IPR02	16	H'FFFE081A	16, 32
	Interrupt priority register 05	IPR05	16	H'FFFE0820	16, 32
	Interrupt priority register 06	IPR06	16	H'FFFE0C00	16, 32
	Interrupt priority register 07	IPR07	16	H'FFFE0C02	16, 32
	Interrupt priority register 08	IPR08	16	H'FFFE0C04	16, 32
	Interrupt priority register 09	IPR09	16	H'FFFE0C06	16, 32
	Interrupt priority register 10	IPR10	16	H'FFFE0C08	16, 32
	Interrupt priority register 11	IPR11	16	H'FFFE0C0A	16, 32
	Interrupt priority register 12	IPR12	16	H'FFFE0C0C	16, 32
	Interrupt priority register 13	IPR13	16	H'FFFE0C0E	16, 32
	Interrupt priority register 14	IPR14	16	H'FFFE0C10	16, 32
	Interrupt priority register 15	IPR15	16	H'FFFE0C12	16, 32
	Interrupt priority register 16	IPR16	16	H'FFFE0C14	16, 32
	Interrupt priority register 17	IPR17	16	H'FFFE0C16	16, 32
	Interrupt priority register 18	IPR18	16	H'FFFE0C18	16, 32
	Interrupt priority register 19	IPR19	16	H'FFFE0C1A	16, 32
	Interrupt priority register 20	IPR20	16	H'FFFE0C1C	16, 32
	Interrupt priority register 21	IPR21	16	H'FFFE0C1E	16, 32
	Interrupt priority register 22	IPR22	16	H'FFFE0C20	16, 32

chanhvinh.com

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
Cache	Cache control register 1	CCR1	32	H'FFFC1000	32
	Cache control register 2	CCR2	32	H'FFFC1004	32
Bus state controller	Common control register	CMNCR	32	H'FFFC0000	32
	CS0 space bus control register	CS0BCR	32	H'FFFC0004	32
	CS1 space bus control register	CS1BCR	32	H'FFFC0008	32
	CS2 space bus control register	CS2BCR	32	H'FFFC000C	32
	CS3 space bus control register	CS3BCR	32	H'FFFC0010	32
	CS4 space bus control register	CS4BCR	32	H'FFFC0014	32
	CS5 space bus control register	CS5BCR	32	H'FFFC0018	32
	CS6 space bus control register	CS6BCR	32	H'FFFC001C	32
	CS0 space wait control register	CS0WCR	32	H'FFFC0028	32
	CS1 space wait control register	CS1WCR	32	H'FFFC002C	32
	CS2 space wait control register	CS2WCR	32	H'FFFC0030	32
	CS3 space wait control register	CS3WCR	32	H'FFFC0034	32
	CS4 space wait control register	CS4WCR	32	H'FFFC0038	32
	CS5 space wait control register	CS5WCR	32	H'FFFC003C	32
	CS6 space wait control register	CS6WCR	32	H'FFFC0040	32
	SDRAM control register	SDCR	32	H'FFFC004C	32
	Refresh timer control/status register	RTCSR	16	H'FFFC0050	32
	Refresh timer counter	RTCNT	16	H'FFFC0054	32
	Refresh time constant register	RTCOR	16	H'FFFC0058	32
	AC characteristics switching register	ACSWR	32	H'FFFC180C	32
	AC characteristics switching key register	ACKEYR	8	H'FFFC1BFC	8
Direct memory access controller	DMA source address register_0	SAR_0	32	H'FFFE1000	16, 32
	DMA destination address register_0	DAR_0	32	H'FFFE1004	16, 32
	DMA transfer count register_0	DMATCR_0	32	H'FFFE1008	16, 32
	DMA channel control register_0	CHCR_0	32	H'FFFE100C	8, 16, 32
	DMA reload source address register_0	RSAR_0	32	H'FFFE1100	16, 32
	DMA reload destination address register_0	RDAR_0	32	H'FFFE1104	16, 32
	DMA reload transfer count register_0	RDMATCR_0	32	H'FFFE1108	16, 32
	DMA source address register_1	SAR_1	32	H'FFFE1010	16, 32
	DMA destination address register_1	DAR_1	32	H'FFFE1014	16, 32

chanhvinh.com

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
Direct memory access controller	DMA transfer count register_1	DMATCR_1	32	H'FFFE1018	16, 32
	DMA channel control register_1	CHCR_1	32	H'FFFE101C	8, 16, 32
	DMA reload source address register_1	RSAR_1	32	H'FFFE1110	16, 32
	DMA reload destination address register_1	RDAR_1	32	H'FFFE1114	16, 32
	DMA reload transfer count register_1	RDMATCR_1	32	H'FFFE1118	16, 32
	DMA source address register_2	SAR_2	32	H'FFFE1020	16
	DMA destination address register_2	DAR_2	32	H'FFFE1024	16
	DMA transfer count register_2	DMATCR_2	32	H'FFFE1028	16
	DMA channel control register_2	CHCR_2	32	H'FFFE102C	16
	DMA reload source address register_2	RSAR_2	32	H'FFFE1120	16
	DMA reload destination address register_2	RDAR_2	32	H'FFFE1124	16
	DMA reload transfer count register_2	RDMATCR_2	32	H'FFFE1128	16
	DMA source address register_3	SAR_3	32	H'FFFE1030	16, 32
	DMA destination address register_3	DAR_3	32	H'FFFE1034	16, 32
	DMA transfer count register_3	DMATCR_3	32	H'FFFE1038	16, 32
	DMA channel control register_3	CHCR_3	32	H'FFFE103C	8, 16, 32
	DMA reload source address register_3	RSAR_3	32	H'FFFE1130	16, 32
	DMA reload destination address register_3	RDAR_3	32	H'FFFE1134	16, 32
	DMA reload transfer count register_3	RDMATCR_3	32	H'FFFE1138	16, 32
	DMA source address register_4	SAR_4	32	H'FFFE1040	16, 32
	DMA destination address register_4	DAR_4	32	H'FFFE1044	16, 32
	DMA transfer count register_4	DMATCR_4	32	H'FFFE1048	16, 32
	DMA channel control register_4	CHCR_4	32	H'FFFE104C	8, 16, 32
	DMA reload source address register_4	RSAR_4	32	H'FFFE1140	16, 32
	DMA reload destination address register_4	RDAR_4	32	H'FFFE1144	16, 32
	DMA reload transfer count register_4	RDMATCR_4	32	H'FFFE1148	16, 32
	DMA source address register_5	SAR_5	32	H'FFFE1050	16, 32
	DMA destination address register_5	DAR_5	32	H'FFFE1054	16, 32
	DMA transfer count register_5	DMATCR_5	32	H'FFFE1058	16, 32
	DMA channel control register_5	CHCR_5	32	H'FFFE105C	8, 16, 32
	DMA reload source address register_5	RSAR_5	32	H'FFFE1150	16, 32
	DMA reload destination address register_5	RDAR_5	32	H'FFFE1154	16, 32

chanhvinh.com

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
Direct memory access controller	DMA reload transfer count register_5	RDMATCR_5	32	H'FFFE1158	16, 32
	DMA source address register_6	SAR_6	32	H'FFFE1060	16, 32
	DMA destination address register_6	DAR_6	32	H'FFFE1064	16, 32
	DMA transfer count register_6	DMATCR_6	32	H'FFFE1068	16, 32
	DMA channel control register_6	CHCR_6	32	H'FFFE106C	8, 16, 32
	DMA reload source address register_6	RSAR_6	32	H'FFFE1160	16, 32
	DMA reload destination address register_6	RDAR_6	32	H'FFFE1164	16, 32
	DMA reload transfer count register_6	RDMATCR_6	32	H'FFFE1168	16, 32
	DMA source address register_7	SAR_7	32	H'FFFE1070	16, 32
	DMA destination address register_7	DAR_7	32	H'FFFE1074	16, 32
	DMA transfer count register_7	DMATCR_7	32	H'FFFE1078	16, 32
	DMA channel control register_7	CHCR_7	32	H'FFFE107C	8, 16, 32
	DMA reload source address register_7	RSAR_7	32	H'FFFE1170	16, 32
	DMA reload destination address register_7	RDAR_7	32	H'FFFE1174	16, 32
	DMA reload transfer count register_7	RDMATCR_7	32	H'FFFE1178	16, 32
	DMA source address register_8	SAR_8	32	H'FFFE1080	16, 32
	DMA destination address register_8	DAR_8	32	H'FFFE1084	16, 32
	DMA transfer count register_8	DMATCR_8	32	H'FFFE1088	16, 32
	DMA channel control register_8	RSAR_8	32	H'FFFE1180	16, 32
	DMA reload source address register_8	RDAR_8	32	H'FFFE1184	16, 32
	DMA reload destination address register_8	RDMATCR_8	32	H'FFFE1188	16, 32
	DMA reload transfer count register_8	CHCR_8	32	H'FFFE108C	8, 16, 32
	DMA source address register_9	SAR_9	32	H'FFFE1090	16, 32
	DMA destination address register_9	DAR_9	32	H'FFFE1094	16, 32
	DMA transfer count register_9	DMATCR_9	32	H'FFFE1098	16, 32
	DMA channel control register_9	CHCR_9	32	H'FFFE109C	8, 16, 32
	DMA reload source address register_9	RSAR_9	32	H'FFFE1190	16, 32
	DMA reload destination address register_9	RDAR_9	32	H'FFFE1194	16, 32
	DMA reload transfer count register_9	RDMATCR_9	32	H'FFFE1198	16, 32
	DMA source address register_10	SAR_10	32	H'FFFE10A0	16, 32
	DMA destination address register_10	DAR_10	32	H'FFFE10A4	16, 32
	DMA transfer count register_10	DMATCR_10	32	H'FFFE10A8	16, 32

chanhvinh.com

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
Direct memory access controller	DMA channel control register_10	CHCR_10	32	H'FFFE10AC	8, 16, 32
	DMA reload source address register_10	RSAR_10	32	H'FFFE11A0	16, 32
	DMA reload destination address register_10	RDAR_10	32	H'FFFE11A4	16, 32
	DMA reload transfer count register_10	RDMATCR_10	32	H'FFFE11A8	16, 32
	DMA source address register_11	SAR_11	32	H'FFFE10B0	16, 32
	DMA destination address register_11	DAR_11	32	H'FFFE10B4	16, 32
	DMA transfer count register_11	DMATCR_11	32	H'FFFE10B8	16, 32
	DMA channel control register_11	CHCR_11	32	H'FFFE10BC	8 ,16, 32
	DMA reload source address register_11	RSAR_11	32	H'FFFE11B0	16, 32
	DMA reload destination address register_11	RDAR_11	32	H'FFFE11B4	16, 32
	DMA reload transfer count register_11	RDMATCR_11	32	H'FFFE11B8	16, 32
	DMA source address register_12	SAR_12	32	H'FFFE10C0	16, 32
	DMA destination address register_12	DAR_12	32	H'FFFE10C4	16, 32
	DMA transfer count register_12	DMATCR_12	32	H'FFFE10C8	16, 32
	DMA channel control register_12	CHCR_12	32	H'FFFE10CC	8, 16, 32
	DMA reload source address register_12	RSAR_12	32	H'FFFE11C0	16, 32
	DMA reload destination address register_12	RDAR_12	32	H'FFFE11C4	16, 32
	DMA reload transfer count register_12	RDMATCR_12	32	H'FFFE11C8	16, 32
	DMA source address register_13	SAR_13	32	H'FFFE10D0	16, 32
	DMA destination address register_13	DAR_13	32	H'FFFE10D4	16, 32
	DMA transfer count register_13	DMATCR_13	32	H'FFFE10D8	16, 32
	DMA channel control register_13	CHCR_13	32	H'FFFE10DC	8, 16, 32
	DMA reload source address register_13	RSAR_13	32	H'FFFE11D0	16, 32
	DMA reload destination address register_13	RDAR_13	32	H'FFFE11D4	16, 32
	DMA reload transfer count register_13	RDMATCR_13	32	H'FFFE11D8	16, 32
	DMA source address register_14	SAR_14	32	H'FFFE10E0	16, 32
	DMA destination address register_14	DAR_14	32	H'FFFE10E4	16, 32
	DMA transfer count register_14	DMATCR_14	32	H'FFFE10E8	16, 32
	DMA channel control register_14	CHCR_14	32	H'FFFE10EC	8, 16, 32

chanhvinh.com

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
Direct memory access controller	DMA reload source address register_14	RSAR_14	32	H'FFFE11E0	16, 32
	DMA reload destination address register_14	RDAR_14	32	H'FFFE11E4	16, 32
	DMA reload transfer count register_14	RDMATCR_14	32	H'FFFE11E8	16, 32
	DMA source address register_15	SAR_15	32	H'FFFE10F0	16, 32
	DMA destination address register_15	DAR_15	32	H'FFFE10F4	16, 32
	DMA transfer count register_15	DMATCR_15	32	H'FFFE10F8	16, 32
	DMA channel control register_15	CHCR_15	32	H'FFFE10FC	8, 16, 32
	DMA reload source address register_15	RSAR_15	32	H'FFFE11F0	16, 32
	DMA reload destination address register_15	RDAR_15	32	H'FFFE11F4	16, 32
	DMA reload transfer count register_15	RDMATCR_15	32	H'FFFE11F8	16, 32
	DMA operation register	DMAOR	16	H'FFFE1200	8, 16
	DMA extension resource selector 0	DMARS0	16	H'FFFE1300	16
	DMA extension resource selector 1	DMARS1	16	H'FFFE1304	16
	DMA extension resource selector 2	DMARS2	16	H'FFFE1308	16
	DMA extension resource selector 3	DMARS3	16	H'FFFE130C	16
	DMA extension resource selector 4	DMARS4	16	H'FFFE1310	16
	DMA extension resource selector 5	DMARS5	16	H'FFFE1314	16
	DMA extension resource selector 6	DMARS6	16	H'FFFE1318	16
	DMA extension resource selector 7	DMARS7	16	H'FFFE131C	16
Multi-function timer pulse unit 2	Timer control register_0	TCR_0	8	H'FFFE4300	8
	Timer mode register_0	TMDR_0	8	H'FFFE4301	8
	Timer I/O control register H_0	TIORH_0	8	H'FFFE4302	8
	Timer I/O control register L_0	TIORL_0	8	H'FFFE4303	8
	Timer interrupt enable register_0	TIER_0	8	H'FFFE4304	8
	Timer status register_0	TSR_0	8	H'FFFE4305	8
	Timer counter_0	TCNT_0	16	H'FFFE4306	16
	Timer general register A_0	TGRA_0	16	H'FFFE4308	16
	Timer general register B_0	TGRB_0	16	H'FFFE430A	16
	Timer general register C_0	TGRC_0	16	H'FFFE430C	16
	Timer general register D_0	TGRD_0	16	H'FFFE430E	16

chanhvinh.com

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
Multi-function timer pulse unit 2	Timer general register E_0	TGRE_0	16	H'FFFE4320	16
	Timer general register F_0	TGRF_0	16	H'FFFE4322	16
	Timer interrupt enable register 2_0	TIER2_0	8	H'FFFE4324	8
	Timer status register 2_0	TSR2_0	8	H'FFFE4325	8
	Timer buffer operation transfer mode register_0	TBTM_0	8	H'FFFE4326	8
	Timer control register_1	TCR_1	8	H'FFFE4380	8
	Timer mode register_1	TMDR_1	8	H'FFFE4381	8
	Timer I/O control register_1	TIOR_1	8	H'FFFE4382	8
	Timer interrupt enable register_1	TIER_1	8	H'FFFE4384	8
	Timer status register_1	TSR_1	8	H'FFFE4385	8
	Timer counter_1	TCNT_1	16	H'FFFE4386	16
	Timer general register A_1	TGRA_1	16	H'FFFE4388	16
	Timer general register B_1	TGRB_1	16	H'FFFE438A	16
	Timer input capture control register	TICCR	8	H'FFFE4390	8
	Timer control register_2	TCR_2	8	H'FFFE4000	8
	Timer mode register_2	TMDR_2	8	H'FFFE4001	8
	Timer I/O control register_2	TIOR_2	8	H'FFFE4002	8
	Timer interrupt enable register_2	TIER_2	8	H'FFFE4004	8
	Timer status register_2	TSR_2	8	H'FFFE4005	8
	Timer counter_2	TCNT_2	16	H'FFFE4006	16
	Timer general register A_2	TGRA_2	16	H'FFFE4008	16
	Timer general register B_2	TGRB_2	16	H'FFFE400A	16
	Timer control register_3	TCR_3	8	H'FFFE4200	8
	Timer mode register_3	TMDR_3	8	H'FFFE4202	8
	Timer I/O control register H_3	TIORH_3	8	H'FFFE4204	8
	Timer I/O control register L_3	TIORL_3	8	H'FFFE4205	8
	Timer interrupt enable register_3	TIER_3	8	H'FFFE4208	8
	Timer status register_3	TSR_3	8	H'FFFE422C	8
	Timer counter_3	TCNT_3	16	H'FFFE4210	16
	Timer general register A_3	TGRA_3	16	H'FFFE4218	16
	Timer general register B_3	TGRB_3	16	H'FFFE421A	16

chanhvinh.com

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
Multi-function timer pulse unit 2	Timer general register C_3	TGRC_3	16	H'FFFE4224	16
	Timer general register D_3	TGRD_3	16	H'FFFE4226	16
	Timer buffer operation transfer mode register_3	TBTM_3	8	H'FFFE4238	8
	Timer control register_4	TCR_4	8	H'FFFE4201	8
	Timer mode register_4	TMDR_4	8	H'FFFE4203	8
	Timer I/O control register H_4	TIORH_4	8	H'FFFE4206	8
	Timer I/O control register L_4	TIORL_4	8	H'FFFE4207	8
	Timer interrupt enable register_4	TIER_4	8	H'FFFE4209	8
	Timer status register_4	TSR_4	8	H'FFFE422D	8
	Timer counter_4	TCNT_4	16	H'FFFE4212	16
	Timer general register A_4	TGRA_4	16	H'FFFE421C	16
	Timer general register B_4	TGRB_4	16	H'FFFE421E	16
	Timer general register C_4	TGRC_4	16	H'FFFE4228	16
	Timer general register D_4	TGRD_4	16	H'FFFE422A	16
	Timer buffer operation transfer mode register_4	TBTM_4	8	H'FFFE4239	8
	Timer A/D converter start request control register	TADCR	16	H'FFFE4240	16
	Timer A/D converter start request cycle set register A_4	TADCORA_4	16	H'FFFE4244	16
	Timer A/D converter start request cycle set register B_4	TADCORB_4	16	H'FFFE4246	16
	Timer A/D converter start request cycle set buffer register A_4	TADCOBRA_4	16	H'FFFE4248	16
	Timer A/D converter start request cycle set buffer register B_4	TADCOBRB_4	16	H'FFFE424A	16
	Timer start register	TSTR	8	H'FFFE4280	8
	Timer synchronous register	TSYR	8	H'FFFE4281	8
	Timer read/write enable register	TRWER	8	H'FFFE4284	8
	Timer output master enable register	TOER	8	H'FFFE420A	8
	Timer output control register 1	TOCR1	8	H'FFFE420E	8
	Timer output control register 2	TOCR2	8	H'FFFE420F	8
	Timer gate control register	TGCR	8	H'FFFE420D	8

chanhvinh.com

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
Multi-function timer pulse unit 2	Timer cycle data register	TCDR	16	H'FFFE4214	16
	Timer dead time data register	TDDR	16	H'FFFE4216	16
	Timer subcounter	TCNTS	16	H'FFFE4220	16
	Timer cycle buffer register	TCBR	16	H'FFFE4222	16
	Timer interrupt skipping set register	TITCR	8	H'FFFE4230	8
	Timer interrupt skipping counter	TITCNT	8	H'FFFE4231	8
	Timer buffer transfer set register	TBTER	8	H'FFFE4232	8
	Timer dead time enable register	TDER	8	H'FFFE4234	8
	Timer waveform control register	TWCR	8	H'FFFE4260	8
	Timer output level buffer register	TOLBR	8	H'FFFE4236	8
Compare match timer	Compare match timer start register	CMSTR	16	H'FFFEC000	16
	Compare match timer control/status register_0	CMCSR_0	16	H'FFFEC002	16
	Compare match counter_0	CMCNT_0	16	H'FFFEC004	8, 16
	Compare match constant register_0	CMCOR_0	16	H'FFFEC006	8, 16
	Compare match timer control/status register_1	CMCSR_1	16	H'FFFEC008	16
	Compare match counter_1	CMCNT_1	16	H'FFFEC00A	8, 16
	Compare match constant register_1	CMCOR_1	16	H'FFFEC00C	8, 16
Watchdog timer	Watchdog timer control/status register	WTCSR	8	H'FFFE0000	8, 16*
	Watchdog timer counter	WTCNT	8	H'FFFE0002	8, 16*
	Watchdog reset control/status register	WRCSR	8	H'FFFE0004	8, 16
Realtime clock	64-Hz counter	R64CNT	8	H'FFFE6000	8
	Second counter	RSECCNT	8	H'FFFE6002	8
	Minute counter	RdINCNT	8	H'FFFE6004	8
	Hour counter	RHRCNT	8	H'FFFE6006	8
	Day of week counter	RWKCNT	8	H'FFFE6008	8
	Date counter	RDAYCNT	8	H'FFFE600A	8
	Month counter	RMONCNT	8	H'FFFE600C	8
	Year counter	RYRCNT	16	H'FFFE600E	16
	Second alarm register	RSECAR	8	H'FFFE6010	8
	Minute alarm register	RMINAR	8	H'FFFE6012	8

chanhvinh.com

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
Realtime clock	Hour alarm register	RHRAR	8	H'FFFE6014	8
	Day of week alarm register	RWKAR	8	H'FFFE6016	8
	Date alarm register	RDAYAR	8	H'FFFE6018	8
	Month alarm register	RMONAR	8	H'FFFE601A	8
	Year alarm register	RYRAR	16	H'FFFE6020	16
	Control register 1	RCR1	8	H'FFFE601C	8
	Control register 2	RCR2	8	H'FFFE601E	8
	Control register 3	RCR3	8	H'FFFE6024	8
	Control register 5	RCR5	8	H'FFFE6026	8
	Frequency register H	RFRH	16	H'FFFE602A	16
	Frequency register L	RFRL	16	H'FFFE602C	16
Serial communication interface with FIFO	Serial mode register_0	SCSMR_0	16	H'FFFE8000	16
	Bit rate register_0	SCBRR_0	8	H'FFFE8004	8
	Serial control register_0	SCSCR_0	16	H'FFFE8008	16
	Transmit FIFO data register_0	SCFTDR_0	8	H'FFFE800C	8
	Serial status register_0	SCFSR_0	16	H'FFFE8010	16
	Receive FIFO data register_0	SCFRDR_0	8	H'FFFE8014	8
	FIFO control register_0	SCFCR_0	16	H'FFFE8018	16
	FIFO data count set register_0	SCFDR_0	16	H'FFFE801C	16
	Serial port register_0	SCSPTR_0	16	H'FFFE8020	16
	Line status register_0	SCLSR_0	16	H'FFFE8024	16
	Serial extension mode register_0	SCEMR_0	16	H'FFFE8028	16
	Serial mode register_1	SCSMR_1	16	H'FFFE8800	16
	Bit rate register_1	SCBRR_1	8	H'FFFE8804	8
	Serial control register_1	SCSCR_1	16	H'FFFE8808	16
	Transmit FIFO data register_1	SCFTDR_1	8	H'FFFE880C	8
	Serial status register_1	SCFSR_1	16	H'FFFE8810	16
	Receive FIFO data register_1	SCFRDR_1	8	H'FFFE8814	8
	FIFO control register_1	SCFCR_1	16	H'FFFE8818	16
	FIFO data count register_1	SCFDR_1	16	H'FFFE881C	16
	Serial port register_1	SCSPTR_1	16	H'FFFE8820	16
	Line status register_1	SCLSR_1	16	H'FFFE8824	16

chanhvinh.com

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
Serial communication interface with FIFO	Serial extension mode register_1	SCEMR_1	16	H'FFFE8828	16
	Serial mode register_2	SCSMR_2	16	H'FFFE9000	16
	Bit rate register_2	SCBRR_2	8	H'FFFE9004	8
	Serial control register_2	SCSCR_2	16	H'FFFE9008	16
	Transmit FIFO data register_2	SCFTDR_2	8	H'FFFE900C	8
	Serial status register_2	SCFSR_2	16	H'FFFE9010	16
	Receive FIFO data register_2	SCFRDR_2	8	H'FFFE9014	8
	FIFO control register_2	SCFCR_2	16	H'FFFE9018	16
	FIFO data count register_2	SCFDR_2	16	H'FFFE901C	16
	Serial port register_2	SCSPTR_2	16	H'FFFE9020	16
	Line status register_2	SCLSR_2	16	H'FFFE9024	16
	Serial extension mode register_2	SCEMR_2	16	H'FFFE9028	16
	Serial mode register_3	SCSMR_3	16	H'FFFE9800	16
	Bit rate register_3	SCBRR_3	8	H'FFFE9804	8
	Serial control register_3	SCSCR_3	16	H'FFFE9808	16
	Transmit FIFO data register_3	SCFTDR_3	8	H'FFFE980C	8
	Serial status register_3	SCFSR_3	16	H'FFFE9810	16
	Receive FIFO data register_3	SCFRDR_3	8	H'FFFE9814	8
	FIFO control register_3	SCFCR_3	16	H'FFFE9818	16
	FIFO data count register_3	SCFDR_3	16	H'FFFE981C	16
	Serial port register_3	SCSPTR_3	16	H'FFFE9820	16
	Line status register_3	SCLSR_3	16	H'FFFE9824	16
	Serial extension mode register_3	SCEMR_3	16	H'FFFE9828	16
	Serial mode register_4	SCSMR_4	16	H'FFFEA000	16
	Bit rate register_4	SCBRR_4	8	H'FFFEA004	8
	Serial control register_4	SCSCR_4	16	H'FFFEA008	16
	Transmit FIFO data register_4	SCFTDR_4	8	H'FFFEA00C	8
	Serial status register_4	SCFSR_4	16	H'FFFEA010	16
	Receive FIFO data register_4	SCFRDR_4	8	H'FFFEA014	8
	FIFO control register_4	SCFCR_4	16	H'FFFEA018	16
	FIFO data count register_4	SCFDR_4	16	H'FFFEA01C	16

chanhvinh.com

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
Serial communication interface with FIFO	Serial port register_4	SCSPTR_4	16	H'FFFEA020	16
	Line status register_4	SCLSR_4	16	H'FFFEA024	16
	Serial extension mode register_4	SCEMR_4	16	H'FFFEA028	16
	Serial mode register_5	SCSMR_5	16	H'FFFEA800	16
	Bit rate register_5	SCBRR_5	8	H'FFFEA804	8
	Serial control register_5	SCSCR_5	16	H'FFFEA808	16
	Transmit FIFO data register_5	SCFTDR_5	8	H'FFFEA80C	8
	Serial status register_5	SCFSR_5	16	H'FFFEA810	16
	Receive FIFO data register_5	SCFRDR_5	8	H'FFFEA814	8
	FIFO control register_5	SCFCR_5	16	H'FFFEA818	16
	FIFO data count register_5	SCFDR_5	16	H'FFFEA81C	16
	Serial port register_5	SCSPTR_5	16	H'FFFEA820	16
	Line status register_5	SCLSR_5	16	H'FFFEA824	16
	Serial extension mode register_5	SCEMR_5	16	H'FFFEA828	16
	Serial mode register_6	SCSMR_6	16	H'FFFEB000	16
	Bit rate register_6	SCBRR_6	8	H'FFFEB004	8
	Serial control register_6	SCSCR_6	16	H'FFFEB008	16
	Transmit FIFO data register_6	SCFTDR_6	8	H'FFFEB00C	8
	Serial status register_6	SCFSR_6	16	H'FFFEB010	16
	Receive FIFO data register_6	SCFRDR_6	8	H'FFFEB014	8
	FIFO control register_6	SCFCR_6	16	H'FFFEB018	16
	FIFO data count register_6	SCFDR_6	16	H'FFFEB01C	16
	Serial port register_6	SCSPTR_6	16	H'FFFEB020	16
	Line status register_6	SCLSR_6	16	H'FFFEB024	16
	Serial extension mode register_6	SCEMR_6	16	H'FFFEB028	16
	Serial mode register_7	SCSMR_7	16	H'FFFEB800	16
	Bit rate register_7	SCBRR_7	8	H'FFFEB804	8
	Serial control register_7	SCSCR_7	16	H'FFFEB808	16
	Transmit FIFO data register_7	SCFTDR_7	8	H'FFFEB80C	8
	Serial status register_7	SCFSR_7	16	H'FFFEB810	16
	Receive FIFO data register_7	SCFRDR_7	8	H'FFFEB814	8

chanhvinh.com

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
Serial communication interface with FIFO	FIFO control register_7	SCFCR_7	16	H'FFFEB818	16
	FIFO data count register_7	SCFDR_7	16	H'FFFEB81C	16
	Serial port register_7	SCSPTR_7	16	H'FFFEB820	16
	Line status register_7	SCLSR_7	16	H'FFFEB824	16
	Serial extension mode register_7	SCEMR_7	16	H'FFFEB828	16
Renesas serial peripheral interface	Control register_0	SPCR_0	8	H'FFFF8000	8, 16
	Slave select polarity register_0	SSLP_0	8	H'FFFF8001	8, 16
	Pin control register_0	SPPCR_0	8	H'FFFF8002	8, 16
	Status register_0	SPSR_0	8	H'FFFF8003	8, 16
	Data register_0	SPDR_0	32	H'FFFF8004	8, 16, 32
	Sequence control register_0	SPSCR_0	8	H'FFFF8008	8, 16
	Sequence status register_0	SPSSR_0	8	H'FFFF8009	8, 16
	Bit rate register_0	SPBR_0	8	H'FFFF800A	8, 16
	Data control register_0	SPDCR_0	8	H'FFFF800B	8, 16
	Clock delay register_0	SPCKD_0	8	H'FFFF800C	8, 16
	Slave select negation delay register_0	SSLND_0	8	H'FFFF800D	8, 16
	Next-access delay register_0	SPND_0	8	H'FFFF800E	8
	Command register_00	SPCMD_00	16	H'FFFF8010	16
	Command register_01	SPCMD_01	16	H'FFFF8012	16
	Command register_02	SPCMD_02	16	H'FFFF8014	16
	Command register_03	SPCMD_03	16	H'FFFF8016	16
	Buffer control register_0	SPBFCR_0	8	H'FFFF8020	8, 16
	Buffer data count setting register_0	SPBFDR_0	16	H'FFFF8022	8, 16
	Control register_1	SPCR_1	8	H'FFFF8800	8, 16
	Slave select polarity register_1	SSLP_1	8	H'FFFF8801	8, 16
	Pin control register_1	SPPCR_1	8	H'FFFF8802	8, 16
	Status register_1	SPSR_1	8	H'FFFF8803	8, 16
	Data register_1	SPDR_1	32	H'FFFF8804	8, 16, 32
	Sequence control register_1	SPSCR_1	8	H'FFFF8808	8, 16
	Sequence status register_1	SPSSR_1	8	H'FFFF8809	8, 16
	Bit rate register_1	SPBR_1	8	H'FFFF880A	8, 16

chanhvinh.com

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
Renesas serial peripheral interface	Data control register_1	SPDCR_1	8	H'FFFF880B	8, 16
	Clock delay register_1	SPCKD_1	8	H'FFFF880C	8, 16
	Slave select negation delay register_1	SSLND_1	8	H'FFFF880D	8, 16
	Next-access delay register_1	SPND_1	8	H'FFFF880E	8
	Command register_10	SPCMD_10	16	H'FFFF8810	16
	Command register_11	SPCMD_11	16	H'FFFF8812	16
	Command register_12	SPCMD_12	16	H'FFFF8814	16
	Command register_13	SPCMD_13	16	H'FFFF8816	16
	Buffer control register_1	SPBFCR_1	8	H'FFFF8820	8, 16
	Buffer data count setting register_1	SPBFDR_1	16	H'FFFF8822	16
I^2C bus interface 3	I^2C bus control register 1_0	ICCR1_0	8	H'FFFEE000	8
	I^2C bus control register 2_0	ICCR2_0	8	H'FFFEE001	8
	I^2C bus mode register_0	ICMR_0	8	H'FFFEE002	8
	I^2C bus interrupt enable register_0	ICIER_0	8	H'FFFEE003	8
	I^2C bus status register_0	ICSR_0	8	H'FFFEE004	8
	Slave address register_0	SAR_0	8	H'FFFEE005	8
	I^2C bus transmit data register_0	ICDRT_0	8	H'FFFEE006	8
	I^2C bus receive data register_0	ICDRR_0	8	H'FFFEE007	8
	NF2CYC register_0	NF2CYC_0	8	H'FFFEE008	8
	I^2C bus control register 1_1	ICCR1_1	8	H'FFFEE400	8
	I^2C bus control register 2_1	ICCR2_1	8	H'FFFEE401	8
	I^2C bus mode register_1	ICMR_1	8	H'FFFEE402	8
	I^2C bus interrupt enable register_1	ICIER_1	8	H'FFFEE403	8
	I^2C bus status register_1	ICSR_1	8	H'FFFEE404	8
	Slave address register_1	SAR_1	8	H'FFFEE405	8
	I^2C bus transmit data register_1	ICDRT_1	8	H'FFFEE406	8
	I^2C bus receive data register_1	ICDRR_1	8	H'FFFEE407	8
	NF2CYC register_1	NF2CYC_1	8	H'FFFEE408	8
	I^2C bus control register 1_2	ICCR1_2	8	H'FFFEE800	8
	I^2C bus control register 2_2	ICCR2_2	8	H'FFFEE801	8
	I^2C bus mode register_2	ICMR_2	8	H'FFFEE802	8

chanhvinh.com

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
I²C bus interface 3	I²C bus interrupt enable register_2	ICIER_2	8	H'FFFEE803	8
	I²C bus status register_2	ICSR_2	8	H'FFFEE804	8
	Slave address register_2	SAR_2	8	H'FFFEE805	8
	I²C bus transmit data register_2	ICDRT_2	8	H'FFFEE806	8
	I²C bus receive data register_2	ICDRR_2	8	H'FFFEE807	8
	NF2CYC register_2	NF2CYC_2	8	H'FFFEE808	8
Serial sound interface	Control register_0	SSICR_0	32	H'FFFF0000	8, 16, 32
	Status register_0	SSISR_0	32	H'FFFF0004	8, 16, 32
	FIFO control register_0	SSIFCR_0	32	H'FFFF0010	8, 16, 32
	FIFO status register_0	SSIFSR_0	32	H'FFFF0014	8, 16, 32
	Transmit FIFO data register 0	SSIFTDR_0	32	H'FFFF0018	32
	Receive FIFO data register 0	SSIFRDR_0	32	H'FFFF001C	8, 16, 32
	Control register_1	SSICR_1	32	H'FFFF0800	8, 16, 32
	Status register_1	SSISR_1	32	H'FFFF0804	8, 16, 32
	FIFO control register_1	SSIFCR_1	32	H'FFFF0810	8, 16, 32
	FIFO status register_1	SSIFSR_1	32	H'FFFF0814	8, 16, 32
	Transmit FIFO data register 1	SSIFTDR_1	32	H'FFFF0818	32
	Receive FIFO data register 1	SSIFRDR_1	32	H'FFFF081C	32
	Control register_2	SSICR_2	32	H'FFFF1000	8, 16, 32
	Status register_2	SSISR_2	32	H'FFFF1004	8, 16, 32
	FIFO control register_2	SSIFCR_2	32	H'FFFF1010	8, 16, 32
	FIFO status register_2	SSIFSR_2	32	H'FFFF1014	8, 16, 32
	Transmit FIFO data register 2	SSIFTDR_2	32	H'FFFF1018	32
	Receive FIFO data register 2	SSIFRDR_2	32	H'FFFF101C	32
	Control register_3	SSICR_3	32	H'FFFF1800	8, 16, 32
	Status register_3	SSISR_3	32	H'FFFF1804	8, 16, 32
	FIFO control register_3	SSIFCR_3	32	H'FFFF1810	8, 16, 32
	FIFO status register_3	SSIFSR_3	32	H'FFFF1814	8, 16, 32
	Transmit FIFO data register 3	SSIFTDR_3	32	H'FFFF1818	32
	Receive FIFO data register 3	SSIFRDR_3	32	H'FFFF181C	32

chanhvinh.com

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
Serial I/O with FIFO	Mode register	SIMDR	16	H'FFFF4800	16
	Clock select register	SISCR	16	H'FFFF4802	16
	Transmit data assign register	SITDAR	16	H'FFFF4804	16
	Receive data assign register	SIRDAR	16	H'FFFF4806	16
	Control register	SICTR	16	H'FFFF480C	16
	FIFO control register	SIFCTR	16	H'FFFF4810	16
	Status register	SISTR	16	H'FFFF4814	16
	Interrupt enable register	SIIER	16	H'FFFF4816	16
	Transmit data register	SITDR	32	H'FFFF4820	8, 16, 32
	Receive data register	SIRDR	32	H'FFFF4824	8, 16, 32
Controller area network	Master Control Register_0	MCR_0	16	H'FFFE5000	16
	General Status Register_0	GSR_0	16	H'FFFE5002	16
	Bit Configuration Register 1_0	BCR1_0	16	H'FFFE5004	16
	Bit Configuration Register 0_0	BCR0_0	16	H'FFFE5006	16
	Interrupt Request Register_0	IRR_0	16	H'FFFE5008	16
	Interrupt Mask Register_0	IMR_0	16	H'FFFE500A	16
	Error Counter Register_0	TEC_REC_0	16	H'FFFE500C	8, 16
	Transmit Pending Register 1_0	TXPR1_0	16	H'FFFE5020	32
	Transmit Pending Register 0_0	TXPR0_0	16	H'FFFE5022	16
	Transmit Cancel Register 1_0	TXCR1_0	16	H'FFFE5028	16
	Transmit Cancel Register 0_0	TXCR0_0	16	H'FFFE502A	16
	Transmit Acknowledge Register 1_0	TXACK1_0	16	H'FFFE5030	16
	Transmit Acknowledge Register 0_0	TXACK0_0	16	H'FFFE5032	16
	Abort Acknowledge Register 1_0	ABACK1_0	16	H'FFFE5038	16
	Abort Acknowledge Register 0_0	ABACK0_0	16	H'FFFE503A	16
	Data Frame Receive Pending Register 1_0	RXPR1_0	16	H'FFFE5040	16
	Data Frame Receive Pending Register 0_0	RXPR0_0	16	H'FFFE5042	16
	Remote Frame Receive Pending Register 1_0	RFPR1_0	16	H'FFFE5048	16
	Remote Frame Receive Pending Register 0_0	RFPR0_0	16	H'FFFE504A	16

chanhvinh.com

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
Controller area network	Mailbox Interrupt Mask Register 1_0	MBIMR1_0	16	H'FFFE5050	16
	Mailbox Interrupt Mask Register 0_0	MBIMR0_0	16	H'FFFE5052	16
	Unread Message Status Register 1_0	UMSR1_0	16	H'FFFE5058	16
	Unread Message Status Register 0_0	UMSR0_0	16	H'FFFE505A	16
	Timer Trigger Control Register 0_0	TTCR0_0	16	H'FFFE5080	16
	Cycle Maximum/Tx-Enable Window Register_0	CMAX_TEW_0	16	H'FFFE5084	16
	Reference Trigger Offset Register_0	RFTROFF_0	16	H'FFFE5086	16
	Timer Status Register_0	TSR_0	16	H'FFFE5088	16
	Cycle Counter Register_0	CCR_0	16	H'FFFE508A	16
	Timer Counter Register_0	TCNTR_0	16	H'FFFE508C	16
	Cycle Time Register_0	CYCTR_0	16	H'FFFE5090	16
	Reference Mark Register_0	RFMK_0	16	H'FFFE5094	16
	Timer Compare Match Register 0_0	TCMR0_0	16	H'FFFE5098	16
	Timer Compare Match Register 1_0	TCMR1_0	16	H'FFFE509C	16
	Timer Compare Match Register 2_0	TCMR2_0	16	H'FFFE50A0	16
	Tx-Trigger Time Selection Register_0	TTTSEL_0	16	H'FFFE50A4	16
	Mailbox n Control 0_H_0 (n = 0 to 31)	MBn_ CONTROL0_H_0 (n = 0 to 31)	16	H'FFFE5100 + n × 32	16, 32
	Mailbox n Control 0_L_0 (n = 0 to 31)	MBn_ CONTROL0_L_0 (n = 0 to 31)	16	H'FFFE5102 + n × 32	16
	Mailbox n Local Acceptance Filter Mask 0_0 (n = 0 to 31)	MBn_LAFM0_0 (n = 0 to 31)	16	H'FFFE5104 + n × 32	16, 32
	Mailbox n Local Acceptance Filter Mask 1_0 (n = 0 to 31)	MBn_LAFM1_0 (n = 0 to 31)	16	H'FFFE5106 + n × 32	16
	Mailbox n Data 01_0 (n = 0 to 31)	MBn_ DATA_01_0 (n = 0 to 31)	16	H'FFFE5108 + n × 32	8, 16, 32

chanhvinh.com

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
Controller area network	Mailbox n Data 23_0 (n = 0 to 31)	MBn_ DATA_23_0 (n = 0 to 31)	16	H'FFFE510A + n × 32	8, 16
	Mailbox n Data 45_0 (n = 0 to 31)	MBn_ DATA_45_0 (n = 0 to 31)	16	H'FFFE510C + n × 32	8, 16, 32
	Mailbox n Data 67_0 (n = 0 to 31)	MBn_ DATA_67_0 (n = 0 to 31)	16	H'FFFE510E + n × 32	8, 16
	Mailbox n Control 1_0 (n = 0 to 31)	MBn_ CONTROL1_0 (n = 0 to 31)	16	H'FFFE5110 + n × 32	8, 16
	Mailbox n Time Stamp_0 (n = 0 to 15, 30, 31)	MBn_ TIMESTAMP_0 (n=0 to 15, 30, 31)	16	H'FFFE5112 + n × 32	16
	Mailbox n Trigger Time_0 (n = 24 to 30)	MBn_TTT_0 (n = 24 to 31)	16	H'FFFE5114 + n × 32	16
	Mailbox n TT Control_0 (n = 24 to 29)	MBn_ TTCONTROL_0 (n = 24 to 29)	16	H'FFFE5116 + n × 32	16
	Master Control Register_1	MCR_1	16	H'FFFE5800	16
	General Status Register_1	GSR_1	16	H'FFFE5802	16
	Bit Configuration Register 1_1	BCR1_1	16	H'FFFE5804	16
	Bit Configuration Register 0_1	BCR0_1	16	H'FFFE5806	16
	Interrupt Request Register_1	IRR_1	16	H'FFFE5808	16
	Interrupt Mask Register_1	IMR_1	16	H'FFFE580A	16
	Error Counter Register_1	TEC_REC_1	16	H'FFFE580C	8, 16
	Transmit Pending Register 1_1	TXPR1_1	16	H'FFFE5820	32
	Transmit Pending Register 0_1	TXPR0_1	16	H'FFFE5822	16
	Transmit Cancel Register 1_1	TXCR1_1	16	H'FFFE5828	16
	Transmit Cancel Register 0_1	TXCR0_1	16	H'FFFE582A	16
	Transmit Acknowledge Register 1_1	TXACK1_1	16	H'FFFE5830	16
	Transmit Acknowledge Register 0_1	TXACK0_1	16	H'FFFE5832	16
	Abort Acknowledge Register 1_1	ABACK1_1	16	H'FFFE5838	16
	Abort Acknowledge Register 0_1	ABACK0_1	16	H'FFFE583A	16
	Data Frame Receive Pending Register 1_1	RXPR1_1	16	H'FFFE5840	16

chanhvinh.com

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
Controller area network	Data Frame Receive Pending Register 0_1	RXPR0_1	16	H'FFFE5842	16
	Remote Frame Receive Pending Register 1_1	RFPR1_1	16	H'FFFE5848	16
	Remote Frame Receive Pending Register 0_1	RFPR0_1	16	H'FFFE584A	16
	Mailbox Interrupt Mask Register 1_1	MBIMR1_1	16	H'FFFE5850	16
	Mailbox Interrupt Mask Register 0_1	MBIMR0_1	16	H'FFFE5852	16
	Unread Message Status Register 1_1	UMSR1_1	16	H'FFFE5858	16
	Unread Message Status Register 0_1	UMSR0_1	16	H'FFFE585A	16
	Timer Trigger Control Register 0_1	TTCR0_1	16	H'FFFE5880	16
	Cycle Maximum/Tx-Enable Window Register_1	CMAX_TEW_1	16	H'FFFE5884	16
	Reference Trigger Offset Register_1	RFTROFF_1	16	H'FFFE5886	16
	Timer Status Register_1	TSR_1	16	H'FFFE5888	16
	Cycle Counter Register_1	CCR_1	16	H'FFFE588A	16
	Timer Counter Register_1	TCNTR_1	16	H'FFFE588C	16
	Cycle Time Register_1	CYCTR_1	16	H'FFFE5890	16
	Reference Mark Register_1	RFMK_1	16	H'FFFE5894	16
	Timer Compare Match Register 0_1	TCMR0_1	16	H'FFFE5898	16
	Timer Compare Match Register 1_1	TCMR1_1	16	H'FFFE589C	16
	Timer Compare Match Register 2_1	TCMR2_1	16	H'FFFE58A0	16
	Tx-Trigger Time Selection Register_1	TTTSEL_1	16	H'FFFE58A4	16
	Mailbox n Control 0_H_1 (n = 0 to 31)	MBn_ CONTROL0_H_1 (n = 0 to 31)	16	H'FFFE5900 + n × 32	16, 32
	Mailbox n Control 0_L_1 (n = 0 to 31)	MBn_ CONTROL0_L_1 (n = 0 to 31)	16	H'FFFE5902 + n × 32	16
	Mailbox n Local Acceptance Filter Mask 0_1 (n = 0 to 31)	MBn_LAFM0_1 (n = 0 to 31)	16	H'FFFE5904 + n × 32	16, 32

chanhvinh.com

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
Controller area network	Mailbox n Local Acceptance Filter Mask 1_1 (n = 0 to 31)	MBn_LAFM1_1 (n = 0 to 31)	16	H'FFFE5906 + n × 32	16
	Mailbox n Data 11_1 (n = 0 to 31)	MBn_ DATA_01_1 (n = 0 to 31)	16	H'FFFE5908 + n × 32	8, 16, 32
	Mailbox n Data 23_1 (n = 0 to 31)	MBn_ DATA_23_1 (n = 0 to 31)	16	H'FFFE590A + n × 32	8, 16
	Mailbox n Data 45_1 (n = 0 to 31)	MBn_ DATA_45_1 (n = 0 to 31)	16	H'FFFE590C + n × 32	8, 16, 32
	Mailbox n Data 67_1 (n = 0 to 31)	MBn_ DATA_67_1 (n = 0 to 31)	16	H'FFFE590E + n × 32	8, 16
	Mailbox n Control 1_1 (n = 0 to 31)	MBn_ CONTROL1_1 (n = 0 to 31)	16	H'FFFE5910 + n × 32	8, 16
	Mailbox n Time Stamp_1 (n = 0 to 15, 30, 31)	MBn_ TIMESTAMP_1 (n = 0 to 15, 30, 31)	16	H'FFFE5912 + n × 32	16
	Mailbox n Trigger Time_1 (n = 24 to 30)	MBn_TTT_1 (n = 24 to 30)	16	H'FFFE5914 + n × 32	16
	Mailbox n TT Control_1 (n = 24 to 29)	MBn_ TTCONTROL_1 (n = 24 to 29)	16	H'FFFE5916 + n × 32	16
IEBus controller	IEBus control register	IECTR	8	H'FFFEF000	8
	IEBus command register	IECMR	8	H'FFFEF001	8
	IEBus master control register	IEMCR	8	H'FFFEF002	8
	IEBus master unit address register 1	IEAR1	8	H'FFFEF003	8
	IEBus master unit address register 2	IEAR2	8	H'FFFEF004	8
	IEBus slave address setting register 1	IESA1	8	H'FFFEF005	8
	IEBus slave address setting register 2	IESA2	8	H'FFFEF006	8
	IEBus transmit message length register	IETBFL	8	H'FFFEF007	8
	IEBus reception master address register 1	IEMA1	8	H'FFFEF009	8

chanhvinh.com

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
IEBus controller	IEBus reception master address register 2	IEMA2	8	H'FFFEF00A	8
	IEBus receive control field register	IERCTL	8	H'FFFEF00B	8
	IEBus receive message length register	IERBFL	8	H'FFFEF00C	8
	IEBus lock address register 1	IELA1	8	H'FFFEF00E	8
	IEBus lock address register 2	IELA2	8	H'FFFEF00F	8
	IEBus general flag register	IEFLG	8	H'FFFEF010	8
	IEBus transmit status register	IETSR	8	H'FFFEF011	8
	IEBus transmit interrupt enable register	IEIET	8	H'FFFEF012	8
	IEBus receive status register	IERSR	8	H'FFFEF014	8
	IEBus receive interrupt enable register	IEIER	8	H'FFFEF015	8
	IEBus clock select register	IECKSR	8	H'FFFEF018	8
	IEBus transmit data buffer registers 001 to 128	IETB001 to IETB128	8	H'FFFEF100 to H'FFFEF17F	8
	IEBus receive data buffer registers 001 to 128	IERB001 to IERB128	8	H'FFFEF200 to H'FFFEF27F	8
Renesas SPDIF interface	Transmitter channel 1 audio register	TLCA	32	H'FFFF5800	32
	Transmitter channel 2 audio register	TRCA	32	H'FFFF5804	32
	Transmitter channel 1 status register	TLCS	32	H'FFFF5808	32
	Transmitter channel 2 status register	TRCS	32	H'FFFF580C	32
	Transmitter user data register	TUI	32	H'FFFF5810	32
	Receiver channel 1 audio register	RLCA	32	H'FFFF5814	32
	Receiver channel 2 audio register	RRCA	32	H'FFFF5818	32
	Receiver channel 1 status register	RLCS	32	H'FFFF581C	32
	Receiver channel 2 status register	RRCS	32	H'FFFF5820	32
	Receiver user data register	RUI	32	H'FFFF5824	32
	Control register	CTRL	32	H'FFFF5828	32
	Status register	STAT	32	H'FFFF582C	32
	Transmitter DMA audio data register	TDAD	32	H'FFFF5830	32
	Receiver DMA audio data register	RDAD	32	H'FFFF5834	32

chanhvinh.com

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
CD-ROM decoder	Enable control register	CROMEN	8	H'FFFF9000	8
	Sync code-based synchronization control register	CROMSY0	8	H'FFFF9001	8
	Decoding mode control register	CROMCTL0	8	H'FFFF9002	8
	EDC/ECC check control register	CROMCTL1	8	H'FFFF9003	8
	Automatic decoding stop control register	CROMCTL3	8	H'FFFF9005	8
	Decoding option setting control register	CROMCTL4	8	H'FFFF9006	8
	HEAD20 to HEAD22 representation control register	CROMCTL5	8	H'FFFF9007	8
	Sync code status register	CROMST0	8	H'FFFF9008	8
	Post-ECC header error status register	CROMST1	8	H'FFFF9009	8
	Post-ECC subheader error status register	CROMST3	8	H'FFFF900B	8
	Header/subheader validity check status register	CROMST4	8	H'FFFF900C	8
	Mode determination and link sector detection status register	CROMST5	8	H'FFFF900D	8
	ECC/EDC error status register	CROMST6	8	H'FFFF900E	8
	Buffer status register	CBUFST0	8	H'FFFF9014	8
	Decoding stoppage source status register	CBUFST1	8	H'FFFF9015	8
	Buffer overflow status register	CBUFST2	8	H'FFFF9016	8
	Pre-ECC correction header: minutes data register	HEAD00	8	H'FFFF9018	8
	Pre-ECC correction header: seconds data register	HEAD01	8	H'FFFF9019	8
	Pre-ECC correction header: frames (1/75 second) data register	HEAD02	8	H'FFFF901A	8
	Pre-ECC correction header: mode data register	HEAD03	8	H'FFFF901B	8
	Pre-ECC correction subheader: file number (byte 16) data register	SHEAD00	8	H'FFFF901C	8

chanhvinh.com

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
CD-ROM decoder	Pre-ECC correction subheader: channel number (byte 17) data register	SHEAD01	8	H'FFFF901D	8
	Pre-ECC correction subheader: sub-mode (byte 18) data register	SHEAD02	8	H'FFFF901E	8
	Pre-ECC correction subheader: data type (byte 19) data register	SHEAD03	8	H'FFFF901F	8
	Pre-ECC correction subheader: file number (byte 20) data register	SHEAD04	8	H'FFFF9020	8
	Pre-ECC correction subheader: channel number (byte 21) data register	SHEAD05	8	H'FFFF9021	8
	Pre-ECC correction subheader: sub-mode (byte 22) data register	SHEAD06	8	H'FFFF9022	8
	Pre-ECC correction subheader: data type (byte 23) data register	SHEAD07	8	H'FFFF9023	8
	Post-ECC correction header: minutes data register	HEAD20	8	H'FFFF9024	8
	Post-ECC correction header: seconds data register	HEAD21	8	H'FFFF9025	8
	Post-ECC correction header: frames (1/75 second) data register	HEAD22	8	H'FFFF9026	8
	Post-ECC correction header: mode data register	HEAD23	8	H'FFFF9027	8
	Post-ECC correction subheader: file number (byte 16) data register	SHEAD20	8	H'FFFF9028	8
	Post-ECC correction subheader: channel number (byte 17) data register	SHEAD21	8	H'FFFF9029	8
	Post-ECC correction subheader: sub-mode (byte 18) data register	SHEAD22	8	H'FFFF902A	8
	Post-ECC correction subheader: data type (byte 19) data register	SHEAD23	8	H'FFFF902B	8
	Post-ECC correction subheader: file number (byte 20) data register	SHEAD24	8	H'FFFF902C	8
	Post-ECC correction subheader: channel number (byte 21) data register	SHEAD25	8	H'FFFF902D	8

chanhvinh.com

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
CD-ROM decoder	Post-ECC correction subheader: sub-mode (byte 22) data register	SHEAD26	8	H'FFFF902E	8
	Post-ECC correction subheader: data type (byte 23) data register	SHEAD27	8	H'FFFF902F	8
	Automatic buffering setting control register	CBUFCTL0	8	H'FFFF9040	8
	Automatic buffering start sector setting:minutes control register	CBUFCTL1	8	H'FFFF9041	8
	Automatic buffering start sector setting:seconds control register	CBUFCTL2	8	H'FFFF9042	8
	Automatic buffering start sector setting:frames control register	CBUFCTL3	8	H'FFFF9043	8
	ISY interrupt source mask control register	CROMST0M	8	H'FFFF9045	8
	CD-ROM decoder reset control register	ROMDECRST	8	H'FFFF9100	8
	CD-ROM decoder reset status register	RSTSTAT	8	H'FFFF9101	8
	Serial sound interface data control register	SSI	8	H'FFFF9102	8
	Interrupt flag register	INTHOLD	8	H'FFFF9108	8
	Interrupt source mask control register	INHINT	8	H'FFFF9109	8
	CD-ROM decoder stream data input register	STRMDIN0	16	H'FFFF9200	16, 32*
	CD-ROM decoder stream data input register	STRMDIN2	16	H'FFFF9202	16
	CD-ROM decoder stream data output register	STRMDOUT0	16	H'FFFF9204	16, 32
A/D converter	A/D data register A	ADDRA	16	H'FFFF9800	16
	A/D data register B	ADDRB	16	H'FFFF9802	16
	A/D data register C	ADDRC	16	H'FFFF9804	16
	A/D data register D	ADDRD	16	H'FFFF9806	16
	A/D data register E	ADDRE	16	H'FFFF9808	16
	A/D data register F	ADDRF	16	H'FFFF980A	16
	A/D data register G	ADDRG	16	H'FFFF980C	16
	A/D data register H	ADDRH	16	H'FFFF980E	16
	A/D control/status register	ADCSR	16	H'FFFF9820	16

chanhvinh.com

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
NAND flash memory controller	Common control register	FLCMNCR	32	H'FFFF4000	32
	Command control register	FLCMDCR	32	H'FFFF4004	32
	Command code register	FLCMCDR	32	H'FFFF4008	32
	Address register	FLADR	32	H'FFFF400C	32
	Address register 2	FLADR2	32	H'FFFF403C	32
	Data register	FLDATAR	32	H'FFFF4010	32
	Data counter register	FLDTCNTR	32	H'FFFF4014	32
	Interrupt DMA control register	FLINTDMACR	32	H'FFFF4018	32
	Ready busy timeout setting register	FLBSYTMR	32	H'FFFF401C	32
	Ready busy timeout counter	FLBSYCNT	32	H'FFFF4020	32
	Data FIFO register	FLDTFIFO	32	H'FFFF4050	32
	Control code FIFO register	FLECFIFO	32	H'FFFF4060	32
	Transfer control register	FLTRCR	8	H'FFFF402C	8
	Bus hold time setting register	FLHOLDCR	32	H'FFFF4038	32
	4-symbol ECC processing result register 1	FL4ECCRES1	32	H'FFFF4080	32
	4-symbol ECC processing result register 2	FL4ECCRES2	32	H'FFFF4084	32
	4-symbol ECC processing result register 3	FL4ECCRES3	32	H'FFFF4088	32
	4-symbol ECC processing result register 4	FL4ECCRES4	32	H'FFFF408C	32
	4-symbol ECC control register	FL4ECCCR	32	H'FFFF4090	32
	4-symbol ECC error count register	FL4ECCCNT	32	H'FFFF4094	32
USB 2.0 host/function module	System configuration control register	SYSCFG	16	H'FFFFC000	16
	CPU bus wait setting register	BUSWAIT	16	H'FFFFC002	16
	System configuration status register	SYSSTS	16	H'FFFFC004	16
	Device state control register	DVSTCTR	16	H'FFFFC008	16
	Test mode register	TESTMODE	16	H'FFFFC00C	16
	DMA0-FIFO bus configuration register	D0FBCFG	16	H'FFFFC010	16
	DMA1-FIFO bus configuration register	D1FBCFG	16	H'FFFFC012	16

chanhvinh.com

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
USB 2.0 host/function module	CFIFO port register	CFIFO	32	H'FFFFC014	8, 16, 32
	D0FIFO port register	D0FIFO	32	H'FFFFC018	8, 16, 32
	D1FIFO port register	D1FIFO	32	H'FFFFC01C	8, 16, 32
	CFIFO port select register	CFIFOSEL	16	H'FFFFC020	16
	CFIFO port control register	CFIFOCTR	16	H'FFFFC022	16
	D0FIFO port select register	D0FIFOSEL	16	H'FFFFC028	16
	D0FIFO port control register	D0FIFOCTR	16	H'FFFFC02A	16
	D1FIFO port select register	D1FIFOSEL	16	H'FFFFC02C	16
	D1FIFO port control register	D1FIFOCTR	16	H'FFFFC02E	16
	Interrupt enable register 0	INTENB0	16	H'FFFFC030	16
	Interrupt enable register 1	INTENB1	16	H'FFFFC032	16
	BRDY interrupt enable register	BRDYENB	16	H'FFFFC036	16
	NRDY interrupt enable register	NRDYENB	16	H'FFFFC038	16
	BEMP interrupt enable register	BEMPENB	16	H'FFFFC03A	16
	SOF output configuration register	SOFCFG	16	H'FFFFC03C	16
	Interrupt status register 0	INTSTS0	16	H'FFFFC040	16
	Interrupt status register 1	INTSTS1	16	H'FFFFC042	16
	BRDY interrupt status register	BRDYSTS	16	H'FFFFC046	16
	NRDY interrupt status register	NRDYSTS	16	H'FFFFC048	16
	BEMP interrupt status register	BEMPSTS	16	H'FFFFC04A	16
	Frame number register	FRMNUM	16	H'FFFFC04C	16
	μFrame number register Frame number register Frame number register	UFRMNUM	16	H'FFFFC04E	16
	USB address register	USBADDR	16	H'FFFFC050	16
	USB request type register	USBREQ	16	H'FFFFC054	16
	USB request value register	USBVAL	16	H'FFFFC056	16
	USB request index register	USBINDX	16	H'FFFFC058	16
	USB request length register	USBLENG	16	H'FFFFC05A	16
	DCP configuration register	DCPCFG	16	H'FFFFC05C	16
	DCP maximum packet size register	DCPMAXP	16	H'FFFFC05E	16
	DCP control register	DCPCTR	16	H'FFFFC060	16

chanhvinh.com

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
USB 2.0 host/function module	Pipe window select register	PIPESEL	16	H'FFFFC064	16
	Pipe configuration register	PIPECFG	16	H'FFFFC068	16
	Pipe buffer setting register	PIPEBUF	16	H'FFFFC06A	16
	Pipe maximum packet size register	PIPEMAXP	16	H'FFFFC06C	16
	Pipe cycle control register	PIPEPERI	16	H'FFFFC06E	16
	Pipe 1 control register	PIPE1CTR	16	H'FFFFC070	16
	Pipe 2 control register	PIPE2CTR	16	H'FFFFC072	16
	Pipe 3 control register	PIPE3CTR	16	H'FFFFC074	16
	Pipe 4 control register	PIPE4CTR	16	H'FFFFC076	16
	Pipe 5 control register	PIPE5CTR	16	H'FFFFC078	16
	Pipe 6 control register	PIPE6CTR	16	H'FFFFC07A	16
	Pipe 7 control register	PIPE7CTR	16	H'FFFFC07C	16
	Pipe 8 control register	PIPE8CTR	16	H'FFFFC07E	16
	Pipe 9 control register	PIPE9CTR	16	H'FFFFC080	16
	Pipe 1 transaction counter enable register	PIPE1TRE	16	H'FFFFC090	16
	Pipe 1 transaction counter register	PIPE1TRN	16	H'FFFFC092	16
	Pipe 2 transaction counter enable register	PIPE2TRE	16	H'FFFFC094	16
	Pipe 2 transaction counter register	PIPE2TRN	16	H'FFFFC096	16
	Pipe 3 transaction counter enable register	PIPE3TRE	16	H'FFFFC098	16
	Pipe 3 transaction counter register	PIPE3TRN	16	H'FFFFC09A	16
	Pipe 4 transaction counter enable register	PIPE4TRE	16	H'FFFFC09C	16
	Pipe 4 transaction counter register	PIPE4TRN	16	H'FFFFC09E	16
	Pipe 5 transaction counter enable register	PIPE5TRE	16	H'FFFFC0A0	16
	Pipe 5 transaction counter register	PIPE5TRN	16	H'FFFFC0A2	16
	USB AC characteristics switching register 1	USBACSWR1	16	H'FFFFC0C2	16
	Device address 0 configuration register	DEVADD0	16	H'FFFFC0D0	16

chanhvinh.com

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
USB 2.0 host/function module	Device address 1 configuration register	DEVADD1	16	H'FFFFC0D2	16
	Device address 2 configuration register	DEVADD2	16	H'FFFFC0D4	16
	Device address 3 configuration register	DEVADD3	16	H'FFFFC0D6	16
	Device address 4 configuration register	DEVADD4	16	H'FFFFC0D8	16
	Device address 5 configuration register	DEVADD5	16	H'FFFFC0DA	16
	Device address 6 configuration register	DEVADD6	16	H'FFFFC0DC	16
	Device address 7 configuration register	DEVADD7	16	H'FFFFC0DE	16
	Device address 8 configuration register	DEVADD8	16	H'FFFFC0E0	16
	Device address 9 configuration register	DEVADD9	16	H'FFFFC0E2	16
	Device address A configuration register	DEVADDA	16	H'FFFFC0E4	16
Video display controller 3	Video operating mode register	VIDEO_MODE	32	H'FFFF2000	8, 16, 32
	Video interrupt control register	VIDEO_INT_CNT	32	H'FFFF2004	8, 16, 32
	Video input timing control register	VIDEO_TIM_CNT	32	H'FFFF2008	8, 16, 32
	Valid video size register	VIDEO_SIZE	32	H'FFFF2100	8, 16, 32
	Vertical valid video start position register	VIDEO_VSTART	32	H'FFFF2104	8, 16, 32
	Horizontal valid video start position register	VIDEO_HSTART	32	H'FFFF2108	8, 16, 32
	Timing control register 1 for vertical sync signal for video	VIDEO_VSYNC_TIM1	32	H'FFFF210C	8, 16, 32
	Video storing field count register	VIDEO_SAVE_NUM	32	H'FFFF2110	8, 16, 32
	Video scaling and correction register	VIDEO_IMAGE_CNT	32	H'FFFF2114	8, 16, 32
	Video base address register	VIDEO_BASEADR	32	H'FFFF2118	8, 16, 32

chanhvinh.com

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
Video display controller 3	Video line offset register	VIDEO_LINE_OFFSET	32	H'FFFF211C	8, 16, 32
	Video field offset register	VIDEO_FIELD_OFFSET	32	H'FFFF2120	8, 16, 32
	Video line buffer count register	VIDEO_LINEBUFF_NUM	32	H'FFFF2124	8, 16, 32
	Video display and recording size register	VIDEO_DISP_SIZE	32	H'FFFF2128	8, 16, 32
	Horizontal video display position register	VIDEO_DISP_HSTART	32	H'FFFF212C	8, 16, 32
	Graphics block control register	GRCMEN1	32	H'FFFF2800	8, 16, 32
	Bus control register	GRCBUSCNT1	32	H'FFFF2804	8, 16, 32
	Graphics block interrupt control register	GRCINTCNT1	32	H'FFFF2808	8, 16, 32
	Graphics image base address register	GROPSADR1	32	H'FFFF2B08	8, 16, 32
	Graphics image size register	GROPSWH1	32	H'FFFF2B0C	8, 16, 32
	Graphics image line offset register	GROPSOFST1	32	H'FFFF2B10	8, 16, 32
	Graphics image start position register	GROPDPHV1	32	H'FFFF2B14	8, 16, 32
	α control area size register	GROPEWH1	32	H'FFFF2B18	8, 16, 32
	α control area start position register	GRPEDPHV1	32	H'FFFF2B1C	8, 16, 32
	α control register	GROPEDPA1	32	H'FFFF2B20	8, 16, 32
	Chroma-key control register	GROPCRKY0_1	32	H'FFFF2B24	8, 16, 32
	Chroma-key color register	GROPCRKY1_1	32	H'FFFF2B28	8, 16, 32
	Color register for outside of graphic image area	GROPBASERGB1	32	H'FFFF2B2C	8, 16, 32
	Graphics block control register	GRCMEN2	32	H'FFFF3000	8, 16, 32
	Bus control register	GRCBUSCNT2	32	H'FFFF3004	8, 16, 32
	Graphics block interrupt control register	GRCINTCNT2	32	H'FFFF3008	8, 16, 32
	Graphics image base address register	GROPSADR2	32	H'FFFF3308	8, 16, 32
	Graphics image size register	GROPSWH2	32	H'FFFF330C	8, 16, 32
	Graphics image line offset register	GROPSOFST2	32	H'FFFF3310	8, 16, 32
	Graphics image start position register	GROPDPHV2	32	H'FFFF3314	8, 16, 32

chanhvinh.com

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
Video display controller 3	α control area size register	GROPEWH2	32	H'FFFF3318	8, 16, 32
	α control area start position register	GRPEDPHV2	32	H'FFFF331C	8, 16, 32
	α control register	GROPEDPA2	32	H'FFFF3320	8, 16, 32
	Chroma-key control register	GROPCRKY0_2	32	H'FFFF3324	8, 16, 32
	Chroma-key color register	GROPCRKY1_2	32	H'FFFF3328	8, 16, 32
	Color register for outside of graphic image area	GROPBASERGB2	32	H'FFFF332C	8, 16, 32
	SG mode register	SGMODE	32	H'FFFF3800	8, 16, 32
	Interrupt output control register	SGINTCNT	32	H'FFFF3804	8, 16, 32
	Sync signal control register	SYNCNT	32	H'FFFF3808	8, 16, 32
	Panel clock select register	PANEL_CLKSEL	32	H'FFFF380C	8, 16, 32
	Sync signal size register	SYN_SIZE	32	H'FFFF3900	8, 16, 32
	Timing control register for vertical sync signal for output to panel	PANEL_ VSYNC_TIM	32	H'FFFF3904	8, 16, 32
	Timing control register for horizontal sync signal for output to panel	PANEL_ HSYNC_TIM	32	H'FFFF3908	8, 16, 32
	Timing control register 2 for vertical sync signal for video	VIDEO_ VSYNC_TIM2	32	H'FFFF390C	8, 16, 32
	Timing control register for vertical sync signal for graphic image	GRA_ VSYNC_TIM	32	H'FFFF3910	8, 16, 32
	AC modulation signal toggle line count	AC_LINE_NUM	32	H'FFFF3914	8, 16, 32
	DE area size register	DE_SIZE	32	H'FFFF3920	8, 16, 32
	DE area start position register	DE_START	32	H'FFFF3924	8, 16, 32
Sampling rate converter	Input data register_0	SRCID_0	32	H'FFFE7000	16, 32
	Output data register_0	SRCOD_0	32	H'FFFE7004	16, 32
	Input data control register_0	SRCIDCTRL_0	16	H'FFFE7008	16
	Output data control register_0	SRCODCTRL_0	16	H'FFFE700A	16
	Control register_0	SRCCTRL_0	16	H'FFFE700C	16
	Status register_0	SRCSTAT_0	16	H'FFFE700E	16
	Input data register_1	SRCID_1	16	H'FFFE7800	16, 32
	Output data register_1	SRCOD_1	32	H'FFFE7804	16, 32
	Input data control register_1	SRCIDCTRL_1	16	H'FFFE7808	16

chanhvinh.com

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
Sampling rate converter	Output data control register_1	SRCODCTRL_1	16	H'FFFE780A	16
	Control register_1	SRCCTRL_1	16	H'FFFE780C	16
	Status register_1	SRCSTAT_1	16	H'FFFE780E	16
General purpose I/O ports	Port A I/O register 0	PAIOR0	16	H'FFFE3812	8, 16
	Port A data register 1	PADR1	16	H'FFFE3814	8, 16, 32*
	Port A data register 0	PADR0	16	H'FFFE3816	8, 16*
	Port A port register 0	PAPR0	16	H'FFFE381A	8, 16
	Port B control register 5	PBCR5	16	H'FFFE3824	8, 16, 32
	Port B control register 4	PBCR4	16	H'FFFE3826	8, 16
	Port B control register 3	PBCR3	16	H'FFFE3828	8, 16, 32
	Port B control register 2	PBCR2	16	H'FFFE382A	8, 16
	Port B control register 1	PBCR1	16	H'FFFE382C	8, 16, 32
	Port B control register 0	PBCR0	16	H'FFFE382E	8, 16
	Port B I/O register 1	PBIOR1	16	H'FFFE3830	8, 16, 32
	Port B I/O register 0	PBIOR0	16	H'FFFE3832	8, 16
	Port B data register 1	PBDR1	16	H'FFFE3834	8, 16, 32
	Port B data register 0	PBDR0	16	H'FFFE3836	8, 16
	Port B port register 1	PBPR1	16	H'FFFE3838	8, 16, 32
	Port B port register 0	PBPR0	16	H'FFFE383A	8, 16
	Port C control register 2	PCCR2	16	H'FFFE384A	8, 16
	Port C control register 1	PCCR1	16	H'FFFE384C	8, 16, 32
	Port C control register 0	PCCR0	16	H'FFFE384E	8, 16
	Port C I/O register 0	PCIOR0	16	H'FFFE3852	8, 16
	Port C data register 0	PCDR0	16	H'FFFE3856	8, 16
	Port C port register 0	PCPR0	16	H'FFFE385A	8, 16
	Port D control register 3	PDCR3	16	H'FFFE3868	8, 16, 32
	Port D control register 2	PDCR2	16	H'FFFE386A	8, 16
	Port D control register 1	PDCR1	16	H'FFFE386C	8, 16, 32
	Port D control register 0	PDCR0	16	H'FFFE386E	8, 16
	Port D I/O register 0	PDIOR0	16	H'FFFE3872	8, 16
	Port D data register 0	PDDR0	16	H'FFFE3876	8, 16
	Port D port register 0	PDPR0	16	H'FFFE387A	8, 16

chanhvinh.com

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
General purpose I/O ports	Port E control register 1	PECR1	16	H'FFFE388C	8, 16, 32
	Port E control register 0	PECR0	16	H'FFFE388E	8, 16
	Port E I/O register 0	PEIOR0	16	H'FFFE3892	8, 16
	Port E data register 0	PEDR0	16	H'FFFE3896	8, 16
	Port E port register 0	PEPR0	16	H'FFFE389A	8, 16
	Port F control register 3	PFCR3	16	H'FFFE38A8	8, 16, 32
	Port F control register 2	PFCR2	16	H'FFFE38AA	8, 16
	Port F control register 1	PFCR1	16	H'FFFE38AC	8, 16, 32
	Port F control register 0	PFCR0	16	H'FFFE38AE	8, 16
	Port F I/O register 0	PFIOR0	16	H'FFFE38B2	8, 16
	Port F data register 0	PFDR0	16	H'FFFE38B6	8, 16
	Port F port register 0	PFPR0	16	H'FFFE38BA	8, 16
	Port G control register 7	PGCR7	16	H'FFFE38C0	8, 16, 32*
	Port G control register 6	PGCR6	16	H'FFFE38C2	8, 16
	Port G control register 5	PGCR5	16	H'FFFE38C4	8, 16, 32
	Port G control register 4	PGCR4	16	H'FFFE38C6	8, 16
	Port G control register 3	PGCR3	16	H'FFFE38C8	8, 16, 32
	Port G control register 2	PGCR2	16	H'FFFE38CA	8, 16
	Port G control register 1	PGCR1	16	H'FFFE38CC	8, 16, 32
	Port G control register 0	PGCR0	16	H'FFFE38CE	8, 16
	Port G I/O register 1	PGIOR1	16	H'FFFE38D0	8, 16, 32
	Port G I/O register 0	PGIOR0	16	H'FFFE38D2	8, 16
	Port G data register 1	PGDR1	16	H'FFFE38D4	8, 16, 32
	Port G data register 0	PGDR0	16	H'FFFE38D6	8, 16
	Port G port register 1	PGPR1	16	H'FFFE38D8	8, 16, 32
	Port G port register 0	PGPR0	16	H'FFFE38DA	8, 16
	Port H control register 1	PHCR1	16	H'FFFE38EC	8, 16, 32
	Port H control register 0	PHCR0	16	H'FFFE38EE	8, 16
	Port H port register 0	PHPR0	16	H'FFFE38FA	8, 16
	Port J control register 2	PJCR2	16	H'FFFE390A	8, 16
	Port J control register 1	PJCR1	16	H'FFFE390C	8, 16, 32
	Port J control register 0	PJCR0	16	H'FFFE390E	8, 16

chanhvinh.com

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
General purpose I/O ports	Port J I/O register 0	PJIOR0	16	H'FFFE3912	8, 16
	Port J data register 0	PJDR0	16	H'FFFE3916	8, 16
	Port J port register 0	PJPR0	16	H'FFFE391A	8, 16
	Port K control register 2	PKCR2	16	H'FFFE392A	8, 16
	Port K control register 1	PKCR1	16	H'FFFE392C	8, 16, 32
	Port K control register 0	PKCR0	16	H'FFFE392E	8, 16
	Port K I/O register 0	PKIOR0	16	H'FFFE3932	8, 16
	Port K data register 0	PKDR0	16	H'FFFE3936	8, 16
	Port K port register 0	PKPR0	16	H'FFFE393A	8, 16
Power-down modes	Standby control register 1	STBCR1	8	H'FFFE0014	8
	Standby control register 2	STBCR2	8	H'FFFE0018	8
	Standby control register 3	STBCR3	8	H'FFFE0408	8
	Standby control register 4	STBCR4	8	H'FFFE040C	8
	Standby control register 5	STBCR5	8	H'FFFE0410	8
	Standby control register 6	STBCR6	8	H'FFFE0414	8
	Standby control register 7	STBCR7	8	H'FFFE0418	8
	Standby control register 8	STBCR8	8	H'FFFE041C	8
	Software reset control register	SWRSTCR	8	H'FFFE0430	8
	System control register 1	SYSCR1	8	H'FFFE0400	8
	System control register 2	SYSCR2	8	H'FFFE0404	8
	System control register 3	SYSCR3	8	H'FFFE0420	8
	System control register 4	SYSCR4	8	H'FFFE0424	8
	System control register 5	SYSCR5	8	H'FFFE0428	8
	Data retention on-chip RAM area specification register	RRAMKP	8	H'FFFE6800	8
	Deep standby control register	DSCTR	8	H'FFFE6802	8
	Deep standby cancel source select register	DSSSR	16	H'FFFE6804	16
	Deep standby cancel edge select register	DSESR	16	H'FFFE6806	16
	Deep standby cancel source flag register 1	DSFR	16	H'FFFE6808	16
	XTAL crystal oscillator gain control register	XTALCTR	8	H'FFFE6810	8
User debugging interface	Instruction register	SDIR	16	H'FFFE2000	16

chanhvinh.com

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
Motor control PWM timer	PWM control register_1	PWCR_1	8	H'FFFEF4E0	8, 16
	PWM polarity register_1	PWPR_1	8	H'FFFEF4E4	8, 16
	PWM cycle register_1	PWCYR_1	16	H'FFFEF4E6	16
	PWM buffer register_1A	PWBFR_1A	16	H'FFFEF4E8	16
	PWM buffer register_1C	PWBFR_1C	16	H'FFFEF4EA	16
	PWM buffer register_1E	PWBFR_1E	16	H'FFFEF4EC	16
	PWM buffer register_1G	PWBFR_1G	16	H'FFFEF4EE	16
	PWM control register_2	PWCR_2	8	H'FFFEF4F0	8, 16
	PWM polarity register_2	PWPR_2	8	H'FFFEF4F4	8, 16
	PWM cycle register_2	PWCYR_2	16	H'FFFEF4F6	16
	PWM buffer register_2A	PWBFR_2A	16	H'FFFEF4F8	16
	PWM buffer register_2C	PWBFR_2C	16	H'FFFEF4FA	16
	PWM buffer register_2E	PWBFR_2E	16	H'FFFEF4FC	16
	PWM buffer register_2G	PWBFR_2G	16	H'FFFEF4FE	16
	PWM buffer transfer control register	PWBTCR	8	H'FFFEF406	8, 16

chanhvinh.com

36.2 Register Bits

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Clock pulse generator	FRQCR	—	CKOEN2	CKOEN[1]	CKOEN[0]	—	—	—	STC
		—	—	IFC[1]	IFC[0]	—	PFC[2]	PFC[1]	PFC[0]
Interrupt controller	ICR0	NMIL	—	—	—	—	—	—	NMIE
		—	—	—	—	—	—	NMIF	NMIM
	ICR1	IRQ71S	IRQ70S	IRQ61S	IRQ60S	IRQ51S	IRQ50S	IRQ41S	IRQ40S
		IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S
	ICR2	—	—	—	—	—	—	—	—
		PINT7S	PINT6S	PINT5S	PINT4S	PINT3S	PINT2S	PINT1S	PINT0S
	IRQRR	—	—	—	—	—	—	—	—
		IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
	PINTER	—	—	—	—	—	—	—	—
		PINT7E	PINT6E	PINT5E	PINT4E	PINT3E	PINT2E	PINT1E	PINT0E
	PIRR	—	—	—	—	—	—	—	—
		PINT7R	PINT6R	PINT5R	PINT4R	PINT3R	PINT2R	PINT1R	PINT0R
	IBCR	E15	E14	E13	E12	E11	E10	E9	E8
		E7	E6	E5	E4	E3	E2	E1	—
	IBNR	BE[1]	BE[0]	BOVE	—	—	—	—	—
		—	—	—	—	BN[3]	BN[2]	BN[1]	BN[0]
	IPR01								
	IPR02								
	IPR05								
	IPR06								
	IPR07								

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Interrupt controller	IPR08								
	IPR09								
	IPR10								
	IPR11								
	IPR12								
	IPR13								
	IPR14								
	IPR15								
	IPR16								
	IPR17								
	IPR18								
	IPR19								
	IPR20								
	IPR21								
	IPR22								

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Cache	CCR1	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	ICF	—	—	ICE
		—	—	—	—	OCF	—	WT	OCE
	CCR2	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	LE
		—	—	—	—	—	—	W3LOAD	W3LOCK
		—	—	—	—	—	—	W2LOAD	W2LOCK
Bus state controller	CMNCR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	BLOCK	DPRTY[1]	DPRTY[0]	DMAIW[2]
		DMAIW[1]	DMAIW[0]	DMAIWA	—	—	—	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]	ENDIAN	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]	ENDIAN	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]	ENDIAN	BSZ[1]	BSZ[0]	—
		—	—	—	—	—	—	—	—
	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]	ENDIAN	BSZ[1]	BSZ[0]	—
		—	—	—	—	—	—	—	—

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Bus state controller	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]	ENDIAN	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]	ENDIAN	BSZ[1]	BSZ[0]	—
		—	—	—	—	—	—	—	—
	CS6BCR	—	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]	ENDIAN	BSZ[1]	BSZ[0]	—
		—	—	—	—	—	—	—	—
	CS0WCR	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	—	—	—
		—	—	—	SW[1]	SW[0]	WR[3]	WR[2]	WR[1]
		WR[0]	WM	—	—	—	—	HW[1]	HW[0]
	CS0WCR	—	—	—	—	—	—	—	—
		—	—	BST[1]	BST[0]	—	—	BW[1]	BW[0]
		—	—	—	—	—	W[3]	W[2]	W[1]
		W[0]	WM	—	—	—	—	—	—
	CS0WCR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	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]

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Bus state controller	CS2WCR	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	—	—	—
		—	—	—	—	—	WR[3]	WR[2]	WR[1]
		WR[0]	WM	—	—	—	—	—	—
	CS2WCR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	A2CL1
		A2CL0	—	—	—	—	—	—	—
	CS3WCR	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	—	—	—
		—	—	—	—	—	WR[3]	WR[2]	WR[1]
		WR[0]	WM	—	—	—	—	—	—
	CS3WCR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	WTRP[1]	WTRP[0]	—	WTRCD[1]	WTRCD[0]	—	A3CL1
		A3CL0	—	—	TRWL[1]	TRWL[0]	—	WTRC[1]	WTRC[0]
	CS4WCR	—	—	—	—	—	—	—	—
		—	—	—	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	—	—	—	—	—	—	—	—
		—	—	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	—	—	—	—	—	—	—	—
		—	—	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

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Bus state controller	CS5WCR	—	—	—	—	—	—	—	—
		—	—	SA[1]	SA[0]	—	—	—	—
		—	TED[3]	TED[2]	TED[1]	TED[0]	PCW[1]	PCW[0]	PCW[1]
		PCW[0]	WM	—	—	THE[3]	THE[2]	THE[1]	THE[0]
	CS6WCR	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	—	—	—
		—	—	—	SW[1]	SW[0]	WR[3]	WR[2]	WR[1]
		WR[0]	WM	—	—	—	—	HW[1]	HW[0]
	CS6WCR	—	—	—	—	—	—	—	—
		—	—	MPXAW[1]	MPXAW[0]	MPXMD	—	BW[1]	BW[0]
		—	—	—	—	—	W[3]	W[2]	W[1]
		W[0]	WM	—	—	—	—	—	—
	CS6WCR	—	—	—	—	—	—	—	—
		—	—	SA[1]	SA[0]	—	—	—	—
		—	TED[3]	TED[2]	TED[1]	TED[0]	PCW[1]	PCW[0]	PCW[1]
		PCW[0]	WM	—	—	THE[3]	THE[2]	THE[1]	THE[0]
	SDCR	—	—	—	—	—	—	—	—
		—	—	—	A2ROW[1]	A2ROW[0]	—	A2COL[1]	A2COL[0]
		—	—	DEEP	SLOW	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	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Bus state controller	RTCOR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	ACSWR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	ACOSW[3:0]			
	ACKEYR	ACKEY[7:0]							
Direct memory access controller	SAR_0								
	DAR_0								
	DMATCR_0	—	—	—	—	—	—	—	—
	CHCR_0	TC	—	RLDSAR	RLDDAR	—	DAF	SAF	—
		DO	TL	—	TEMASK	HE	HIE	AM	AL
		DM[1]	DM[0]	SM[1]	SM[0]	RS[3]	RS[2]	RS[1]	RS[0]
		DL	DS	TB	TS[1]	TS[0]	IE	TE	DE
	RSAR_0								
	RDAR_0								

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Direct memory access controller	RDMATCR_0	—	—	—	—	—	—	—	—
	SAR_1								
	DAR_1								
	DMATCR_1	—	—	—	—	—	—	—	—
	CHCR_1	TC	—	RLDSAR	RLDDAR	—	DAF	SAF	—
		DO	TL	—	TEMASK	HE	HIE	AM	AL
		DM[1]	DM[0]	SM[1]	SM[0]	RS[3]	RS[2]	RS[1]	RS[0]
		DL	DS	TB	TS[1]	TS[0]	IE	TE	DE
	RSAR_1								
	RDAR_1								

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Direct memory access controller	RDMATCR_1	—	—	—	—	—	—	—	—
	SAR_2								
	DAR_2								
	DMATCR_2	—	—	—	—	—	—	—	—
	CHCR_2	TC	—	RLDSAR	RLDDAR	—	DAF	SAF	—
		—	—	—	TEMASK	HE	HIE	—	—
		DM[1]	DM[0]	SM[1]	SM[0]	RS[3]	RS[2]	RS[1]	RS[0]
		—	—	TB	TS[1]	TS[0]	IE	TE	DE
	RSAR_2								
	RDAR_2								

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Direct memory access controller	RDMATCR_2	—	—	—	—	—	—	—	—
	SAR_3								
	DAR_3								
	DMATCR_3	—	—	—	—	—	—	—	—
	CHCR_3	TC	—	RLDSAR	RLDDAR	—	DAF	SAF	—
		—	—	—	TEMASK	HE	HIE	—	—
		DM[1]	DM[0]	SM[1]	SM[0]	RS[3]	RS[2]	RS[1]	RS[0]
		—	—	TB	TS[1]	TS[0]	IE	TE	DE
	RSAR_3								
	RDAR_3								

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Direct memory access controller	RDMATCR_3	—	—	—	—	—	—	—	—
	SAR_4								
	DAR_4								
	DMATCR_4	—	—	—	—	—	—	—	—
	CHCR_4	TC	—	RLDSAR	RLDDAR	—	DAF	SAF	—
		—	—	—	TEMASK	HE	HIE	—	—
		DM[1]	DM[0]	SM[1]	SM[0]	RS[3]	RS[2]	RS[1]	RS[0]
		—	—	TB	TS[1]	TS[0]	IE	TE	DE
	RSAR_4								
	RDAR_4								

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Direct memory access controller	RDMATCR_4	—	—	—	—	—	—	—	—
	SAR_5								
	DAR_5								
	DMATCR_5	—	—	—	—	—	—	—	—
	CHCR_5	TC	—	RLDSAR	RLDDAR	—	DAF	SAF	—
		—	—	—	TEMASK	HE	HIE	—	—
		DM[1]	DM[0]	SM[1]	SM[0]	RS[3]	RS[2]	RS[1]	RS[0]
		—	—	TB	TS[1]	TS[0]	IE	TE	DE
	RSAR_5								
	RDAR_5								

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Direct memory access controller	RDMATCR_5	—	—	—	—	—	—	—	—
	SAR_6								
	DAR_6								
	DMATCR_6	—	—	—	—	—	—	—	—
	CHCR_6	TC	—	RLDSAR	RLDDAR	—	DAF	SAF	—
		—	—	—	TEMASK	HE	HIE	—	—
		DM[1]	DM[0]	SM[1]	SM[0]	RS[3]	RS[2]	RS[1]	RS[0]
		—	—	TB	TS[1]	TS[0]	IE	TE	DE
	RSAR_6								
	RDAR_6								

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Direct memory access controller	RDMATCR_6	—	—	—	—	—	—	—	—
	SAR_7								
	DAR_7								
	DMATCR_7	—	—	—	—	—	—	—	—
	CHCR_7	TC	—	RLDSAR	RLDDAR	—	DAF	SAF	—
		—	—	—	TEMASK	HE	HIE	—	—
		DM[1]	DM[0]	SM[1]	SM[0]	RS[3]	RS[2]	RS[1]	RS[0]
		—	—	TB	TS[1]	TS[0]	IE	TE	DE
	RSAR_7								
	RDAR_7								

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Direct memory access controller	RDMATCR_7	—	—	—	—	—	—	—	—
	SAR_8								
	DAR_8								
	DMATCR_8	—	—	—	—	—	—	—	—
	CHCR_8	TC	—	RLDSAR	RLDDAR	—	DAF	SAF	—
		—	—	—	TEMASK	HE	HIE	—	—
		DM[1]	DM[0]	SM[1]	SM[0]	RS[3]	RS[2]	RS[1]	RS[0]
		—	—	TB	TS[1]	TS[0]	IE	TE	DE
	RSAR_8								
	RDAR_8								

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Direct memory access controller	RDMATCR_8	—	—	—	—	—	—	—	—
	SAR_9								
	DAR_9								
	DMATCR_9	—	—	—	—	—	—	—	—
	CHCR_9	TC	—	RLDSAR	RLDDAR	—	DAF	SAF	—
		—	—	—	TEMASK	HE	HIE	—	—
		DM[1]	DM[0]	SM[1]	SM[0]	RS[3]	RS[2]	RS[1]	RS[0]
		—	—	TB	TS[1]	TS[0]	IE	TE	DE
	RSAR_9								
	RDAR_9								

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Direct memory access controller	RDMATCR_9	—	—	—	—	—	—	—	—
	SAR_10								
	DAR_10								
	DMATCR_10	—	—	—	—	—	—	—	—
	CHCR_10	TC	—	RLDSAR	RLDDAR	—	DAF	SAF	—
		—	—	—	TEMASK	HE	HIE	—	—
		DM[1]	DM[0]	SM[1]	SM[0]	RS[3]	RS[2]	RS[1]	RS[0]
		—	—	TB	TS[1]	TS[0]	IE	TE	DE
	RSAR_10								
	RDAR_10								

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Direct memory access controller	RDMATCR_10	—	—	—	—	—	—	—	—
	SAR_11								
	DAR_11								
	DMATCR_11	—	—	—	—	—	—	—	—
	CHCR_11	TC	—	RLDSAR	RLDDAR	—	DAF	SAF	—
		—	—	—	TEMASK	HE	HIE	—	—
		DM[1]	DM[0]	SM[1]	SM[0]	RS[3]	RS[2]	RS[1]	RS[0]
		—	—	TB	TS[1]	TS[0]	IE	TE	DE
	RSAR_11								
	RDAR_11								

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Direct memory access controller	RDMATCR_11	—	—	—	—	—	—	—	—
	SAR_12								
	DAR_12								
	DMATCR_12	—	—	—	—	—	—	—	—
	CHCR_12	TC	—	RLDSAR	RLDDAR	—	DAF	SAF	—
		—	—	—	TEMASK	HE	HIE	—	—
		DM[1]	DM[0]	SM[1]	SM[0]	RS[3]	RS[2]	RS[1]	RS[0]
		—	—	TB	TS[1]	TS[0]	IE	TE	DE
	RSAR_12								
	RDAR_12								

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Direct memory access controller	RDMATCR_12	—	—	—	—	—	—	—	—
	SAR_13								
	DAR_13								
	DMATCR_13	—	—	—	—	—	—	—	—
	CHCR_13	TC	—	RLDSAR	RLDDAR	—	DAF	SAF	—
		—	—	—	TEMASK	HE	HIE	—	—
		DM[1]	DM[0]	SM[1]	SM[0]	RS[3]	RS[2]	RS[1]	RS[0]
		—	—	TB	TS[1]	TS[0]	IE	TE	DE
	RSAR_13								
	RDAR_13								

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Direct memory access controller	RDMATCR_13	—	—	—	—	—	—	—	—
	SAR_14								
	DAR_14								
	DMATCR_14	—	—	—	—	—	—	—	—
	CHCR_14	TC	—	RLDSAR	RLDDAR	—	DAF	SAF	—
		—	—	—	TEMASK	HE	HIE	—	—
		DM[1]	DM[0]	SM[1]	SM[0]	RS[3]	RS[2]	RS[1]	RS[0]
		—	—	TB	TS[1]	TS[0]	IE	TE	DE
	RSAR_14								
	RDAR_14								

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Direct memory access controller	RDMATCR_14	—	—	—	—	—	—	—	—
	SAR_15								
	DAR_15								
	DMATCR_15	—	—	—	—	—	—	—	—
	CHCR_15	TC	—	RLDSAR	RLDDAR	—	DAF	SAF	—
		—	—	—	TEMASK	HE	HIE	—	—
		DM[1]	DM[0]	SM[1]	SM[0]	RS[3]	RS[2]	RS[1]	RS[0]
		—	—	TB	TS[1]	TS[0]	IE	TE	DE
	RSAR_15								
	RDAR_15								

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Direct memory access controller	RDMATCR_15	—	—	—	—	—	—	—	—
	DMAOR	—	—	CMS[1]	CMS[0]	—	—	PR[1]	PR[0]
		—	—	—	—	—	AE	NMIF	DME
	DMARS0	CH1MID[5]	CH1MID[4]	CH1MID[3]	CH1MID[2]	CH1MID[1]	CH1MID[0]	CH1RID[1]	CH1RID[0]
		CH0MID[5]	CH0MID[4]	CH0MID[3]	CH0MID[2]	CH0MID[1]	CH0MID[0]	CH0RID[1]	CH0RID[0]
	DMARS1	CH3MID[5]	CH3MID[4]	CH3MID[3]	CH3MID[2]	CH3MID[1]	CH3MID[0]	CH3RID[1]	CH3RID[0]
		CH2MID[5]	CH2MID[4]	CH2MID[3]	CH2MID[2]	CH2MID[1]	CH2MID[0]	CH2RID[1]	CH2RID[0]
	DMARS2	CH5MID[5]	CH5MID[4]	CH5MID[3]	CH5MID[2]	CH5MID[1]	CH5MID[0]	CH5RID[1]	CH5RID[0]
		CH4MID[5]	CH4MID[4]	CH4MID[3]	CH4MID[2]	CH4MID[1]	CH4MID[0]	CH4RID[1]	CH4RID[0]
	DMARS3	CH7MID[5]	CH7MID[4]	CH7MID[3]	CH7MID[2]	CH7MID[1]	CH7MID[0]	CH7RID[1]	CH7RID[0]
		CH6MID[5]	CH6MID[4]	CH6MID[3]	CH6MID[2]	CH6MID[1]	CH6MID[0]	CH6RID[1]	CH6RID[0]
	DMARS4	CH9MID[5]	CH9MID[4]	CH9MID[3]	CH9MID[2]	CH9MID[1]	CH9MID[0]	CH9RID[1]	CH9RID[0]
		CH8MID[5]	CH8MID[4]	CH8MID[3]	CH8MID[2]	CH8MID[1]	CH8MID[0]	CH8RID[1]	CH8RID[0]
	DMARS5	CH11MID[5]	CH11MID[4]	CH11MID[3]	CH11MID[2]	CH11MID[1]	CH11MID[0]	CH11RID[1]	CH11RID[0]
		CH10MID[5]	CH10MID[4]	CH10MID[3]	CH10MID[2]	CH10MID[1]	CH10MID[0]	CH10RID[1]	CH10RID[0]
	DMARS6	CH13MID[5]	CH13MID[4]	CH13MID[3]	CH13MID[2]	CH13MID[1]	CH13MID[0]	CH13RID[1]	CH13RID[0]
		CH12MID[5]	CH12MID[4]	CH12MID[3]	CH12MID[2]	CH12MID[1]	CH12MID[0]	CH12RID[1]	CH12RID[0]
	DMARS7	CH15MID[5]	CH15MID[4]	CH15MID[3]	CH15MID[2]	CH15MID[1]	CH15MID[0]	CH15RID[1]	CH15RID[0]
		CH14MID[5]	CH14MID[4]	CH14MID[3]	CH14MID[2]	CH14MID[1]	CH14MID[0]	CH14RID[1]	CH14RID[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								

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Multi-function timer pulse unit 2	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

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Multi-function timer pulse unit 2	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]
	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

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Multi-function timer pulse unit 2	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								

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Multi-function timer pulse unit 2	TDDR								
	TCNTS								
	TCBR								
	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
Compare match timer	CMSTR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	STR1	STR0
	CMCSR_0	—	—	—	—	—	—	—	—
		CMF	CMIE	—	—	—	—	CKS[1]	CKS[0]
	CMCNT_0								
	CMCOR_0								
	CMCSR_1	—	—	—	—	—	—	—	—
		CMF	CMIE	—	—	—	—	CKS[1]	CKS[0]
	CMCNT_1								
	CMCOR_1								
Watchdog timer	WTCNT								
	WTCSR	IOVF	WT/IT	TME	—	—	CKS[2]	CKS[1]	CKS[0]
	WRCSR	WOVF	RSTE	RSTS	—	—	—	—	—

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
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
	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]
Serial communication interface with FIFO	SCSMR_0	—	—	—	—	—	—	—	—
		C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS[1]	CKS[0]
	SCBRR_0								
	SCBSCSCR_0	—	—	—	—	—	—	—	—
		TIE	RIE	TE	RE	REIE	—	CKE[1]	CKE[0]
	SCFTDR_0								

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Serial communication interface with FIFO	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	—	—	—	—	—	—	—	—
		—	—	—	—	SCKIO	SCKDT	SPB2IO	SPB2DT
	SCLSR_0	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ORER
	SCEMR_0	—	—	—	—	—	—	—	—
		BGDM	—	—	—	—	—	—	ABCS
	SCSMR_1	—	—	—	—	—	—	—	—
		C/A	CHR	PE	O/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

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Serial communication interface with FIFO	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	—	—	—	—	—	—	—	—
		—	—	—	—	SCKIO	SCKDT	SPB2IO	SPB2DT
	SCLSR_2	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ORER
	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								

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Serial communication interface with FIFO	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
	SCEMR_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	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	SPB2IO	SPB2DT
	SCLSR_4	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ORER
	SCEMR_4	—	—	—	—	—	—	—	—
		BGDM	—	—	—	—	—	—	ABCS
	SCSMR_5	—	—	—	—	—	—	—	—
		C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS[1]	CKS[0]

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Serial communication interface with FIFO	SCBRR_5								
	SCSCR_5	—	—	—	—	—	—	—	—
		TIE	RIE	TE	RE	REIE	—	CKE[1]	CKE[0]
	SCFTDR_5								
	SCFSR_5	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_5								
	SCFCR_5	—	—	—	—	—	RSTRG[2]	RSTRG[1]	RSTRG[0]
		RTRG[1]	RTRG[0]	TTRG[1]	TTRG[0]	MCE	TFRST	RFRST	LOOP
	SCFDR_5	—	—	—	T[4]	T[3]	T[2]	T[1]	T[0]
		—	—	—	R[4]	R[3]	R[2]	R[1]	R[0]
	SCSPTR_5	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	SPB2IO	SPB2DT
	SCLSR_5	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ORER
	SCEMR_5	—	—	—	—	—	—	—	—
		BGDM	—	—	—	—	—	—	ABCS
	SCSMR_6	—	—	—	—	—	—	—	—
		C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS[1]	CKS[0]
	SCBRR_6								
	SCSCR_6	—	—	—	—	—	—	—	—
		TIE	RIE	TE	RE	REIE	—	CKE[1]	CKE[0]
	SCFTDR_6								
	SCFSR_6	PER[3]	PER[2]	PER[1]	PER[0]	FER[3]	FER[2]	FER[1]	FER[0]
	SCFRDR_6								
	SCFCR_6	—	—	—	—	—	RSTRG[2]	RSTRG[1]	RSTRG[0]
		RTRG[1]	RTRG[0]	TTRG[1]	TTRG[0]	MCE	TFRST	RFRST	LOOP
	SCFDR_6	—	—	—	T[4]	T[3]	T[2]	T[1]	T[0]
		—	—	—	R[4]	R[3]	R[2]	R[1]	R[0]
	SCSPTR_6	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	SPB2IO	SPB2DT

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Serial communication interface with FIFO	SCLSR_6	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ORER
	SCEMR_6	—	—	—	—	—	—	—	—
		BGDM	—	—	—	—	—	—	ABCS
	SCSMR_7	—	—	—	—	—	—	—	—
		C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS[1]	CKS[0]
	SCBRR_7								
	SCSCR_7	—	—	—	—	—	—	—	—
		TIE	RIE	TE	RE	REIE	—	CKE[1]	CKE[0]
	SCFTDR_7								
	SCFSR_7	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_7								
	SCFCR_7	—	—	—	—	—	RSTRG[2]	RSTRG[1]	RSTRG[0]
		RTRG[1]	RTRG[0]	TTRG[1]	TTRG[0]	MCE	TFRST	RFRST	LOOP
	SCFDR_7	—	—	—	T[4]	T[3]	T[2]	T[1]	T[0]
		—	—	—	R[4]	R[3]	R[2]	R[1]	R[0]
	SCSPTR_7	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	SPB2IO	SPB2DT
	SCLSR_7	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ORER
	SCEMR_7	—	—	—	—	—	—	—	—
		BGDM	—	—	—	—	—	—	ABCS
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

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Renesas serial peripheral interface	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
	SPCMD_00	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB3	SPB2	SPB1	SPB0
		SSLKP	—	—	—	BRDV1	BRDV0	CPOL	CPHA
	SPCMD_01	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB3	SPB2	SPB1	SPB0
		SSLKP	—	—	—	BRDV1	BRDV0	CPOL	CPHA
	SPCMD_02	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB3	SPB2	SPB1	SPB0
		SSLKP	—	—	—	BRDV1	BRDV0	CPOL	CPHA
	SPCMD_03	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

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Renesas serial peripheral interface	SSLND_1	—	—	—	—	—	SLNDL2	SLNDL1	SLNDL0
	SPND_1	—	—	—	—	—	SPNDL2	SPNDL1	SPNDL0
	SPCMD_10	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB3	SPB2	SPB1	SPB0
		SSLKP	—	—	—	BRDV1	BRDV0	CPOL	CPHA
	SPCMD_11	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB3	SPB2	SPB1	SPB0
		SSLKP	—	—	—	BRDV1	BRDV0	CPOL	CPHA
	SPCMD_12	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB3	SPB2	SPB1	SPB0
		SSLKP	—	—	—	BRDV1	BRDV0	CPOL	CPHA
	SPCMD_13	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]
	SPBFDR_1	—	—	—	—	T[3]	T[2]	T[1]	T[0]
		—	—	R[5]	R[4]	R[3]	R[2]	R[1]	R[0]
I²C bus interface 3	ICCR1_0	ICE	RCVD	MST	TRS	CKS[3]	CKS[2]	CKS[1]	CKS[0]
	ICCR2_0	BBSY	SCP	SDAO	SDAOP	SCLO	—	IICRST	—
	ICMR_0	MLS	—	—	—	BCWP	BC[2]	BC[1]	BC[0]
	ICIER_0	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT
	ICSR_0	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ
	SAR_0	SVA[6]	SVA[5]	SVA[4]	SVA[3]	SVA[2]	SVA[1]	SVA[0]	FS
	ICDRT_0								
	ICDRR_0								
	NF2CYC_0	—	—	—	CKS4	—	—	PRS	NF2CYC
	ICCR1_1	ICE	RCVD	MST	TRS	CKS[3]	CKS[2]	CKS[1]	CKS[0]
	ICCR2_1	BBSY	SCP	SDAO	SDAOP	SCLO	—	IICRST	—
	ICMR_1	MLS	—	—	—	BCWP	BC[2]	BC[1]	BC[0]
	ICIER_1	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT
	ICSR_1	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ
	SAR_1	SVA[6]	SVA[5]	SVA[4]	SVA[3]	SVA[2]	SVA[1]	SVA[0]	FS
	ICDRT_1								
	ICDRR_1								
	NF2CYC_1	—	—	—	CKS4	—	—	PRS	NF2CYC

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
I²C bus interface 3	ICCR1_2	ICE	RCVD	MST	TRS	CKS[3]	CKS[2]	CKS[1]	CKS[0]
	ICCR2_2	BBSY	SCP	SDAO	SDAOP	SCLO	—	IICRST	—
	ICMR_2	MLS	—	—	—	BCWP	BC[2]	BC[1]	BC[0]
	ICIER_2	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT
	ICSR_2	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ
	SAR_2	SVA[6]	SVA[5]	SVA[4]	SVA[3]	SVA[2]	SVA[1]	SVA[0]	FS
	ICDRT_2								
	ICDRR_2								
	NF2CYC_2	—	—	—	CKS4	—	—	PRS	NF2CYC
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								

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Serial sound interface	SSIFRDR_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	TOI	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
	SSIFSR_1	—	—	—	—	TDC[3]	TDC[2]	TDC[1]	TDC[0]
		—	—	—	—	—	—	—	TDE
		—	—	—	—	RDC[3]	RDC[2]	RDC[1]	RDC[0]
		—	—	—	—	—	—	—	RDF
	SSIFTDR_1								
	SSIFRDR_1								

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Serial sound interface	SSICR_2	—	CKS	TUIEN	TOIMEN	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[1]	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								
	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

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Serial sound interface	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								
Serial I/O with FIFO	SIMDR	TRMD[1]	TRMD[0]	SYNCAT	REDG	FL[3]	FL[2]	FL[1]	FL[0]
		TXDIZ	—	SYNCAC	SYNCDL	—	—	—	—
	SISCR	MSSEL	—	—	BRPS[4]	BRPS[3]	BRPS[2]	BRPS[1]	BRPS[0]
		—	—	—	—	—	BRDV[2]	BRDV[1]	BRDV[0]
	SITDAR	TDLE	—	—	—	TDLA[3]	TDLA[2]	TDLA[1]	TDLA[0]
		TDRE	TLREP	—	—	TDRA[3]	TDRA[2]	TDRA[1]	TDRA[0]
	SIRDAR	RDLE	—	—	—	RDLA[3]	RDLA[2]	RDLA[1]	RDLA[0]
		RDRE	—	—	—	RDRA[3]	RDRA[2]	RDRA[1]	RDRA[0]

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Serial I/O with FIFO	SICTR	SCKE	FSE	—	—	—	—	TXE	RXE
		—	—	—	—	—	—	TXRST	RXRST
	SIFCTR	TFWM[2]	TFWM[1]	TFWM[0]	TFUA[4]	TFUA[3]	TFUA[2]	TFUA[1]	TFUA[0]
		RFWM[2]	RFWM[1]	RFWM[0]	RFUA[4]	RFUA[3]	RFUA[2]	RFUA[1]	RFUA[0]
	SISTR	—	—	TFEMP	TDREQ	—	—	RFFUL	RDREQ
		—	—	—	FSERR	TFOVF	TFUDF	RFUDF	RFOVF
	SIIER	TDMAE	—	TFEMPE	TDREQE	RDMAE	—	RFFULE	RDREQE
		—	—	—	FSERRE	TFOVFE	TFUDFE	RFUDFE	RFOVFE
	SITDR	SITDL[15]	SITDL[14]	SITDL[13]	SITDL[12]	SITDL[11]	SITDL[10]	SITDL[9]	SITDL[8]
		SITDL[7]	SITDL[6]	SITDL[5]	SITDL[4]	SITDL[3]	SITDL[2]	SITDL[1]	SITDL[0]
		SITDR[15]	SITDR[14]	SITDR[13]	SITDR[12]	SITDR[11]	SITDR[10]	SITDR[9]	SITDR[8]
		SITDR[7]	SITDR[6]	SITDR[5]	SITDR[4]	SITDR[3]	SITDR[2]	SITDR[1]	SITDR[0]
	SIRDR	SIRDL[15]	SIRDL[14]	SIRDL[13]	SIRDL[12]	SIRDL[11]	SIRDL[10]	SIRDL[9]	SIRDL[8]
		SIRDL[7]	SIRDL[6]	SIRDL[5]	SIRDL[4]	SIRDL[3]	SIRDL[2]	SIRDL[1]	SIRDL[0]
		SIRDR[15]	SIRDR[14]	SIRDR[13]	SIRDR[12]	SIRDR[11]	SIRDR[10]	SIRDR[9]	SIRDR[8]
		SIRDR[7]	SIRDR[6]	SIRDR[5]	SIRDR[4]	SIRDR[3]	SIRDR[2]	SIRDR[1]	SIRDR[0]
Controller area network	MCR_0	MCR15	MCR14	—	—	—	TST[2]	TST[1]	TST[0]
		MCR7	MCR6	MCR5	—	—	MCR2	MCR1	MCR0
	GSR_0	—	—	—	—	—	—	—	—
		—	—	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0
	BCR1_0	TSG1[3]	TSG1[2]	TSG1[1]	TSG1[0]	—	TSG2[2]	TSG2[1]	TSG2[0]
		—	—	SJW[1]	SJW[0]	—	—	—	BSP
	BCR0_0	—	—	—	—	—	—	—	—
		BRP[7]	BRP[6]	BRP[5]	BRP[4]	BRP[3]	BRP[2]	BRP[1]	BRP[0]
	IRR_0	IRR15	IRR14	IRR13	IRR12	IRR11	IRR10	IRR9	IRR8
		IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0
	IMR_0	IMR15	IMR14	IMR13	IMR12	IMR11	IMR10	IMR9	IMR8
		IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0
	TEC_REC_0	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]

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Controller area network	TXPR1_0	TXPR1[15]	TXPR1[14]	TXPR1[13]	TXPR1[12]	TXPR1[11]	TXPR1[10]	TXPR1[9]	TXPR1[8]
		TXPR1[7]	TXPR1[6]	TXPR1[5]	TXPR1[4]	TXPR1[3]	TXPR1[2]	TXPR1[1]	TXPR1[0]
	TXPR0_0	TXPR0[15]	TXPR0[14]	TXPR0[13]	TXPR0[12]	TXPR0[11]	TXPR0[10]	TXPR0[9]	TXPR0[8]
		TXPR0[7]	TXPR0[6]	TXPR0[5]	TXPR0[4]	TXPR0[3]	TXPR0[2]	TXPR0[1]	—
	TXCR1_0	TXCR1[15]	TXCR1[14]	TXCR1[13]	TXCR1[12]	TXCR1[11]	TXCR1[10]	TXCR1[9]	TXCR1[8]
		TXCR1[7]	TXCR1[6]	TXCR1[5]	TXCR1[4]	TXCR1[3]	TXCR1[2]	TXCR1[1]	TXCR1[0]
	TXCR0_0	TXCR0[15]	TXCR0[14]	TXCR0[13]	TXCR0[12]	TXCR0[11]	TXCR0[10]	TXCR0[9]	TXCR0[8]
		TXCR0[7]	TXCR0[6]	TXCR0[5]	TXCR0[4]	TXCR0[3]	TXCR0[2]	TXCR0[1]	—
	TXACK1_0	TXACK1[15]	TXACK1[14]	TXACK1[13]	TXACK1[12]	TXACK1[11]	TXACK1[10]	TXACK1[9]	TXACK1[8]
		TXACK1[7]	TXACK1[6]	TXACK1[5]	TXACK1[4]	TXACK1[3]	TXACK1[2]	TXACK1[1]	TXACK1[0]
	TXACK0_0	TXACK0[15]	TXACK0[14]	TXACK0[13]	TXACK0[12]	TXACK0[11]	TXACK0[10]	TXACK0[9]	TXACK0[8]
		TXACK0[7]	TXACK0[6]	TXACK0[5]	TXACK0[4]	TXACK0[3]	TXACK0[2]	TXACK0[1]	—
	ABACK1_0	ABACK1[15]	ABACK1[14]	ABACK1[13]	ABACK1[12]	ABACK1[11]	ABACK1[10]	ABACK1[9]	ABACK1[8]
		ABACK1[7]	ABACK1[6]	ABACK1[5]	ABACK1[4]	ABACK1[3]	ABACK1[2]	ABACK1[1]	ABACK1[0]
	ABACK0_0	ABACK0[15]	ABACK0[14]	ABACK0[13]	ABACK0[12]	ABACK0[11]	ABACK0[10]	ABACK0[9]	ABACK0[8]
		ABACK0[7]	ABACK0[6]	ABACK0[5]	ABACK0[4]	ABACK0[3]	ABACK0[2]	ABACK0[1]	—
	RXPR1_0	RXPR1[15]	RXPR1[14]	RXPR1[13]	RXPR1[12]	RXPR1[11]	RXPR1[10]	RXPR1[9]	RXPR1[8]
		RXPR1[7]	RXPR1[6]	RXPR1[5]	RXPR1[4]	RXPR1[3]	RXPR1[2]	RXPR1[1]	RXPR1[0]
	RXPR0_0	RXPR0[15]	RXPR0[14]	RXPR0[13]	RXPR0[12]	RXPR0[11]	RXPR0[10]	RXPR0[9]	RXPR0[8]
		RXPR0[7]	RXPR0[6]	RXPR0[5]	RXPR0[4]	RXPR0[3]	RXPR0[2]	RXPR0[1]	RXPR0[0]
	RFPR1_0	RFPR1[15]	RFPR1[14]	RFPR1[13]	RFPR1[12]	RFPR1[11]	RFPR1[10]	RFPR1[9]	RFPR1[8]
		RFPR1[7]	RFPR1[6]	RFPR1[5]	RFPR1[4]	RFPR1[3]	RFPR1[2]	RFPR1[1]	RFPR1[0]
	RFPR0_0	RFPR0[15]	RFPR0[14]	RFPR0[13]	RFPR0[12]	RFPR0[11]	RFPR0[10]	RFPR0[9]	RFPR0[8]
		RFPR0[7]	RFPR0[6]	RFPR0[5]	RFPR0[4]	RFPR0[3]	RFPR0[2]	RFPR0[1]	RFPR0[0]
	MBIMR1_0	MBIMR1[15]	MBIMR1[14]	MBIMR1[13]	MBIMR1[12]	MBIMR1[11]	MBIMR1[10]	MBIMR1[9]	MBIMR1[8]
		MBIMR1[7]	MBIMR1[6]	MBIMR1[5]	MBIMR1[4]	MBIMR1[3]	MBIMR1[2]	MBIMR1[1]	MBIMR1[0]
	MBIMR0_0	MBIMR0[15]	MBIMR0[14]	MBIMR0[13]	MBIMR0[12]	MBIMR0[11]	MBIMR0[10]	MBIMR0[9]	MBIMR0[8]
		MBIMR0[7]	MBIMR0[6]	MBIMR0[5]	MBIMR0[4]	MBIMR0[3]	MBIMR0[2]	MBIMR0[1]	MBIMR0[0]
	UMSR1_0	UMSR1[15]	UMSR1[14]	UMSR1[13]	UMSR1[12]	UMSR1[11]	UMSR1[10]	UMSR1[9]	UMSR1[8]
		UMSR1[7]	UMSR1[6]	UMSR1[5]	UMSR1[4]	UMSR1[3]	UMSR1[2]	UMSR1[1]	UMSR1[0]

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Controller area network	UMSR0_0	UMSR0[15]	UMSR0[14]	UMSR0[13]	UMSR0[12]	UMSR0[11]	UMSR0[10]	UMSR0[9]	UMSR0[8]
		UMSR0[7]	UMSR0[6]	UMSR0[5]	UMSR0[4]	UMSR0[3]	UMSR0[2]	UMSR0[1]	UMSR0[0]
	TTCR0_0	TCR15	TCR14	TCR13	TCR12	TCR11	TCR10	—	—
		—	TCR6	TPSC5	TPSC4	TPSC3	TPSC2	TPSC1	TPSC0
	CMAX_TEW_0	—	—	—	—	—	CMAX[2]	CMAX[1]	CMAX[0]
		—	—	—	—	TEW[3]	TEW[2]	TEW[1]	TEW[0]
	RFTROFF_0	RFTROFF[7]	RFTROFF[6]	RFTROFF[5]	RFTROFF[4]	RFTROFF[3]	RFTROFF[2]	RFTROFF[1]	RFTROFF[0]
		—	—	—	—	—	—	—	—
	TSR_0	—	—	—	—	—	—	—	—
		—	—	—	TSR4	TSR3	TSR2	TSR1	TSR0
	CCR_0	—	—	—	—	—	—	—	—
		—	—	CCR[5]	CCR[4]	CCR[3]	CCR[2]	CCR[1]	CCR[0]
	TCNTR_0	TCNTR[15]	TCNTR[14]	TCNTR[13]	TCNTR[12]	TCNTR[11]	TCNTR[10]	TCNTR[9]	TCNTR[8]
		TCNTR[7]	TCNTR[6]	TCNTR[5]	TCNTR[4]	TCNTR[3]	TCNTR[2]	TCNTR[1]	TCNTR[0]
	CYCTR_0	CYCTR[15]	CYCTR[14]	CYCTR[13]	CYCTR[12]	CYCTR[11]	CYCTR[10]	CYCTR[9]	CYCTR[8]
		CYCTR[7]	CYCTR[6]	CYCTR[5]	CYCTR[4]	CYCTR[3]	CYCTR[2]	CYCTR[1]	CYCTR[0]
	RFMK_0	RFMK[15]	RFMK[14]	RFMK[13]	RFMK[12]	RFMK[11]	RFMK[10]	RFMK[9]	RFMK[8]
		RFMK[7]	RFMK[6]	RFMK[5]	RFMK[4]	RFMK[3]	RFMK[2]	RFMK[1]	RFMK[0]
	TCMR0_0	TCMR0[15]	TCMR0[14]	TCMR0[13]	TCMR0[12]	TCMR0[11]	TCMR0[10]	TCMR0[9]	TCMR0[8]
		TCMR0[7]	TCMR0[6]	TCMR0[5]	TCMR0[4]	TCMR0[3]	TCMR0[2]	TCMR0[1]	TCMR0[0]
	TCMR1_0	TCMR1[15]	TCMR1[14]	TCMR1[13]	TCMR1[12]	TCMR1[11]	TCMR1[10]	TCMR1[9]	TCMR1[8]
		TCMR1[7]	TCMR1[6]	TCMR1[5]	TCMR1[4]	TCMR1[3]	TCMR1[2]	TCMR1[1]	TCMR1[0]
	TCMR2_0	TCMR2[15]	TCMR2[14]	TCMR2[13]	TCMR2[12]	TCMR2[11]	TCMR2[10]	TCMR2[9]	TCMR2[8]
		TCMR2[7]	TCMR2[6]	TCMR2[5]	TCMR2[4]	TCMR2[3]	TCMR2[2]	TCMR2[1]	TCMR2[0]
	TTTSEL_0	—	TTTSEL[14]	TTTSEL[13]	TTTSEL[12]	TTTSEL[11]	TTTSEL[10]	TTTSEL[9]	TTTSEL[8]
		—	—	—	—	—	—	—	—
	MBn_CONTROL0_H_0 (n = 0 to 31)*1	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]
		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]
	MBn_CONTROL0_H_0 (n = 0 to 31)*2	IDE	RTR	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]
		STDID[5]	STDID[4]	STDID[3]	STDID[2]	STDID[1]	STDID[0]	EXTID[17]	EXTID[16]

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Controller area network	MBn_CONTROL 0_L_0 (n = 0 to 31)	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]
		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]
	MBn_LAFM0_0 (n = 0 to 31)*1	—	STDID_ LAFM[10]	STDID_ LAFM[9]	STDID_ LAFM[8]	STDID_ LAFM[7]	STDID_ LAFM[6]	STDID_ LAFM[5]	STDID_ LAFM[4]
		STDID_ LAFM[3]	STDID_ LAFM[2]	STDID_ LAFM[1]	STDID_ LAFM[0]	—	IDE	EXTID_ LAFM[17]	EXTID_ LAFM[16]
	MBn_LAFM0_0 (n = 0 to 31)*2	IDE	—	—	STDID_ LAFM[10]	STDID_ LAFM[9]	STDID_ LAFM[8]	STDID_ LAFM[7]	STDID_ LAFM[6]
		STDID_ LAFM[5]	STDID_ LAFM[4]	STDID_ LAFM[3]	STDID_ LAFM[2]	STDID_ LAFM[1]	STDID_ LAFM[0]	EXTID_ LAFM[17]	EXTID_ LAFM[16]
	MBn_LAFM1_0 (n = 0 to 31)	EXTID_ LAFM[15]	EXTID_ LAFM[14]	EXTID _LAFM[13]	EXTID_ LAFM[12]	EXTID_ LAFM[11]	EXTID_ LAFM[10]	EXTID_ LAFM[9]	EXTID_ LAFM[8]
		EXTID_ LAFM[7]	EXTID_ LAFM[6]	EXTID_ LAFM[5]	EXTID_ LAFM[4]	EXTID_ LAFM[3]	EXTID_ LAFM[2]	EXTID_ LAFM[1]	EXTID_ LAFM [0]
	MBn_DATA_01_ 0 (n = 0 to 31)	MSG_DATA0	MSG_DATA0	MSG_DATA0	MSG_DATA0	MSG_DATA0	MSG_DATA0	MSG_DATA0	MSG_DATA0
		MSG_DATA1	MSG_DATA1	MSG_DATA1	MSG_DATA1	MSG_DATA1	MSG_DATA1	MSG_DATA1	MSG_DATA1
	MBn_DATA_23_ 0 (n = 0 to 31)	MSG_DATA2	MSG_DATA2	MSG_DATA2	MSG_DATA2	MSG_DATA2	MSG_DATA2	MSG_DATA2	MSG_DATA2
		MSG_DATA3	MSG_DATA3	MSG_DATA3	MSG_DATA3	MSG_DATA3	MSG_DATA3	MSG_DATA3	MSG_DATA3
	MBn_DATA_45_ 0 (n = 0 to 31)	MSG_DATA4	MSG_DATA4	MSG_DATA4	MSG_DATA4	MSG_DATA4	MSG_DATA4	MSG_DATA4	MSG_DATA4
		MSG_DATA5	MSG_DATA5	MSG_DATA5	MSG_DATA5	MSG_DATA5	MSG_DATA5	MSG_DATA5	MSG_DATA5
	MBn_DATA_67_ 0 (n = 0 to 31)	MSG_DATA6	MSG_DATA6	MSG_DATA6	MSG_DATA6	MSG_DATA6	MSG_DATA6	MSG_DATA6	MSG_DATA6
		MSG_DATA7	MSG_DATA7	MSG_DATA7	MSG_DATA7	MSG_DATA7	MSG_DATA7	MSG_DATA7	MSG_DATA7
	MBn_CONTROL 1_0 (n =0)	—	—	NMC	—	—	MBC[2]	MBC[1]	MBC[0]
		—	—	—	—	DLD[3]	DLD[2]	DLD[1]	DLD[0]
	MBn_CONTROL 1_0 (n = 0 to 31)	—	—	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]
		—	—	—	—	DLC[3]	DLC[2]	DLC[1]	DLC[0]
	MBn_TIMESTA MP_0 (n = 0 to 15, 30, 31)	TS15	TS14	TS13	TS12	TS11	TS10	TS9	TS8
		TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0
	MBn_TTT_0 (n = 24 to 30)	TTT15	TTT14	TTT13	TTT12	TTT11	TTT10	TTT9	TTT8
		TTT7	TTT6	TTT5	TTT4	TTT3	TTT2	TTT1	TTT0

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Controller area network	MBn_TTCONTROL_0 (n = 24 to 29	TTW[1]	TTW[0]	OFFSET[5]	OFFSET[4]	OFFSET[3]	OFFSET[2]	OFFSET[1]	OFFSET[0]
		—	—	—	—	—	REP_FACTOR[2]	REP_FACTOR[1]	REP_FACTOR[0]
	MCR_1	MCR15	MCR14	—	—	—	TST2	TST1	TST0
		MCR7	MCR6	MCR5	—	—	MCR2	MCR1	MCR0
	GSR_1	—	—	—	—	—	—	—	—
		—	—	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0
	BCR1_1	TSEG13	TSEG12	TSEG11	TSEG10	—	TSEG22	TSEG21	TSEG20
		—	—	SJW1	SJW0	—	—	—	BSP
	BCR0_1	—	—	—	—	—	—	—	—
		BRP7	BRP6	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0
	IRR_1	IRR15	IRR14	IRR13	IRR12	IRR11	IRR10	IRR9	IRR8
		IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0
	IMR_1	IMR15	IMR14	IMR13	IMR12	IMR11	IMR10	IMR9	IMR8
		IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0
	TEC_REC_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]
	TXPR1_1	TXPR1[15]	TXPR1[14]	TXPR1[13]	TXPR1[12]	TXPR1[11]	TXPR1[10]	TXPR1[9]	TXPR1[8]
		TXPR1[7]	TXPR1[6]	TXPR1[5]	TXPR1[4]	TXPR1[3]	TXPR1[2]	TXPR1[1]	TXPR1[0]
	TXPR0_1	TXPR0[15]	TXPR0[14]	TXPR0[13]	TXPR0[12]	TXPR0[11]	TXPR0[10]	TXPR0[9]	TXPR0[8]
		TXPR0[7]	TXPR0[6]	TXPR0[5]	TXPR0[4]	TXPR0[3]	TXPR0[2]	TXPR0[1]	—
	TXCR1_1	TXCR1[15]	TXCR1[14]	TXCR1[13]	TXCR1[12]	TXCR1[11]	TXCR1[10]	TXCR1[9]	TXCR1[8]
		TXCR1[7]	TXCR1[6]	TXCR1[5]	TXCR1[4]	TXCR1[3]	TXCR1[2]	TXCR1[1]	TXCR1[0]
	TXCR0_1	TXCR0[15]	TXCR0[14]	TXCR0[13]	TXCR0[12]	TXCR0[11]	TXCR0[10]	TXCR0[9]	TXCR0[8]
		TXCR0[7]	TXCR0[6]	TXCR0[5]	TXCR0[4]	TXCR0[3]	TXCR0[2]	TXCR0[1]	—
	TXACK1_1	TXACK1[15]	TXACK1[14]	TXACK1[13]	TXACK1[12]	TXACK1[11]	TXACK1[10]	TXACK1[9]	TXACK1[8]
		TXACK1[7]	TXACK1[6]	TXACK1[5]	TXACK1[4]	TXACK1[3]	TXACK1[2]	TXACK1[1]	TXACK1[0]
	TXACK0_1	TXACK0[15]	TXACK0[14]	TXACK0[13]	TXACK0[12]	TXACK0[11]	TXACK0[10]	TXACK0[9]	TXACK0[8]
		TXACK0[7]	TXACK0[6]	TXACK0[5]	TXACK0[4]	TXACK0[3]	TXACK0[2]	TXACK0[1]	—
	ABACK1_1	ABACK1[15]	ABACK1[14]	ABACK1[13]	ABACK1[12]	ABACK1[11]	ABACK1[10]	ABACK1[9]	ABACK1[8]
		ABACK1[7]	ABACK1[6]	ABACK1[5]	ABACK1[4]	ABACK1[3]	ABACK1[2]	ABACK1[1]	ABACK1[0]

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Controller area network	ABACK0_1	ABACK0[15]	ABACK0[14]	ABACK0[13]	ABACK0[12]	ABACK0[11]	ABACK0[10]	ABACK0[9]	ABACK0[8]
		ABACK0[7]	ABACK0[6]	ABACK0[5]	ABACK0[4]	ABACK0[3]	ABACK0[2]	ABACK0[1]	—
	RXPR1_1	RXPR1[15]	RXPR1[14]	RXPR1[13]	RXPR1[12]	RXPR1[11]	RXPR1[10]	RXPR1[9]	RXPR1[8]
		RXPR1[7]	RXPR1[6]	RXPR1[5]	RXPR1[4]	RXPR1[3]	RXPR1[2]	RXPR1[1]	RXPR1[0]
	RXPR0_1	RXPR0[15]	RXPR0[14]	RXPR0[13]	RXPR0[12]	RXPR0[11]	RXPR0[10]	RXPR0[9]	RXPR0[8]
		RXPR0[7]	RXPR0[6]	RXPR0[5]	RXPR0[4]	RXPR0[3]	RXPR0[2]	RXPR0[1]	RXPR0[0]
	RFPR1_1	RFPR1[15]	RFPR1[14]	RFPR1[13]	RFPR1[12]	RFPR1[11]	RFPR1[10]	RFPR1[9]	RFPR1[8]
		RFPR1[7]	RFPR1[6]	RFPR1[5]	RFPR1[4]	RFPR1[3]	RFPR1[2]	RFPR1[1]	RFPR1[0]
	RFPR0_1	RFPR0[15]	RFPR0[14]	RFPR0[13]	RFPR0[12]	RFPR0[11]	RFPR0[10]	RFPR0[9]	RFPR0[8]
		RFPR0[7]	RFPR0[6]	RFPR0[5]	RFPR0[4]	RFPR0[3]	RFPR0[2]	RFPR0[1]	RFPR0[0]
	MBIMR1_1	MBIMR1[15]	MBIMR1[14]	MBIMR1[13]	MBIMR1[12]	MBIMR1[11]	MBIMR1[10]	MBIMR1[9]	MBIMR1[8]
		MBIMR1[7]	MBIMR1[6]	MBIMR1[5]	MBIMR1[4]	MBIMR1[3]	MBIMR1[2]	MBIMR1 [1]	MBIMR1[0]
	MBIMR0_1	MBIMR0[15]	MBIMR0[14]	MBIMR0[13]	MBIMR0[12]	MBIMR0[11]	MBIMR0[10]	MBIMR0[9]	MBIMR0[8]
		MBIMR0[7]	MBIMR0[6]	MBIMR0[5]	MBIMR04]	MBIMR0[3]	MBIMR0[2]	MBIMR0[1]	MBIMR0[0]
	UMSR1_1	UMSR1[15]	UMSR1[14]	UMSR1[13]	UMSR1[12]	UMSR1[11]	UMSR1[10]	UMSR1[9]	UMSR1[8]
		UMSR1[7]	UMSR1[6]	UMSR1[5]	UMSR1[4]	UMSR1[3]	UMSR1[2]	UMSR1[1]	UMSR1[0]
	UMSR0_1	UMSR0[15]	UMSR0[14]	UMSR0[13]	UMSR0[12]	UMSR0[11]	UMSR0[10]	UMSR0[9]	UMSR0[8]
		UMSR0[7]	UMSR0[6]	UMSR0[5]	UMSR0[4]	UMSR0[3]	UMSR0[2]	UMSR0[1]	UMSR0[0]
	TTCR0_1	TCR[15]	TCR[14]	TCR[13]	TCR[12]	TCR[11]	TCR[10]	—	—
		—	TCR[6]	TPSC[5]	TPSC[4]	TPSC[3]	TPSC[2]	TPSC[1]	TPSC[0]
	CMAX_TEW_1	—	—	—	—	—	CMAX[2]	CMAX[1]	CMAX[0]
		—	—	—	TEW[4]	TEW[3]	TEW[2]	TEW[1]	TEW[0]
	RFTROFF_1	RTROFF[7]	RTROFF[6]	RTROFF[5]	RTROFF[4]	RTROFF[3]	RTROFF[2]	RTROFF[1]	RTROFF[0]
		—	—	—	—	—	—	—	—
	TSR_1	—	—	—	—	—	—	—	—
		—	—	—	TSR[4]	TSR[3]	TSR[2]	TSR[1]	TSR[0]
	CCR_1	—	—	—	—	—	—	—	—
		—	—	CCR[5]	CCR[4]	CCR[3]	CCR[2]	CCR[1]	CCR[0]
	TCNTR_1	TCNTR[15]	TCNTR[14]	TCNTR[13]	TCNTR[12]	TCNTR[11]	TCNTR[10]	TCNTR[9]	TCNTR[8]
		TCNTR[7]	TCNTR[6]	TCNTR[5]	TCNTR[4]	TCNTR[3]	TCNTR[2]	TCNTR[1]	TCNTR[0]

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Controller area network	CYCTR_1	CYCTR[15]	CYCTR[14]	CYCTR[13]	CYCTR[12]	CYCTR[11]	CYCTR[10]	CYCTR[9]	CYCTR[8]
		CYCTR[7]	CYCTR[6]	CYCTR[5]	CYCTR[4]	CYCTR[3]	CYCTR[2]	CYCTR[1]	CYCTR[0]
	RFMK_1	RFMK[15]	RFMK[14]	RFMK[13]	RFMK[12]	RFMK[11]	RFMK[10]	RFMK[9]	RFMK[8]
		RFMK[7]	RFMK[6]	RFMK[5]	RFMK[4]	RFMK[3]	RFMK[2]	RFMK[1]	RFMK[0]
	TCMR0_1	TCMR0[15]	TCMR0[14]	TCMR0[13]	TCMR0[12]	TCMR0[11]	TCMR0[10]	TCMR0[9]	TCMR0[8]
		TCMR0[7]	TCMR0[6]	TCMR0[5]	TCMR0[4]	TCMR0[3]	TCMR0[2]	TCMR0[1]	TCMR0[0]
	TCMR1_1	TCMR1[15]	TCMR1[14]	TCMR1[13]	TCMR1[12]	TCMR1[11]	TCMR1[10]	TCMR1[9]	TCMR1[8]
		TCMR1[7]	TCMR1[6]	TCMR1[5]	TCMR1[4]	TCMR1[3]	TCMR1[2]	TCMR1[1]	TCMR1[0]
	TCMR2_1	TCMR2[15]	TCMR2[14]	TCMR2[13]	TCMR2[12]	TCMR2[11]	TCMR2[10]	TCMR2[9]	TCMR2[8]
		TCMR2[7]	TCMR2[6]	TCMR2[5]	TCMR2[4]	TCMR2[3]	TCMR2[2]	TCMR2[1]	TCMR2[0]
	TTTSEL_1	—	TTTSEL[14]	TTTSEL[13]	TTTSEL[12]	TTTSEL[11]	TTTSEL[10]	TTTSEL[9]	TTTSEL[8]
		—	—	—	—	—	—	—	—
	MBn_CONTROL0_H_1 (n = 0 to 31)*1	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]
		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]
	MBn_CONTROL0_H_1 (n = 0 to 31)*2	IDE	RTR	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]
		STDID[5]	STDID[4]	STDID[3]	STDID[2]	STDID[1]	STDID[0]	EXTID[17]	EXTID[16]
	MBn_CONTROL0_L_1 (n = 0 to 31)	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]
		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]
	MBn_LAFM0_1 (n = 0 to 31)*1	—	STDID_LAFM[10]	STDID_LAFM[9]	STDID_LAFM[8]	STDID_LAFM[7]	STDID_LAFM[6]	STDID_LAFM[5]	STDID_LAFM[4]
		STDID_LAFM[3]	STDID_LAFM[2]	STDID_LAFM[1]	STDID_LAFM[0]	—	IDE	EXTID_LAFM[17]	EXTID_LAFM[16]
	MBn_LAFM0_1 (n = 0 to 31)*2	IDE	—	—	STDID_LAFM[10]	STDID_LAFM[9]	STDID_LAFM[8]	STDID_LAFM[7]	STDID_LAFM[6]
		STDID_LAFM[5]	STDID_LAFM[4]	STDID_LAFM[3]	STDID_LAFM[2]	STDID_LAFM[1]	STDID_LAFM[0]	EXTID_LAFM[17]	EXTID_LAFM[16]
	MBn_LAFM1_1 (n = 0 to 31)	EXTID_LAFM[15]	EXTID_LAFM[14]	EXTID_LAFM[13]	EXTID_LAFM[12]	EXTID_LAFM[11]	EXTID_LAFM[10]	EXTID_LAFM[9]	EXTID_LAFM[8]
		EXTID_LAFM[7]	EXTID_LAFM[6]	EXTID_LAFM[5]	EXTID_LAFM[4]	EXTID_LAFM[3]	EXTID_LAFM[2]	EXTID_LAFM[1]	EXTID_LAFM[0]

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Controller area network	MBn_DATA_01_1(n = 0 to 31)	MSG_DATA0	MSG_DATA0	MSG_DATA0	MSG_DATA0	MSG_DATA0	MSG_DATA0	MSG_DATA0	MSG_DATA0
		MSG_DATA1	MSG_DATA1	MSG_DATA1	MSG_DATA1	MSG_DATA1	MSG_DATA1	MSG_DATA1	MSG_DATA1
	MBn_DATA_23_1 (n = 0 to 31)	MSG_DATA2	MSG_DATA2	MSG_DATA2	MSG_DATA2	MSG_DATA2	MSG_DATA2	MSG_DATA2	MSG_DATA2
		MSG_DATA3	MSG_DATA3	MSG_DATA3	MSG_DATA3	MSG_DATA3	MSG_DATA3	MSG_DATA3	MSG_DATA3
	MBn_DATA_45_1 (n = 0 to 31)	MSG_DATA4	MSG_DATA4	MSG_DATA4	MSG_DATA4	MSG_DATA4	MSG_DATA4	MSG_DATA4	MSG_DATA4
		MSG_DATA5	MSG_DATA5	MSG_DATA5	MSG_DATA5	MSG_DATA5	MSG_DATA5	MSG_DATA5	MSG_DATA5
	MBn_DATA_67_1 (n = 0 to 31)	MSG_DATA6	MSG_DATA6	MSG_DATA6	MSG_DATA6	MSG_DATA6	MSG_DATA6	MSG_DATA6	MSG_DATA6
		MSG_DATA7	MSG_DATA7	MSG_DATA7	MSG_DATA7	MSG_DATA7	MSG_DATA7	MSG_DATA7	MSG_DATA7
	MBn_CONTROL1_1(n = 0)	—	—	NMC	—	—	MBC[2]	MBC[1]	MBC[0]
		—	—	—	—	DLD[3]	DLD[2]	DLD[1]	DLD[0]
	MBn_CONTROL1_1 (n = 0 to 31)	—	—	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]
		—	—	—	—	DLD[3]	DLD[2]	DLD[1]	DLD[0]
	MBn_TIMESTAMP_1 (n = 0 to 15, 30, 31)	TS15	TS14	TS13	TS12	TS11	TS10	TS9	TS8
		TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0
	MBn_TTT_1 (n = 24 to 30)	TTT15	TTT14	TTT13	TTT12	TTT11	TTT10	TTT9	TTT8
		TTT7	TTT6	TTT5	TTT4	TTT3	TTT2	TTT1	TTT0
	MBn_TTCONTROL_1 (n = 24 to 29)	TTW[1]	TTW[0]	OFFSET[5]	OFFSET[4]	OFFSET[3]	OFFSET[2]	OFFSET[1]	OFFSET[0]
		—	—	—	—	—	REP_FACTOR[2]	REP_FACTOR[1]	REP_FACTOR[0]
IEBus controller	IECTR	—	IOL	DEE	—	RE	—	—	—
	IECMR	—	—	—	—	—	CMD[2]	CMD[1]	CMD[0]
	IEMCR	SS	RN[2]	RN[1]	RN[0]	CTL[3]	CTL[2]	CTL[1]	CTL[0]
	IEAR1	IARL4[3]	IARL4[2]	IARL4[1]	IARL4[0]	IMD[1]	IMD[0]	—	STE
	IEAR2	IARU8[7]	IARU8[6]	IARU8[5]	IARU8[4]	IARU8[3]	IARU8[2]	IARU8[1]	IARU8[0]
	IESA1	ISAL4[3]	ISAL4[2]	ISAL4[1]	ISAL4[0]	—	—	—	—
	IESA2	ISAU8[7]	ISAU8[6]	ISAU8[5]	ISAU8[4]	ISAU8[3]	ISAU8[2]	ISAU8[1]	ISAU8[0]
	IETBFL	IBFL[7]	IBFL[6]	IBFL[5]	IBFL[4]	IBFL[3]	IBFL[2]	IBFL[1]	IBFL[0]
	IEMA1	IMAL4[3]	IMAL4[2]	IMAL4[1]	IMAL4[0]	—	—	—	—
	IEMA2	IMAU8[7]	IMAU8[6]	IMAU8[5]	IMAU8[4]	IMAU8[3]	IMAU8[2]	IMAU8[1]	IMAU8[0]
	IERCTL	—	—	—	—	RCTL[3]	RCTL[2]	RCTL[1]	RCTL[0]

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
IEBus controller	IERBFL	RBFL[7]	RBFL[6]	RBFL[5]	RBFL[4]	RBFL[3]	RBFL[2]	RBFL[1]	RBFL[0]
	IELA1	ILAL8[7]	ILAL8[6]	ILAL8[5]	ILAL8[4]	ILAL8[3]	ILAL8[2]	ILAL8[1]	ILAL8[0]
	IELA2	—	—	—	—	ILAU4[3]	ILAU4[2]	ILAU4[1]	ILAU4[0]
	IEFLG	CMX	MRQ	SRQ	SRE	LCK	—	RSS	GG
	IETSR	—	TXS	TXF	—	TXEAL	TXETTME	TXERO	TXEACK
	IEIET	—	TXSE	TXFE	—	TXEALE	TXETTMEE	TXEROE	TXEACKE
	IERSR	RXBSY	RXS	RXF	RXEDE	RXEOVE	RXERTME	RXEDLE	RXEPE
	IEIER	RXBSYE	RXSE	RXFE	RXEDEE	RXEOVEE	RXERTMEE	RXEDLEE	RXEPEE
	IECKSR	—	—	—	CKS3	—	CKS[2]	CKS[1]	CKS[0]
	IETB001 to IETB128								
	IERB001 to IERB128								
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]	—

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Renesas SPDIF interface	TUI								
	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

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Renesas SPDIF interface	TDAD	—	—	—	—	—	—	—	—
	RDAD	—	—	—	—	—	—	—	—
CD-ROM decoder	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]
	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]

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
CD-ROM decoder	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_LINK	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	PREINH REQDM	PREINHI READY
	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]
A/D converter	ADDRA								
				—	—	—	—	—	—
	ADDRB								
				—	—	—	—	—	—
	ADDRC								
				—	—	—	—	—	—
	ADDRD								
				—	—	—	—	—	—
	ADDRE								
				—	—	—	—	—	—

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
A/D converter	ADDRF								
				—	—	—	—	—	—
	ADDRG								
				—	—	—	—	—	—
	ADDRH								
				—	—	—	—	—	—
	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]
NAND flash memory controller	FLCMNCR	—	—	—	—	—	—	ECCPOS[2]	4ECCCNTEN
		4ECCEN	4ECCCORRECT	BUSYON	—	—	SNAND	QTSEL	—
		—	—	ECCPOS[1]	ECCPOS[0]	ACM[1]	ACM[0]	NANDWF	—
		—	—	—	—	CE	—	—	—
	FLCMDCR	ADRCNT2	SCTCNT[19]	SCTCNT[18]	SCTCNT[17]	SCTCNT[16]	ADRMD	CDSRC	DOSR
		—	—	SELRW	DOADR	ADRCNT[1]	ADRCNT[0]	DOCMD2	DOCMD1
		SCTCNT[15]	SCTCNT[14]	SCTCNT[13]	SCTCNT[12]	SCTCNT[11]	SCTCNT[10]	SCTCNT[9]	SCTCNT[8]
		SCTCNT[7]	SCTCNT[6]	SCTCNT[5]	SCTCNT[4]	SCTCNT[3]	SCTCNT[2]	SCTCNT[1]	SCTCNT[0]
	FLCMCDR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		CMD2[7]	CMD2[6]	CMD2[5]	CMD2[4]	CMD2[3]	CMD2[2]	CMD2[1]	CMD2[0]
		CMD1[7]	CMD1[6]	CMD1[5]	CMD1[4]	CMD1[3]	CMD1[2]	CMD1[1]	CMD1[0]
	FLADR*3	ADR4[7]	ADR4[6]	ADR4[5]	ADR4[4]	ADR4[3]	ADR4[2]	ADR4[1]	ADR4[0]
		ADR3[7]	ADR3[6]	ADR3[5]	ADR3[4]	ADR3[3]	ADR3[2]	ADR3[1]	ADR3[0]
		ADR2[7]	ADR2[6]	ADR2[5]	ADR2[4]	ADR2[3]	ADR2[2]	ADR2[1]	ADR2[0]
		ADR1[7]	ADR1[6]	ADR1[5]	ADR1[4]	ADR1[3]	ADR1[2]	ADR1[1]	ADR1[0]
	FLADR*4	—	—	—	—	—	—	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]

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
NAND flash memory controller	FLADR2	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		ADR5[7]	ADR5[6]	ADR5[5]	ADR5[4]	ADR5[3]	ADR5[2]	ADR5[1]	ADR5[0]
	FLDTCNTR	ECFLW[7]	ECFLW[6]	ECFLW[5]	ECFLW[4]	ECFLW[3]	ECFLW[2]	ECFLW[1]	ECFLW[0]
		DTFLW[7]	DTFLW[6]	DTFLW[5]	DTFLW[4]	DTFLW[3]	DTFLW[2]	DTFLW[1]	DTFLW[0]
		—	—	—	—	DTCNT[11]	DTCNT[10]	DTCNT[9]	DTCNT[8]
		DTCNT[7]	DTCNT[6]	DTCNT[5]	DTCNT[4]	DTCNT[3]	DTCNT[2]	DTCNT[1]	DTCNT[0]
	FLDATAR	DT4[7]	DT4[6]	DT4[5]	DT4[4]	DT4[3]	DT4[2]	DT4[1]	DT4[0]
		DT3[7]	DT3[6]	DT3[5]	DT3[4]	DT3[3]	DT3[2]	DT3[1]	DT3[0]
		DT2[7]	DT2[6]	DT2[5]	DT2[4]	DT2[3]	DT2[2]	DT2[1]	DT2[0]
		DT1[7]	DT1[6]	DT1[5]	DT1[4]	DT1[3]	DT1[2]	DT1[1]	DT1[0]
	FLINTDMACR	—	—	—	—	—	—	4ECEINTE	ECERINTE
		—	—	FIFOTRG[1]	FIFOTRG[0]	AC1CLR	AC0CLR	DREQ1EN	DREQ0EN
		—	—	—	—	—	—	ECERB	STERB
		BTOERB	TRREQF1	TRREQF0	STERINTE	RBERINTE	TEINTE	TRINTE1	TRINTE0
	FLBSYTMR	—	—	—	—	—	—	—	—
		—	—	—	—	RBTMOUT [19]	RBTMOUT [18]	RBTMOUT [17]	RBTMOUT [16]
		RBTMOUT [15]	RBTMOUT [14]	RBTMOUT [13]	RBTMOUT [12]	RBTMOUT [11]	RBTMOUT [10]	RBTMOUT[9]	RBTMOUT[8]
		RBTMOUT[7]	RBTMOUT[6]	RBTMOUT[5]	RBTMOUT[4]	RBTMOUT[3]	RBTMOUT[2]	RBTMOUT[1]	RBTMOUT[0]
	FLBSYCNT	STAT[7]	STAT[6]	STAT[5]	STAT[4]	STAT[3]	STAT[2]	STAT[1]	STAT[0]
		—	—	—	—	RBTIMCNT [19]	RBTIMCNT [18]	RBTIMCNT [17]	RBTIMCNT [16]
		RBTIMCNT [15]	RBTIMCNT [14]	RBTIMCNT [13]	RBTIMCNT [12]	RBTIMCNT [11]	RBTIMCNT [10]	RBTIMCNT[9]	RBTIMCNT[8]
		RBTIMCNT[7]	RBTIMCNT[6]	RBTIMCNT[5]	RBTIMCNT[4]	RBTIMCNT[3]	RBTIMCNT[2]	RBTIMCNT[1]	RBTIMCNT[0]
	FLDTFIFO	DTFO[31]	DTFO[30]	DTFO[29]	DTFO[28]	DTFO[27]	DTFO[26]	DTFO[25]	DTFO[24]
		DTFO[23]	DTFO[22]	DTFO[21]	DTFO[20]	DTFO[19]	DTFO[18]	DTFO[17]	DTFO[16]
		DTFO[15]	DTFO[14]	DTFO[13]	DTFO[12]	DTFO[11]	DTFO[10]	DTFO[9]	DTFO[8]
		DTFO[7]	DTFO[6]	DTFO[5]	DTFO[4]	DTFO[3]	DTFO[2]	DTFO[1]	DTFO[0]

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
NAND flash memory controller	FLECFIFO	ECFO[31]	ECFO[30]	ECFO[29]	ECFO[28]	ECFO[27]	ECFO[26]	ECFO[25]	ECFO[24]
		ECFO[23]	ECFO[22]	ECFO[21]	ECFO[20]	ECFO[19]	ECFO[18]	ECFO[17]	ECFO[16]
		ECFO[15]	ECFO[14]	ECFO[13]	ECFO[12]	ECFO[11]	ECFO[10]	ECFO[9]	ECFO[8]
		ECFO[7]	ECFO[6]	ECFO[5]	ECFO[4]	ECFO[3]	ECFO[2]	ECFO[1]	ECFO[0]
	FLTRCR	—	—	—	—	—	TRSTAT	TREND	TRSTRT
	FLHOLDCR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	HOLDEN
	FL4ECCRES1	—	—	—	—	—	—	LOC1[9]	LOC1[8]
		LOC1[7]	LOC1[6]	LOC1[5]	LOC1[4]	LOC1[3]	LOC1[2]	LOC1[1]	LOC1[0]
		—	—	—	—	—	—	PAT1[9]	PAT1[8]
		PAT1[7]	PAT1[6]	PAT1[5]	PAT1[4]	PAT1[3]	PAT1[2]	PAT1[1]	PAT1[0]
	FL4ECCRES2	—	—	—	—	—	—	LOC2[9]	LOC2[8]
		LOC2[7]	LOC2[6]	LOC2[5]	LOC2[4]	LOC2[3]	LOC2[2]	LOC2[1]	LOC2[0]
		—	—	—	—	—	—	PAT2[9]	PAT2[8]
		PAT2[7]	PAT2[6]	PAT2[5]	PAT2[4]	PAT2[3]	PAT2[2]	PAT2[1]	PAT2[0]
	FL4ECCRES3	—	—	—	—	—	—	LOC3[9]	LOC3[8]
		LOC3[7]	LOC3[6]	LOC3[5]	LOC3[4]	LOC3[3]	LOC3[2]	LOC3[1]	LOC3[0]
		—	—	—	—	—	—	PAT3[9]	PAT3[8]
		PAT3[7]	PAT3[6]	PAT3[5]	PAT3[4]	PAT3[3]	PAT3[2]	PAT3[1]	PAT3[0]
	FL4ECCRES4	—	—	—	—	—	—	LOC4[9]	LOC4[8]
		LOC4[7]	LOC4[6]	LOC4[5]	LOC4[4]	LOC4[3]	LOC4[2]	LOC4[1]	LOC4[0]
		—	—	—	—	—	—	PAT4[9]	PAT4[8]
		PAT4[7]	PAT4[6]	PAT4[5]	PAT4[4]	PAT4[3]	PAT4[2]	PAT4[1]	PAT4[0]
	FL4ECCCR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	4ECCFA	4ECCEND	4ECCEXST

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
NAND flash memory controller	FL4ECCCNT	—	—	—	—	—	ERRCNT[10]	ERRCNT[9]	ERRCNT[8]
		ERRCNT[7]	ERRCNT[6]	ERRCNT[5]	ERRCNT[4]	ERRCNT[3]	ERRCNT[2]	ERRCNT[1]	ERRCNT[0]
		—	—	—	—	—	—	—	—
		—	—	—	—	—	ERRMAX[2]	ERRMAX[1]	ERRMAX[0]
USB 2.0 host/function module	SYSCFG	—	—	—	—	—	SCKE	—	—
		HSE	DCFM	DMRPD	DPRPU	—	—	—	USBE
	BUSWAIT		—	—	—	—	—	—	—
		—	—	—	—	BWAIT[3]	BWAIT[2]	BWAIT[1]	BWAIT[0]
	SYSSTS	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	LNST[1]	LNST[0]
	DVSTCTR	—	—	—	—	—	—	—	WKUP
		RWUPE	USBRST	RESUME	UACT	—	RHST[2]	RHST[1]	RHST[0]
	TESTMODE	—	—	—	—	—	—	—	—
		—	—	—	—	UTST[3]	UTST[2]	UTST[1]	UTST[0]
	D0FBCFG	—	—	—	—	—	—	—	—
		—	—	—	TENDE	—	—	—	—
	D1FBCFG	—	—	—	—	—	—	—	—
		—	—	—	TENDE	—	—	—	—
	CFIFO	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	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

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
USB 2.0 host/function module	D1FIFO	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	RCNT	REW	—	—	MBW[1]	MBW[0]	—	BIGEND
		—	—	ISEL	—	CURPIPE[3]	CURPIPE[2]	CURPIPE[1]	CURPIPE[0]
	CFIFOCTR	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	RCNT	REW	DCLRM	DREQE	MBW[1]	MBW[0]	—	BIGEND
		—	—	—	—	CURPIPE[3]	CURPIPE[2]	CURPIPE[1]	CURPIPE[0]
	D0FIFOCTR	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	RCNT	REW	DCLRM	DREQE	MBW[1]	MBW[0]	—	BIGEND
		—	—	—	—	CURPIPE[3]	CURPIPE[2]	CURPIPE[1]	CURPIPE[0]
	D1FIFOCTR	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	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE
		—	—	—	—	—	—	—	—
	INTENB1	—	BCHGE	—	DTCHE	ATTCHE	—	—	—
		—	EOFEPRE	SIGNE	SACKE	—	—	—	—
	BRDYENB	—	—	—	—	—	—	PIPE9BRDYE	PIPE8BRDYE
		PIPE7BRDYE	PIPE6BRDYE	PIPE5BRDYE	PIPE4BRDYE	PIPE3BRDYE	PIPE2BRDYE	PIPE1BRDYE	PIPE0BRDYE
	NRDYENB	—	—	—	—	—	—	PIPE9NRDYE	PIPE8NRDYE
		PIPE7NRDYE	PIPE6NRDYE	PIPE5NRDYE	PIPE4NRDYE	PIPE3NRDYE	PIPE2NRDYE	PIPE1NRDYE	PIPE0NRDYE
	BEMPENB	—	—	—	—	—	—	PIPE9BEMPE	PIPE8BEMPE
		PIPE7BEMPE	PIPE6BEMPE	PIPE5BEMPE	PIPE4BEMPE	PIPE3BEMPE	PIPE2BEMPE	PIPE1BEMPE	PIPE0BEMPE
	SOFCFG	—	—	—	—	—	—	—	TRNENSEL
		—	BRDYM	—	—	—	—	—	—

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
USB 2.0 host/function module	INTSTS0	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY
		VBSTS	DVSQ[2]	DVSQ[1]	DVSQ[0]	VALID	CTSQ[2]	CTSQ[1]	CTSQ[0]
	INTSTS1	—	BCHG	—	DTCH	ATTCH	—	—	—
		—	EOFEPR	SIGN	SACK	—	—	—	—
	BRDYSTS	—	—	—	—	—	—	PIPE9BRDY	PIPE8BRDY
		PIPE7BRDY	PIPE6BRDY	PIPE5BRDY	PIPE4BRDY	PIPE3BRDY	PIPE2BRDY	PIPE1BRDY	PIPE0BRDY
	NRDYSTS	—	—	—	—	—	—	PIPE9NRDY	PIPE8NRDY
		PIPE7NRDY	PIPE6NRDY	PIPE5NRDY	PIPE4NRDY	PIPE3NRDY	PIPE2NRDY	PIPE1NRDY	PIPE0NRDY
	BEMPSTS	—	—	—	—	—	—	PIPE9BEMP	PIPE8BEMP
		PIPE7BEMP	PIPE6BEMP	PIPE5BEMP	PIPE4BEMP	PIPE3BEMP	PIPE2BEMP	PIPE1BEMP	PIPE0BEMP
	FRMNUM	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	—	—	—	—	—	—	—	—
		—	—	—	—	—	UFRNM[2]	UFRNM[1]	UFRNM[0]
	USBADDR	—	—	—	—	—	—	—	—
		—	USBADDR[6]	USBADDR[5]	USBADDR[4]	USBADDR[3]	USBADDR[2]	USBADDR[1]	USBADDR[0]
	USBREQ	BREQUEST [7]	BREQUEST [6]	BREQUEST [5]	BREQUEST [4]	BREQUEST [3]	BREQUEST [2]	BREQUEST [1]	BREQUEST [0]
		BMREQUEST TYPE[7]	BMREQUEST TYPE[6]	BMREQUEST TYPE[5]	BMREQUEST TYPE[4]	BMREQUEST TYPE[3]	BMREQUEST TYPE[2]	BMREQUEST TYPE[1]	BMREQUEST TYPE[0]
	USBVAL	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	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	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	—	—	—	—	—	—	—	—
		—	—	—	DIR	—	—	—	—
	DCPMAXP	DEVSEL[3]	DEVSEL[2]	DEVSEL[1]	DEVSEL[0]	—	—	—	—
		—	MXPS[6]	MXPS[5]	MXPS[4]	MXPS[3]	MXPS[2]	MXPS[1]	MXPS[0]

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
USB 2.0 host/function module	DCPCTR	BSTS	SUREQ	CSCLR	CSSTS	SUREQCLR	—	—	SQCLR
		SQSET	SQMON	PBUSY	PINGE	—	CCPL	PID[1]	PID[0]
	PIPESEL	—	—	—	—	—	—	—	—
		—	—	—	—	PIPESEL[3]	PIPESEL[2]	PIPESEL[1]	PIPESEL[0]
	PIPECFG	TYPE[1]	TYPE[0]	—	—	—	BFRE	DBLB	CNTMD
		SHTNAK	—	—	DIR	EPNUM[3]	EPNUM[2]	EPNUM[1]	EPNUM[0]
	PIPEBUF	—	BUFSIZE[4]	BUFSIZE[3]	BUFSIZE[2]	BUFSIZE[1]	BUFSIZE[0]	—	—
		—	BUFNMB[6]	BUFNMB[5]	BUFNMB[4]	BUFNMB[3]	BUFNMB[2]	BUFNMB[1]	BUFNMB[0]
	PIPEMAXP	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	—	—	—	IFIS	—	—	—	—
		—	—	—	—	—	IITV[2]	IITV[1]	IITV[0]
	PIPE1CTR	BSTS	INBUFM	CSCLR	CSSTS	—	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1]	PID[0]
	PIPE2CTR	BSTS	INBUFM	CSCLR	CSSTS	—	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1]	PID[0]
	PIPE3CTR	BSTS	INBUFM	CSCLR	CSSTS	—	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1]	PID[0]
	PIPE4CTR	BSTS	INBUFM	CSCLR	CSSTS	—	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1]	PID[0]
	PIPE5CTR	BSTS	INBUFM	CSCLR	CSSTS	—	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1]	PID[0]
	PIPE6CTR	BSTS	—	CSCLR	CSSTS	—	—	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1]	PID[0]
	PIPE7CTR	BSTS	—	CSCLR	CSSTS	—	—	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1]	PID[0]
	PIPE8CTR	BSTS	—	CSCLR	CSSTS	—	—	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1]	PID[0]
	PIPE9CTR	BSTS	—	CSCLR	CSSTS	—	—	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1]	PID[0]

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
USB 2.0 host/function module	PIPE1TRE	—	—	—	—	—	—	TRENB	TRCLR
		—	—	—	—	—	—	—	—
	PIPE1TRN	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	—	—	—	—	—	—	TRENB	TRCLR
		—	—	—	—	—	—	—	—
	PIPE2TRN	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	—	—	—	—	—	—	TRENB	TRCLR
		—	—	—	—	—	—	—	—
	IPE3TRN	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	—	—	—	—	—	—	TRENB	TRCLR
		—	—	—	—	—	—	—	—
	PIPE4TRN	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	—	—	—	—	—	—	TRENB	TRCLR
		—	—	—	—	—	—	—	—
	PIPE5TRN	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]
	USBACSWR1	—	—	—	—	—	—	UACS25	—
		—	—	—	—	—	—	—	—
	DEVADD0	—	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	—	—	—	—	—	—
	DEVADD1	—	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	—	—	—	—	—	—
	DEVADD2	—	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	—	—	—	—	—	—
	DEVADD3	—	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	—	—	—	—	—	—
	DEVADD4	—	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	—	—	—	—	—	—

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
USB 2.0 host/function module	DEVADD5	—	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	—	—	—	—	—	—
	DEVADD6	—	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	—	—	—	—	—	—
	DEVADD7	—	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	—	—	—	—	—	—
	DEVADD8	—	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	—	—	—	—	—	—
	DEVADD9	—	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	—	—	—	—	—	—
	DEVADDA	—	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	—	—	—	—	—	—
Video display controller 3	VIDEO_MODE	—	—	RGB565	INV_CbCr	—	—	—	—
		—	—	BURST_ MODE_DISP	BURST_ MODE_MAIN	—	—	ENDIAN_ DISP	ENDIAN_ MAIN
		—	SEL_ EXSYNC	SEL_656601	SEL_525625	—	—	—	—
		—	—	—	VIDEO_ MODE	—	—	VIDEO_ DISP_EXE	VIDEO_ MAIN_EXE
	VIDEO_INT_ CNT	—	—	—	INT_V_EN	—	—	—	INT_F_EN
		—	—	—	INT_UF_EN	—	—	—	INT_OF_EN
		—	—	—	V_PERIOD	—	—	—	F_END
		—	—	—	UNDER_ FLOW	—	—	—	OVER_FLOW
	VIDEO_TIM_ CNT	—	—	—	—	—	—	FIELD_SKEW [9]	FIELD_SKEW [8]
		FIELD_SKEW [7]	FIELD_SKEW [6]	FIELD_SKEW [5]	FIELD_SKEW [4]	FIELD_SKEW [3]	FIELD_SKEW [2]	FIELD_SKEW [1]	FIELD_SKEW [0]
		—	—	—	—	—	—	—	—
		—	—	VSYNC_TYP	HSYNC_TYP	—	VSYNC_TIM	HSYNC_TIM	VIDEO_TIM

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Video display controller 3	VIDEO_SIZE	—	—	—	—	—	—	—	VIDEO_HEIGHT[8]
		VIDEO_HEIGHT[7]	VIDEO_HEIGHT[6]	VIDEO_HEIGHT[5]	VIDEO_HEIGHT[4]	VIDEO_HEIGHT[3]	VIDEO_HEIGHT[2]	VIDEO_HEIGHT[1]	VIDEO_HEIGHT[0]
		—	—	—	—	—	—	VIDEO_WIDTH[9]	VIDEO_WIDTH[8]
		VIDEO_WIDTH[7]	VIDEO_WIDTH[6]	VIDEO_WIDTH[5]	VIDEO_WIDTH[4]	VIDEO_WIDTH[3]	VIDEO_WIDTH[2]	VIDEO_WIDTH[1]	VIDEO_WIDTH[0]
	VIDEO_VSTART	—	—	—	—	—	—	—	VIDEO_VSTART_TOP[8]
		VIDEO_VSTART_TOP[7]	VIDEO_VSTART_TOP[6]	VIDEO_VSTART_TOP[5]	VIDEO_VSTART_TOP[4]	VIDEO_VSTART_TOP[3]	VIDEO_VSTART_TOP[2]	VIDEO_VSTART_TOP[1]	VIDEO_VSTART_TOP[0]
		—	—	—	—	—	—	—	VIDEO_VSTART_BTM[8]
		VIDEO_VSTART_BTM[7]	VIDEO_VSTART_BTM[6]	VIDEO_VSTART_BTM[5]	VIDEO_VSTART_BTM[4]	VIDEO_VSTART_BTM[3]	VIDEO_VSTART_BTM[2]	VIDEO_VSTART_BTM[1]	VIDEO_VSTART_BTM[0]
	VIDEO_HSTART	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	VIDEO_HSTART[8]
		VIDEO_HSTART[7]	VIDEO_HSTART[6]	VIDEO_HSTART[5]	VIDEO_HSTART[4]	VIDEO_HSTART[3]	VIDEO_HSTART[2]	VIDEO_HSTART[1]	VIDEO_HSTART[0]
	VIDEO_VSYNC_TIM1	—	—	—	—	—	—	VIDEO_VSYNC_START1_TOP[9]	VIDEO_VSYNC_START1_TOP[8]
		VIDEO_VSYNC_START1_TOP[7]	VIDEO_VSYNC_START1_TOP[6]	VIDEO_VSYNC_START1_TOP[5]	VIDEO_VSYNC_START1_TOP[4]	VIDEO_VSYNC_START1_TOP[3]	VIDEO_VSYNC_START1_TOP[2]	VIDEO_VSYNC_START1_TOP[1]	VIDEO_VSYNC_START1_TOP[0]
		—	—	—	—	—	—	VIDEO_VSYNC_START1_BTM[9]	VIDEO_VSYNC_START1_BTM[8]
		VIDEO_VSYNC_START1_BTM[7]	VIDEO_VSYNC_START1_BTM[6]	VIDEO_VSYNC_START1_BTM[5]	VIDEO_VSYNC_START1_BTM[4]	VIDEO_VSYNC_START1_BTM[3]	VIDEO_VSYNC_START1_BTM[2]	VIDEO_VSYNC_START1_BTM[1]	VIDEO_VSYNC_START1_BTM[0]

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Video display controller 3	VIDEO_SAVE_NUM	—	—	—	—	—	—	FIELD_NUM[9]	FIELD_NUM[8]
		FIELD_NUM[7]	FIELD_NUM[6]	FIELD_NUM[5]	FIELD_NUM[4]	FIELD_NUM[3]	FIELD_NUM[2]	FIELD_NUM[1]	FIELD_NUM[0]
		—	—	—	—	—	—	FIELD_SAVE_NUM[9]	FIELD_SAVE_NUM[8]
		FIELD_SAVE_NUM[7]	FIELD_SAVE_NUM[6]	FIELD_SAVE_NUM[5]	FIELD_SAVE_NUM[4]	FIELD_SAVE_NUM[3]	FIELD_SAVE_NUM[2]	FIELD_SAVE_NUM[1]	FIELD_SAVE_NUM[0]
	VIDEO_IMAGE_CNT	Contrast[7]	Contrast[6]	Contrast[5]	Contrast[4]	Contrast[3]	Contrast[2]	Contrast[1]	Contrast[0]
		Bright[7]	Bright[6]	Bright[5]	Bright[4]	Bright[3]	Bright[2]	Bright[1]	Bright[0]
		—	—	—	—	—	—	CLIP_Y	CLKP_C
		—	SUB_SCALE_V	SCALE_V[1]	SCALE_V[0]	—	SUB_SCALE_H	SCALE_H[1]	SCALE_H[0]
	VIDEO_BASEADR	VIDEO_BASEADR[31]	VIDEO_BASEADR[30]	VIDEO_BASEADR[29]	VIDEO_BASEADR[28]	VIDEO_BASEADR[27]	VIDEO_BASEADR[26]	VIDEO_BASEADR[25]	VIDEO_BASEADR[24]
		VIDEO_BASEADR[23]	VIDEO_BASEADR[22]	VIDEO_BASEADR[21]	VIDEO_BASEADR[20]	VIDEO_BASEADR[19]	VIDEO_BASEADR[18]	VIDEO_BASEADR[17]	VIDEO_BASEADR[16]
		VIDEO_BASEADR[15]	VIDEO_BASEADR[14]	VIDEO_BASEADR[13]	VIDEO_BASEADR[12]	VIDEO_BASEADR[11]	VIDEO_BASEADR[10]	VIDEO_BASEADR[9]	VIDEO_BASEADR[8]
		VIDEO_BASEADR[7]	VIDEO_BASEADR[6]	VIDEO_BASEADR[5]	VIDEO_BASEADR[4]	VIDEO_BASEADR[3]	VIDEO_BASEADR[2]	VIDEO_BASEADR[1]	VIDEO_BASEADR[0]
	VIDEO_LINE_OFFSET	VIDEO_LINE_OFFSET[31]	VIDEO_LINE_OFFSET[30]	VIDEO_LINE_OFFSET[29]	VIDEO_LINE_OFFSET[28]	VIDEO_LINE_OFFSET[27]	VIDEO_LINE_OFFSET[26]	VIDEO_LINE_OFFSET[25]	VIDEO_LINE_OFFSET[24]
		VIDEO_LINE_OFFSET[23]	VIDEO_LINE_OFFSET[22]	VIDEO_LINE_OFFSET[21]	VIDEO_LINE_OFFSET[20]	VIDEO_LINE_OFFSET[19]	VIDEO_LINE_OFFSET[18]	VIDEO_LINE_OFFSET[17]	VIDEO_LINE_OFFSET[16]
		VIDEO_LINE_OFFSET[15]	VIDEO_LINE_OFFSET[14]	VIDEO_LINE_OFFSET[13]	VIDEO_LINE_OFFSET[12]	VIDEO_LINE_OFFSET[11]	VIDEO_LINE_OFFSET[10]	VIDEO_LINE_OFFSET[9]	VIDEO_LINE_OFFSET[8]
		VIDEO_LINE_OFFSET[7]	VIDEO_LINE_OFFSET[6]	VIDEO_LINE_OFFSET[5]	VIDEO_LINE_OFFSET[4]	VIDEO_LINE_OFFSET[3]	VIDEO_LINE_OFFSET[2]	VIDEO_LINE_OFFSET[1]	VIDEO_LINE_OFFSET[0]

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Video display controller 3	VIDEO_FIELD_OFFSET	VIDEO_FIELD_OFFSET [31]	VIDEO_FIELD_OFFSET [30]	VIDEO_FIELD_OFFSET [29]	VIDEO_FIELD_OFFSET [28]	VIDEO_FIELD_OFFSET [27]	VIDEO_FIELD_OFFSET [26]	VIDEO_FIELD_OFFSET [25]	VIDEO_FIELD_OFFSET [24]
		VIDEO_FIELD_OFFSET [23]	VIDEO_FIELD_OFFSET [22]	VIDEO_FIELD_OFFSET [21]	VIDEO_FIELD_OFFSET [20]	VIDEO_FIELD_OFFSET [19]	VIDEO_FIELD_OFFSET [18]	VIDEO_FIELD_OFFSET [17]	VIDEO_FIELD_OFFSET [16]
		VIDEO_FIELD_OFFSET [15]	VIDEO_FIELD_OFFSET [14]	VIDEO_FIELD_OFFSET [13]	VIDEO_FIELD_OFFSET [12]	VIDEO_FIELD_OFFSET [11]	VIDEO_FIELD_OFFSET [10]	VIDEO_FIELD_OFFSET[9]	VIDEO_FIELD_OFFSET[8]
		VIDEO_FIELD_OFFSET[7]	VIDEO_FIELD_OFFSET[6]	VIDEO_FIELD_OFFSET[5]	VIDEO_FIELD_OFFSET[4]	VIDEO_FIELD_OFFSET[3]	VIDEO_FIELD_OFFSET[2]	VIDEO_FIELD_OFFSET[1]	VIDEO_FIELD_OFFSET[0]
	VIDEO_LINE BUFF_NUM	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	VIDEO_LINE BUFF_NUM [8]
		VIDEO_LINE BUFF_NUM [7]	VIDEO_LINE BUFF_NUM [6]	VIDEO_LINE BUFF_NUM [5]	VIDEO_LINE BUFF_NUM [4]	VIDEO_LINE BUFF_NUM [3]	VIDEO_LINE BUFF_NUM [2]	VIDEO_LINE BUFF_NUM [1]	VIDEO_LINE BUFF_NUM [0]
	VIDEO_DISP_SIZE	—	—	—	—	—	—	—	VIDEO_DISP_HEIGHT[8]
		VIDEO_DISP_HEIGHT[7]	VIDEO_DISP_HEIGHT[6]	VIDEO_DISP_HEIGHT[5]	VIDEO_DISP_HEIGHT[4]	VIDEO_DISP_HEIGHT[3]	VIDEO_DISP_HEIGHT[2]	VIDEO_DISP_HEIGHT[1]	VIDEO_DISP_HEIGHT[0]
		—	—	—	—	—	—	VIDEO_DISP_WIDTH[9]	VIDEO_DISP_WIDTH[8]
		VIDEO_DISP_WIDTH[7]	VIDEO_DISP_WIDTH[6]	VIDEO_DISP_WIDTH[5]	VIDEO_DISP_WIDTH[4]	VIDEO_DISP_WIDTH[3]	VIDEO_DISP_WIDTH[2]	VIDEO_DISP_WIDTH[1]	VIDEO_DISP_WIDTH[0]
	VIDEO_DISP_HSTART	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	VIDEO_DISP_HSTART[9]	VIDEO_DISP_HSTART[8]
		VIDEO_DISP_HSTART[7]	VIDEO_DISP_HSTART[6]	VIDEO_DISP_HSTART[5]	VIDEO_DISP_HSTART[4]	VIDEO_DISP_HSTART[3]	VIDEO_DISP_HSTART[2]	VIDEO_DISP_HSTART[1]	VIDEO_DISP_HSTART[0]

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Video display controller 3	GRCMEN1 to 2	WE	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	DEN	VEN
	GRCBUSCNT 1 to 2	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	BURST_ MODE
		—	—	—	BUS_ FORMAT	—	—	—	ENDIAN
	GRCINTCNT 1 to 2	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	INT_UF_EN
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	UNDER_ FLOW
	GROPSADR 1 to 2	—	—	—	GROPSADR [28]	GROPSADR [27]	GROPSADR [26]	GROPSADR [25]	GROPSADR [24]
		GROPSADR [23]	GROPSADR [22]	GROPSADR [21]	GROPSADR [20]	GROPSADR [19]	GROPSADR [18]	GROPSADR [17]	GROPSADR [16]
		GROPSADR [15]	GROPSADR [14]	GROPSADR [13]	GROPSADR [12]	GROPSADR [11]	GROPSADR [10]	GROPSADR [9]	GROPSADR [8]
		GROPSADR [7]	GROPSADR [6]	GROPSADR [5]	GROPSADR [4]	GROPSADR [3]	GROPSADR [2]	GROPSADR [1]	GROPSADR [0]
	GROPSWH 1 to 2	—	—	—	—	—	—	GROPSH[9]	GROPSH[8]
		GROPSH[7]	GROPSH[6]	GROPSH[5]	GROPSH[4]	GROPSH[3]	GROPSH[2]	GROPSH[1]	GROPSH[0]
		—	—	—	—	—	—	GROPSW[9]	GROPSW[8]
		GROPSW[7]	GROPSW[6]	GROPSW[5]	GROPSW[4]	GROPSW[3]	GROPSW[2]	GROPSW[1]	GROPSW[0]

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Video display controller 3	GROPSOFST 1 to 2	—	—	—	GROPSOFST [28]	GROPSOFST [27]	GROPSOFST [26]	GROPSOFST [25]	GROPSOFST [24]
		GROPSOFST [23]	GROPSOFST [22]	GROPSOFST [21]	GROPSOFST [20]	GROPSOFST [19]	GROPSOFST [18]	GROPSOFST [17]	GROPSOFST [16]
		GROPSOFST [15]	GROPSOFST [14]	GROPSOFST [13]	GROPSOFST [12]	GROPSOFST [11]	GROPSOFST [10]	GROPSOFST [9]	GROPSOFST [8]
		GROPSOFST [7]	GROPSOFST [6]	GROPSOFST [5]	GROPSOFST [4]	GROPSOFST [3]	GROPSOFST [2]	GROPSOFST [1]	GROPSOFST [0]
	GROPDPHV 1 to 2	—	—	—	—	—	—	GROPDPV[9]	GROPDPV[8]
		GROPDPV[7]	GROPDPV[6]	GROPDPV[5]	GROPDPV[4]	GROPDPV[3]	GROPDPV[2]	GROPDPV[1]	GROPDPV[0]
		—	—	—	—	—	—	GROPDPH[9]	GROPDPH[8]
		GROPDPH[7]	GROPDPH[6]	GROPDPH[5]	GROPDPH[4]	GROPDPH[3]	GROPDPH[2]	GROPDPH[1]	GROPDPH[0]
	GROPEWH 1 to 2	—	—	—	—	—	—	GROPEH[9]	GROPEH[8]
		GROPEH[7]	GROPEH[6]	GROPEH[5]	GROPEH[4]	GROPEH[3]	GROPEH[2]	GROPEH[1]	GROPEH[0]
		—	—	—	—	—	—	GROPEW[9]	GROPEW[8]
		GROPEW[7]	GROPEW[6]	GROPEW[5]	GROPEW[4]	GROPEW[3]	GROPEW[2]	GROPEW[1]	GROPEW[0]
	GROPEDPHV 1 to 2	—	—	—	—	—	—	GROPEDPV [9]	GROPEDPV [8]
		GROPEDPV [7]	GROPEDPV [6]	GROPEDPV [5]	GROPEDPV [4]	GROPEDPV [3]	GROPEDPV [2]	GROPEDPV [1]	GROPEDPV [0]
		—	—	—	—	—	—	GROPEDPH [9]	GROPEDPH [8]
		GROPEDPH [7]	GROPEDPH [6]	GROPEDPH [5]	GROPEDPH [4]	GROPEDPH [3]	GROPEDPH [2]	GROPEDPH [1]	GROPEDPH [0]
	GROPEDPA 1 to 2	DEFA[7]	DEFA[6]	DEFA[5]	DEFA[4]	DEFA[3]	DEFA[2]	DEFA[1]	DEFA[0]
		ACOEF[7]	ACOEF[6]	ACOEF[5]	ACOEF[4]	ACOEF[3]	ACOEF[2]	ACOEF[1]	ACOEF[0]
		ARATE[7]	ARATE[6]	ARATE[5]	ARATE[4]	ARATE[3]	ARATE[2]	ARATE[1]	ARATE[0]
		WE	—	—	AST	—	AMOD[1]	AMOD[0]	AEN
	GROPCRKY0_ 1 to 2	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	CKEN
		CROMAKR[4]	CROMAKR[3]	CROMAKR[2]	CROMAKR[1]	CROMAKR[0]	CROMAKG[5]	CROMAKG[4]	CROMAKG[3]
		CROMAKG[2]	CROMAKG[1]	CROMAKG[0]	CROMAKB[4]	CROMAKB[3]	CROMAKB[2]	CROMAKB[1]	CROMAKB[0]

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Video display controller 3	GROPCRKY1_1 to 2	—	—	—	—	—	—	—	—
		ALPHA[7]	ALPHA[6]	ALPHA[5]	ALPHA[4]	ALPHA[3]	ALPHA[2]	ALPHA[1]	ALPHA[0]
		R[4]	R[3]	R[2]	R[1]	R[0]	G[5]	G[4]	G[3]
		G[2]	G[1]	G[0]	B[4]	B[3]	B[2]	B[1]	B[0]
	GROPBASERGB1 to 2	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		BASE_R[4]	BASE_R[3]	BASE_R[2]	BASE_R[1]	BASE_R[0]	BASE_G[5]	BASE_G[4]	BASE_G[3]
		BASE_G[2]	BASE_G[1]	BASE_G[0]	BASE_B[4]	BASE_B[3]	BASE_B[2]	BASE_B[1]	BASE_B[0]
	SGMODE	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	EX_SYNC_MODE
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	RGB565
	SGINTCNT	—	—	INT_LINE_NUM[9]	INT_LINE_NUM[8]	INT_LINE_NUM[7]	INT_LINE_NUM[6]	INT_LINE_NUM[5]	INT_LINE_NUM[4]
		INT_LINE_NUM[3]	INT_LINE_NUM[2]	INT_LINE_NUM[1]	INT_LINE_NUM[0]	—	—	—	INT_LINE_EN
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	LINE_STATUS
	SYNCNT	—	—	—	—	—	—	—	RGB_TIM
		—	—	—	—	VSYNC_TIM	HSYNC_TIM	DE_TIM	M_DISP_TIM
		—	—	—	—	—	—	—	—
		—	—	—	—	VSYNC_TYPE	HSYNC_TYPE	DE_TYPE	M_DISP_TYPE
	PANEL_CLKSEL	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	ICKSEL	ICKEN	—	—	—	—
		—	—	DCDR[5]	DCDR[4]	DCDR[3]	DCDR[2]	DCDR[1]	DCDR[0]

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Video display controller 3	SYN_SIZE	—	—	—	—	—	—	SYN_HEIGHT [9]	SYN_HEIGHT [8]
		SYN_HEIGHT [7]	SYN_HEIGHT [6]	SYN_HEIGHT [5]	SYN_HEIGHT [4]	SYN_HEIGHT [3]	SYN_HEIGHT [2]	SYN_HEIGHT [1]	SYN_HEIGHT [0]
		—	—	—	—	—	SYN_WIDTH [10]	SYN_WIDTH [9]	SYN_WIDTH [8]
		SYN_WIDTH [7]	SYN_WIDTH [6]	SYN_WIDTH [5]	SYN_WIDTH [4]	SYN_WIDTH [3]	SYN_WIDTH [2]	SYN_WIDTH [1]	SYN_WIDTH [0]
	PANEL_VSYNC_TIM	—	—	—	—	—	—	VSYNC_START[9]	VSYNC_START[8]
		VSYNC_START[7]	VSYNC_START[6]	VSYNC_START[5]	VSYNC_START[4]	VSYNC_START[3]	VSYNC_START[2]	VSYNC_START[1]	VSYNC_START[0]
		—	—	—	—	—	—	VSYNC_END [9]	VSYNC_END [8]
		VSYNC_END [7]	VSYNC_END [6]	VSYNC_END [5]	VSYNC_END [4]	VSYNC_END [3]	VSYNC_END [2]	VSYNC_END [1]	VSYNC_END [0]
	PANEL_HSYNC_TIM	—	—	—	—	—	HSYNC_START[10]	HSYNC_START[9]	HSYNC_START[8]
		HSYNC_START[7]	HSYNC_START[6]	HSYNC_START[5]	HSYNC_START[4]	HSYNC_START[3]	HSYNC_START[2]	HSYNC_START[1]	HSYNC_START[0]
		—	—	—	—	—	HSYNC_END [10]	HSYNC_END [9]	HSYNC_END [8]
		HSYNC_END [7]	HSYNC_END[6]	HSYNC_END [5]	HSYNC_END [4]	HSYNC_END [3]	HSYNC_END [2]	HSYNC_END [1]	HSYNC_END [0]
	VIDEO_VSYNC_TIM2	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	VIDEO_VSYNC_START2[9]	VIDEO_VSYNC_START2[8]
		VIDEO_VSYNC_START2[7]	VIDEO_VSYNC_START2[6]	VIDEO_VSYNC_START2[5]	VIDEO_VSYNC_START2[4]	VIDEO_VSYNC_START2[3]	VIDEO_VSYNC_START2[2]	VIDEO_VSYNC_START2[1]	VIDEO_VSYNC_START2[0]
	GRA_VSYNC_TIM	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	GRA_VSYNC_START[9]	GRA_VSYNC_START[8]
		GRA_VSYNC_START[7]	GRA_VSYNC_START[6]	GRA_VSYNC_START[5]	GRA_VSYNC_START[4]	GRA_VSYNC_START[3]	GRA_VSYNC_START[2]	GRA_VSYNC_START[1]	GRA_VSYNC_START[0]

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Video display controller 3	AC_LINE_NUM	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	AC_LINE_NUM[4]	AC_LINE_NUM[3]	AC_LINE_NUM[2]	AC_LINE_NUM[1]	AC_LINE_NUM[0]
	DE_SIZE	—	—	—	—	—	—	DE_HEIGHT[9]	DE_HEIGHT[8]
		DE_HEIGHT[7]	DE_HEIGHT[6]	DE_HEIGHT[5]	DE_HEIGHT[4]	DE_HEIGHT[3]	DE_HEIGHT[2]	DE_HEIGHT[1]	DE_HEIGHT[0]
		—	—	—	—	—	DE_WIDTH[10]	DE_WIDTH[9]	DE_WIDTH[8]
		DE_WIDTH[7]	DE_WIDTH[6]	DE_WIDTH[5]	DE_WIDTH[4]	DE_WIDTH[3]	DE_WIDTH[2]	DE_WIDTH[1]	DE_WIDTH[0]
	DE_START	—	—	—	—	—	—	DE_START_V[9]	DE_START_V[8]
		DE_START_V[7]	DE_START_V[6]	DE_START_V[5]	DE_START_V[4]	DE_START_V[3]	DE_START_V[2]	DE_START_V[1]	DE_START_V[0]
		—	—	—	—	—	DE_START_H[10]	DE_START_H[9]	DE_START_H[8]
		DE_START_H[7]	DE_START_H[6]	DE_START_H[5]	DE_START_H[4]	DE_START_H[3]	DE_START_H[2]	DE_START_H[1]	DE_START_H[0]
Sampling rate converter	SRCID_0								
	SRCOD_0								
	SRCIDCTRL_0	—	—	—	—	—	—	IED	IEN
		—	—	—	—	—	—	IFTRG[1]	IFTRG[0]
	SRCODCTRL_0	—	—	—	—	—	OCH	OED	OEN
		—	—	—	—	—	—	OFTRG[1]	OFTRG[0]

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Sampling rate converter	SRCCTRL_0	—	—	CEEN	SRCEN	UDEN	OVEN	FL	CL
		IFS[3]	IFS[2]	IFS[1]	IFS[0]	—	—	—	OFS
	SRCSTAT_0	OFDN[4]	OFDN[3]	OFDN[2]	OFDN[1]	OFDN[0]	IFDN[3]	IFDN[2]	IFDN[1]
		IFDN[0]	—	CEF	FLF	UDF	OVF	IINT	OINT
	SRCID_1								
	SRCOD_1								
	SRCIDCTRL_1	—	—	—	—	—	—	IED	IEN
		—	—	—	—	—	—	IFTRG[1]	IFTRG[0]
	SRCODCTRL_1	—	—	—	—	—	OCH	OED	OEN
		—	—	—	—	—	—	OFTRG[1]	OFTRG[0]
	SRCCTRL_1	—	—	CEEN	SRCEN	UDEN	OVEN	FL	CL
		—	—	—	—	—	—	—	OFS
	SRCSTAT_1	—	—	OFDN[2]	OFDN[1]	OFDN[0]	IFDN[3]	IFDN[2]	IFDN[1]
		IFDN[0]	—	CEF	FLF	UDF	OVF	IINT	OINT

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
General purpose I/O ports	PAIOR0	—	—	—	—	—	—	—	—
		—	—	—	—	PA3IOR	PA2IOR	PA1IOR	PA0IOR
	PADR1	—	—	—	—	—	—	—	PA3DR
		—	—	—	—	—	—	—	PA2DR
	PADR0	—	—	—	—	—	—	—	PA1DR
		—	—	—	—	—	—	—	PA0DR
	PAPR0	—	—	—	—	—	—	—	—
		—	—	—	—	PA3PR	PA2PR	PA1PR	PA0PR
	PBCR5	—	—	—	—	—	—	PB22MD1	PB22MD0
		—	—	PB21MD1	PB21MD0	—	—	PB20MD1	PB20MD0
	PBCR4	—	—	PB19MD1	PB19MD0	—	—	PB18MD1	PB18MD0
		—	—	PB17MD1	PB17MD0	—	—	PB16MD1	PB16MD0
	PBCR3	—	—	PB15MD1	PB15MD0	—	—	PB14MD1	PB14MD0
		—	—	PB13MD1	PB13MD0	—	—	PB12MD1	PB12MD0
	PBCR2	—	—	PB11MD1	PB11MD0	—	—	PB10MD1	PB10MD0
		—	—	PB9MD1	PB9MD0	—	—	PB8MD1	PB8MD0
	PBCR1	—	—	PB7MD1	PB7MD0	—	—	PB6MD1	PB6MD0
		—	—	PB5MD1	PB5MD0	—	—	PB4MD1	PB4MD0
	PBCR0	—	—	—	PB3MD0	—	—	—	PB2MD0
		—	—	—	PB1MD0	—	—	—	—
	PBIOR1	—	—	—	—	—	—	—	—
		—	PB22IOR	PB21IOR	PB20IOR	PB19IOR	PB18IOR	PB17IOR	PB16IOR
	PBIOR0	PB15IOR	PB14IOR	PB13IOR	PB12IOR	PB11IOR	PB10IOR	PB9IOR	PB8IOR
		PB7IOR	PB6IOR	PB5IOR	PB4IOR	PB3IOR	PB2IOR	PB1IOR	—
	PBDR1	—	—	—	—	—	—	—	—
		—	PB22DR	PB21DR	PB20DR	PB19DR	PB18DR	PB17DR	PB16DR
	PBDR0	PB15DR	PB14DR	PB13DR	PB12DR	PB11DR	PB10DR	PB9DR	PB8DR
		PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	—
	PBPR1	—	—	—	—	—	—	—	—
		—	PB22PR	PB21PR	PB20PR	PB19PR	PB18PR	PB17PR	PB16PR

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
General purpose I/O ports	PBPR0	PB15PR	PB14PR	PB13PR	PB12PR	PB11PR	PB10PR	PB9PR	PB8PR
		PB7PR	PB6PR	PB5PR	PB4PR	PB3PR	PB2PR	PB1PR	—
	PCCR2	—	—	—	—	—	—	—	PC10MD0
		—	—	—	PC9MD0	—	—	PC8MD1	PC8MD0
	PCCR1	—	—	PC7MD1	PC7MD0	—	—	PC6MD1	PC6MD0
		—	—	PC5MD1	PC5MD0	—	—	—	PC4MD0
	PCCR0	—	—	—	PC3MD0	—	—	—	PC2MD0
		—	—	—	PC1MD0	—	—	—	PC0MD0
	PCIOR0	—	—	—	—	—	PC10IOR	PC9IOR	PC8IOR
		PC7IOR	PC6IOR	PC5IOR	PC4IOR	PC3IOR	PC2IOR	PC1IOR	PC0IOR
	PCDR0	—	—	—	—	—	PC10DR	PC9DR	PC8DR
		PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR
	PCPR0	—	—	—	—	—	PC10PR	PC9PR	PC8PR
		PC7PR	PC6PR	PC5PR	PC4PR	PC3PR	PC2PR	PC1PR	PC0PR
	PDCR3	—	—	PD15MD1	PD15MD0	—	—	PD14MD1	PD14MD0
		—	—	PD13MD1	PD13MD0	—	—	PD12MD1	PD12MD0
	PDCR2	—	—	PD11MD1	PD11MD0	—	—	PD10MD1	PD10MD0
		—	—	PD9MD1	PD9MD0	—	—	PD8MD1	PD8MD0
	PDCR1	—	—	PD7MD1	PD7MD0	—	—	PD6MD1	PD6MD0
		—	—	PD5MD1	PD5MD0	—	—	PD4MD1	PD4MD0
	PDCR0	—	—	PD3MD1	PD3MD0	—	—	PD2MD1	PD2MD0
		—	—	PD1MD1	PD1MD0	—	—	PD0MD1	PD0MD0
	PDIOR0	PD15IOR	PD14IOR	PD13IOR	PD12IOR	PD11IOR	PD10IOR	PD9IOR	PD8IOR
		PD7IOR	PD6IOR	PD5IOR	PD4IOR	PD3IOR	PD2IOR	PD1IOR	PD0IOR
	PDDR0	PD15DR	PD14DR	PD13DR	PD12DR	PD11DR	PD10DR	PD9DR	PD8DR
		PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR
	PDPR0	PD15PR	PD14PR	PD13PR	PD12PR	PD11PR	PD10PR	PD9PR	PD8PR
		PD7PR	PD6PR	PD5PR	PD4PR	PD3PR	PD2PR	PD1PR	PD0PR
	PECR1	—	—	—	—	—	—	—	—
		—	—	PE5MD1	PE5MD0	—	—	PE4MD1	PE4MD0

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
General purpose I/O ports	PECR0	—	—	PE3MD1	PE3MD0	—	—	PE2MD1	PE2MD0
		—	PE1MD2	PE1MD1	PE1MD0	—	—	PE0MD1	PE0MD0
	PEIOR0	—	—	—	—	—	—	—	—
		—	—	PE5IOR	PE4IOR	PE3IOR	PE2IOR	PE1IOR	PE0IOR
	PEDR0	—	—	—	—	—	—	—	—
		—	—	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR
	PEPR0	—	—	—	—	—	—	—	—
		—	—	PE5PR	PE4PR	PE3PR	PE2PR	PE1PR	PE0PR
	PFCR3	—	—	—	—	—	—	—	—
		—	—	—	—	—	PF12MD2	PF12MD1	PF12MD0
	PFCR2	—	PF11MD2	PF11MD1	PF11MD0	—	PF10MD2	PF10MD1	PF10MD0
		—	PF9MD2	PF9MD1	PF9MD0	—	PF8MD2	PF8MD1	PF8MD0
	PFCR1	—	PF7MD2	PF7MD1	PF7MD0	—	PF6MD2	PF6MD1	PF6MD0
		—	PF5MD2	PF5MD1	PF5MD0	—	PF4MD2	PF4MD1	PF4MD0
	PFCR0	—	PF3MD2	PF3MD1	PF3MD0	—	PF2MD2	PF2MD1	PF2MD0
		—	PF1MD2	PF1MD1	PF1MD0	—	PF0MD2	PF0MD1	PF0MD0
	PFIOR0	—	—	—	PF12IOR	PF11IOR	PF10IOR	PF9IOR	PF8IOR
		PF7IOR	PF6IOR	PF5IOR	PF4IOR	PF3IOR	PF2IOR	PF1IOR	PF0IOR
	PFDR0	—	—	—	PF12DR	PF11DR	PF10DR	PF9DR	PF8DR
		PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR
	PFPR0	—	—	—	PF12PR	PF11PR	PF10PR	PF9PR	PF8PR
		PF7PR	PF6PR	PF5PR	PF4PR	PF3PR	PF2PR	PF1PR	PF0PR
	PGCR7	—	—	—	—	—	—	—	—
		—	—	—	—	—	PG0MD2	PG0MD1	PG0MD0
	PGCR6	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	PG24MD1	PG24MD0
	PGCR5	—	—	PG23MD1	PG23MD0	—	—	PG22MD1	PG22MD0
		—	—	PG21MD1	PG21MD0	—	PG20MD2	PG20MD1	PG20MD0
	PGCR4	—	PG19MD2	PG19MD1	PG19MD0	—	PG18MD2	PG18MD1	PG18MD0
		—	PG17MD2	PG17MD1	PG17MD0	—	PG16MD2	PG16MD1	PG16MD0

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
General purpose I/O ports	PGCR3	—	PG15MD2	PG15MD1	PG15MD0	—	PG14MD2	PG14MD1	PG14MD0
		—	PG13MD2	PG13MD1	PG13MD0	—	PG12MD2	PG12MD1	PG12MD0
	PGCR2	—	PG11MD2	PG11MD1	PG11MD0	—	PG10MD2	PG10MD1	PG10MD0
		—	PG9MD2	PG9MD1	PG9MD0	—	PG8MD2	PG8MD1	PG8MD0
	PGCR1	—	PG7MD2	PG7MD1	PG7MD0	—	PG6MD2	PG6MD1	PG6MD0
		—	PG5MD2	PG5MD1	PG5MD0	—	PG4MD2	PG4MD1	PG4MD0
	PGCR0	—	—	PG3MD1	PG3MD0	—	—	PG2MD1	PG2MD0
		—	—	PG1MD1	PG1MD0	—	—	—	—
	PGIOR1	—	—	—	—	—	—	—	PG24IOR
		PG23IOR	PG22IOR	PG21IOR	PG20IOR	PG19IOR	PG18IOR	PG17IOR	PG16IOR
	PGIOR0	PG15IOR	PG14IOR	PG13IOR	PG12IOR	PG11IOR	PG10IOR	PG9IOR	PG8IOR
		PG7IOR	PG6IOR	PG5IOR	PG4IOR	PG3IOR	PG2IOR	PG1IOR	PG0IOR
	PGDR1	—	—	—	—	—	—	—	PG24DR
		PG23DR	PG22DR	PG21DR	PG20DR	PG19DR	PG18DR	PG17DR	PG16DR
	PGDR0	PG15DR	PG14DR	PG13DR	PG12DR	PG11DR	PG10DR	PG9DR	PG8DR
		PG7DR	PG6DR	PG5DR	PG4DR	PG3DR	PG2DR	PG1DR	PG0DR
	PGPR1	—	—	—	—	—	—	—	PG24PR
		PG23PR	PG22PR	PG21PR	PG20PR	PG19PR	PG18PR	PG17PR	PG16PR
	PGPR0	PG15PR	PG14PR	PG13PR	PG12PR	PG11PR	PG10PR	PG9PR	PG8PR
		PG7PR	PG6PR	PG5PR	PG4PR	PG3PR	PG2PR	PG1PR	PG0PR
	PHCR1	—	—	—	PH7MD0	—	—	—	PH6MD0
		—	—	—	PH5MD0	—	—	—	PH4MD0
	PHCR0	—	—	—	PH3MD0	—	—	—	PH2MD0
		—	—	—	PH1MD0	—	—	—	PH0MD0
	PHPR0	—	—	—	—	—	—	—	—
		PH7PR	PH6PR	PH5PR	PH4PR	PH3PR	PH2PR	PH1PR	PH0PR
	PJCR2	—	—	PJ11MD1	PJ11MD0	—	—	PJ10MD1	PJ10MD0
		—	—	PJ9MD1	PJ9MD0	—	—	PJ8MD1	PJ8MD0
	PJCR1	—	—	PJ7MD1	PJ7MD0	—	—	PJ6MD1	PJ6MD0
		—	—	PJ5MD1	PJ5MD0	—	—	PJ4MD1	PJ4MD0

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
General purpose I/O ports	PJCR0	—	—	PJ3MD1	PJ3MD0	—	PJ2MD2	PJ2MD1	PJ2MD0
		—	PJ1MD2	PJ1MD1	PJ1MD0	—	PJ0MD2	PJ0MD1	PJ0MD0
	PJIOR0	—	—	—	—	PJ11IOR	PJ10IOR	PJ9IOR	PJ8IOR
		PJ7IOR	PJ6IOR	PJ5IOR	PJ4IOR	PJ3IOR	PJ2IOR	PJ1IOR	PJ0IOR
	PJDR0	—	—	—	—	PJ11DR	PJ10DR	PJ9DR	PJ8DR
		PJ7DR	PJ6DR	PJ5DR	PJ4DR	PJ3DR	PJ2DR	PJ1DR	PJ0DR
	PJPR0	—	—	—	—	PJ11PR	PJ10PR	PJ9PR	PJ8PR
		PJ7PR	PJ6PR	PJ5PR	PJ4PR	PJ3PR	PJ2PR	PJ1PR	PJ0PR
	PKCR2	—	—	PK11MD1	PK11MD0	—	—	PK10MD1	PK10MD0
		—	—	PK9MD1	PK9MD0	—	—	PK8MD1	PK8MD0
	PKCR1	—	—	PK7MD1	PK7MD0	—	—	PK6MD1	PK6MD0
		—	—	PK5MD1	PK5MD0	—	—	PK4MD1	PK4MD0
	PKCR0	—	—	PK3MD1	PK3MD0	—	—	PK2MD1	PK2MD0
		—	—	PK1MD1	PK1MD0	—	—	PK0MD1	PK0MD0
	PKIOR0	—	—	—	—	PK11IOR	PK10IOR	PK9IOR	PK8IOR
		PK7IOR	PK6IOR	PK5IOR	PK4IOR	PK3IOR	PK2IOR	PK1IOR	PK0IOR
	PKDR0	—	—	—	—	PK11DR	PK10DR	PK9DR	PK8DR
		PK7DR	PK6DR	PK5DR	PK4DR	PK3DR	PK2DR	PK1DR	PK0DR
	PKPR0	—	—	—	—	PK11PR	PK10PR	PK9PR	PK8PR
		PK7PR	PK6PR	PK5PR	PK4PR	PK3PR	PK2PR	PK1PR	PK0PR
Power-down modes	STBCR1	STBY	DEEP	—	—	—	—	—	—
	STBCR2	MSTP10	—	MSTP8	MSTP7	—	—	—	—
	STBCR3	HIZ	MSTP36	MSTP35	MSTP34	MSTP33	MSTP32	—	MSTP30
	STBCR4	MSTP47	MSTP46	MSTP45	MSTP44	MSTP43	MSTP42	MSTP41	MSTP40
	STBCR5	MSTP57	MSTP56	MSTP55	—	MSTP53	MSTP52	MSTP51	MSTP50
	STBCR6	MSTP67	MSTP66	MSTP65	MSTP64	MSTP63	MSTP62	MSTP61	MSTP60
	STBCR7	MSTP77	MSTP76	—	MSTP74	—	MSTP72	—	MSTP70
	STBCR8	MSTP87	—	—	—	—	—	—	MSTP80
	SWRSTCR	AXTALE	—	—	IEBSRST	SSIF3SRST	SSIF2SRST	SSIF1SRST	SSIF0SRST
	SYSCR1	—	—	—	—	RAME3	RAME2	RAME1	RAME0
	SYSCR2	—	—	—	—	RAMWE3	RAMWE2	RAMWE1	RAMWE0

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Power-down modes	SYSCR3	—	—	VRAME5	VRAME4	VRAME3	VRAME2	VRAME1	VRAME0
	SYSCR4	—	—	VRAMWE5	VRAMWE4	VRAMWE3	VRAMWE2	VRAMWE1	VRAMWE0
	SYSCR5	—	—	—	—	RRAMWE3	RRAMWE2	RRAMWE1	RRAMWE0
	RRAMKP	—	—	—	—	RRAMKP3	RRAMKP2	RRAMKP1	RRAMKP0
	DSCTR	EBUSKEEPE	RAMBOOT	—	—	—	—	—	—
	DSSSR	—	—	—	—	—	PG11	PG10	—
		—	RTCAR	PC8	PC7	PC6	PC5	PJ3	PJ1
	DSESR	—	—	—	—	—	PG11E	PG10E	NMIE
		—	—	PC8E	PC7E	PC6E	PC5E	PJ3E	PJ1E
	DSFR	IOKEEP	—	—	—	—	PG11F	PG10F	NMIF
		—	RTCARF	PC8F	PC7F	PC6F	PC5F	PJ3F	PJ1F
	XTALCTR	—	—	—	—	—	—	—	GAIN
User debugging interface	SDIR	TI[7]	TI[6]	TI[5]	TI[4]	TI[3]	TI[2]	TI[1]	TI[0]
		—	—	—	—	—	—	—	—
Motor control PWM timer	PWCR_1	—	—	IE	CMF	CST	CKS2	CKS1	CKS0
	PWPR_1	OPS1H	OPS1G	OPS1F	OPS1E	OPS1D	OPS1C	OPS1B	OPS1A
	PWCYR_1	PWCY15	PWCY14	PWCY13	PWCY12	PWCY11	PWCY10	PWCY9	PWCY8
		PWCY7	PWCY6	PWCY5	PWCY4	PWCY3	PWCY2	PWCY1	PWCY0
	PWBFR_1A	—	—	—	OTS	—	—	DT9	DT8
		DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
	PWBFR_1C	—	—	—	OTS	—	—	DT9	DT8
		DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
	PWBFR_1E	—	—	—	OTS	—	—	DT9	DT8
		DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
	PWBFR_1G	—	—	—	OTS	—	—	DT9	DT8
		DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
	PWCR_2	—	—	IE	CMF	CST	CKS2	CKS1	CKS0
	PWPR_2	OPS2H	OPS2G	OPS2F	OPS2E	OPS2D	OPS2C	OPS2B	OPS2A
	PWCYR_2	PWCY15	PWCY14	PWCY13	PWCY12	PWCY11	PWCY10	PWCY9	PWCY8
		PWCY7	PWCY6	PWCY5	PWCY4	PWCY3	PWCY2	PWCY1	PWCY0

chanhvinh.com

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Motor control PWM timer	PWBFR_2A	—	—	—	OTS	—	—	DT9	DT8
		DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
	PWBFR_2C	—	—	—	OTS	—	—	DT9	DT8
		DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
	PWBFR_2E	—	—	—	OTS	—	—	DT9	DT8
		DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
	PWBFR_2G	—	—	—	OTS	—	—	DT9	DT8
		DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
	PWBTCR	BTC2G	BTC2E	BTC2C	BTC2A	BTC1G	BTC1E	BTC1C	BTC1A
		—	—	—	—	—	—	—	—

Notes: 1. When MCR15=0
2. When MCR15=1
3. In command access mode
4. In sector access mode

chanhvinh.com

36.3 Register States in Each Operating Mode

Module Name	Register Abbreviation	Power-On Reset	Manual Reset	Deep Standby	Software Standby	Module Standby	Sleep
Clock pulse generator	FRQCR	Initialized*1	Retained	Initialized	Retained	—	Retained
Interrupt control register	IBNR	Initialized	Retained*2	Initialized	Retained	—	Retained
	Other than above	Initialized	Retained	Initialized	Retained	—	Retained
Cache	All registers	Initialized	Retained	Initialized	Retained	—	Retained
Bus state controller	RTCSR	Initialized	Retained*3	Initialized	Retained	—	Retained*3
	RTCNT	Initialized	Retained*4	Initialized	Retained	—	Retained*4
	Other than above	Initialized	Retained	Initialized	Retained	—	Retained
Direct memory access controller	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained*7
Multi-function timer pulse unit 2	All registers	Initialized	Retained	Initialized	Initialized	Initialized	Retained
Compare match timer	All registers	Initialized	Retained	Initialized	Initialized	Retained	Retained
Watchdog timer	WRCSR	Initialized*1	Retained	Initialized	Retained	—	Retained
	Other than above	Initialized	Retained	Initialized	Retained	—	Retained
Realtime clock	R64CNT	Retained*4	Retained*4	Retained*4	Retained*4	Retained	Retained*4
	RSECCNT						
	RMINCNT						
	RHRCNT						
	RWKCNT						
	RDAYCNT						
	RMONCNT						
	RYRCNT						

chanhvinh.com

Module Name	Register Abbreviation	Power-On Reset	Manual Reset	Deep Standby	Software Standby	Module Standby	Sleep
Realtime clock	RSECAR	Retained	Retained	Initialized	Retained	Retained	Retained
	RMINAR						
	RHRAR						
	RWKAR						
	RDAYAR						
	RMONAR						
	RYRAR						
	RCR1	Initialized	Initialized	Initialized	Retained	Retained	Retained
	RCR2	Initialized	Initialized*5	Initialized	Retained	Retained	Retained
	RCR3	Retained	Retained	Retained	Retained	Retained	Retained
	RCR5	Retained	Retained	Retained	Retained	Retained	Retained
	RFRH	Retained	Retained	Retained	Retained	Retained	Retained
	RFRL	Retained	Retained	Retained	Retained	Retained	Retained
Serial communication interface with FIFO	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained
Renesas serial peripheral interface	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained
I^2C bus interface 3	ICMR_0 to 2	Initialized	Retained	Initialized	Retained*6	Retained*6	Retained
	Other than above	Initialized	Retained	Initialized	Retained	Retained	Retained
Serial sound interface	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained
Serial I/O with FIFO	All registers	Initialized	Initialized	Initialized	Retained	Retained	Retained
Controller area network	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained
IEBus controller	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained
Renesas SPDIF interface	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained

chanhvinh.com

Module Name	Register Abbreviation	Power-On Reset	Manual Reset	Deep Standby	Software Standby	Module Standby	Sleep
CD-ROM decoder	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained
A/D converter	All registers	Initialized	Retained	Initialized	Initialized	Initialized	Retained
NAND flash memory controller	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained
USB 2.0 host/func-tion module	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained
Video display controller 3	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained
Sampling rate converter	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained
General purpose I/O ports	All registers	Initialized	Retained	Initialized	Retained	—	Retained
Power-down modes	DSFR	Initialized	Retained	Retained	Retained	—	Retained
	XTALCTR	Initialized*10	Retained	Retained*9	Retained*9	—	Retained
	Other than above	Initialized	Retained	Initialized	Retained	—	Retained
User debugging interface*8	SDIR	Retained	Retained	Initialized	Retained	Retained	Retained
Motor control PWM timer	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained

Notes: 1. Retains the previous value after an internal power-on reset by means of the watchdog timer.
2. The BN3 to BN0 bits are initialized.
3. Flag handling continues.
4. Counting up continues.
5. Bits RTCEN and START are retained.
6. Bits BC2 to BC0 are initialized.
7. Transfer operations can be continued.
8. Initialized by $\overline{\text{TRST}}$ assertion or in the Test-Logic-Reset state of the TAP controller.
9. Initialized when realtime clock is not using EXTAL.

chanhvinh.com

10. Retains the previous value after an internal power-on reset by the watchdog timer or a user debugging interface reset.

chanhvinh.com

chanhvinh.com

chanhvinh.com
chanhvinh.com
chanhvinh.com
chanhvinh.com

chanhvinh.com

Section 37 Electrical Characteristics

37.1 Absolute Maximum Ratings

Table 37.1 Absolute Maximum Ratings

Item		Symbol	Value	Unit
Power supply voltage (I/O)		PV_{CC}	–0.3 to 4.6	V
Power supply voltage (Internal)		V_{CC}	–0.3 to 1.7	V
PLL power supply voltage		$PLLV_{CC}$	–0.3 to 1.7	V
Analog power supply voltage		AV_{CC}	–0.3 to 4.6	V
Analog reference voltage		AV_{ref}	–0.3 to AV_{CC} +0.3	V
USB transceiver analog power supply voltage (I/O)		$USBAPV_{CC}$	–0.3 to 4.6	V
USB transceiver digital power supply voltage (I/O)		$USBDPV_{CC}$	–0.3 to 4.6	V
USB transceiver analog power supply voltage (internal)		$USBAV_{CC}$	–0.3 to 1.7	V
USB transceiver digital power supply voltage (internal)		$USBDV_{CC}$	–0.3 to 1.7	V
Power supply for USB 480 MHz (internal)		$USBUV_{CC}$	–0.3 to 1.7	V
Input voltage	Analog input pin	V_{AN}	–0.3 to AV_{CC} +0.3	V
	VBUS	V_{in}	–0.3 to 5.5	V
	Other input pins	V_{in}	–0.3 to PV_{CC} +0.3	V
Operating temperature	Regular specifications	T_{opr}	–20 to +85	°C
	Wide-range specifications		–40 to +85	
Storage temperature		T_{stg}	–55 to +125	°C

Caution: Permanent damage to the LSI may result if absolute maximum ratings are exceeded.

chanhvinh.com

37.2 Power-On/Power-Off Sequence

The 1.2-V power supply (V_{CC}, $PLLV_{CC}$, $USBAV_{CC}$, $USBDV_{CC}$, 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 $\overline{\text{TRST}}$ and $\overline{\text{RES}}$ pins low; otherwise, the output pins and input/output pins output undefined levels, resulting in system malfunction.

When turning off the power, drive the $\overline{\text{TRST}}$ and $\overline{\text{RES}}$ pins low if the undefined output may cause a problem.

chanhvinh.com

37.3 DC Characteristics

- Conditions used to obtain DC characteristics (2) in table 37.2 other than current consumption (for both 1-Mbyte and 640-Kbyte versions)
  V_{CC} = $PLLV_{CC}$ = $USBDV_{CC}$ = $USBUV_{CC}$ = 1.1 to 1.3 V,
  PV_{CC} = $USBDPV_{CC}$ = 3.0 to 3.6 V, AV_{CC} = 3.0 to 3.6 V, $USBAV_{CC}$ = 1.1 to 1.3 V,
  $USBAPV_{CC}$ = 3.0 to 3.6 V, V_{SS} = $PLLV_{SS}$ = AV_{SS} = $USBDV_{SS}$ = $USBAV_{SS}$ = $USBDPV_{SS}$ = $USBAPV_{SS}$ = $USBUV_{SS}$ = 0 V, T_a = –20 to 85 °C (regular specifications), –40 to 85 °C (wide-range specifications)

- Conditions used to obtain DC characteristics (2) in table 37.2 for current consumption (1-Mbyte and 640-Kbyte versions)
  V_{CC} = $PLLV_{CC}$ = $USBDV_{CC}$ = $USBUV_{CC}$ = 1.2 V, PV_{CC} = $USBDPV_{CC}$ = 3.3 V, AV_{CC} = 3.3 V, $USBAV_{CC}$ = 1.2 V, $USBAPV_{CC}$ = 3.3 V, V_{SS} = $PLLV_{SS}$ = AV_{SS} = $USBDV_{SS}$ = $USBAV_{SS}$ = $USBDPV_{SS}$ = $USBAPV_{SS}$ = $USBUV_{SS}$ = 0 V, Avref = 3.3 V, VBUS = 5.0 V
  T_a = –20 to 85 °C (regular specifications),
  –40 to 85 °C (wide-range specifications)
  Iϕ = 144.00 MHz, Bϕ = 72.00 MHz, Pϕ = 36.00 MHz

Table 37.2 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}	1.1	1.2	1.3	V	
PLL power supply voltage		$PLLV_{CC}$	1.1	1.2	1.3	V	
Analog power supply voltage		AV_{CC}	3.0	3.3	3.6	V	
USB power supply voltage		$USBAPV_{CC}$ $USBDPV_{CC}$	3.0	3.3	3.6	V	
		$USBAV_{CC}$ $USBDV_{CC}$ $USBUV_{CC}$	1.1	1.2	1.3	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 PE5 to PE0) (off state)	$\|I_{STI}\|$	—	—	1.0	µA	V_{in} = 0.5 to PV_{CC} – 0.5 V
	PE5 to PE0		—	—	10	µA	
Input capacitance	All pins	C_{in}	—	—	20	pF	

chanhvinh.com

Table 37.2 DC Characteristics (2) 1-Mbyte Version [Current Consumption]

Item		Power Supply	Symbol	Typ.	Max.	Unit	Test Conditions
Current consumption in normal operation		Vcc+PLLVcc	Icc	100	140	mA	
		PVcc	PIcc*	75	—	mA	
		AVcc	AIcc	1	4	mA	During A/D conversion
				1	3	µA	Waiting for A/D conversion
		AVref	AIref	1	4	mA	During A/D conversion and waiting for A/D conversion
		USBAVcc+ USBDVcc+ USBUVcc	UIcc	25.5	31	mA	In USB high-speed operation
		USBAPVcc+ USBDPVcc	UPIcc	37	40	mA	In USB high-speed operation
		VBUS	VIcc	6.5	7	µA	
Current consumption in sleep mode		Vcc+PLLVcc	Isleep	60	90	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	Vcc+PLLVcc	Isstby	1.5	20	mA	
		PVcc	PIsstby	1	—	µA	
		For the other power supply, the current consumption is the same as in normal operation.					
	Ta ≤ 50°C	Vcc+PLLVcc	Isstby	0.5	5	mA	
		PVcc	PIsstby	0.5	—	µA	
		For the other power supply, the current consumption is the same as in normal operation.					

chanhvinh.com

Item		Power Supply	Symbol	Typ.	Max.	Unit	Test Conditions
Current consumption in deep standby mode	Ta > 50°C	Vcc+ PLLVcc+ USBAVcc+ USBDVcc+ USBUVcc	Idstby	8	47	µA	RAM 0 Kbytes retained, RTC_X1 selected
				9.5	53	µA	RAM 16 Kbytes retained, RTC_X1 selected
				11	59	µA	RAM 32 Kbytes retained, RTC_X1 selected
				16	55	µA	RAM 0 Kbytes retained, EXTAL selected
				17.5	61	µA	RAM 16 Kbytes retained, EXTAL selected
				19	67	µA	RAM 32 Kbytes retained, EXTAL selected
		PVcc+ Avcc+ Avref+ USBAPVcc+ USBDPVcc	PIdstby	1	11	µA	RTC is not operating
				3.5	21	µA	RTC_X1 selected
				1	—	mA	EXTAL selected, small gain*
		VBUS	VIdstby	6.5	7	µA	
	Ta ≤ 50°C	Vcc+ PLLVcc+ USBAVcc+ USBDVcc+ USBUVcc	Idstby	3	19	µA	RAM 0 Kbytes retained, RTC_X1 selected
				4	23	µA	RAM 16 Kbytes retained, RTC_X1 selected
				5	27	µA	RAM 32 Kbytes retained, RTC_X1 selected
				11	27	µA	RAM 0 Kbytes retained, EXTAL selected
				12	31	µA	RAM 16 Kbytes retained, EXTAL selected
				13	35	µA	RAM 32 Kbytes retained, EXTAL selected
		PVcc+ AVcc+ AVref+ USBAPVcc+ USBDPVcc	PIdstby	0.5	7	µA	RTC is not operating
				3	13	µA	RTC_X1 selected
				1	—	mA	EXTAL selected, small gain*
		VBUS	VIdstby	6.5	7	µA	

Note: 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.

chanhvinh.com

Table 37.2 DC Characteristics (2) 640-Kbyte Version [Current Consumption]

Item		Power Supply	Symbol	Typ.	Max.	Unit	Test Conditions
Current consumption in normal operation		Vcc+ PLLVcc	Icc	75	100	mA	
		PVcc	PIcc*	75	—	mA	
		AVcc	AIcc	1	4	mA	During A/D conversion
				1	3	μA	Waiting for A/D conversion
		AVref	AIref	1	4	mA	During A/D conversion and waiting for A/D conversion
		USBAVcc+ USBDVcc+ USBUVcc	UIcc	25.5	31	mA	In USB high-speed operation
		USBAPVcc+ USBDPVcc	UPIcc	37	40	mA	In USB high-speed operation
		VBUS	VIcc	6.5	7	μA	
Current consumption in sleep mode		Vcc+ PLLVcc	Isleep	45	70	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	Vcc+PLLVcc	Isstby	1.5	20	mA	
		PVcc	PIsstby	1	—	μA	
		For the other power supply, the current consumption is the same as in normal operation.					
	Ta ≤ 50 °C	Vcc+PLLVcc	Isstby	0.5	5	mA	
		PVcc	PIsstby	0.5	—	μA	
		For the other power supply, the current consumption is the same as in normal operation.					

chanhvinh.com

Item		Power Supply	Symbol	Typ.	Max.	Unit	Test Conditions
Current consumption in deep standby mode	Ta > 50 °C	Vcc+ PLLVcc+ USBAVcc+ USBDVcc+ USBUVcc	Idstby	7	57	µA	RAM 0 Kbytes retained, RTC_X1 selected
				8.5	64	µA	RAM 16 Kbytes retained, RTC_X1 selected
				19	113	µA	RAM 120 Kbytes retained, RTC_X1 selected
				23	127	µA	RAM 160 Kbytes retained, RTC_X1 selected
				37	197	µA	RAM 320 Kbytes retained, RTC_X1 selected
				14	65	µA	RAM 0 Kbytes retained, EXTAL selected
				15.5	72	µA	RAM 16 Kbytes retained, EXTAL selected
				26	121	µA	RAM 128 Kbytes retained, EXTAL selected
				30	135	µA	RAM 160 Kbytes retained, EXTAL selected
				44	205	µA	RAM 320 Kbytes retained, EXTAL selected
		PVcc+ AVcc+ Avref+ USBAPVcc+ USBDPVcc	PIdstby	1	11	µA	RTC is not operating
				3.5	21	µA	RTC_X1 selected
				1	—	mA	EXTAL selected, small gain*
		VBUS	VIdstby	6.5	7	µA	

chanhvinh.com

Item		Power Supply	Symbol	Typ.	Max.	Unit	Test Conditions
Current consumption in deep standby mode	Ta ≤ 50 °C	Vcc+ PLLVcc+ USBAVcc+ USBDVcc+ USBUVcc	Idstby	3	23	µA	RAM 0 Kbytes retained, RTC_X1 selected
				4	27	µA	RAM 16 Kbytes retained, RTC_X1 selected
				11	55	µA	RAM 120 Kbytes retained, RTC_X1 selected
				13	63	µA	RAM 160 Kbytes retained, RTC_X1 selected
				23	103	µA	RAM 320 Kbytes retained, RTC_X1 selected
				10	31	µA	RAM 0 Kbytes retained, EXTAL selected
				11	35	µA	RAM 16 Kbytes retained, EXTAL selected
				18	63	µA	RAM 128 Kbytes retained, EXTAL selected
				20	71	µA	RAM 160 Kbytes retained, EXTAL selected
				30	111	µA	RAM 320 Kbytes retained, EXTAL selected
		PVcc+ AVcc+ Avref+ USBAPVcc+ USBDPVcc	PIdstby	0.5	7	µA	RTC is not operating
				3	13	µA	RTC_X1 selected
				1	—	mA	EXTAL selected, small gain*
		VBUS	VIdstby	6.5	7	µA	

Note: 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.

chanhvinh.com

Table 37.2 DC Characteristics (3) [Except I²C Bus Interface 3, and USB 2.0 Host/Function Module-Related Pins]

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input high voltage	$\overline{\text{RES}}$, NMI, MD_BOOT1, MD_BOOT0, MD_CLK1, MD_CLK0, $\overline{\text{ASEMD}}$, $\overline{\text{TRST}}$, EXTAL, AUDIO_X1, RTC_X1, PA3 to PA0	V_{IH}	PV_{CC} – 0.5	—	PV_{CC} + 0.3	V	
	PH7 to PH0		2.2	—	AV_{CC} + 0.3	V	
	Input pins other than above (except Schmitt pins)		2.2	—	PV_{CC} + 0.3	V	
Input low voltage	$\overline{\text{RES}}$, NMI, MD_BOOT1, MD_BOOT0, MD_CLK1, MD_CLK0, $\overline{\text{ASEMD}}$, $\overline{\text{TRST}}$, EXTAL, AUDIO_X1, RTC_X1, PA3 to PA0	V_{IL}	–0.3	—	0.5	V	
	Input pins other than above (except Schmitt pins)		–0.3	—	0.8	V	

chanhvinh.com

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input characteristics	IRQ7 to IRQ0, PINT7 to PINT0, WAIT, $\overline{\text{BREQ}}$, DREQ1, DREQ0, TIOC0A to TIOC0D, TIOC1A, TIOC1B, TIOC2A, TIOC2B. TIOC3A to TIOC3D, TIOC4A to TIOC4D, TCLKB to TCLKD, SCK3 to SCK0, RxD7 to RxD0, $\overline{\text{CTS3}}$, $\overline{\text{RTS3}}$, $\overline{\text{CTS1}}$, $\overline{\text{RTS1}}$, RSPCK1, RSPCK0, MOSI1, MOSI0, MISO1, MISO0, SSL10, SSL00, SSISCK3 to SSISCK0, SSIRxD0, SSIDATA3 to SSIDATA0, SSIWS3 to SSIWS0, SIOFSCK, SIOFSYNC, SIOFRxD, CRx0, CRx1, IERxD, SPDIF_IN, FRB, DV_CLK, DV_DATA7 to DV_DATA0, LCD_EXTCLK, SD_CMD, SD_D3 to SD_D0, SD_CD, SD_WP, PB22 to PB1, PC10 to PC0, PF12 to PF0, PG24 to PG0, PJ11 to PJ0, PK11 to PK0	V_T^+	$PV_{CC} \times 0.75$	—	—	V	
		V_T^-	—	—	0.5	V	
		$V_T^+ - V_T^-$	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

chanhvinh.com

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
RAM standby voltage	Software standby mode (high-speed on-chip RAM and large-capacity on-chip RAM)	V_{RAMS}	0.75	—	—	V	Measured with V_{CC} (= $PLLV_{CC}$) as parameter
	Deep standby mode (only the on-chip RAM for data retention)	V_{RAMD}	1.1	—	—	V	

Table 37.2 DC Characteristics (4) [I^2C Bus Interface 3-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 PE5/SDA2/DV_HSYNC to PE0/SCL0/AUDIO_CLK/IRQ0 pins are open-drain pins.

Table 37.2 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 (VBUS)	V_{IH}	4.02	—	5.25	V	
Input low voltage (VBUS)	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, VBUS, USB_X1, and USB_X2 pins

chanhvinh.com

Table 37.2 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: * DP and DM pins

Table 37.2 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: * DP and DM pins

chanhvinh.com

Table 37.2 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: * DP and DM pins

Table 37.3 Permissible Output Currents

Item		Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (per pin)	PE5 to PE0	I_{OL}	—	—	10	mA
	Output pins other than above				2	mA
Permissible output low current (total)		ΣI_{OL}	—	—	150	mA
Permissible output high current (per pin)		$-I_{OH}$	—	—	2	mA
Permissible output high current (total)		$\Sigma -I_{OH}$	—	—	150	mA

Caution: To protect the LSI's reliability, do not exceed the output current values in table 37.3.

chanhvinh.com

37.4 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} = $PLLV_{CC}$ = $USBDV_{CC}$ = $USBUV_{CC}$ = 1.1 to 1.3 V,
  PV_{CC} = $USBDPV_{CC}$ = 3.0 to 3.6 V, AV_{CC} = 3.0 to 3.6 V, $USBAV_{CC}$ = 1.1 to 1.3 V, $USBAPV_{CC}$ = 3.0 to 3.6 V, V_{SS} = $PLLV_{SS}$ = AV_{SS} = $USBDV_{SS}$ = $USBAV_{SS}$ = $USBDPV_{SS}$ = $USBAPV_{SS}$ = $USBUV_{SS}$ = 0 V, T_a = –20 to 85 °C (regular specifications), –40 to 85 °C (wide-range specifications)

Table 37.4 Maximum Operating Frequency

Item		Symbol	Min.	Max.	Unit	Remarks
Operating frequency	CPU clock (Iϕ)	f	40.00	144.00	MHz	
	Bus clock (Bϕ)		40.00	72.00	MHz	
	Peripheral clock (Pϕ)		10.00	36.00	MHz	

37.4.1 Clock Timing

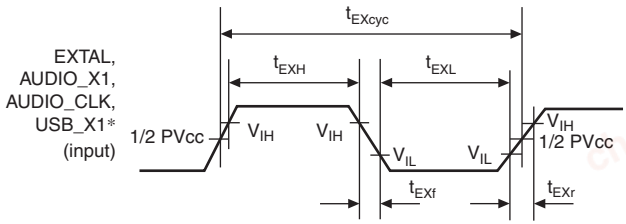
Table 37.5 Clock Timing

Item	Symbol	Min.	Max.	Unit	Figure
EXTAL clock input frequency	f_{EX}	10.00	18.00	MHz	Figure 37.1
EXTAL clock input cycle time	t_{EXcyc}	55.55	100.00	ns	
AUDIO_X1 clock input frequency (crystal resonator connected)	f_{EX}	10.00	25.00	MHz	
AUDIO_X1 clock input cycle time (crystal resonator connected)	t_{EXcyc}	40.00	100.00	ns	
AUDIO_X1, AUDIO_CLK clock input frequency (external clock input)	f_{EX}	1.00	25.00	MHz	
AUDIO_X1, AUDIO_CLK clock input cycle time (external clock input)	t_{EXcyc}	40.00	1000.00	ns	
USB_X1 clock input frequency (when high-speed transfer function is used)	f_{EX}	48 MHz ±100 ppm			
USB_X1 clock input frequency (when high-speed transfer function is not used and host controller function is used)		48 MHz ±500 ppm			

chanhvinh.com

Item	Symbol	Min.	Max.	Unit	Figure
USB_X1 clock input frequency (when neither high-speed transfer function nor host controller function is used)	f_{EX}	48 MHz ±2500 ppm			Figure 37.1
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}	
EXTAL, AUDIO_X1, AUDIO_CLK, USB_ X1 clock input rise time	t_{EXr}	—	4	ns	
EXTAL, AUDIO_X1, AUDIO_CLK, USB_ X1 clock input fall time	t_{EXf}	—	4	ns	
CKIO clock output frequency	f_{OP}	40.00	72.00	MHz	
CKIO clock output cycle time	t_{cyc}	13.88	25	ns	Figures 37.2 (1) and 37.2 (2)
CKIO clock output low pulse width 1	t_{CKOL1}	$t_{cyc}/2 - t_{CKOr1}$	—	ns	Figure 37.2 (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 37.2 (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	
Power-on oscillation settling time	t_{OSC1}	10	—	ms	Figure 37.3
Oscillation settling time 1 on return from standby	t_{OSC2}	10	—	ms	Figure 37.4
Oscillation settling time 2 on return from standby	t_{OSC3}	10	—	ms	Figure 37.5
Real time clock oscillation settling time	t_{ROSC}	3	—	s	Figure 37.6
Mode hold time	t_{MDH}	200	—	ns	Figures 37.3 and 37.4

chanhvinh.com



Note: * When the clock is input on the EXTAL, AUDIO_X1, AUDIO_CLK, or USB_X1 pin.

Figure 37.1 EXTAL, AUDIO_X1, AUDIO_CLK, and USB_X1 Clock Input Timing

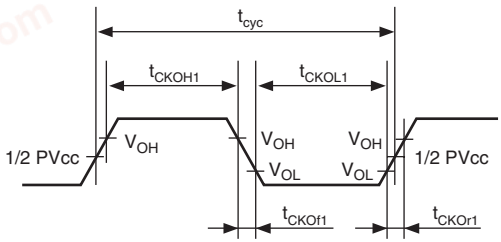


Figure 37.2 (1) CKIO Clock Output Timing 1

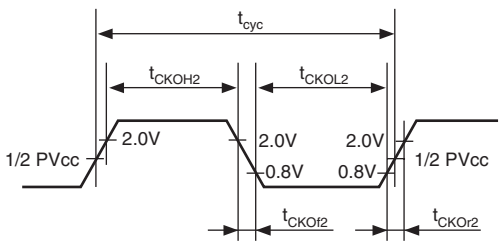
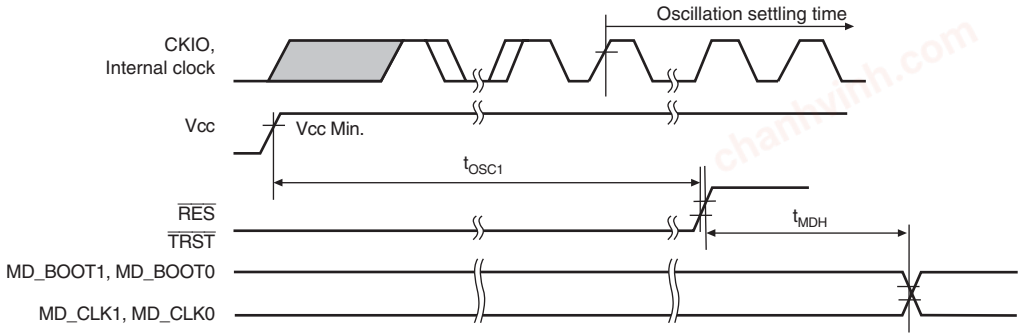


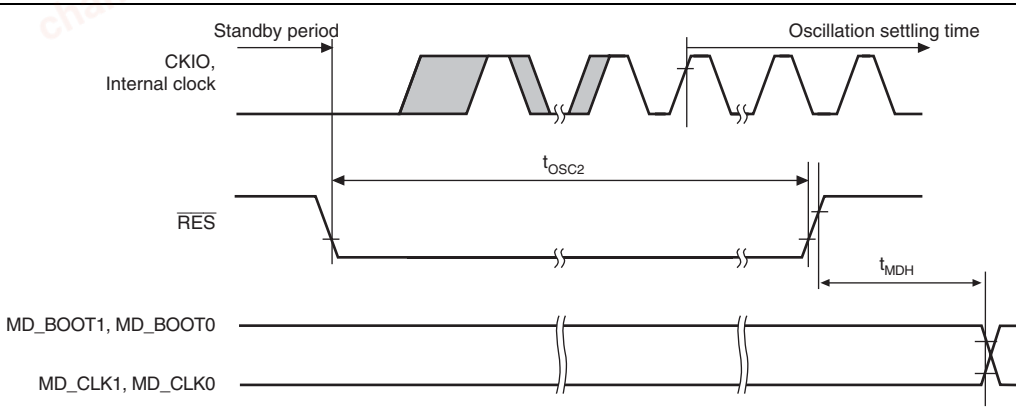
Figure 37.2 (2) CKIO Clock Output Timing 2

chanhvinh.com



Note: Oscillation settling time when the internal oscillator is used.

Figure 37.3 Power-On Oscillation Settling Time



Note: Oscillation settling time when the internal oscillator is used.

Figure 37.4 Oscillation Settling Time on Return from Standby (Return by Reset)

chanhvinh.com

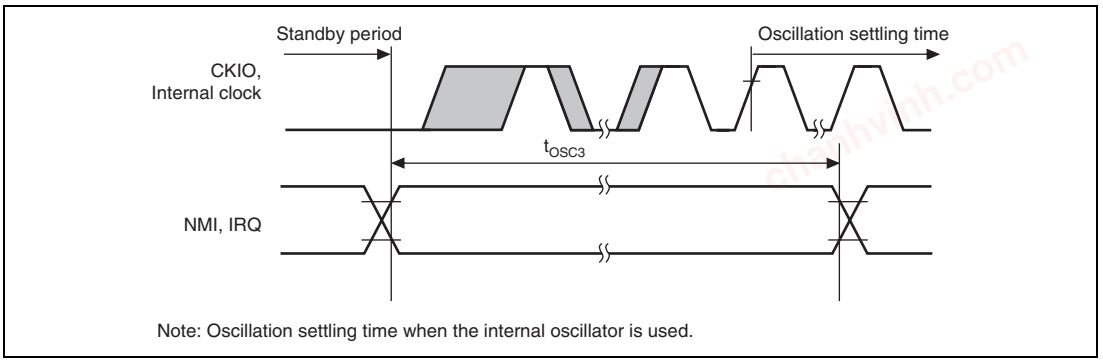


Figure 37.5 Oscillation Settling Time on Return from Standby (Return by NMI or IRQ)

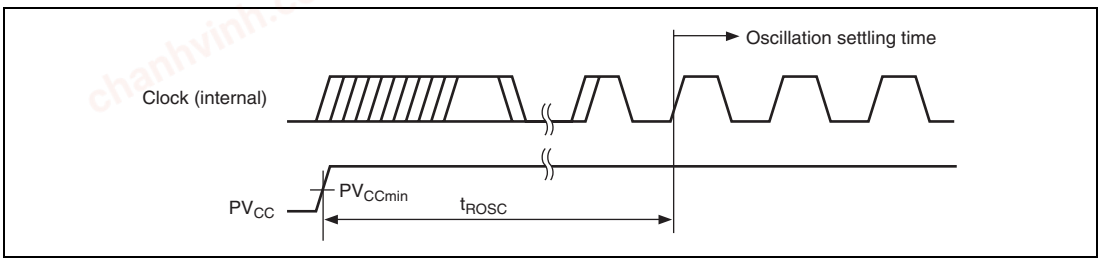


Figure 37.6 Real Time Clock Oscillation Settling Time

chanhvinh.com

37.4.2 Control Signal Timing

Table 37.6 Control Signal Timing

Item		Symbol	Bϕ = 72 MHz Min.	Max.	Unit	Figure
$\overline{\text{RES}}$ pulse width	Exit from standby mode	t_{RESW}	10	—	ms	Figure 37.7 (1)
	Other than above		20	—	t_{cyc}	
$\overline{\text{TRST}}$ pulse width		t_{TRSW}	20	—	t_{cyc}	
NMI pulse width	Exit from standby mode	t_{NMIW}	10	—	ms	Figure 37.7 (2)
	Other than above		20	—	t_{cyc}	
IRQ pulse width	Exit from standby mode	t_{IRQW}	10	—	ms	
	Other than above		20	—	t_{cyc}	
PINT pulse width		t_{PINTW}	20	—	t_{cyc}	
$\overline{\text{BREQ}}$ setup time		t_{BREQS}	$1/2t_{cyc}$ + 7	—	ns	Figure 37.8
$\overline{\text{BREQ}}$ hold time		t_{BREQH}	$1/2t_{cyc}$ + 2	—	ns	
$\overline{\text{BACK}}$ delay time		t_{BACKD}	—	$1/2t_{cyc}$ + 13	ns	
Bus buffer off time 1		t_{BOFF1}	—	15	ns	
Bus buffer off time 2		t_{BOFF2}	—	15	ns	
Bus buffer on time 1		t_{BON1}	—	15	ns	
Bus buffer on time 2		t_{BON2}	—	15	ns	
$\overline{\text{BACK}}$ setup time before the bus buffer off timing		t_{BACKS}	0	—	ns	

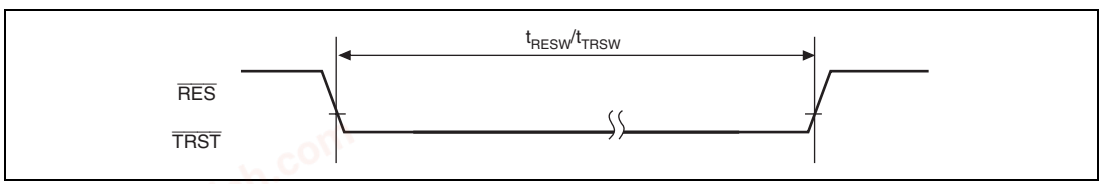


Figure 37.7 (1) Reset Input Timing

chanhvinh.com

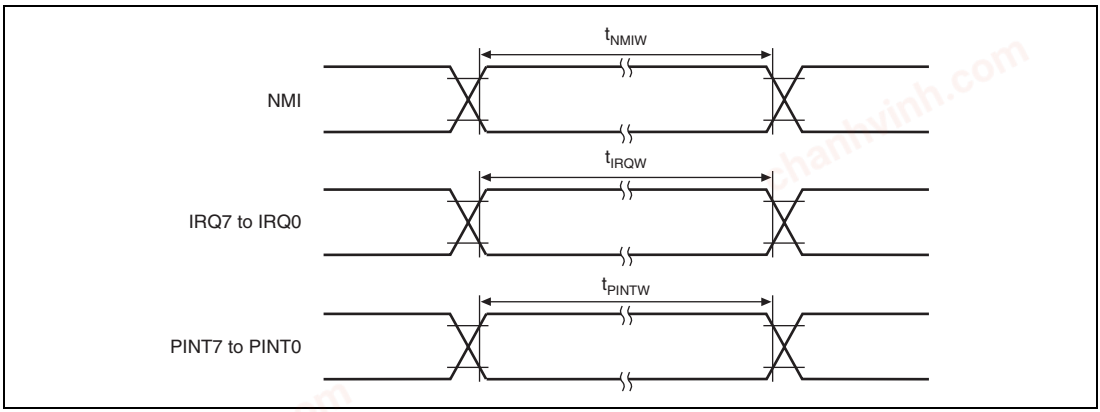


Figure 37.7 (2) Interrupt Signal Input Timing

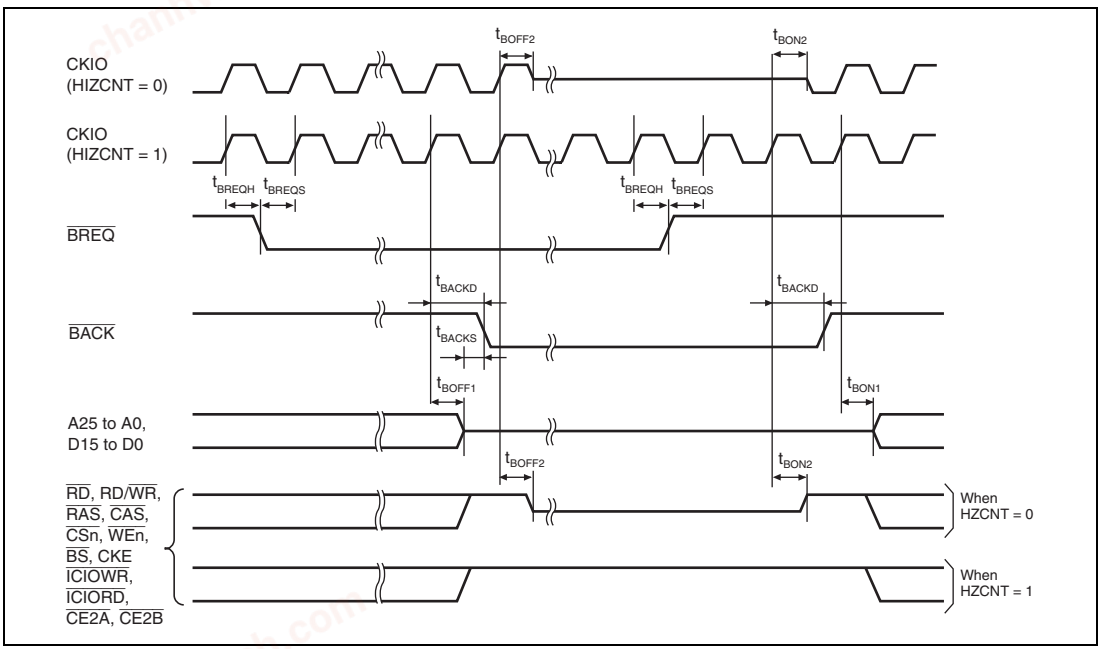


Figure 37.8 Bus Release Timing

chanhvinh.com

37.4.3 Bus Timing

Table 37.7 Bus Timing

		Bϕ = 72 MHz*[1]			
Item	**Symbol**	**Min.**	**Max.**	**Unit**	**Figure**
Address delay time 1	t_{AD1}	0/2*[3]	10.5	ns	Figures 37.9 to 37.33, 37.36 to 37.39
Address delay time 2	t_{AD2}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 10.5	ns	Figure 37.16
Address delay time 3	t_{AD3}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 10.5	ns	Figures 37.34, 37.35
Address setup time	t_{AS}	0	—	ns	Figures 37.9 to 37.12, 37.16
Chip enable setup time	t_{CS}	0	—	ns	Figures 37.9 to 37.12, 37.16
Address hold time	t_{AH}	0	—	ns	Figures 37.9 to 37.12
$\overline{BS}$ delay time	t_{BSD}	—	10.5	ns	Figures 37.9 to 37.30, 37.34, 37.36 to 37.39
$\overline{CS}$ delay time 1	t_{CSD1}	0/2*[3]	10.5	ns	Figures 37.9 to 37.33, 37.36 to 37.39
$\overline{CS}$ delay time 2	t_{CSD2}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 10.5	ns	Figures 37.34, 37.35
Read write delay time 1	t_{RWD1}	0/2*[3]	10.5	ns	Figures 37.9 to 37.33, 37.36 to 37.39
Read write delay time 2	t_{RWD2}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 10.5	ns	Figures 37.34, 37.35
Read strobe delay time	t_{RSD}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 10.5	ns	Figures 37.9 to 37.16, 37.36, 37.37
Read data setup time 1	t_{RDS1}	$1/2t_{cyc}$ + 4	—	ns	Figures 37.9 to 37.15, 37.36 to 37.39

chanhvinh.com

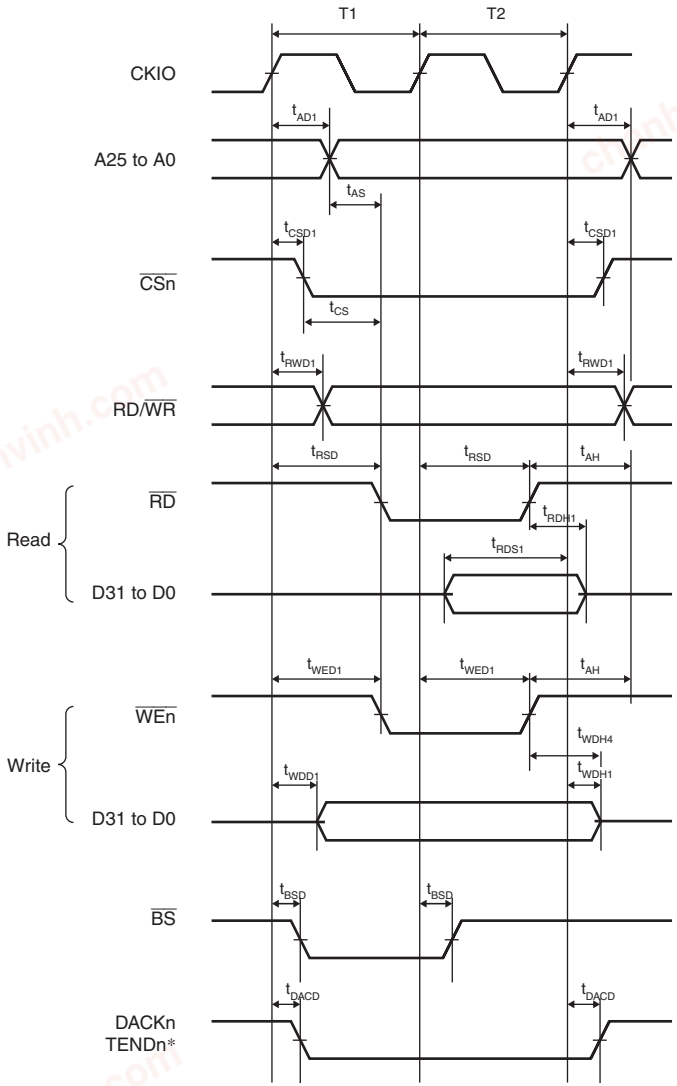
Item	Symbol	Bϕ = 72 MHz*1 Min.	Max.	Unit	Figure
Read data setup time 2	t_{RDS2}	7	—	ns	Figures 37.17 to 37.20, 37.25 to 37.27
Read data setup time 3	t_{RDS3}	$1/2t_{cyc}$ + 4	—	ns	Figure 37.16
Read data setup time 4	t_{RDS4}	$1/2t_{cyc}$ + 4	—	ns	Figure 37.34
Read data hold time 1	t_{RDH1}	0	—	ns	Figures 37.9 to 37.15, 37.36 to 37.39
Read data hold time 2	t_{RDH2}	2	—	ns	Figures 37.19 to 37.20, 37.25 to 37.27
Read data hold time 3	t_{RDH3}	0	—	ns	Figure 37.16
Read data hold time 4	t_{RDH4}	$1/2t_{cyc}$ + 6	—	ns	Figure 37.34
Write enable delay time 1	t_{WED1}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 10.5	ns	Figures 37.9 to 37.14, 37.36, 37.37
Write enable delay time 2	t_{WED2}	—	10.5	ns	Figure 37.15
Write data delay time 1	t_{WDD1}	—	10.5	ns	Figures 37.9 to 37.15, 37.36 to 37.39
Write data delay time 2	t_{WDD2}	—	10.5	ns	Figures 37.21 to 37.24, 37.28 to 37.30
Write data delay time 3	t_{WDD3}	—	$1/2t_{cyc}$ + 10.5	ns	Figure 37.34
Write data hold time 1	t_{WDH1}	1	—	ns	Figures 37.9 to 37.15, 37.36 to 37.39
Write data hold time 2	t_{WDH2}	2*4	—	ns	Figures 37.21 to 37.24, 37.28 to 37.30
Write data hold time 3	t_{WDH3}	$1/2t_{cyc}$	—	ns	Figure 37.34
Write data hold time 4	t_{WDH4}	0	—	ns	Figures 37.9 to 37.13, 37.36, 37.38
$\overline{\text{WAIT}}$ setup time	t_{WTS}	$1/2t_{cyc}$ + 4.5	—	ns	Figures 37.10 to 37.16, 37.37, 37.39
$\overline{\text{WAIT}}$ hold time	t_{WTH}	$1/2t_{cyc}$ + 3.5	—	ns	Figures 37.10 to 37.16, 37.37, 37.39
$\overline{\text{IOIS16}}$ setup time	T_{IO16S}	$1/2t_{cyc}$ + 4.5	—	ns	Figure 37.39

chanhvinh.com

Item	Symbol	Bϕ = 72 MHz*1 Min.	Max.	Unit	Figure
$\overline{IOIS16}$ hold time	T_{IO16H}	$1/2t_{cyc}$ + 3.5	—	ns	Figure 37.39
$\overline{RAS}$ delay time 1	t_{RASD1}	2*4	10.5	ns	Figures 37.17 to 37.33
$\overline{RAS}$ delay time 2	t_{RASD2}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 10.5	ns	Figures 37.34, 37.35
$\overline{CAS}$ delay time 1	t_{CASD1}	2*4	10.5	ns	Figures 37.17 to 37.33
$\overline{CAS}$ delay time 2	t_{CASD2}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 10.5	ns	Figures 37.34, 37.35
DQM delay time 1	t_{DQMD1}	2*4	10.5	ns	Figures 37.17 to 37.30
DQM delay time 2	t_{DQMD2}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 10.5	ns	Figures 37.34, 37.35
CKE delay time 1	t_{CKED1}	2*4	10.5	ns	Figure 37.32
CKE delay time 2	t_{CKED2}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 10.5	ns	Figure 37.35
$\overline{AH}$ delay time	t_{AHD}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 10.5	ns	Figure 37.13
Multiplexed address delay time	t_{MAD}	—	10.5	ns	Figure 37.13
Multiplexed address hold time	t_{MAH}	1	—	ns	Figure 37.13
Address setup time for $\overline{AH}$	t_{AVVH}	$1/2t_{cyc}$ – 2	—	ns	Figure 37.13
DACK, TEND delay time	t_{DACD}	Refer to section 37.4.4, Direct Memory Access Controller Timing		ns	Figures 37.9 to 37.30, 37.34, 37.36 to 37.39
$\overline{ICIORD}$ delay time	t_{ICRSD}	—	$1/2t_{cyc}$ + 10.5	ns	Figures 37.38, 37.39
$\overline{ICIOWR}$ delay time	t_{ICWSD}	—	$1/2t_{cyc}$ + 10.5	ns	Figures 37.38, 37.39

Notes:
1. The maximum value (fmax) of Bϕ (external bus clock) depends on the number of wait cycles and the system configuration of your board.
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.
3. Values when SDRAM is used. Be sure to make necessary settings in ACSWR. (For details, refer to the descriptions from section 9.4.8, AC Characteristics Switching Register (ACSWR), to section 9.4.10, Sequence to Write to ACSWR.)
4. Be sure to make necessary settings in ACSWR. (For details, refer to the descriptions from section 9.4.8, AC Characteristics Switching Register (ACSWR), to section 9.4.10, Sequence to Write to ACSWR.)

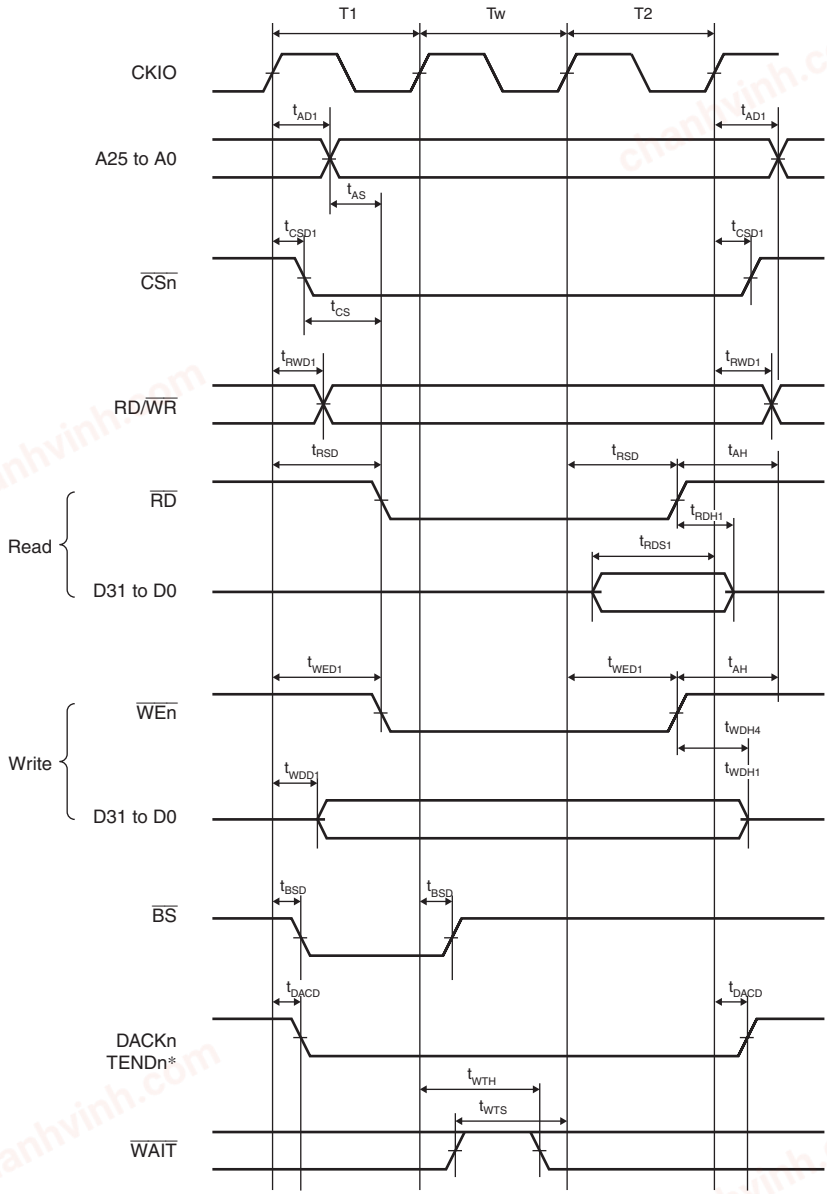
chanhvinh.com



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

Figure 37.9 Basic Bus Timing for Normal Space (No Wait)

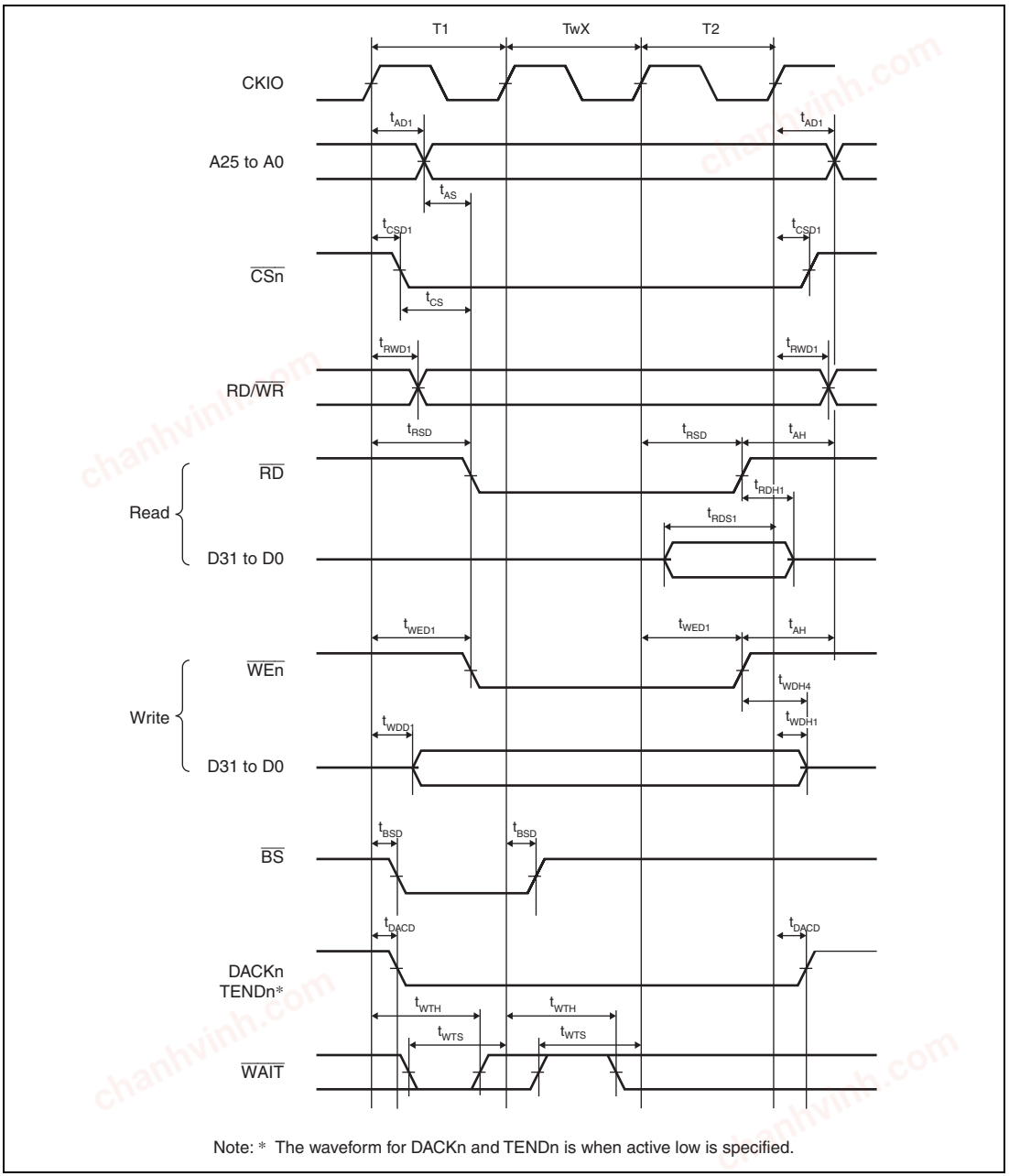
chanhvinh.com



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

Figure 37.10 Basic Bus Timing for Normal Space (One Software Wait Cycle)

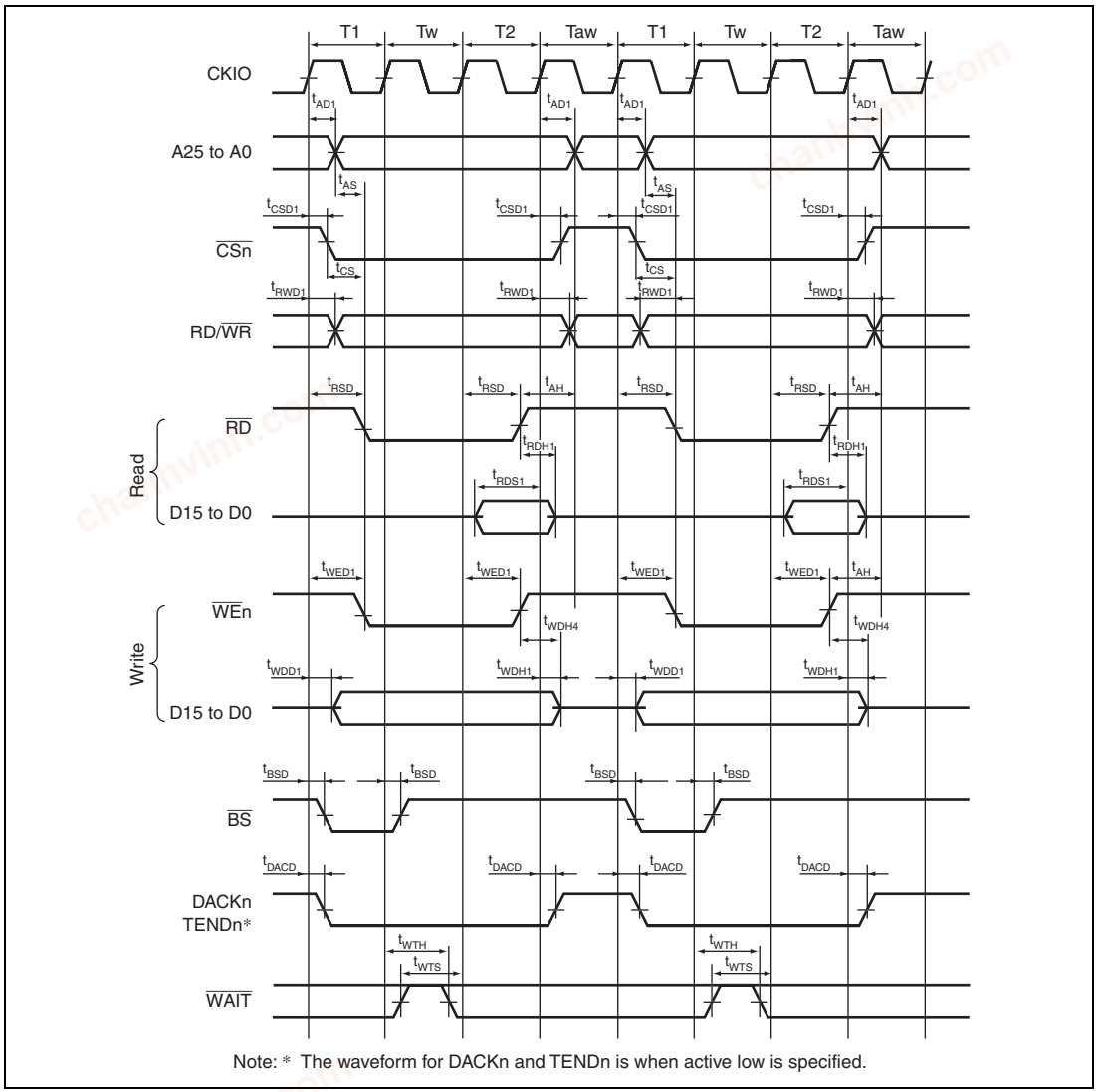
chanhvinh.com



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

Figure 37.11 Basic Bus Timing for Normal Space (One External Wait Cycle)

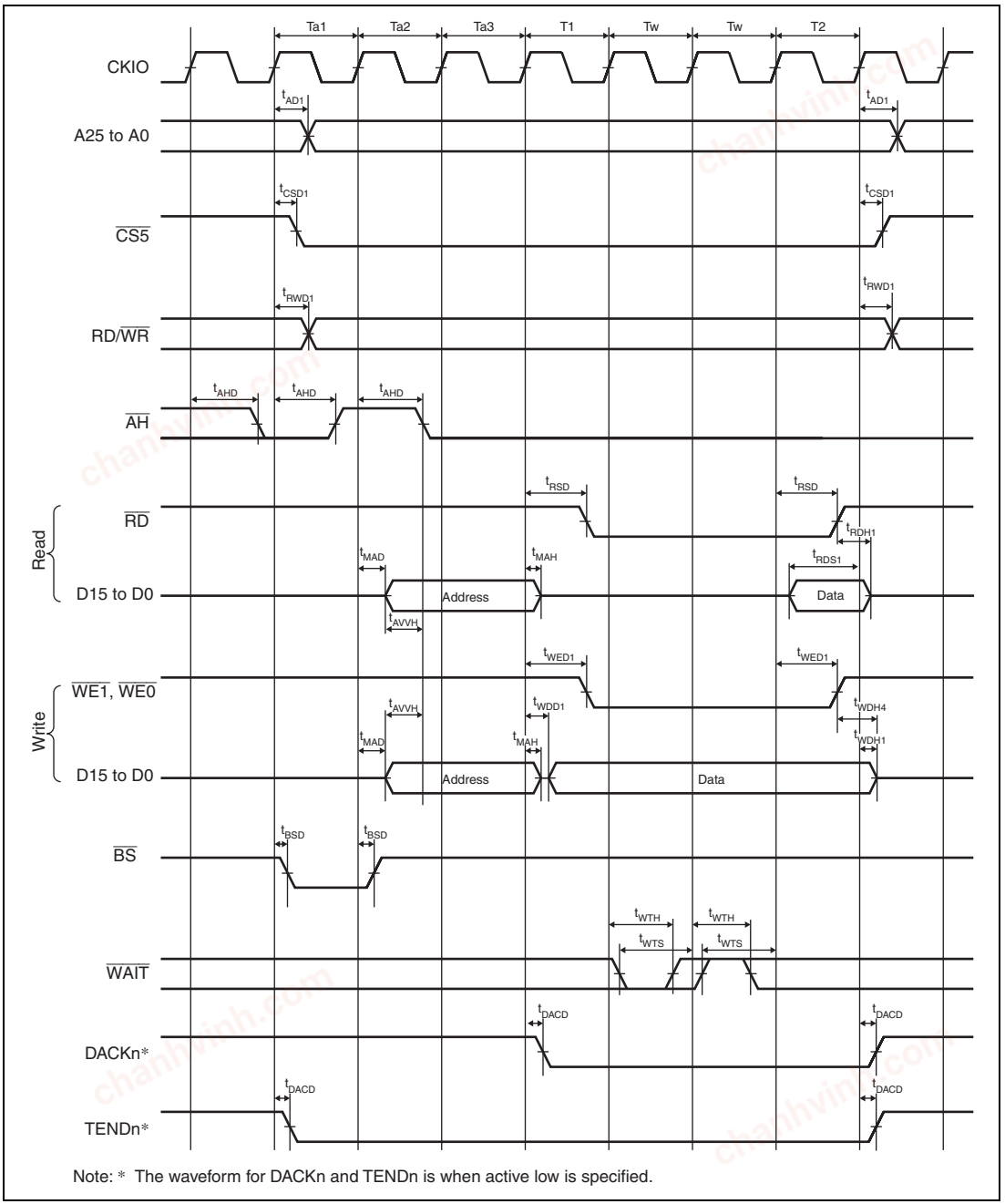
chanhvinh.com



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

Figure 37.12 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 37.13 MPX-I/O Interface Bus Cycle
(Three Address Cycles, One Software Wait Cycle, One External Wait Cycle)

chanhvinh.com

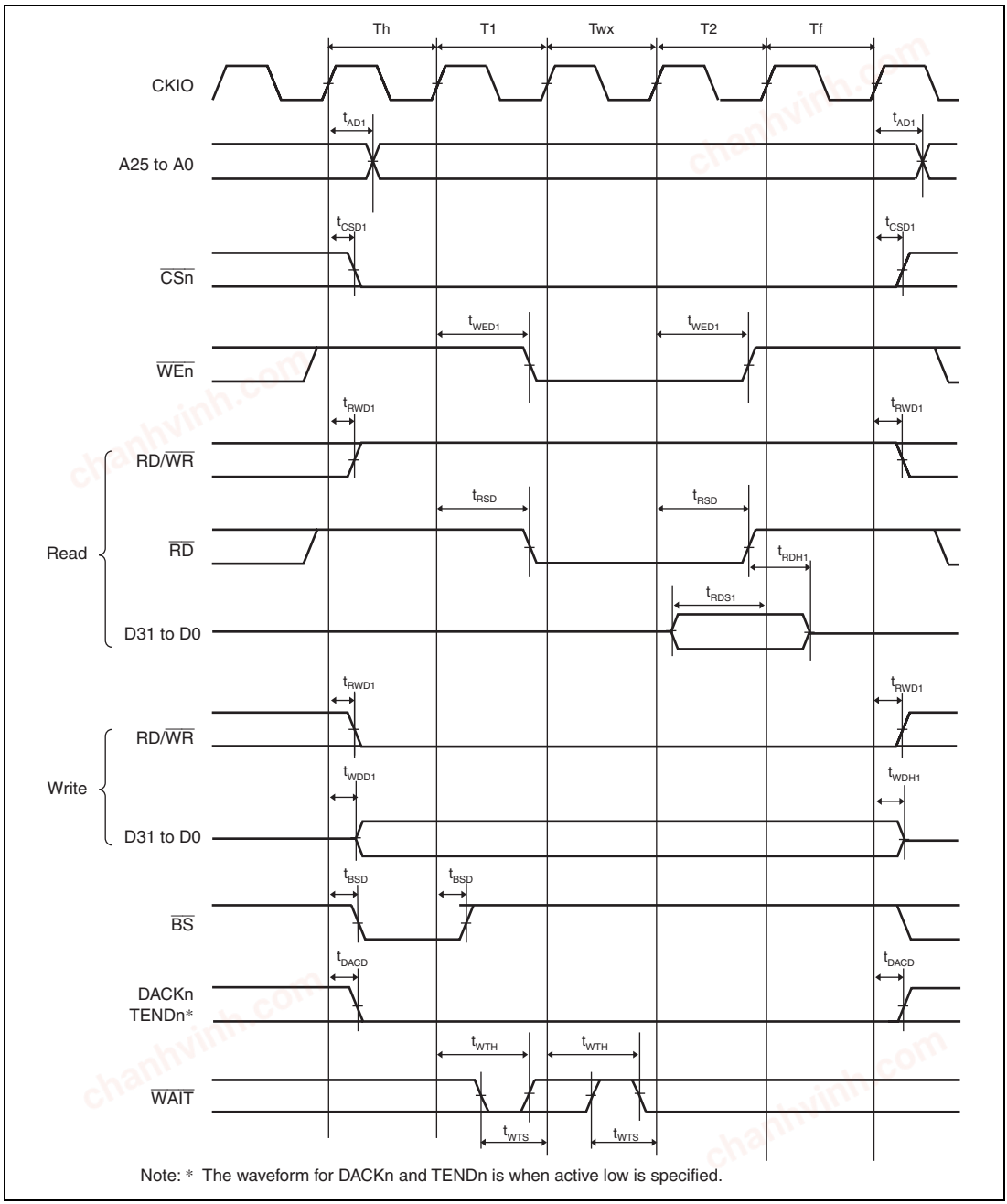


Figure 37.14 Bus Cycle of SRAM with Byte Selection (SW = 1 Cycle, HW = 1 Cycle, One Asynchronous External Wait Cycle, BAS = 0 (Write Cycle UB/LB Control))

chanhvinh.com

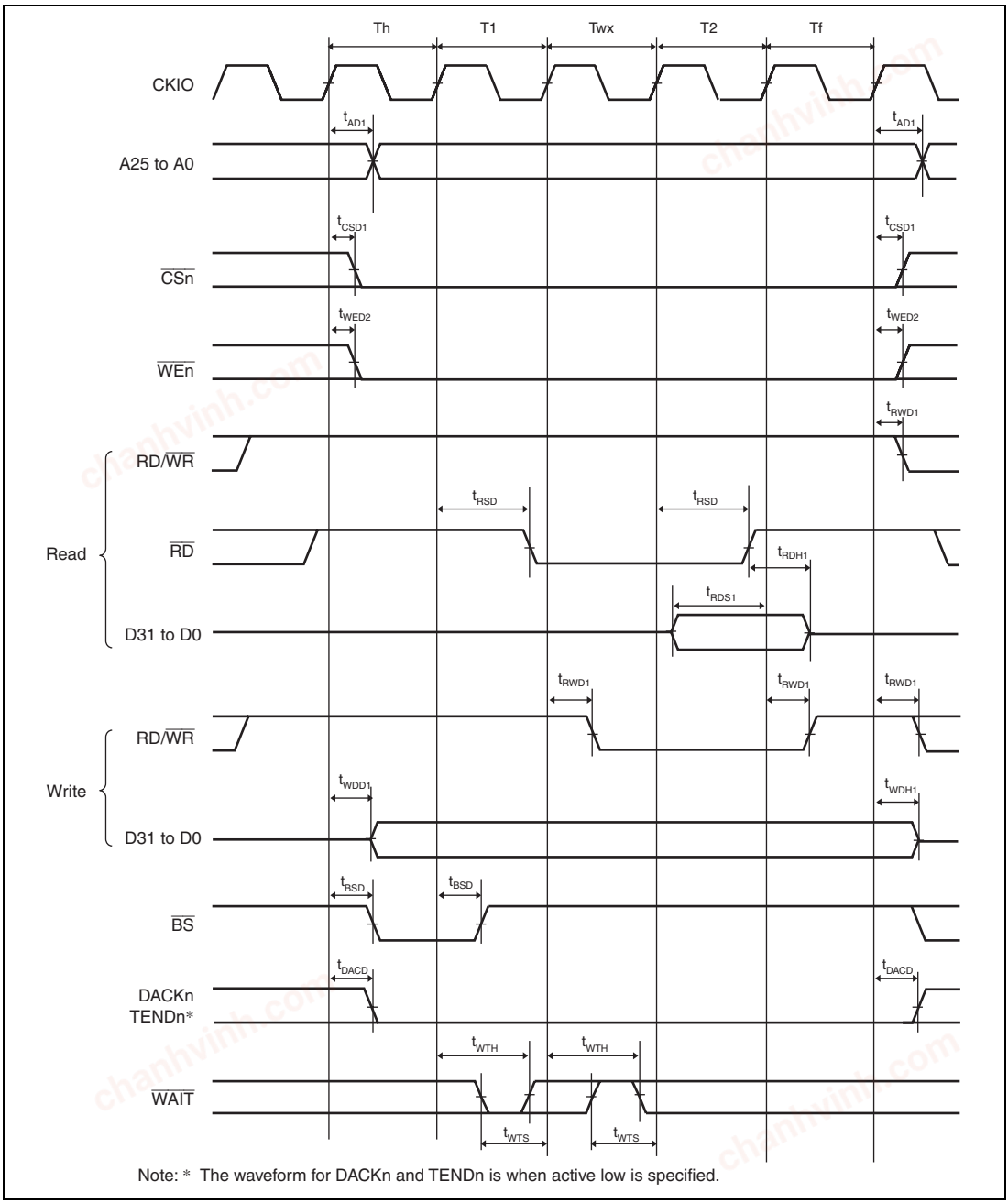
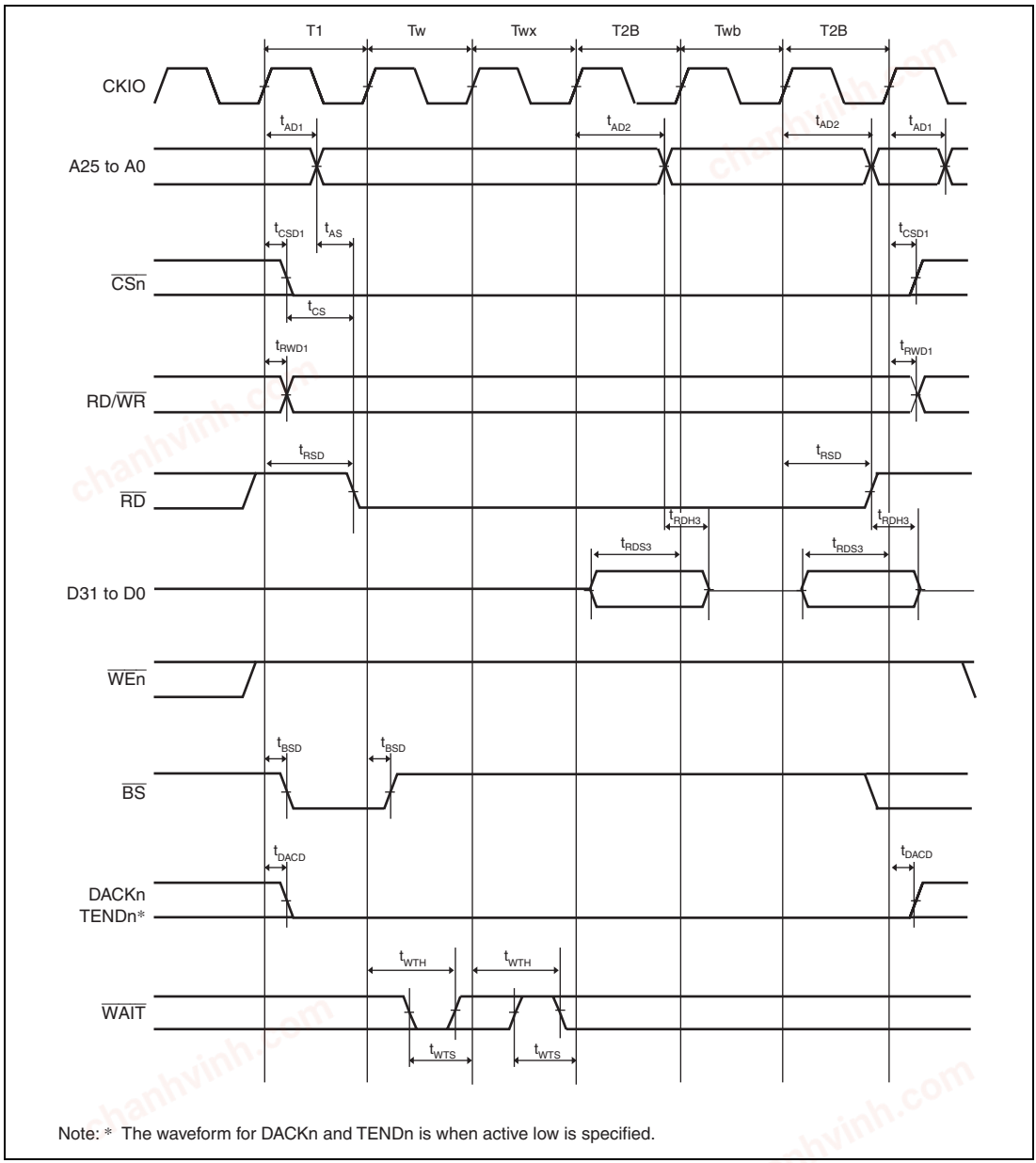


Figure 37.15 Bus Cycle of SRAM with Byte Selection (SW = 1 Cycle, HW = 1 Cycle, One Asynchronous External Wait Cycle, BAS = 1 (Write Cycle WE Control))

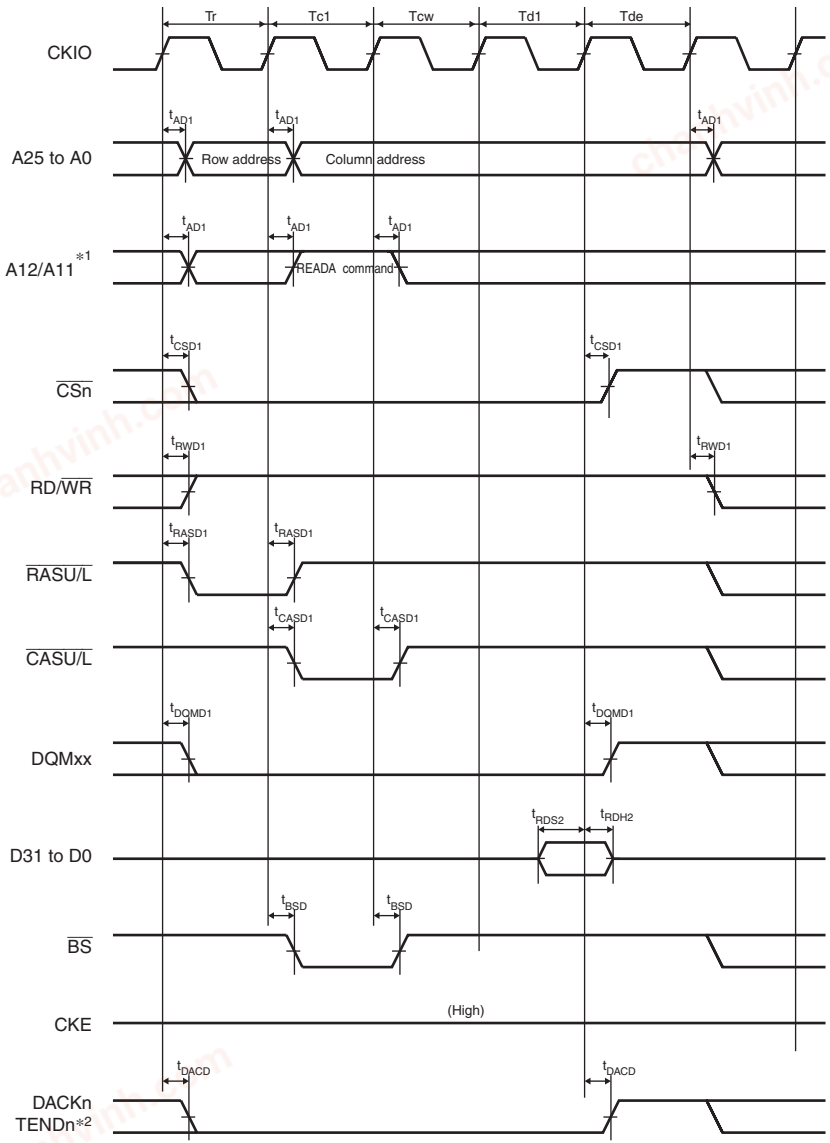
chanhvinh.com



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

Figure 37.16 Burst ROM Read Cycle
(One Software Wait Cycle, One Asynchronous External Burst Wait Cycle, Two Burst)

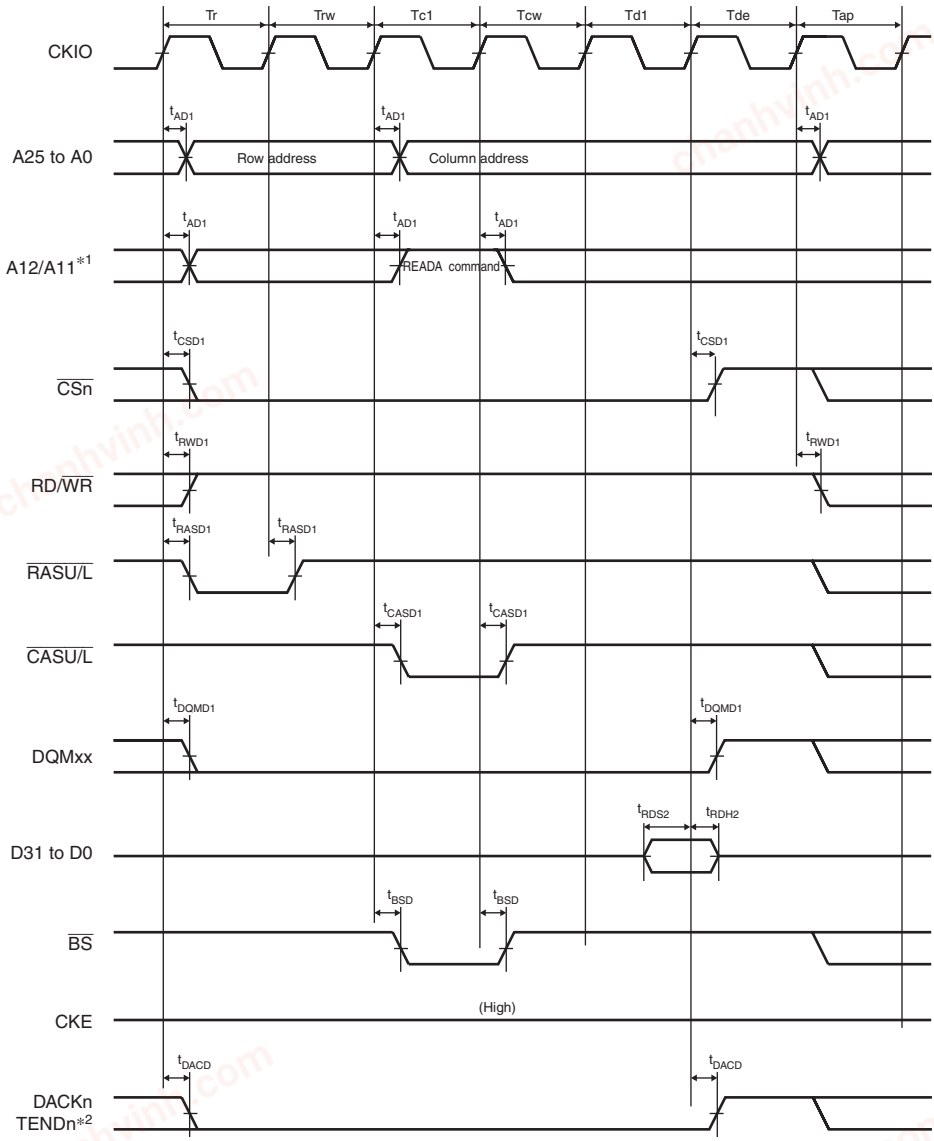
chanhvinh.com



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 37.17 Synchronous DRAM Single Read Bus Cycle (Auto Precharge, CAS Latency 2, WTRCD = 0 Cycle, WTRP = 0 Cycle)

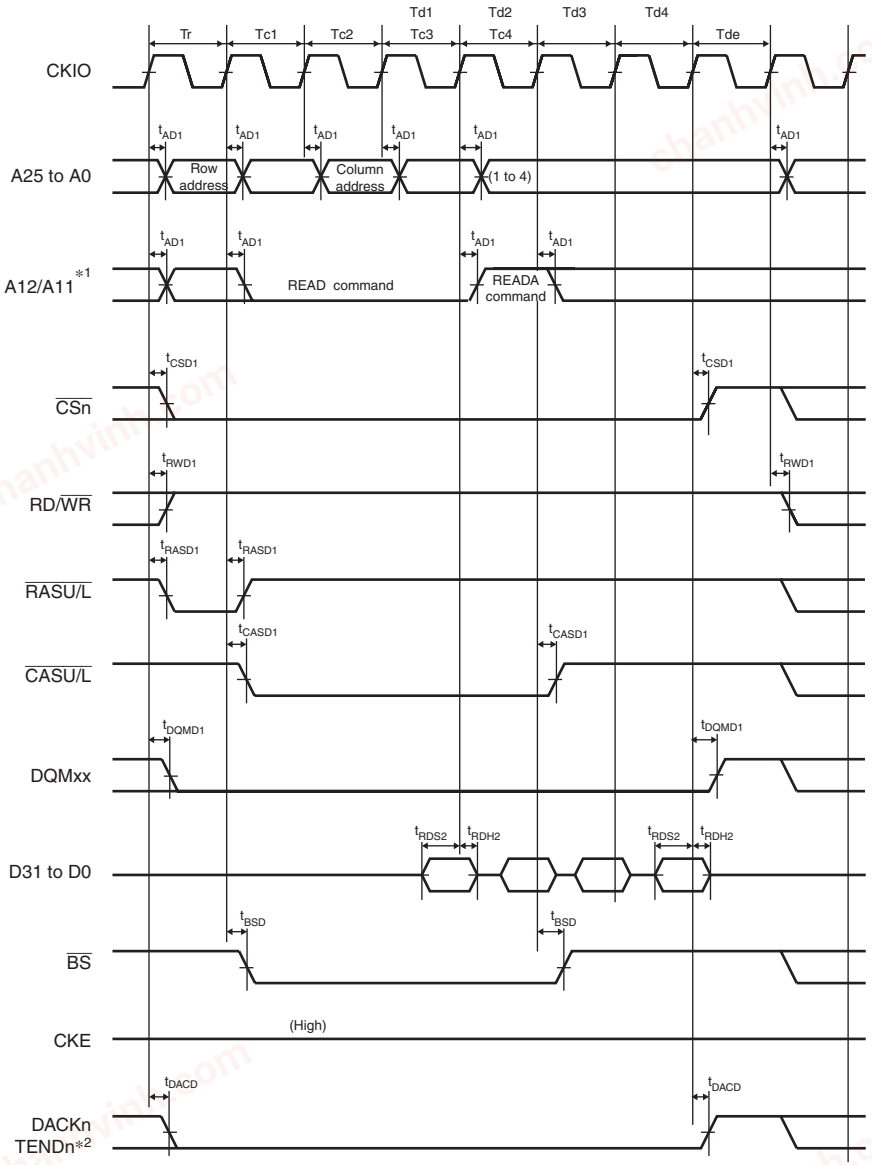
chanhvinh.com



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 37.18 Synchronous DRAM Single Read Bus Cycle (Auto Precharge, CAS Latency 2, WTRCD = 1 Cycle, WTRP = 1 Cycle)

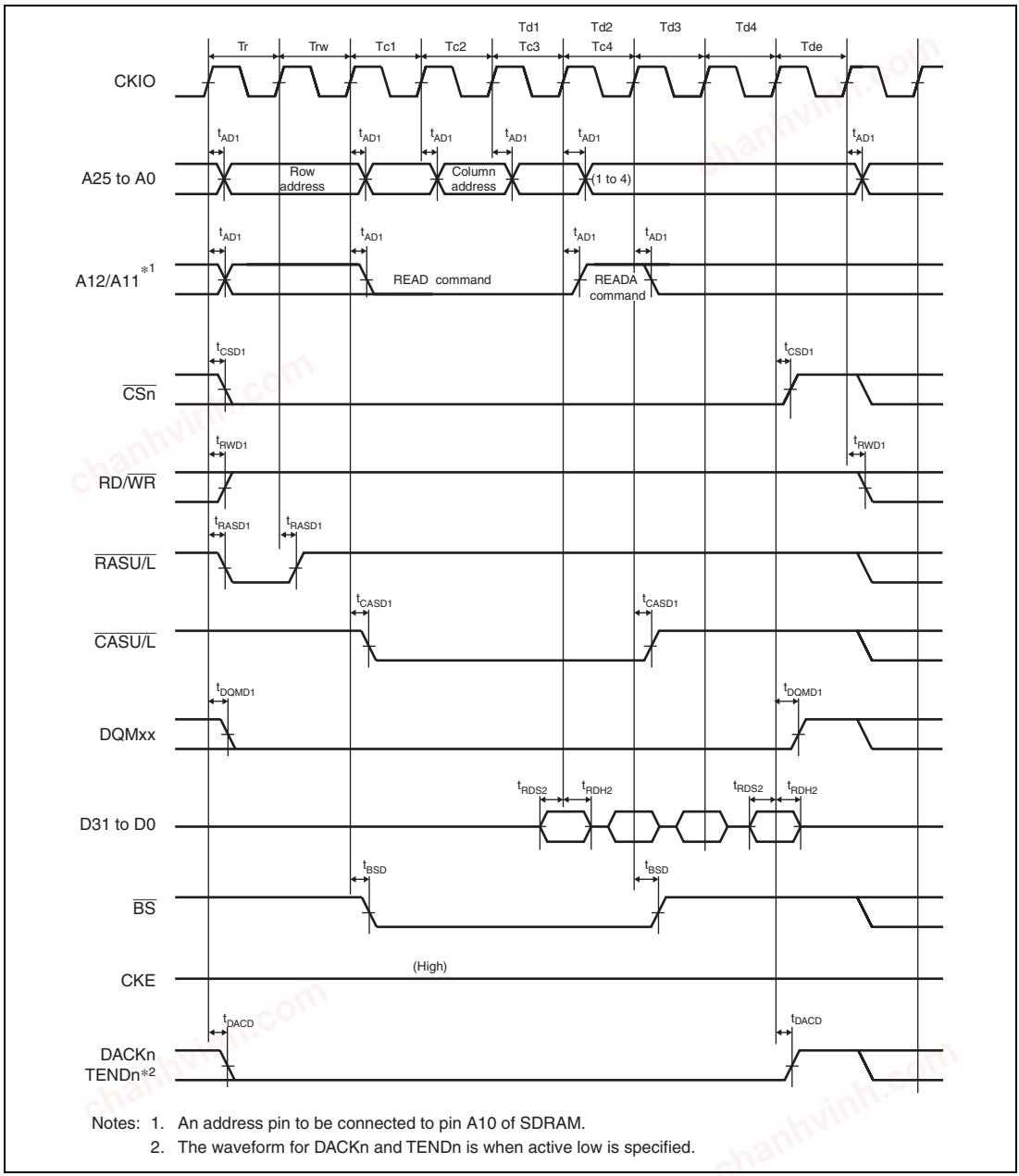
chanhvinh.com



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 37.19 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycles) (Auto Precharge, CAS Latency 2, WTRCD = 0 Cycle, WTRP = 1 Cycle)

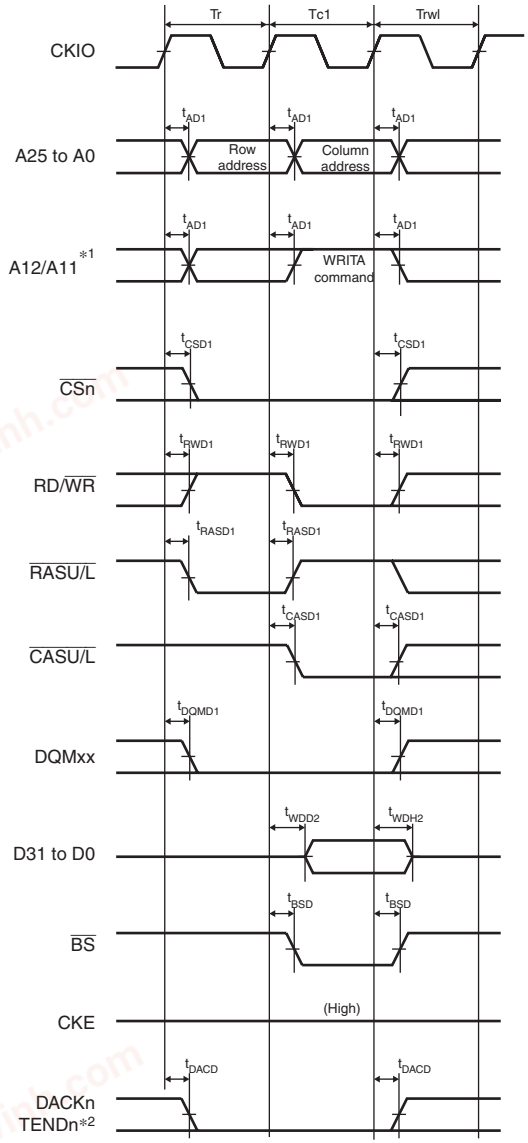
chanhvinh.com



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 37.20 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycles) (Auto Precharge, CAS Latency 2, WTRCD = 1 Cycle, WTRP = 0 Cycle)

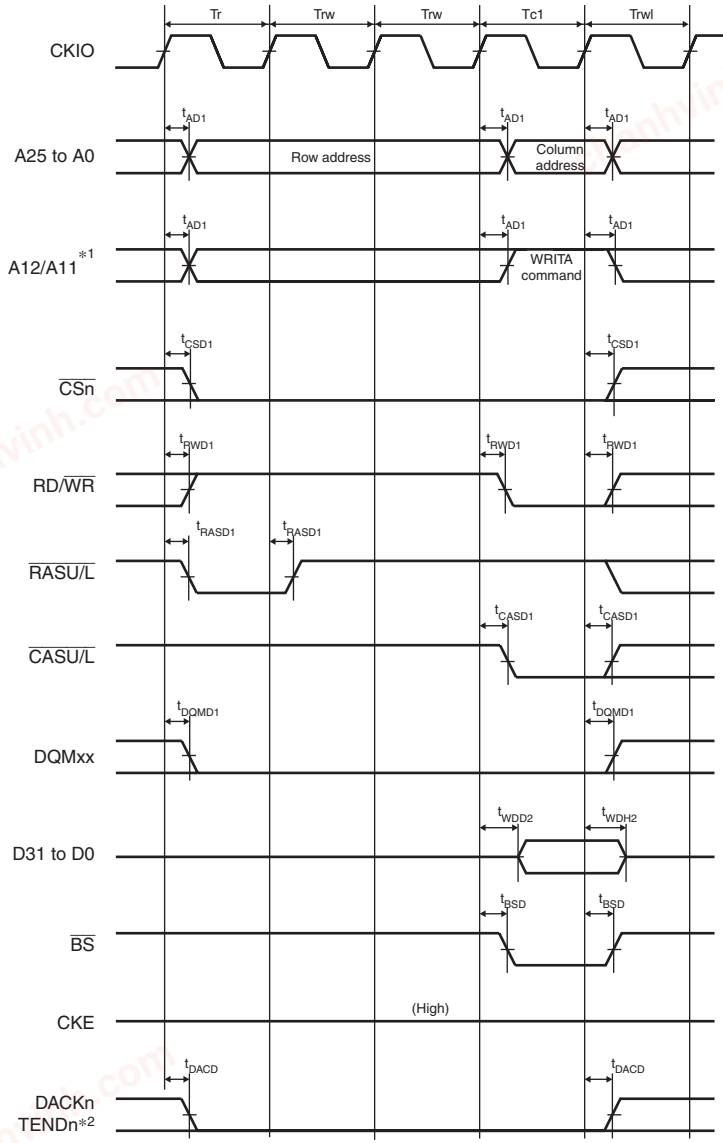
chanhvinh.com



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 37.21 Synchronous DRAM Single Write Bus Cycle (Auto Precharge, TRWL = 1 Cycle)

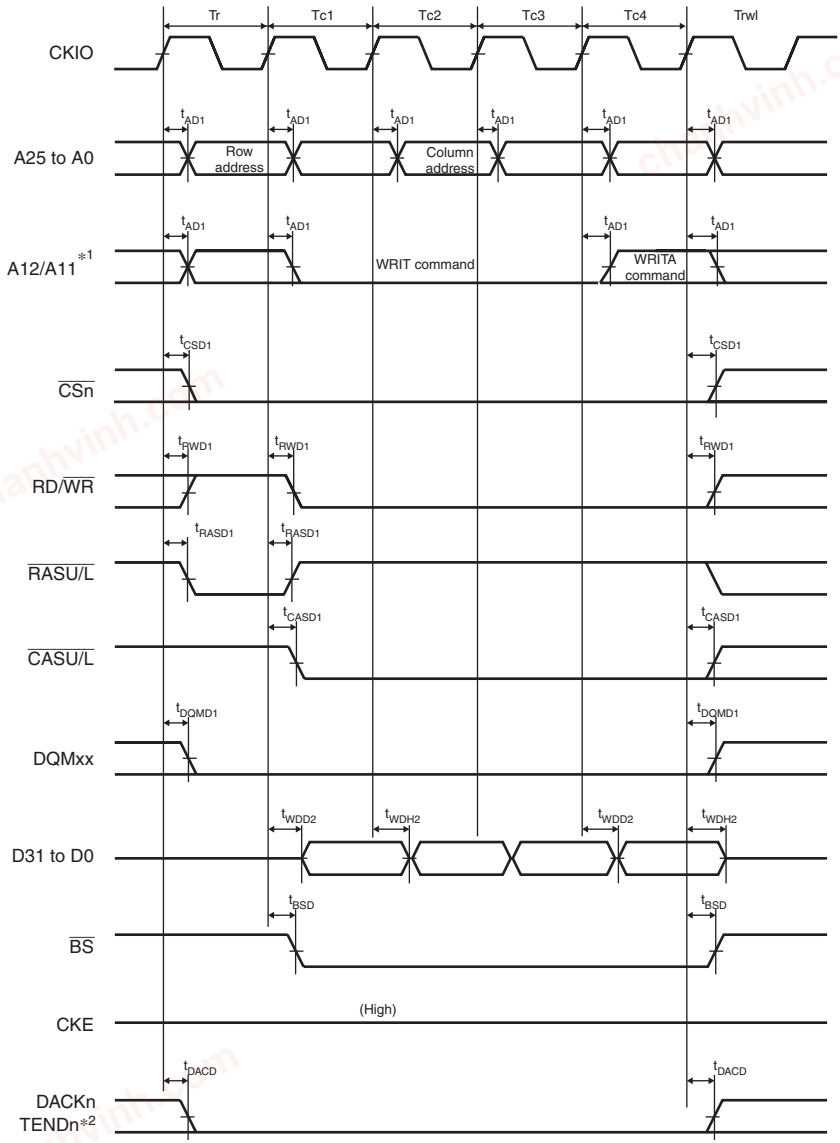
chanhvinh.com



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 37.22 Synchronous DRAM Single Write Bus Cycle (Auto Precharge, WTRCD = 2 Cycles, TRWL = 1 Cycle)

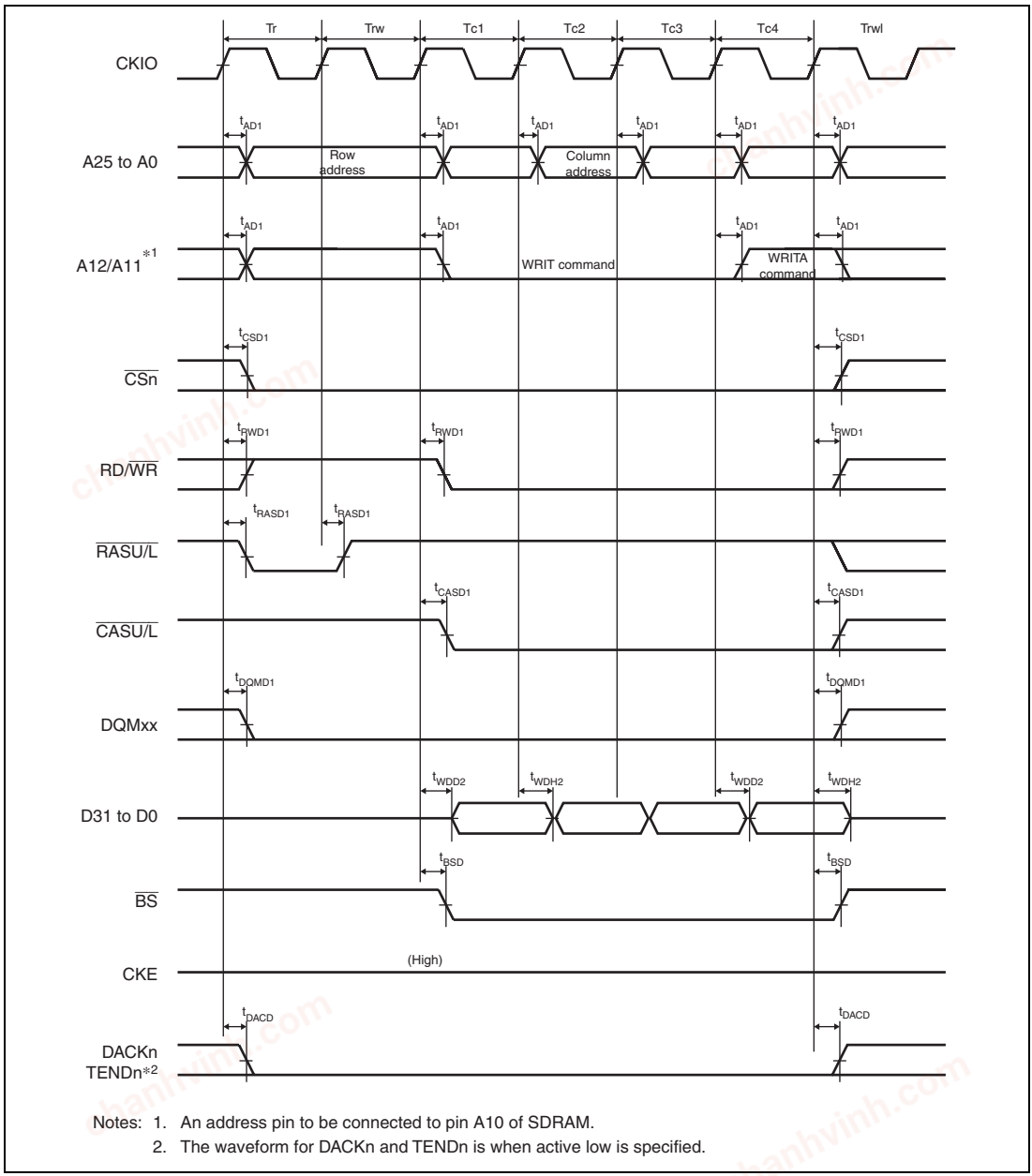
chanhvinh.com



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 37.23 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles) (Auto Precharge, WTRCD = 0 Cycle, TRWL = 1 Cycle)

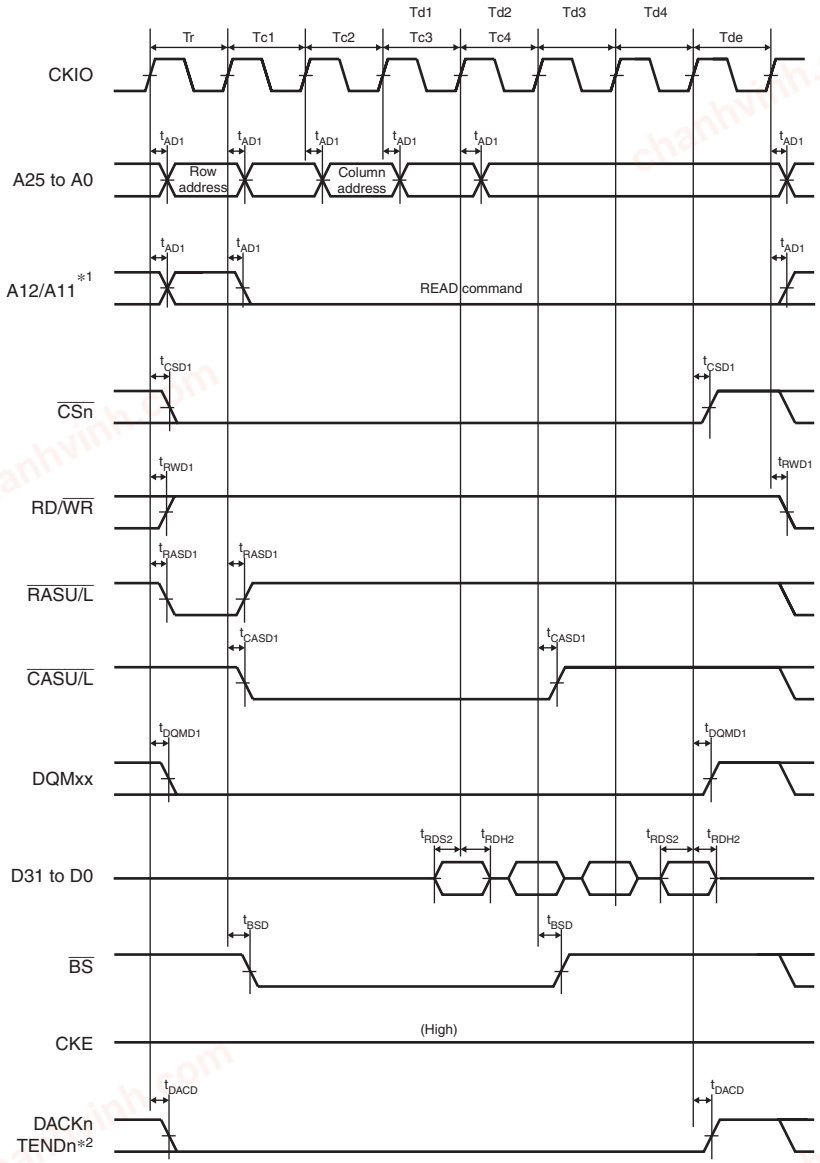
chanhvinh.com



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 37.24 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles) (Auto Precharge, WTRCD = 1 Cycle, TRWL = 1 Cycle)

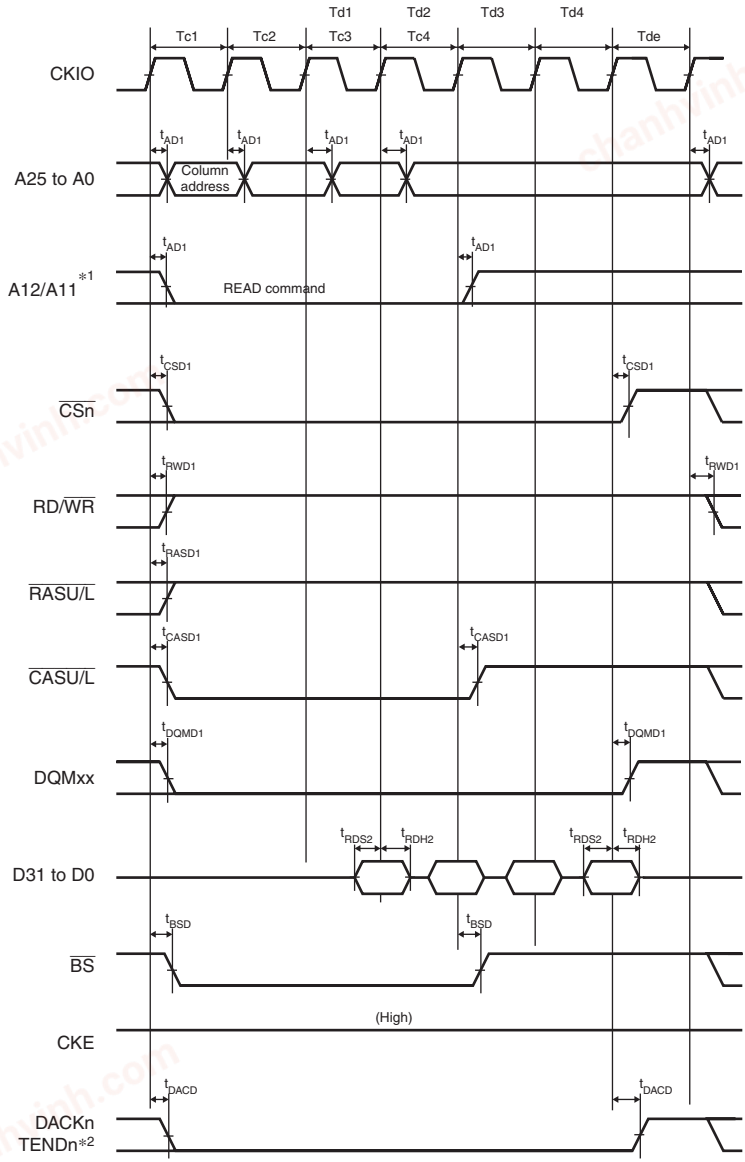
chanhvinh.com



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 37.25 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycles)
(Bank Active Mode: ACT + READ Commands, CAS Latency 2, WTRCD = 0 Cycle)

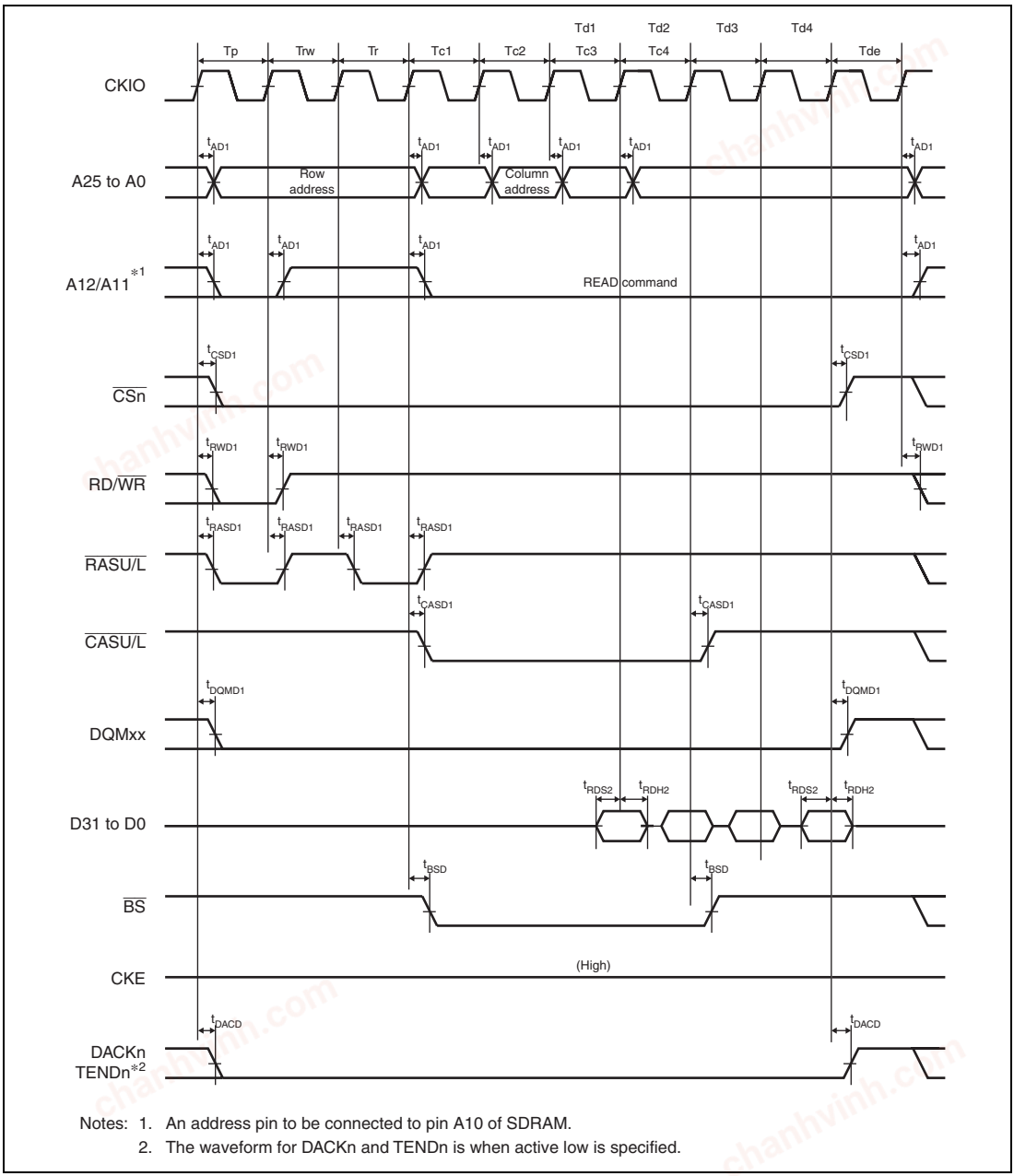
chanhvinh.com



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 37.26 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycles)
(Bank Active Mode: READ Command, Same Row Address, CAS Latency 2, WTRCD = 0 Cycle)

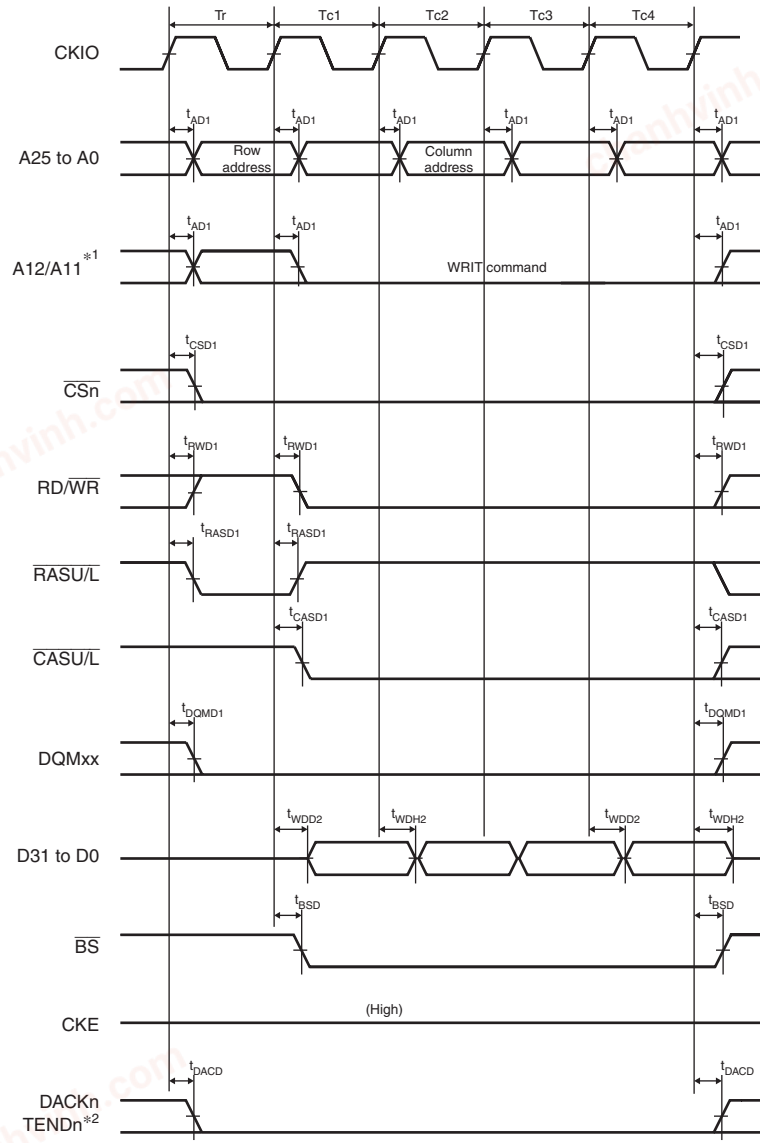
chanhvinh.com



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 37.27 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)

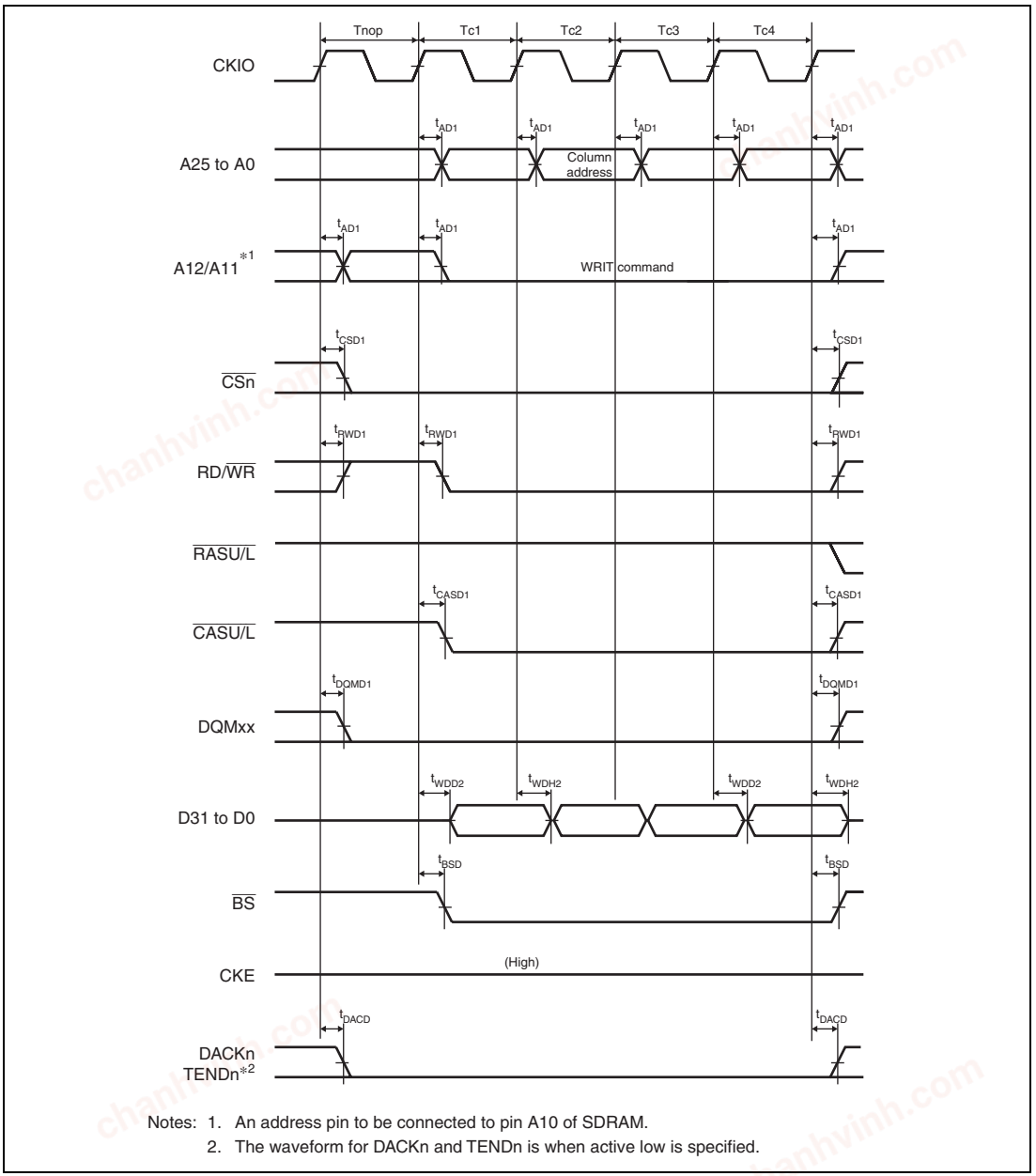
chanhvinh.com



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 37.28 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles)
(Bank Active Mode: ACT + WRITE Commands, WTRCD = 0 Cycle, TRWL = 0 Cycle)

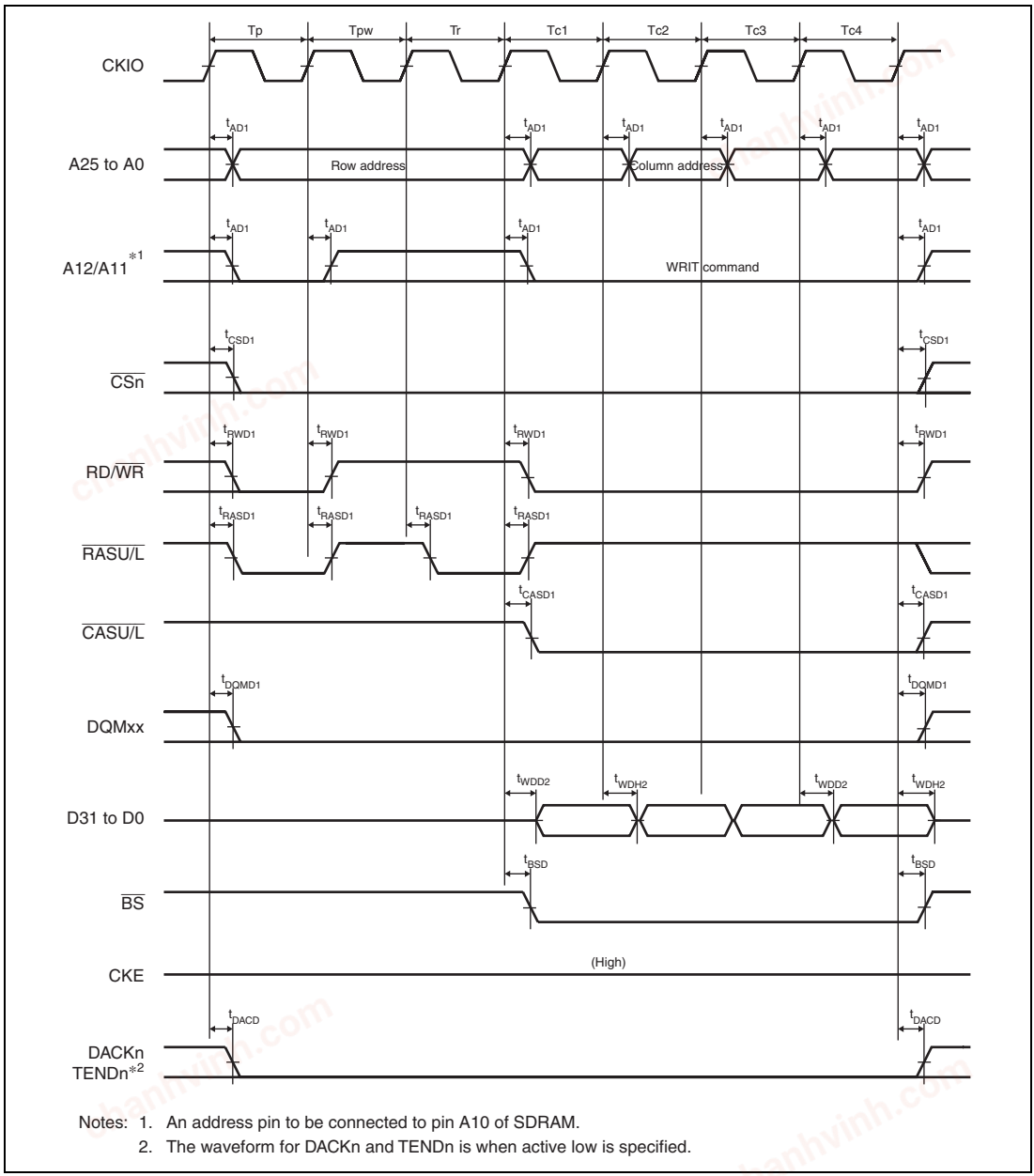
chanhvinh.com



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 37.29 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles) (Bank Active Mode: WRITE Command, Same Row Address, WTRCD = 0 Cycle, TRWL = 0 Cycle)

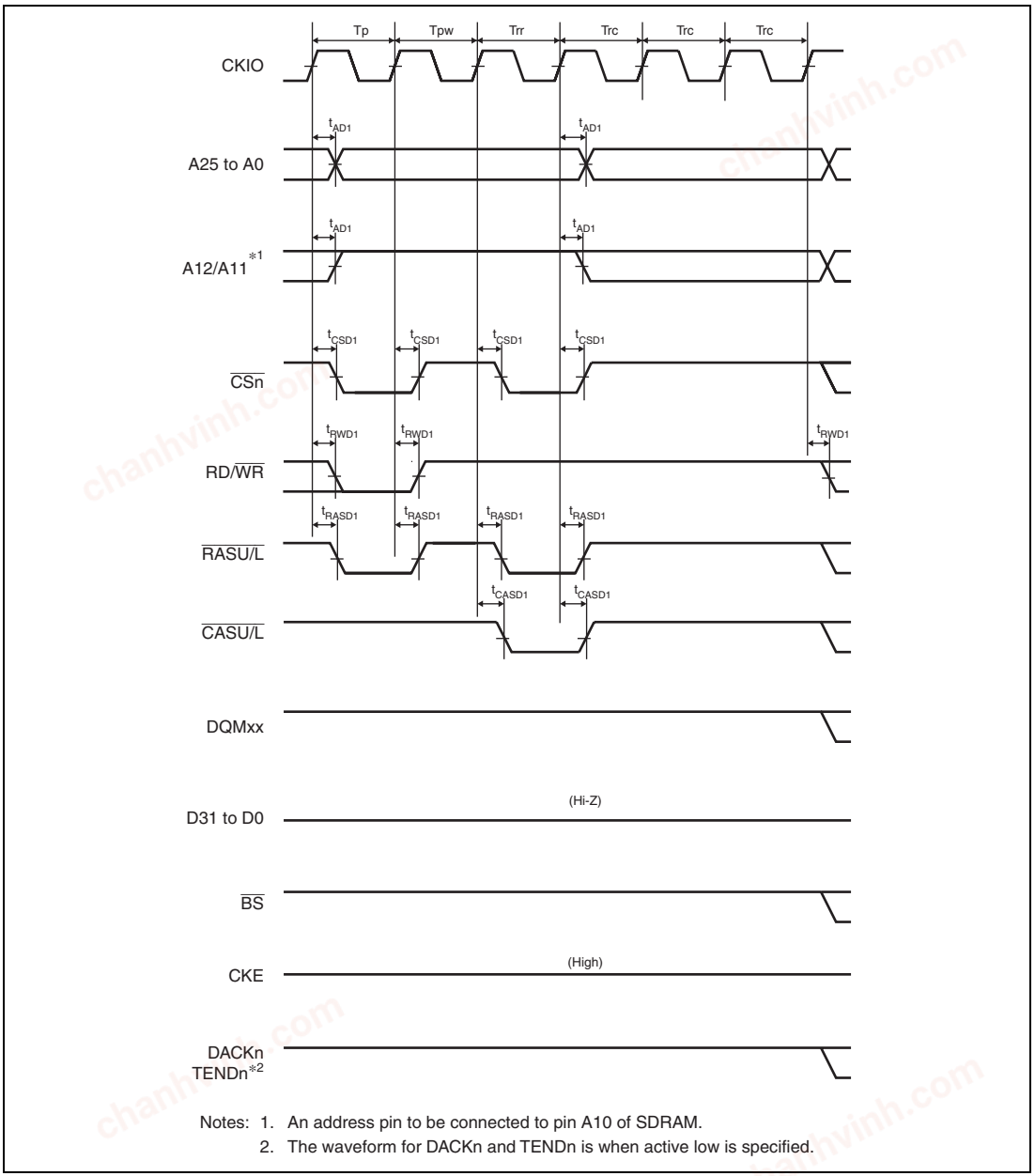
chanhvinh.com



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 37.30 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)

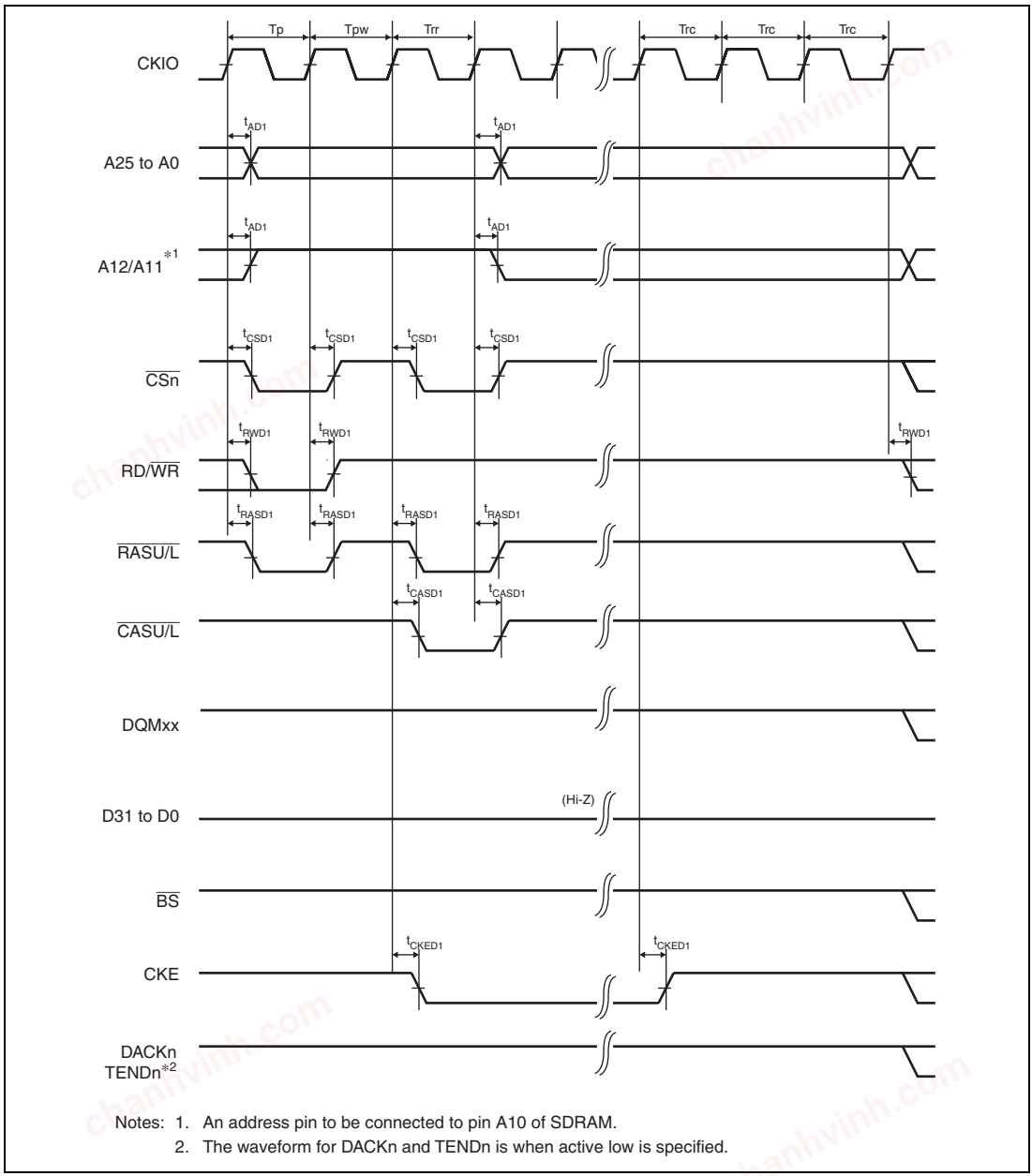
chanhvinh.com



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 37.31 Synchronous DRAM Auto-Refreshing Timing (WTRP = 1 Cycle, WTRC = 3 Cycles)

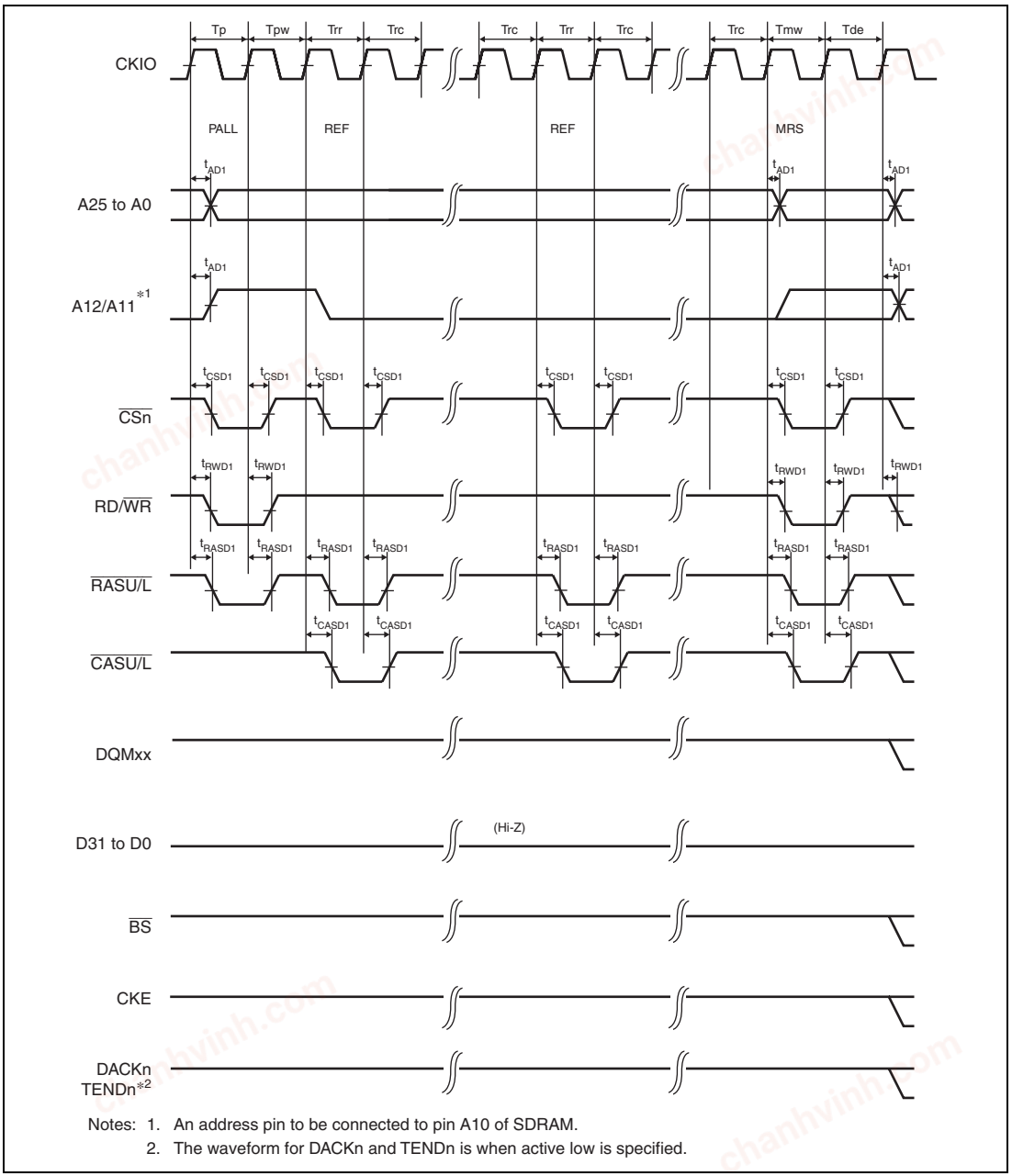
chanhvinh.com



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 37.32 Synchronous DRAM Self-Refreshing Timing (WTRP = 1 Cycle)

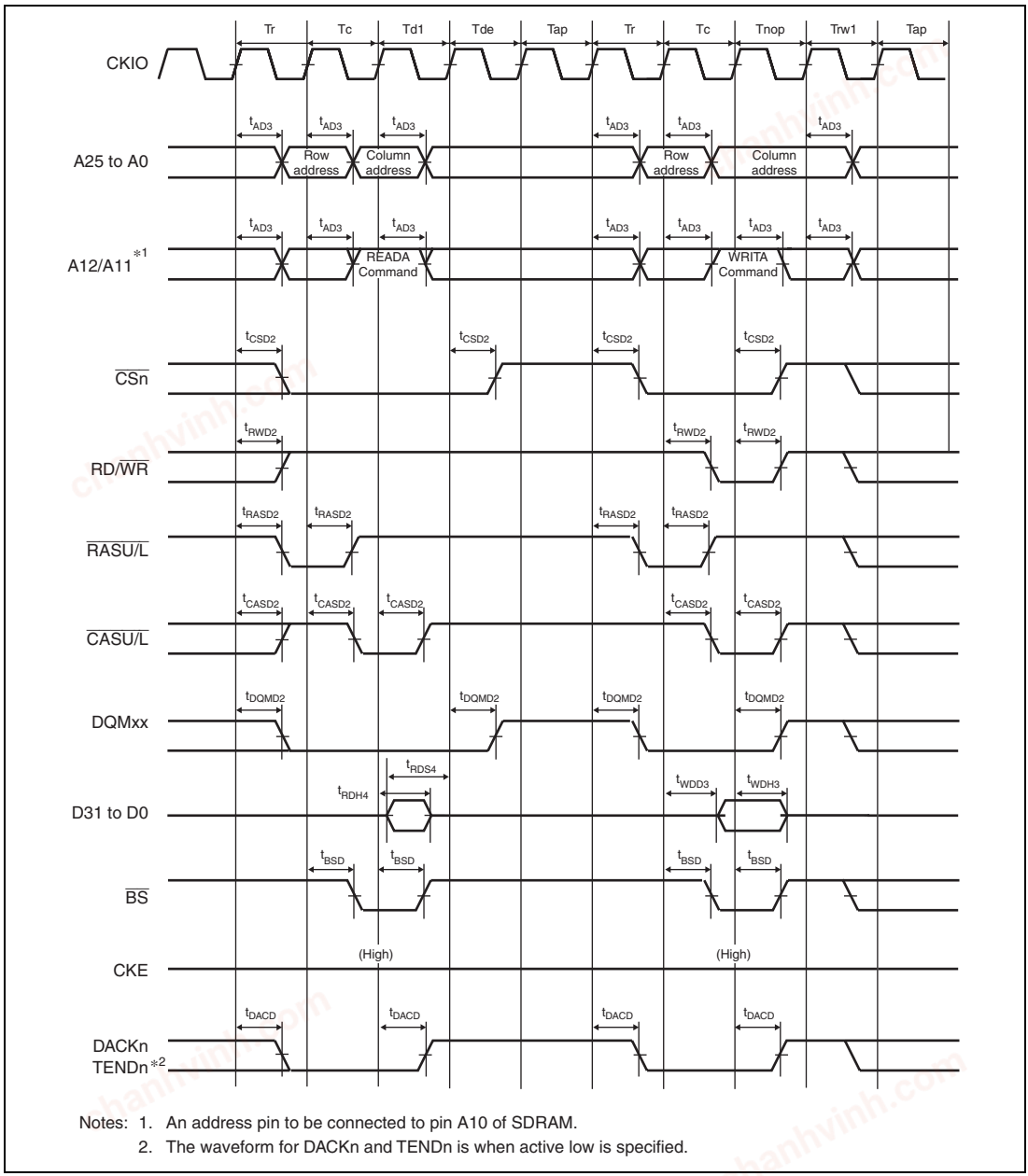
chanhvinh.com



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 37.33 Synchronous DRAM Mode Register Write Timing (WTRP = 1 Cycle)

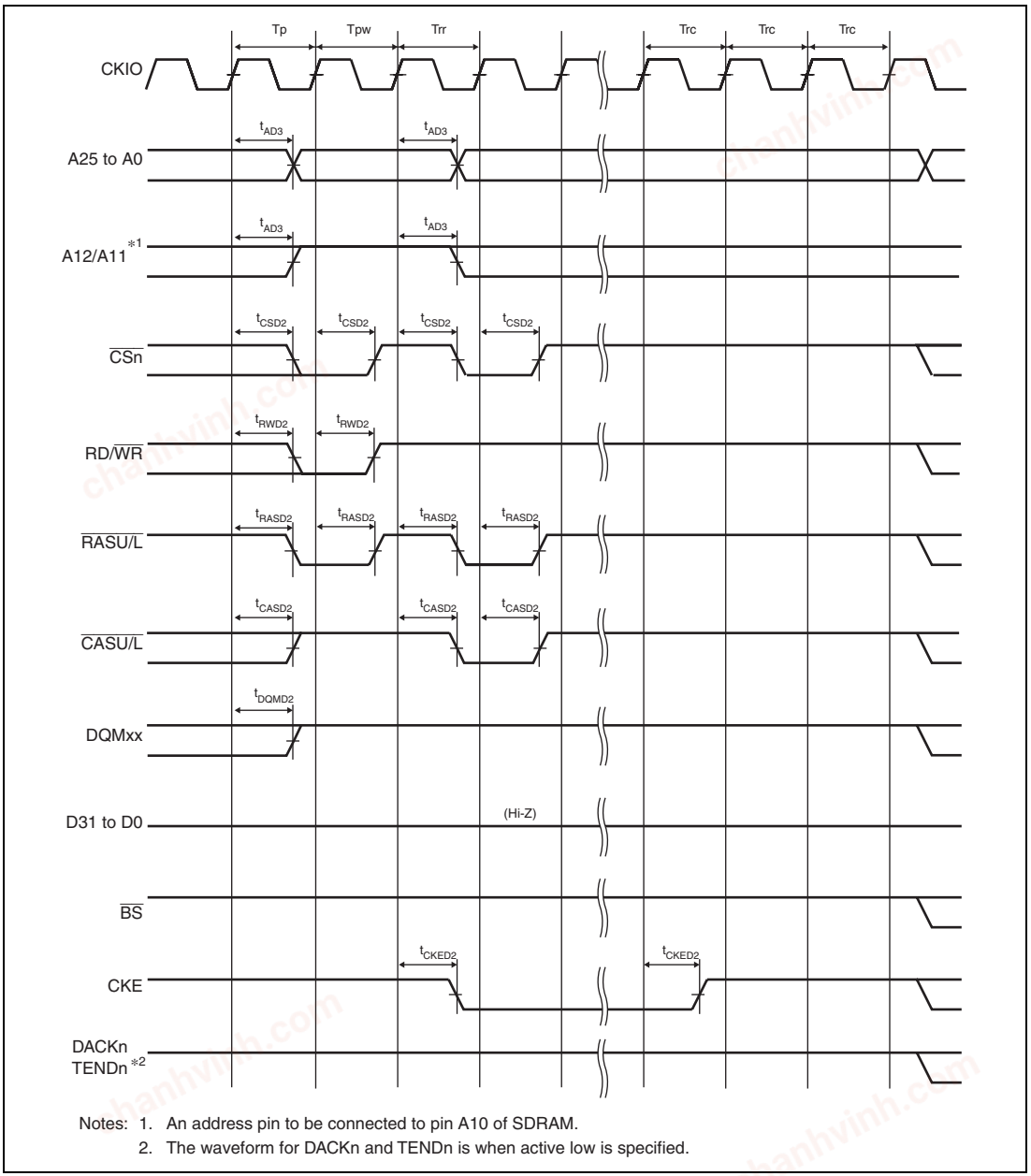
chanhvinh.com



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 37.34 Synchronous DRAM Access Timing in Low-Frequency Mode (Auto-Precharge, TRWL = 2 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 37.35 Synchronous DRAM Self-Refreshing Timing in Low-Frequency Mode (WTRP = 2 Cycles)

chanhvinh.com

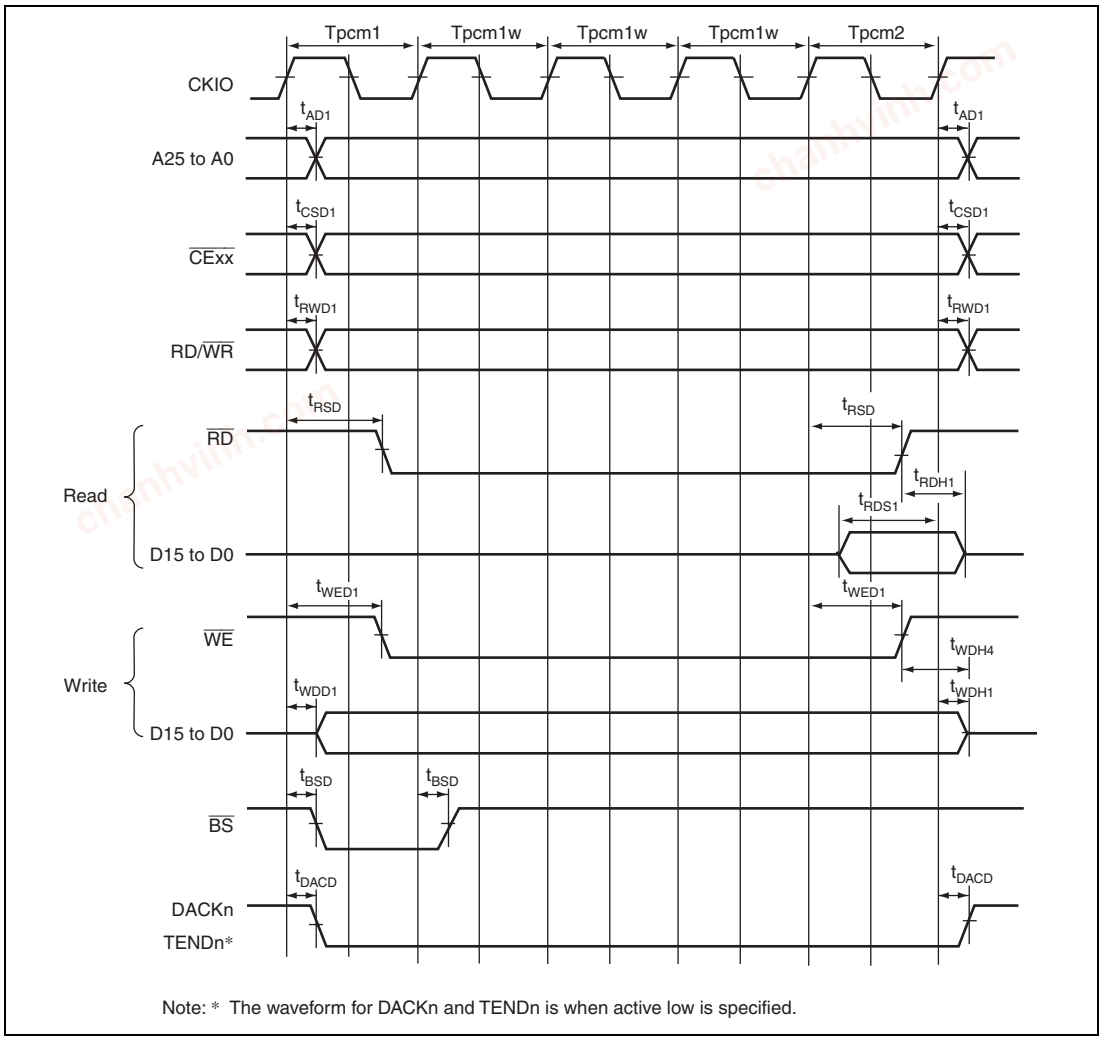


Figure 37.36 PCMCIA Memory Card Bus Cycle (TED = 0 Cycle, TEH = 0 Cycle, No Wait)

chanhvinh.com

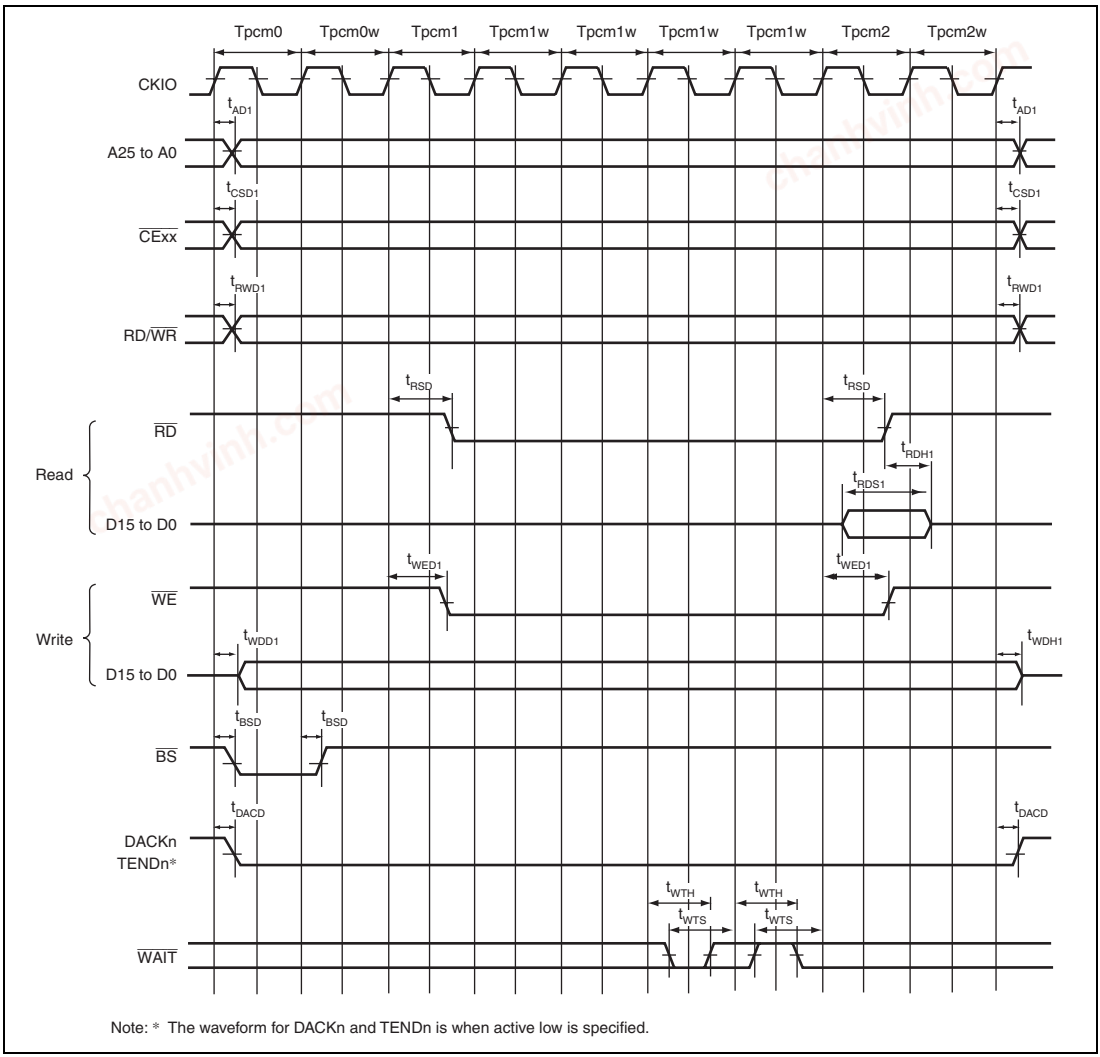


Figure 37.37 PCMCIA Memory Card Bus Cycle
(TED = 2 Cycles, TEH = 1 Cycle, Software Wait Cycle 0, Hardware Wait Cycle 1)

chanhvinh.com

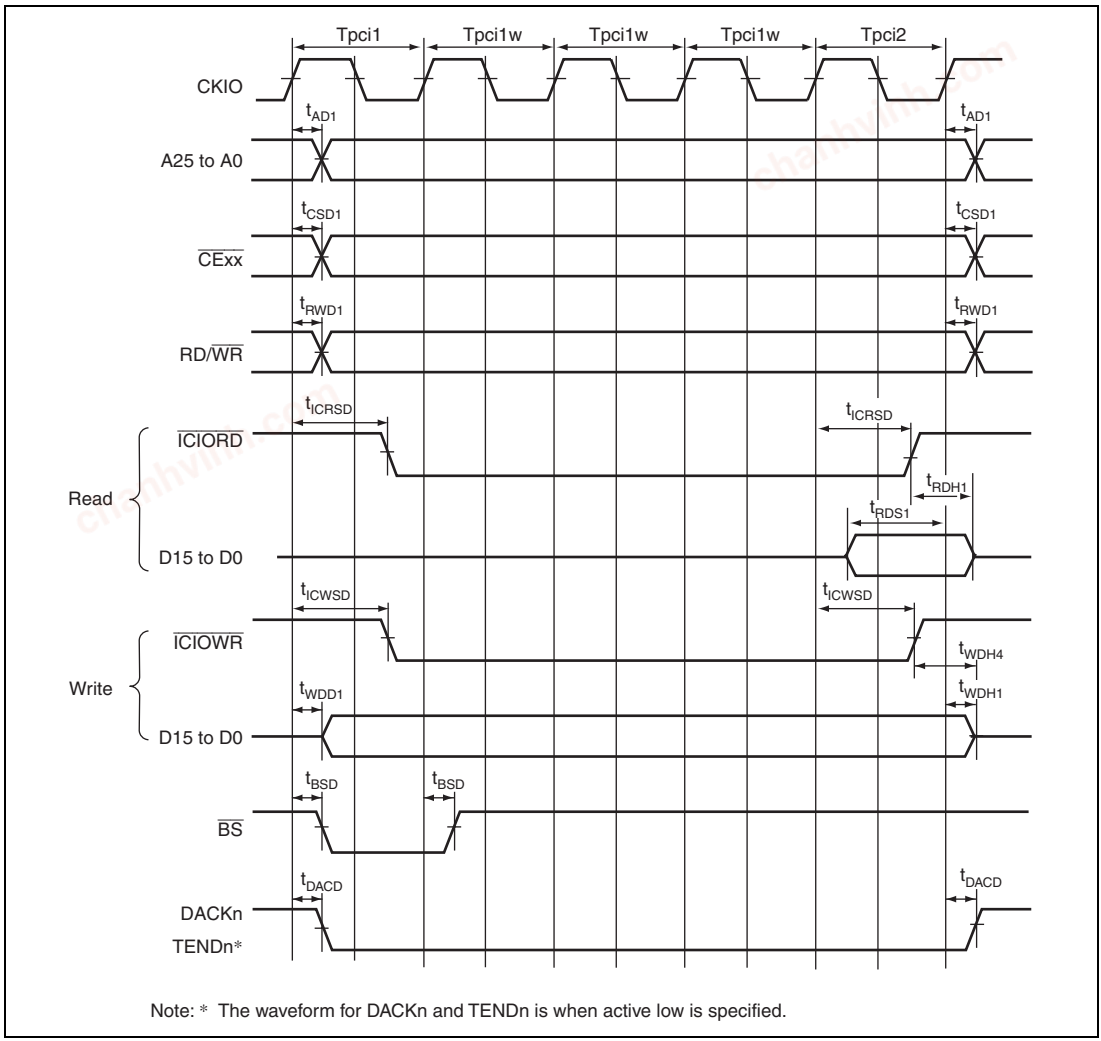


Figure 37.38 PCMCIA I/O Card Bus Cycle (TED = 0 Cycle, TEH = 0 Cycle, No Wait)

chanhvinh.com

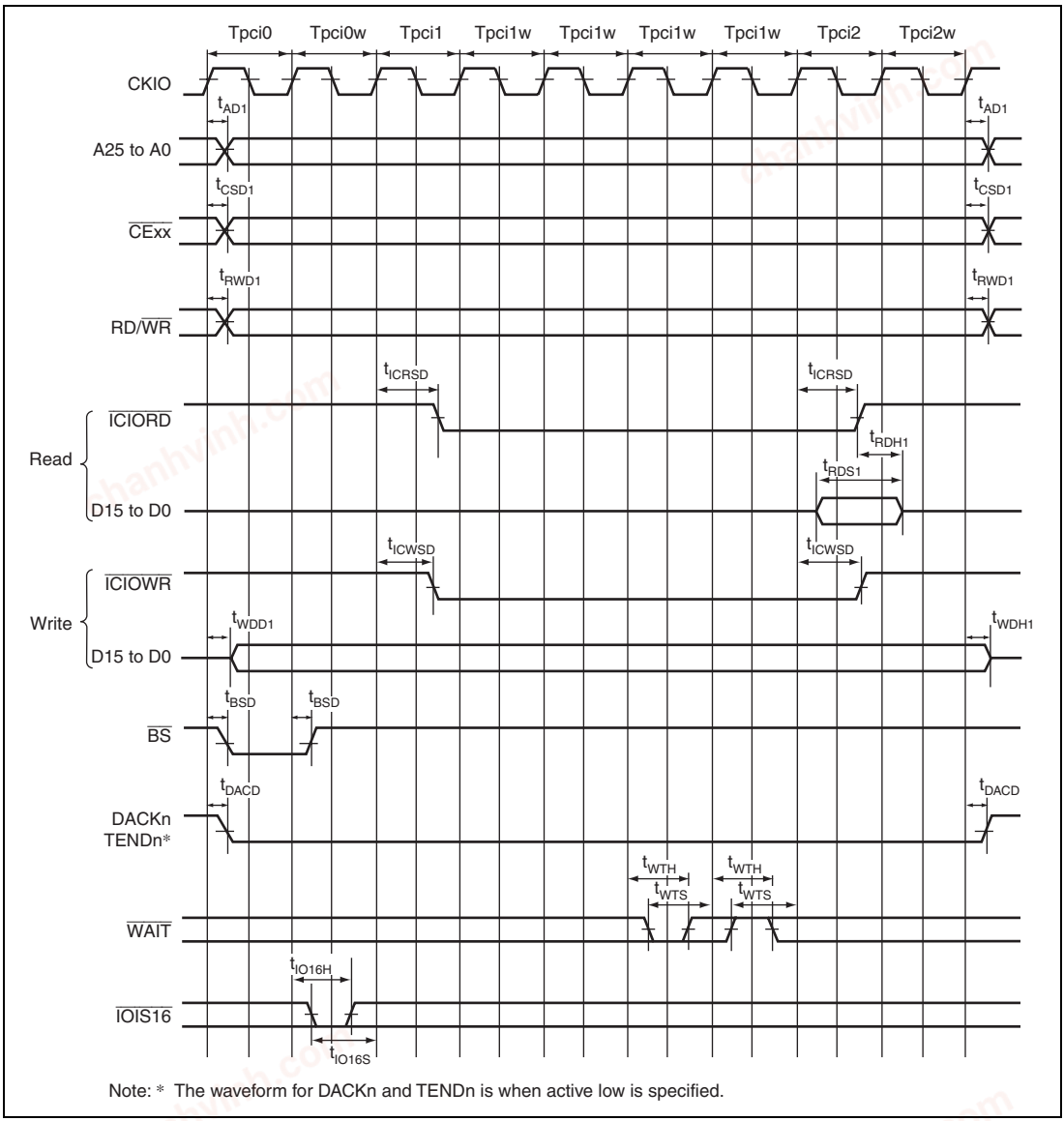


Figure 37.39 PCMCIA I/O Card Bus Cycle
(TED = 2 Cycles, TEH = 1 Cycle, Software Wait Cycle 0, Hardware Wait Cycle 1)

chanhvinh.com

37.4.4 Direct Memory Access Controller Timing

Table 37.8 Direct Memory Access Controller Timing

Item	Symbol	Min.	Max.	Unit	Figure
DREQ setup time	t_{DRQS}	5.5	—	ns	Figure 37.40
DREQ hold time	t_{DRQH}	2.5	—		
DACK, TEND delay time	t_{DACD}	0	10.5		Figure 37.41

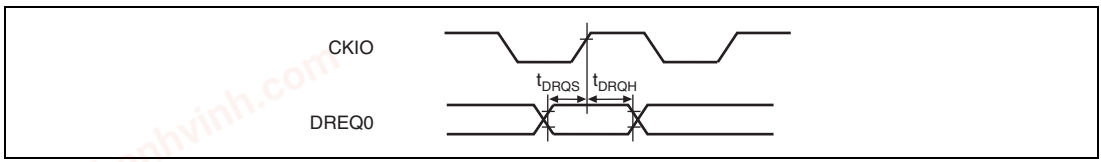


Figure 37.40 DREQ Input Timing

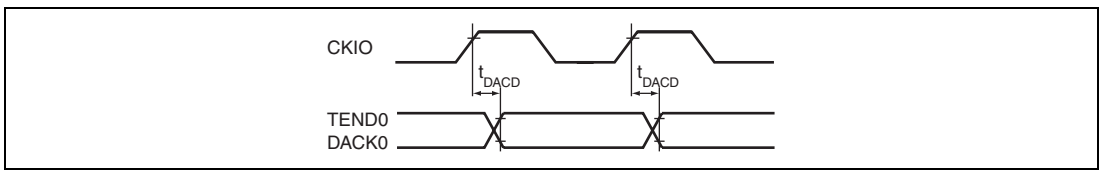


Figure 37.41 DACK, TEND Output Timing

chanhvinh.com

37.4.5 Multi-Function Timer Pulse Unit 2 Timing

Table 37.9 Multi-Function Timer Pulse Unit 2 Timing

Item	Symbol	Min.	Max.	Unit	Figure
Output compare output delay time	t_{TOCD}	—	20	ns	Figure 37.42
Input capture input setup time	t_{TICS}	20	—	ns	
Timer input setup time	t_{TCKS}	20	—	ns	Figure 37.43
Timer clock pulse width (single edge)	$t_{TCKWH/L}$	1.5	—	t_{pcyc}	
Timer clock pulse width (both edges)	$t_{TCKWH/L}$	2.5	—	t_{pcyc}	
Timer clock pulse width (phase counting mode)	$t_{TCKWH/L}$	2.5	—	t_{pcyc}	

Note: t_{pcyc} indicates peripheral clock (Pϕ) cycle.

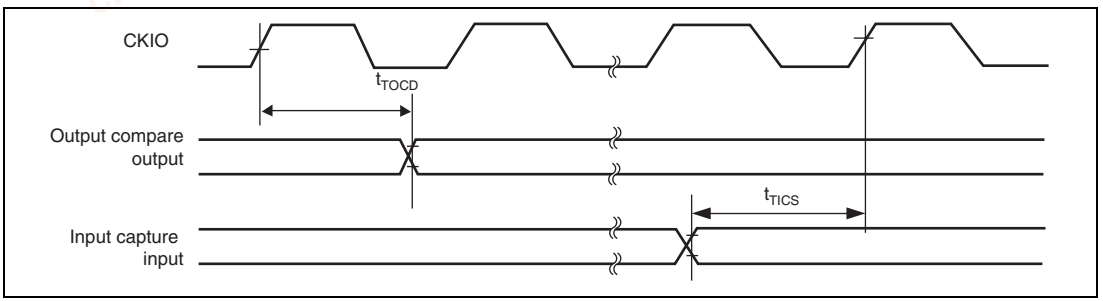


Figure 37.42 Pulse Input/Output Timing

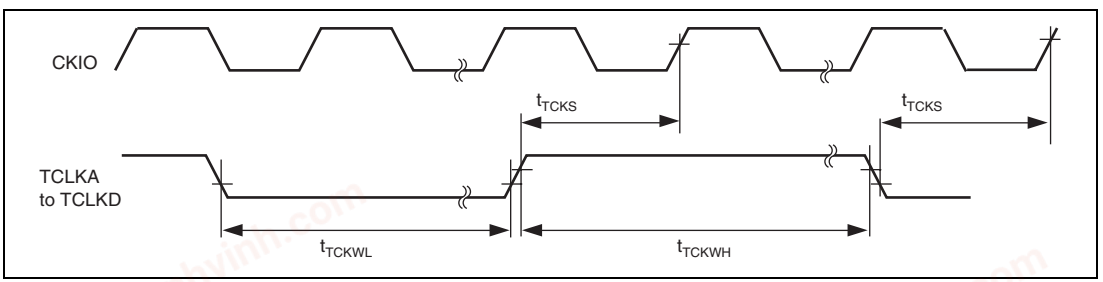


Figure 37.43 Clock Input Timing

chanhvinh.com

37.4.6 Watchdog Timer Timing

Table 37.10 Watchdog Timer Timing

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{\text{WDTOVF}}$ delay time	t_{WOVD}	—	100	ns	Figure 37.44

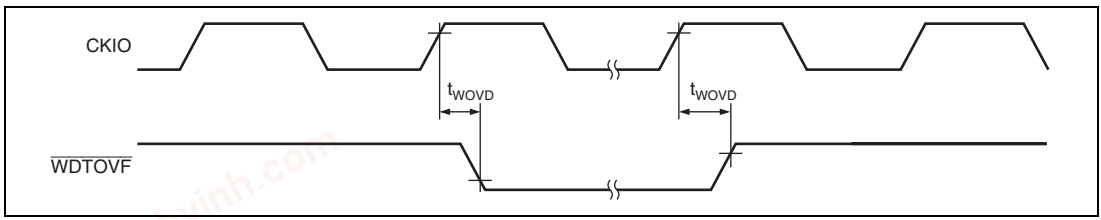


Figure 37.44 $\overline{\text{WDTOVF}}$ Output Timing

chanhvinh.com

37.4.7 Serial Communication Interface with FIFO Timing

Table 37.11 Serial Communication Interface with FIFO Timing

Item		Symbol	Min.	Max.	Unit	Figure
Input clock cycle	(clocked synchronous)	t_{Scyc}	12	—	t_{pcyc}	Figure 37.45
	(asynchronous)		4	—	t_{pcyc}	
Input clock rise time		t_{SCKr}	—	1.5	t_{pcyc}	
Input clock fall time		t_{SCKf}	—	1.5	t_{pcyc}	
Input clock width		t_{SCKW}	0.4	0.6	t_{Scyc}	
Transmit data delay time (clocked synchronous)		t_{TXD}	—	3 t_{pcyc} + 15	ns	Figure 37.46
Receive data setup time (clocked synchronous)		t_{RXS}	4 t_{pcyc} + 15	—	ns	
Receive data hold time (clocked synchronous)		t_{RXH}	1 t_{pcyc} + 15	—	ns	

Note: t_{pcyc} indicates the peripheral clock (Pϕ) cycle.

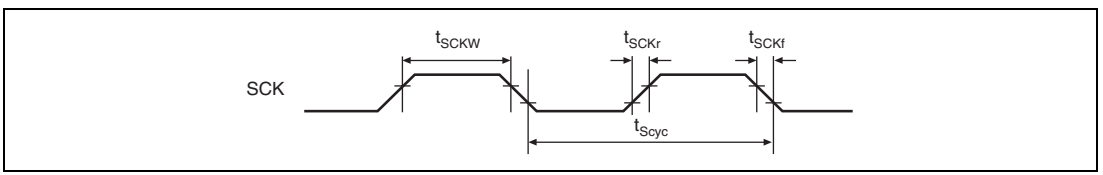


Figure 37.45 SCK Input Clock Timing

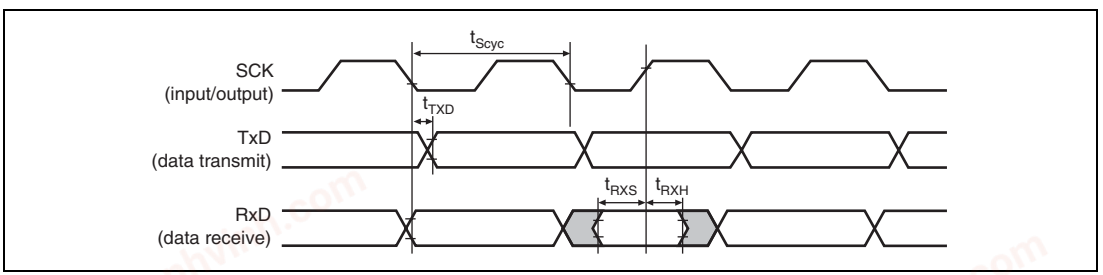


Figure 37.46 Transmit/Receive Data Input/Output Timing in Clocked Synchronous Mode

chanhvinh.com

37.4.8 Renesas Serial Peripheral Interface Timing

Table 37.12 Renesas Serial Peripheral Interface Timing

Item		Symbol	Min.	Max.	Unit	Figure
RSPCK clock cycle	Master	t_{SPcyc}	2	4096	t_{cyc}	Figure 37.47
	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	Figures 37.48 to 37.51
	Slave		0	—	t_{cyc}	
Data input hold time	Master	t_H	0	—	ns	
	Slave		4	—	t_{cyc}	
SSL setup time	Master	t_{LEAD}	1	8	t_{SPcyc}	
	Slave		4	—	t_{cyc}	
SSL hold time	Master	t_{LAG}	1	8	t_{SPcyc}	
	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}	Figures 37.50, 37.51
Slave out release time		t_{REL}	—	3	t_{cyc}	

chanhvinh.com

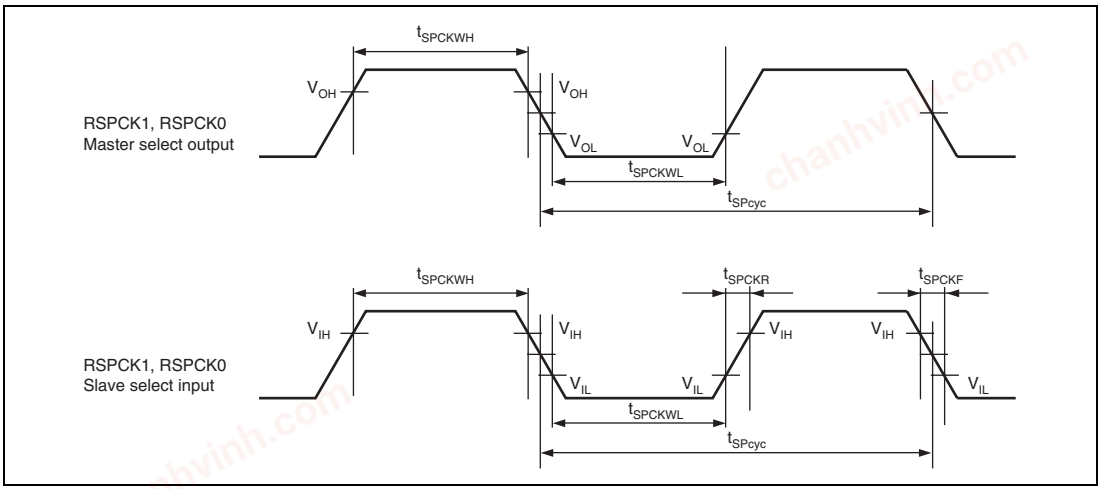


Figure 37.47 Clock Timing

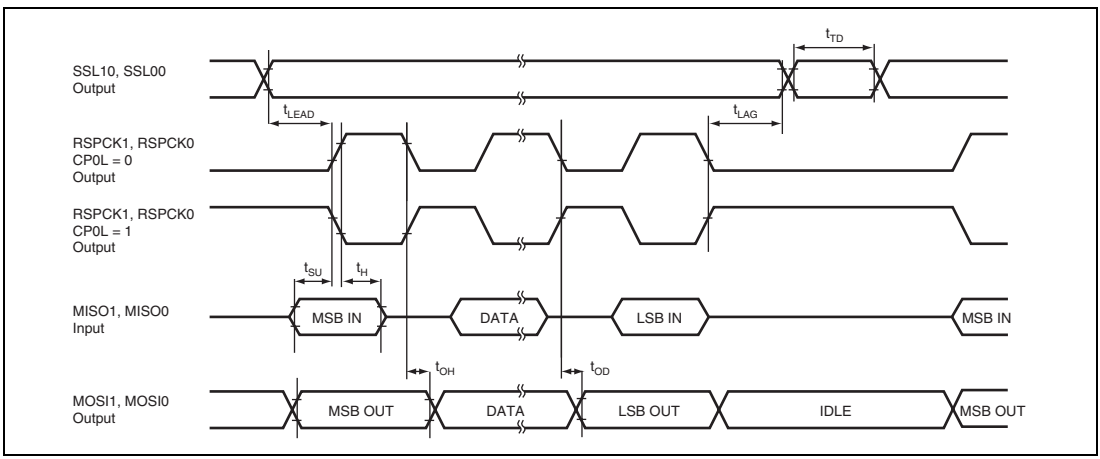


Figure 37.48 Transmission and Reception Timing (Master, CPHA = 0)

chanhvinh.com

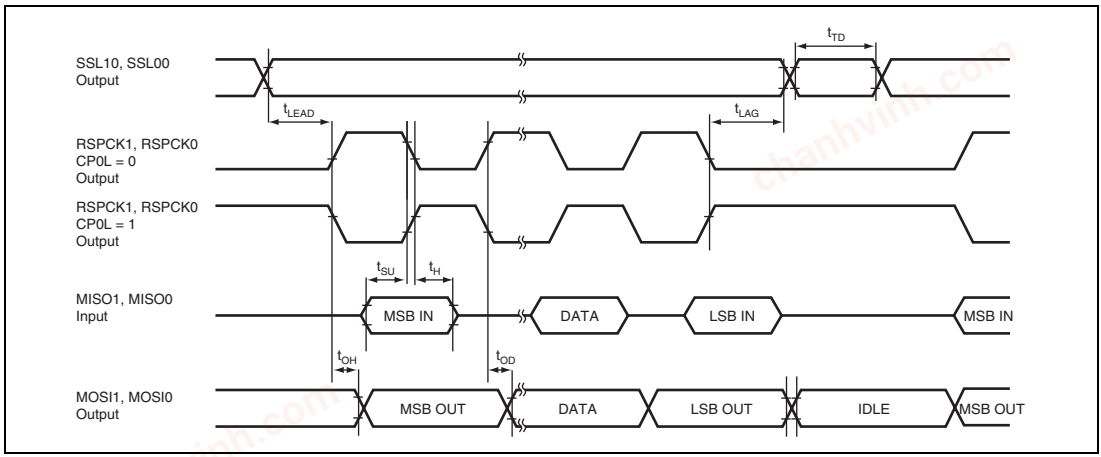


Figure 37.49 Transmission and Reception Timing (Master, CPHA = 1)

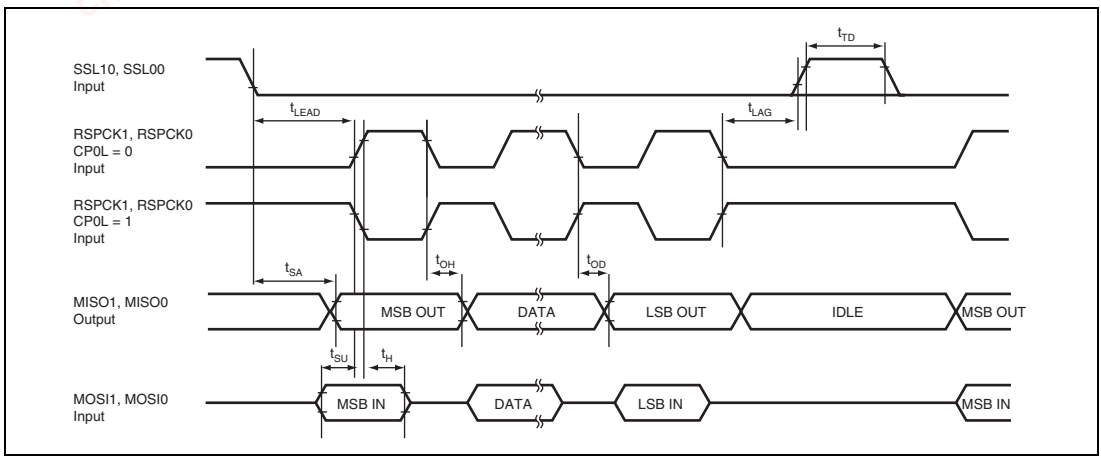


Figure 37.50 Transmission and Reception Timing (Slave, CPHA = 0)

chanhvinh.com

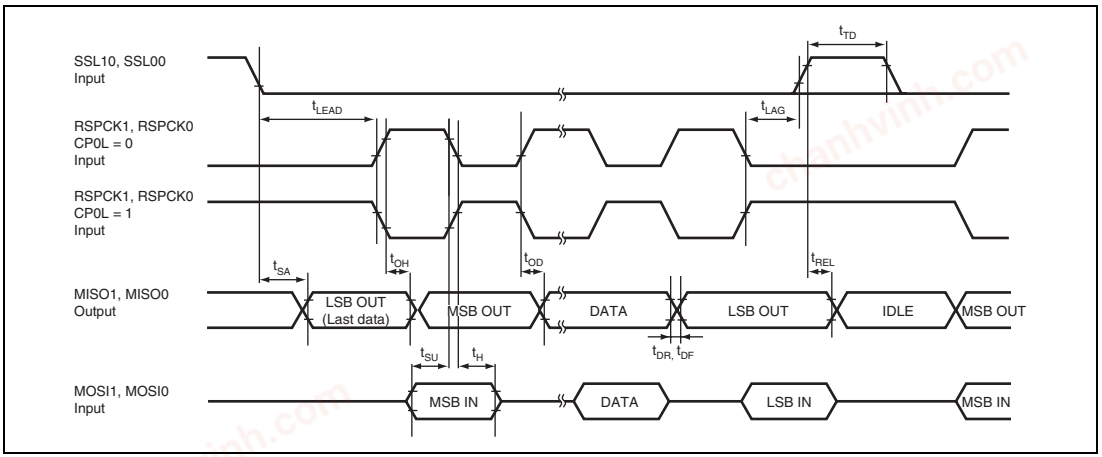


Figure 37.51 Transmission and Reception Timing (Slave, CPHA = 1)

chanhvinh.com

37.4.9 I²C Bus Interface 3 Timing

Table 37.13 I²C Bus Interface 3 Timing

Item	Symbol	Min.	Max.	Unit	Figure
SCL input cycle time	t_{SCL}	12 t_{pcyc} *1 + 600	—	ns	Figure 37.52
SCL input high pulse width	t_{SCLH}	3 t_{pcyc} *1 + 300	—	ns	
SCL input low pulse width	t_{SCLL}	5 t_{pcyc} *1 + 300	—	ns	
SCL, SDA input rise time	t_{Sr}	—	300	ns	
SCL, SDA input fall time	t_{Sf}	—	300	ns	
SCL, SDA input spike pulse removal time*2	t_{SP}	—	1, 2	t_{pcyc} *1	
SDA input bus free time	t_{BUF}	5	—	t_{pcyc} *1	
Start condition input hold time	t_{STAH}	3	—	t_{pcyc} *1	
Retransmit start condition input setup time	t_{STAS}	3	—	t_{pcyc} *1	
Stop condition input setup time	t_{STOS}	3	—	t_{pcyc} *1	
Data input setup time	t_{SDAS}	1 t_{pcyc} *1 + 20	—	ns	
Data input hold time	t_{SDAH}	0	—	ns	
SCL, SDA capacitive load	Cb	0	400	pF	
SCL, SDA output fall time*3	t_{Sf}	—	250	ns	

Notes: 1. t_{pcyc} indicates the peripheral clock (Pϕ) cycle.
2. Depends on the value of NF2CYC.
3. Indicates the I/O buffer characteristic.

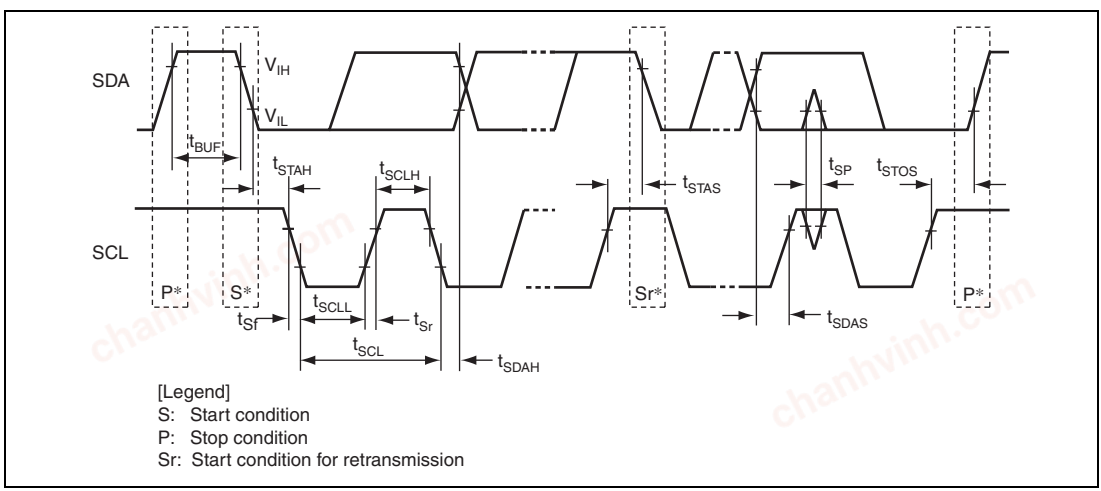


Figure 37.52 Input/Output Timing

chanhvinh.com

37.4.10 Serial Sound Interface Timing

Table 37.14 Serial Sound Interface Timing

Item	Symbol	Min.	Max.	Unit	Remarks	Figure
Output clock cycle	t_O	80	64000	ns	Output	Figure 37.53
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	t_{DTR}	–5	25	ns	Transmit	Figures 37.54, 37.55
Setup time	t_{SR}	25	—	ns	Receive	
Hold time	t_{HTR}	5	—	ns	Receive, transmit	

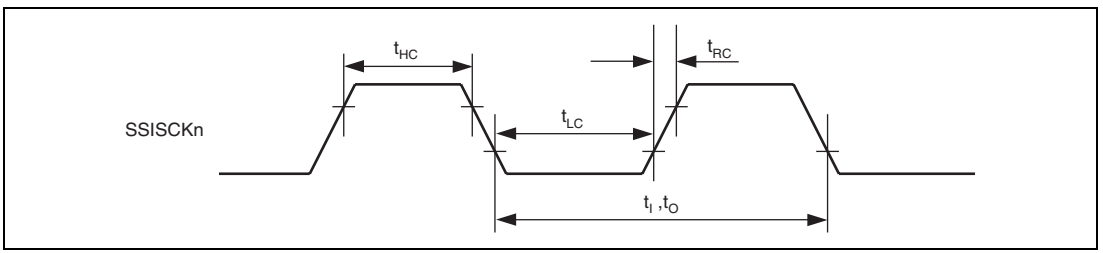


Figure 37.53 Clock Input/Output Timing

chanhvinh.com

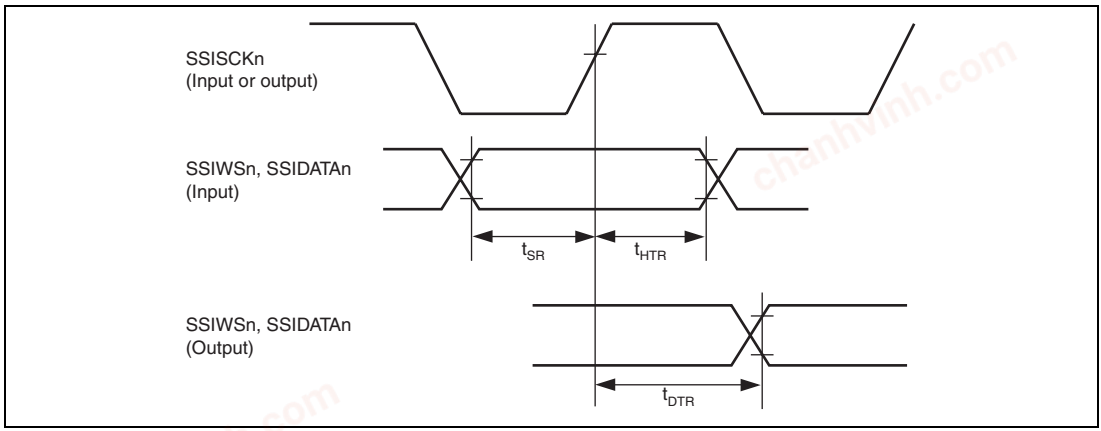


Figure 37.54 Transmission and Reception Timing (Synchronization with Rising Edge of SSISCKn)

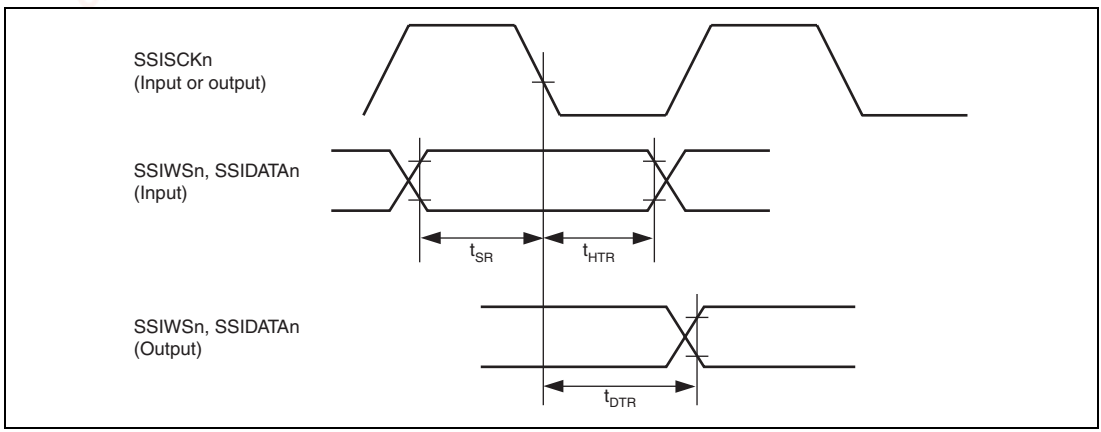


Figure 37.55 Transmission and Reception Timing (Synchronization with Falling Edge of SSISCKn)

chanhvinh.com

37.4.11 Serial I/O with FIFO Timing

Table 37.15 Serial I/O with FIFO Timing

Item	Symbol	Min.	Max.	Unit	Figure
SCK_SIO clock input/output cycle time	t_{SIcyc}	80	—	ns	Figures 37.56 to 37.58
SCK_SIO output high width	t_{SWHO}	0.4 × t_{SIcyc}	—		Figures 37.56, 37.57
SCK_SIO output low width	t_{SWLO}	0.4 × t_{SIcyc}	—		
SIOFSYNC output delay time	t_{FSD}	–5	20		
SCK_SIO input high width	t_{SWHI}	0.4 × t_{SIcyc}	—		Figure 37.58
SCK_SIO input low width	t_{SWLI}	0.4 × t_{SIcyc}	—		
SIOFSYNC input setup time	t_{FSS}	20	—		
SIOFSYNC input hold time	t_{FSH}	20	—		
TXD_SIO output delay time	t_{STDD}	–5	20		Figures 37.56 to 37.58
RXD_SIO input setup time	t_{SRDS}	20	—		
RXD_SIO input hold time	t_{SRDH}	20	—		

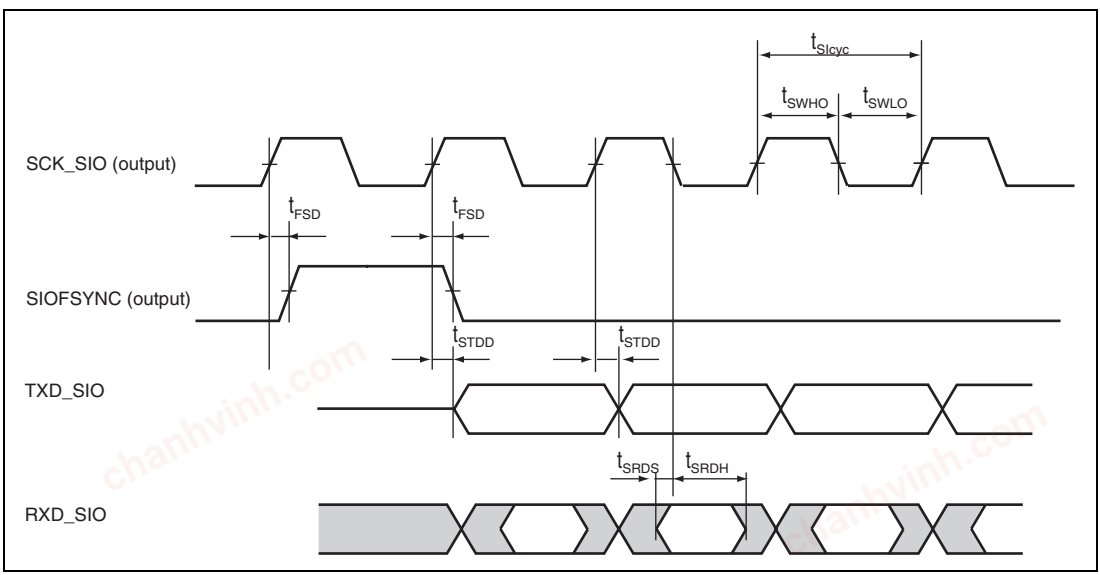


Figure 37.56 Transmission and Reception Timing (Master Mode 1, Sampled at Falling Edge)

chanhvinh.com

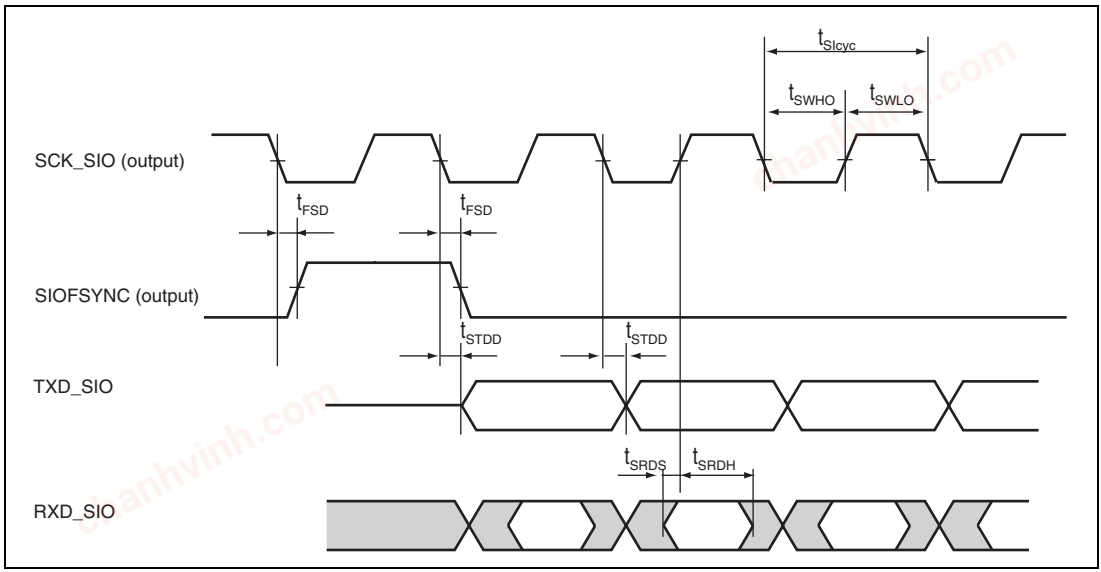


Figure 37.57 Transmission and Reception Timing (Master Mode 1, Sampled at Rising Edge)

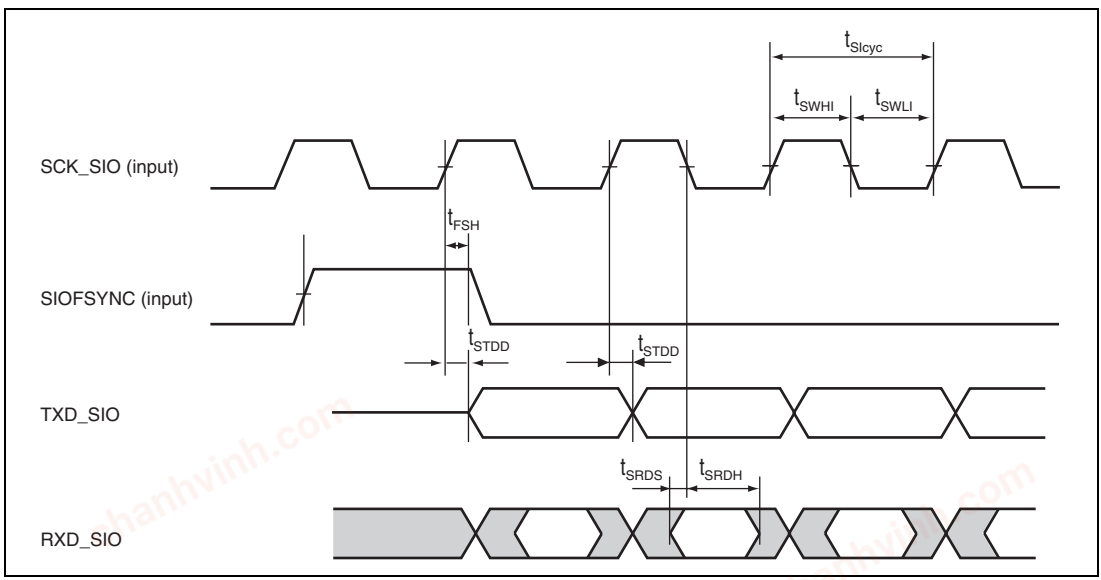


Figure 37.58 Transmission and Reception Timing (Slave Mode 1)

chanhvinh.com

37.4.12 A/D Converter Timing

Table 37.16 A/D Converter Timing

Module	Item		Symbol	Min.	Max.	Unit	Figure
A/D converter	Trigger input setup time	B:P clock ratio = 1:1	t_{TRGS}	17	—	ns	Figure 37.59
		B:P clock ratio = 2:1		t_{cyc} + 17	—		
		B:P clock ratio = 4:1		3 × t_{cyc} + 17	—		

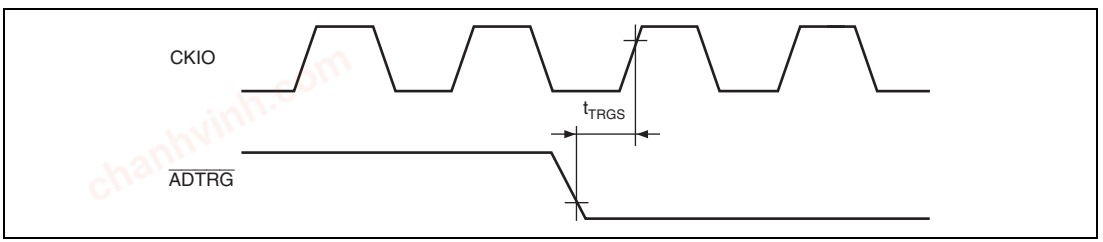


Figure 37.59 A/D Converter External Trigger Input Timing

chanhvinh.com

37.4.13 NAND Type Flash Memory Controller Timing

Table 37.17 NAND Type Flash Memory Interface Timing

Item	Symbol	Min.	Max.	Unit	Figure
Command output setup time	t_{NCDS}	$2 \times t_{fcyc} - 10$	—	ns	Figures 37.60, 37.64
Command output hold time	t_{NCDH}	$1.5 \times t_{fcyc} - 5$	—	ns	
Data output setup time	t_{NDOS}	$0.5 \times t_{wfcyc} - 5$	—	ns	Figures 37.60, 37.61, 37.63, 37.64
Data output hold time	t_{NDOH}	$0.5 \times t_{wfcyc} - 10$	—	ns	
Command to address transition time 1	t_{NCDAD1}	$1.5 \times t_{fcyc} - 10$	—	ns	Figures 37.60, 37.61
Command to address transition time 2	t_{NCDAD2}	$2 \times t_{fcyc} - 10$	—	ns	Figure 37.61
$\overline{FWE}$ cycle time	t_{NWC}	$t_{wfcyc} - 5$	—	ns	Figures 37.61, 37.63
$\overline{FWE}$ low pulse width	t_{NWP}	$0.5 \times t_{wfcyc} - 5$	—	ns	Figures 37.60, 37.61, 37.63, 37.64
$\overline{FWE}$ high pulse width	t_{NWH}	$0.5 \times t_{wfcyc} - 5$	—	ns	Figures 37.61, 37.63
Address to ready/busy transition time	t_{NADRB}	—	$32 \times t_{pcyc}$	ns	Figures 37.61, 37.62
Command to ready/busy transition time	t_{NCDRB}	—	$10 \times t_{pcyc}$	ns	Figures 37.61, 37.62
Ready/busy to data read transition time 1	t_{NRBDR1}	$1.5 \times t_{fcyc}$	—	ns	Figure 37.62
Ready/busy to data read transition time 2	t_{NRBDR2}	$32 \times t_{pcyc}$	—	ns	
$\overline{FRE}$ cycle time	t_{NSCC}	$t_{wfcyc} - 5$	—	ns	
$\overline{FRE}$ low pulse width	t_{NSP}	$0.5 \times t_{wfcyc} - 5$	—	ns	Figures 37.62, 37.64

chanhvinh.com

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{FRE}$ high pulse width	t_{NSPH}	$0.5 \times t_{wfcyc} - 5$	—	ns	Figure 37.62
Read data setup time	t_{NRDS}	16	—	ns	Figures 37.62, 37.64
Read data hold time	t_{NRDH}	5	—	ns	Figures 37.62, 37.64
Data write setup time	t_{NDWS}	$32 \times t_{pcyc}$	—	ns	Figure 37.63
Command to status read transition time	t_{NCDSR}	$4 \times t_{fcyc}$	—	ns	Figure 37.64
Command output off to status read transition time	t_{NCDFSR}	$3.5 \times t_{fcyc}$	—	ns	
Status read setup time	t_{NSTS}	$2.5 \times t_{fcyc}$	—	ns	
$\overline{FCE}$ output setup time	t_{NCES}	$8 \times t_{pcyc}$	—	ns	Figure 37.60
$\overline{FCE}$ output hold time	t_{NCEH}	t_{pcyc}	—	ns	Figure 37.63
$\overline{FCE}$ output access time	t_{NCEA}	$6 \times t_{pcyc}$	—	ns	Figure 37.62
$\overline{FCE}$ output high-level hold time	t_{NCEOH}	$2 \times t_{pcyc}$	—	ns	

Note: t_{fcyc} indicates the period of one cycle of the FLCTL clock.

t_{wfcyc} indicates the period of one cycle of the FLCTL clock when the value of the NANDWF bit is 0. On the other hand, t_{wfcyc} indicates the period of two cycles of the FLCTL clock when the value of the NANDWF bit is 1.

t_{pcyc} indicates the period of one cycle of the peripheral clock (Pϕ).

chanhvinh.com

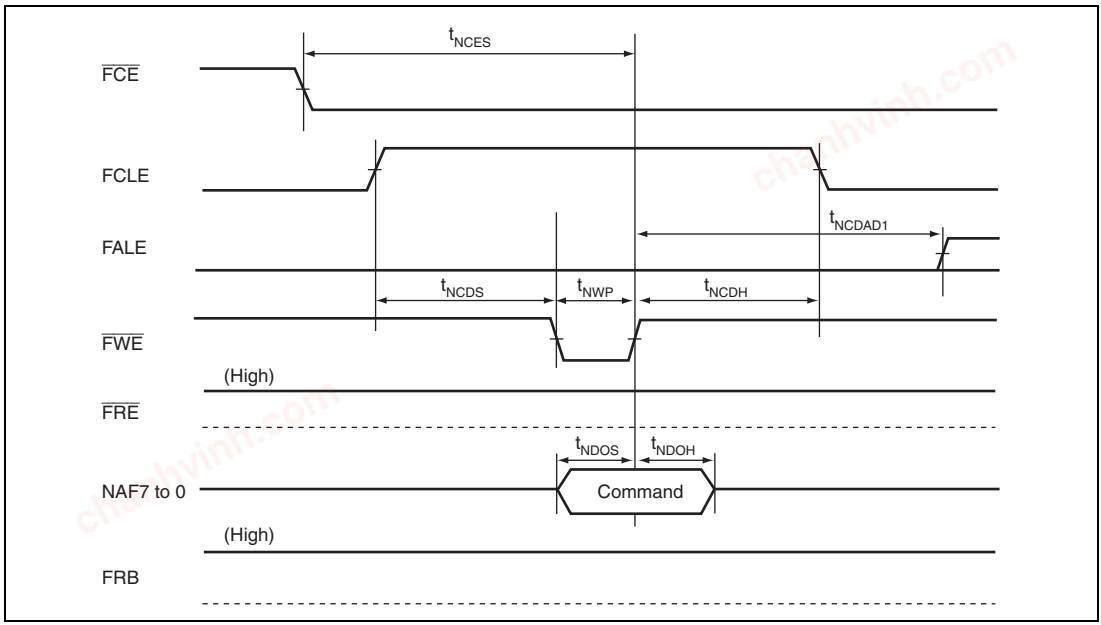


Figure 37.60 NAND Type Flash Memory Command Issuance Timing

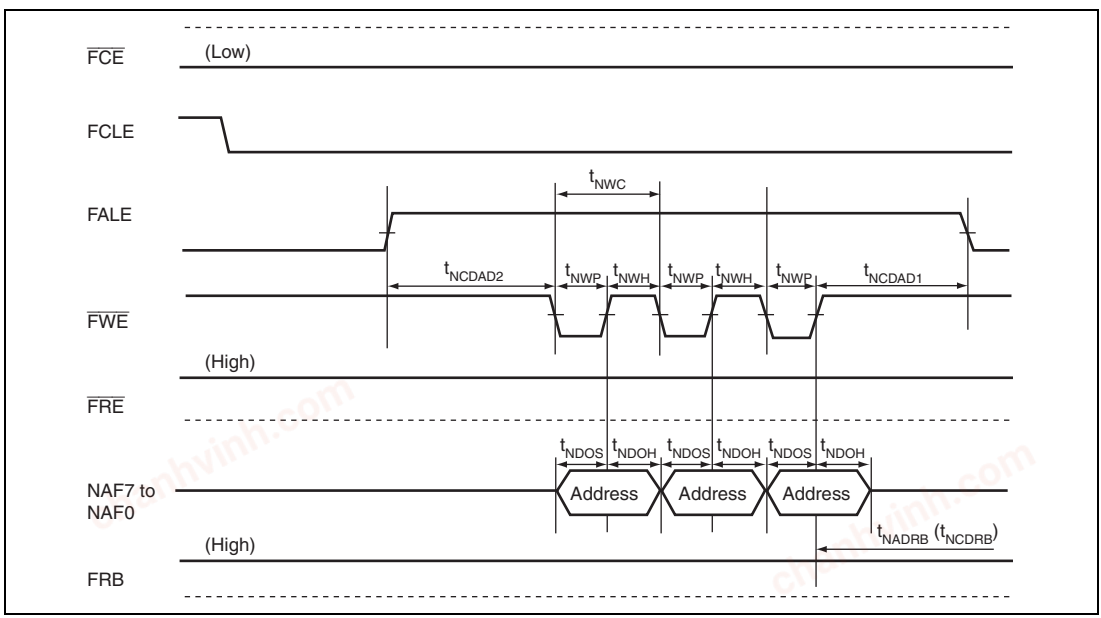
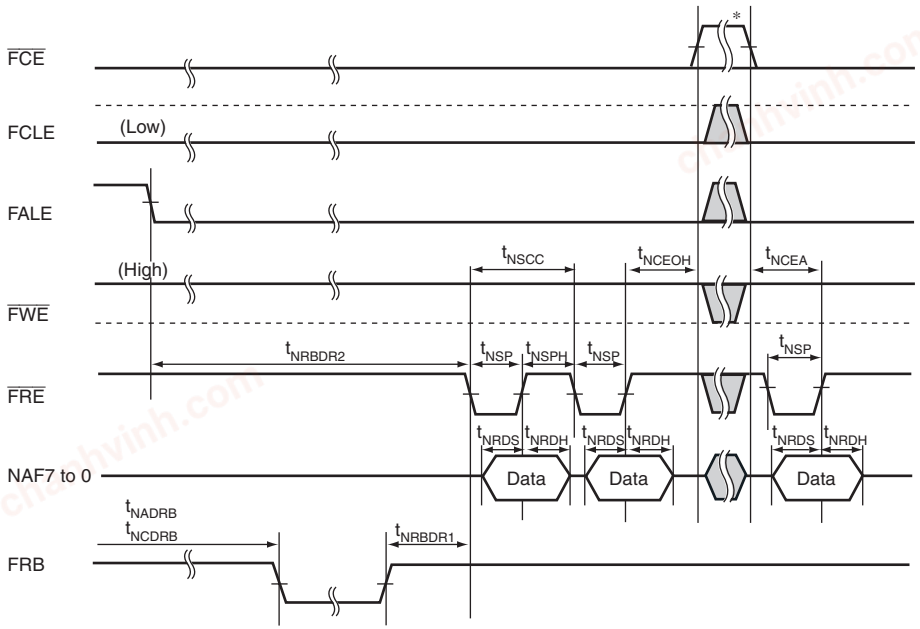


Figure 37.61 NAND Type Flash Memory Address Issuance Timing

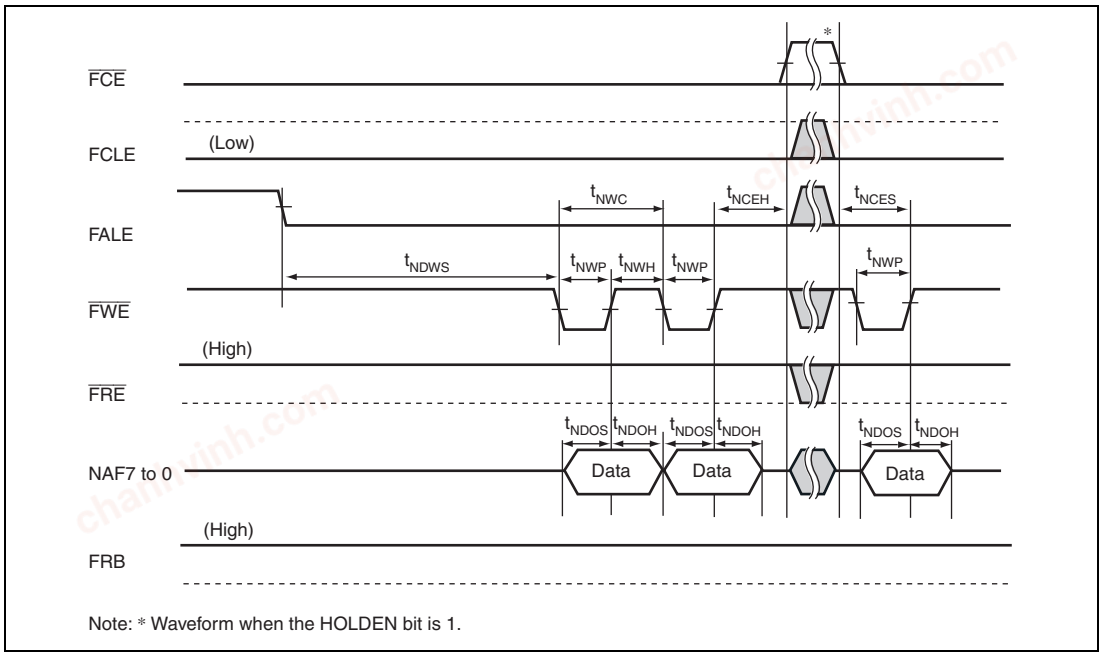
chanhvinh.com



Note: * Waveform when the HOLDEN bit is 1.

Figure 37.62 NAND Type Flash Memory Data Read Timing

chanhvinh.com



Note: * Waveform when the HOLDEN bit is 1.

Figure 37.63 NAND Type Flash Memory Data Write Timing

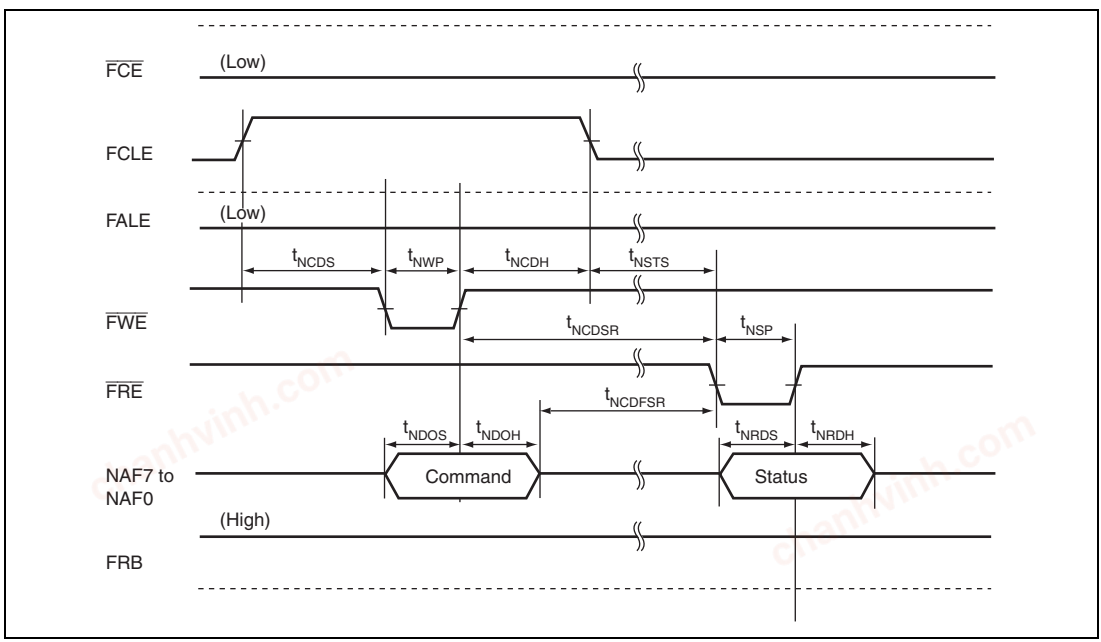


Figure 37.64 NAND Type Flash Memory Status Read Timing

chanhvinh.com

37.4.14 USB 2.0 Host/Function Module Timing

Table 37.18 USB Transceiver Timing (Low-Speed)

Item	Symbol	Min.	Typ.	Max.	Unit	Figure
Rise time	t_{LR}	75	—	300	ns	Figure 37.65
Fall time	t_{LF}	75	—	300	ns	
Rise/fall time lag	t_{LR}/t_{LF}	80	—	125	%	

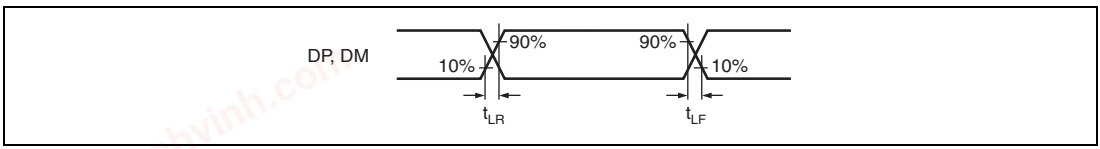


Figure 37.65 DP and DM Output Timing (Low-Speed)

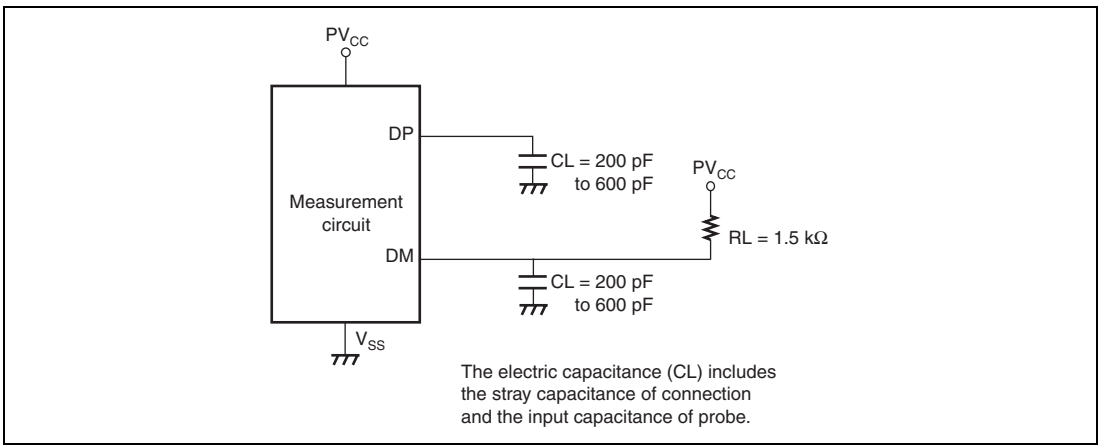


Figure 37.66 Measurement Circuit (Low-Speed)

chanhvinh.com

Table 37.19 USB Transceiver Timing (Full-Speed)

Item	Symbol	Min.	Typ.	Max.	Unit	Figure
Rise time	t_{FR}	4	—	20	ns	Figure 37.67
Fall time	t_{FF}	4	—	20	ns	
Rise/fall time lag	t_{FR}/t_{FF}	90	—	111.11	%	

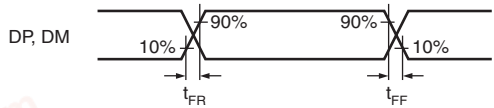


Figure 37.67 DP and DM Output Timing (Full-Speed)

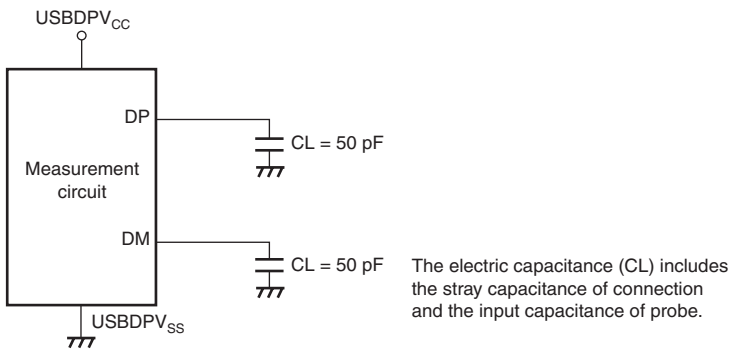


Figure 37.68 Measurement Circuit (Full-Speed)

Table 37.20 USB Transceiver Timing (High-Speed)

Item	Symbol	Min.	Typ.	Max.	Unit	Figure
Rise time	t_{HSR}	500	—	—	ps	Figure 37.69
Fall time	t_{HSF}	500	—	—	ps	
Output driver resistance	Z_{HSDRV}	40.5	—	49.5	Ω	

chanhvinh.com

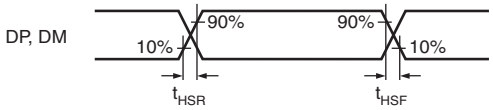


Figure 37.69 DP and DM Output Timing (High-Speed)

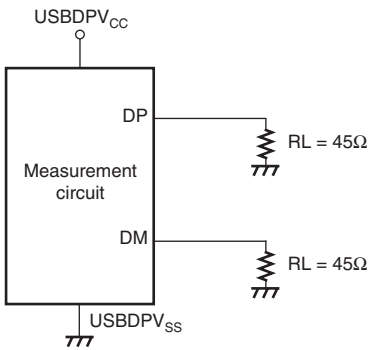


Figure 37.70 Measurement Circuit (High-Speed)

chanhvinh.com

37.4.15 Video Display Controller 3 Timing

Table 37.21 Video Display Controller 3 Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Figure
DV_CLK input clock frequency	t_{cyc}	—	27	—	MHz	Figure 37.71
DV_CLK input clock low pulse width	t_{WIL}	0.4	—	—	t_{cyc}	
DV_CLK input clock high pulse width	t_{WIH}	0.4	—	—		
LCD_EXTCLK input clock frequency	t_{cyc}	4	—	36	MHz	
LCD_EXTCLK input clock low pulse width	t_{WIL}	0.4	—	—	t_{cyc}	
LCD_EXTCLK input clock high pulse width	t_{WIH}	0.4	—	—		
LCD_CLK output clock frequency	t_{cyc}	4	—	36	MHz	Figure 37.72
LCD_CLK output clock low pulse width	t_{WIL}	0.4	—	—	t_{cyc}	
LCD_CLK output clock high pulse width	t_{WIH}	0.4	—	—		
Input data setup time	t_{VS}	10	—	—	ns	Figure 37.73
Input data hold time	t_{VH}	3	—	—	ns	
Output data delay time	t_{DD}	0	—	10	ns	Figure 37.74

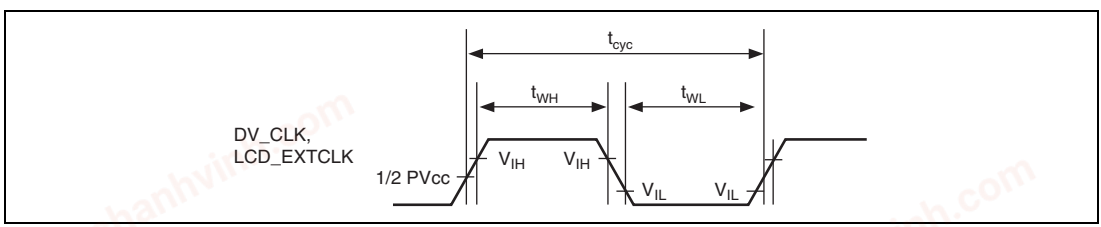


Figure 37.71 DV_CLK and LCD_EXTCLK Clock Input Timing

chanhvinh.com

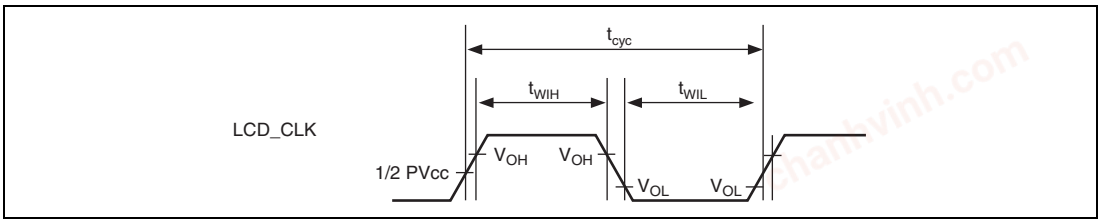


Figure 37.72 LCD_CLK Clock Output Timing

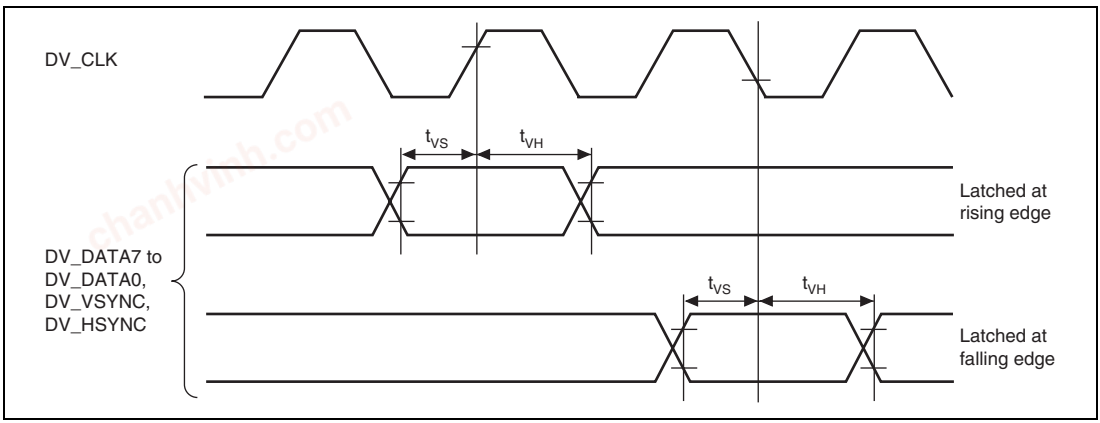


Figure 37.73 Video Input Timing

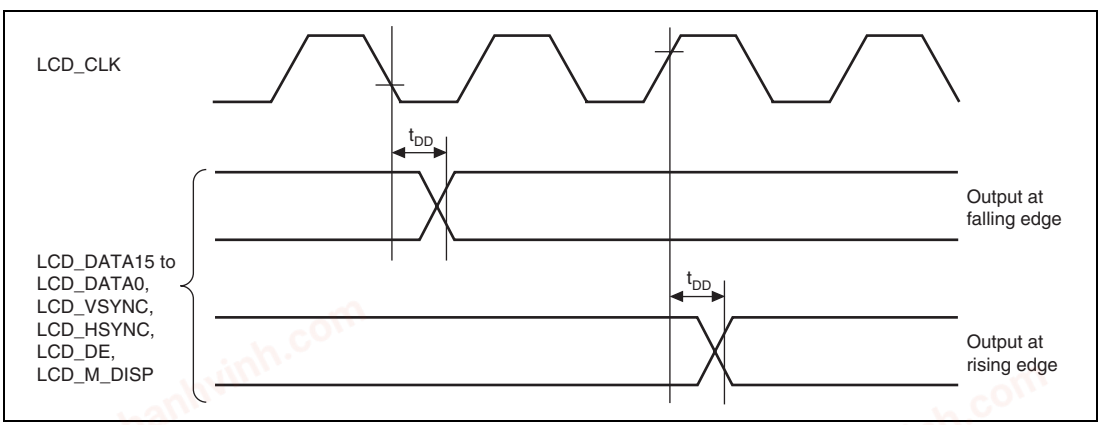


Figure 37.74 Display Output Timing

chanhvinh.com

37.4.16 SD Host Interface Timing

Table 37.22 SD Host Interface Timing

Item	Symbol	Min.	Max.	Unit	Figure
SD_CLK clock cycle	t_{SDPP}	2 × t_{pcyc}	—	ns	Figure 37.75
SD_CLK clock high width	t_{SDWH}	0.4 × t_{SDPP}	—	ns	
SD_CLK clock low width	t_{SDWL}	0.4 × t_{SDPP}	—	ns	
SD_CMD, SD_D3 to SD_D0 output data delay (data transfer mode)	t_{SDODLY}	—	14	ns	
SD_CMD, SD_D3 to SD_D0 input data setup	t_{SDISU}	5	—	ns	
SD_CMD, SD_D3 to SD_D0 input data hold	t_{SDIH}	5	—	ns	

Note: t_{pcyc} indicates peripheral clock (Pϕ) cycle.

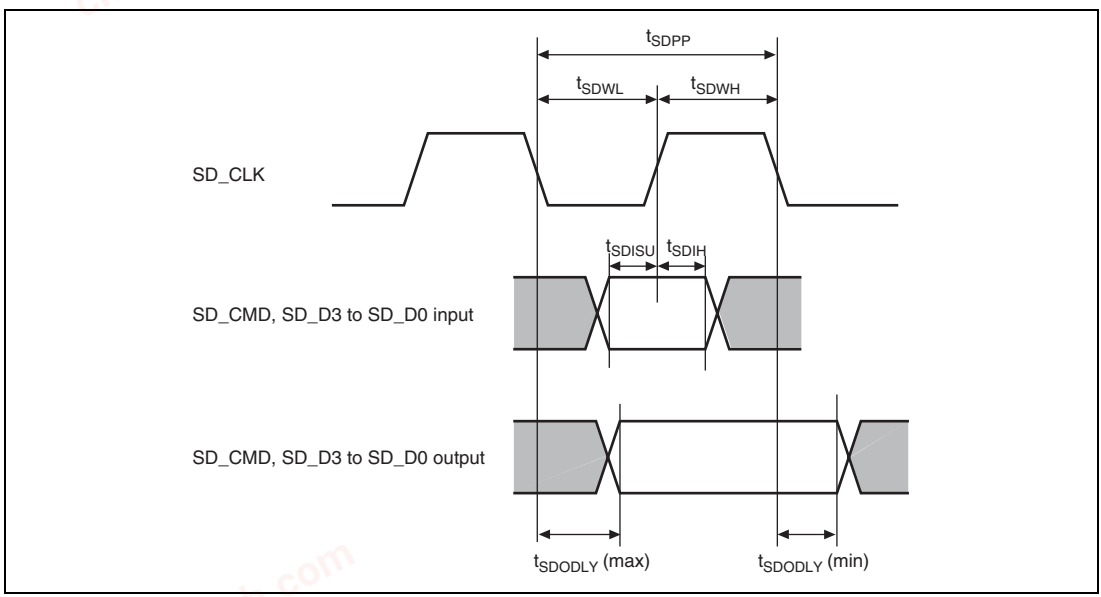


Figure 37.75 SD Card Interface

chanhvinh.com

37.4.17 General Purpose I/O Ports Timing

Table 37.23 General Purpose I/O Ports Timing

Item	Symbol	Min.	Max.	Unit	Figure
Output data delay time	t_{PORTD}	—	100	ns	Figure 37.76
Input data setup time	t_{PORTS}	100	—		
Input data hold time	t_{PORTH}	100	—		

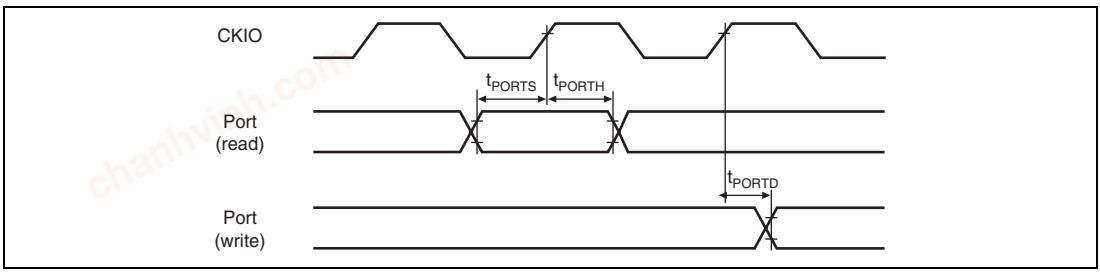


Figure 37.76 General I/O Ports Timing

chanhvinh.com

37.4.18 User Debugging Interface Timing

Table 37.24 User Debugging Interface Timing

Item	Symbol	Min.	Max.	Unit	Figure
TCK cycle time	t_{TCKcyc}	50*	—	ns	Figure 37.77
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 37.78
TDI hold time	t_{TDIH}	10	—	ns	
TMS setup time	t_{TMSS}	10	—	ns	
TMS hold time	t_{TMSH}	10	—	ns	
TDO delay time	t_{TDOD}	—	16	ns	

Note: * Should be greater than the peripheral clock (Pϕ) cycle time.

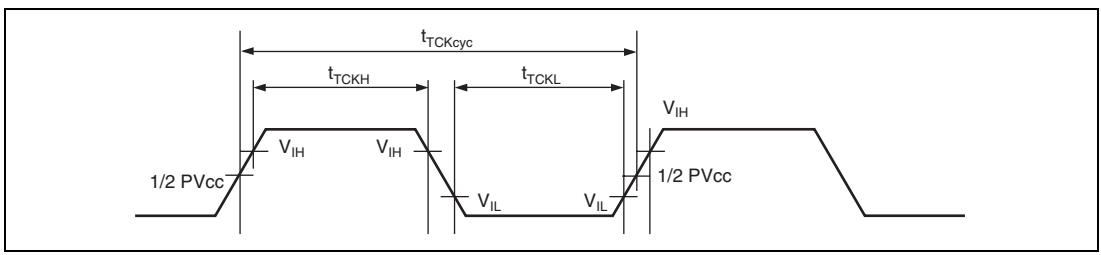


Figure 37.77 TCK Input Timing

chanhvinh.com

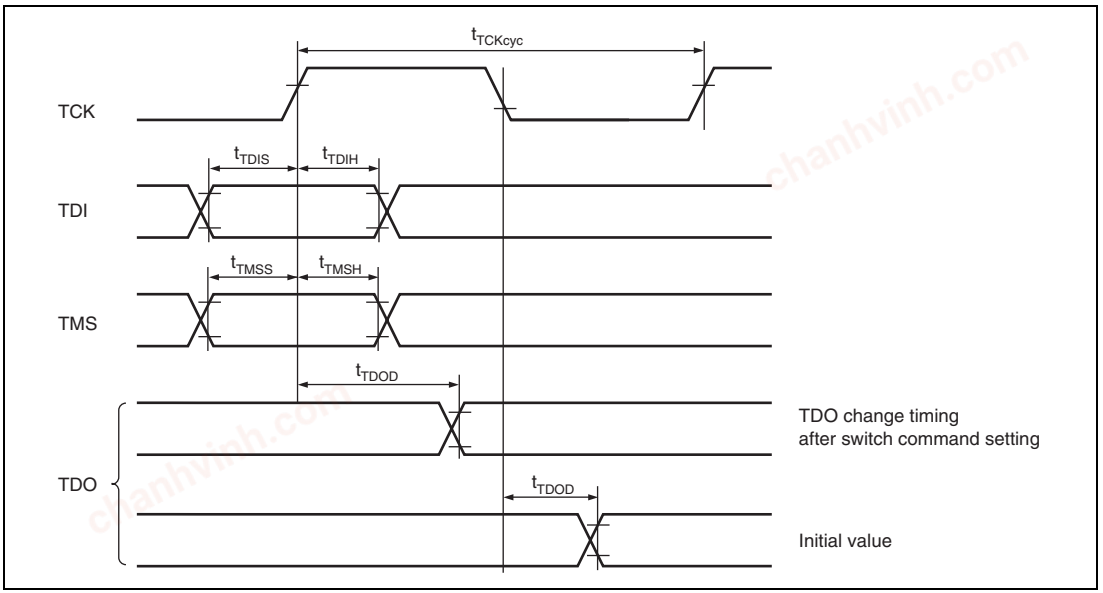


Figure 37.78 Data Transfer Timing

chanhvinh.com

37.4.19 Motor Control PWM Timer

Table 37.25 Motor Control PWM Timer Timing

Item	Symbol	Min.	Max.	Unit	Figure
Pulse output delay time	t_{MPWMOD}	—	20	ns	Figure 37.79

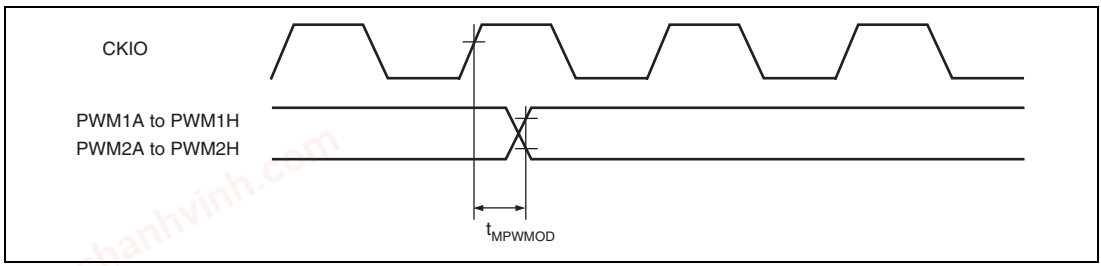
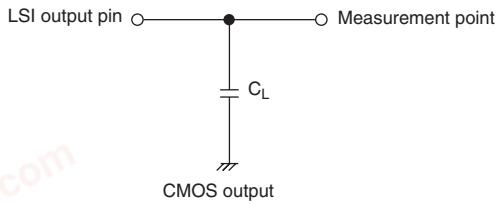


Figure 37.79 Pulse Output Timing

chanhvinh.com

37.4.20 AC Characteristics Measurement Conditions

- I/O signal reference level: $PV_{CC}/2$ (PV_{CC} = 3.0 to 3.6 V, V_{CC} = 1.1 to 1.3 V)
- Input pulse level: PV_{CC}
- Input rise and fall times: 1 ns



Note: C_L is the total value that includes the capacitance of measurement tools.
C_L = 30 pF

Figure 37.80 Output Load Circuit

chanhvinh.com

37.5 A/D Converter Characteristics

Table 37.26 A/D Converter Characteristics

Conditions: V_{CC} = $PLLV_{CC}$ = $USBDV_{CC}$ = $USBUV_{CC}$ = 1.1 to 1.3 V, PV_{CC} = $USBDPV_{CC}$ = 3.0 to 3.6 V, AV_{CC} = 3.0 to 3.6 V, $USBAV_{CC}$ = 1.1 to 1.3 V, $USBAPV_{CC}$ = 3.0 to 3.6 V, V_{SS} = $PLLV_{SS}$ = AV_{SS} = $USBDV_{SS}$ = $USBAV_{SS}$ = $USBDPV_{SS}$ = $USBAPV_{SS}$ = $USBUV_{SS}$ = 0 V, T_a = –20 to 85 °C (regular specifications), –40 to 85 °C (wide-range specifications)

Item	Min.	Typ.	Max.	Unit
Resolution	10	10	10	bits
Conversion time	4	—	—	µs
Analog input capacitance	—	—	20	pF
Permissible signal-source impedance	—	—	5	kΩ
Nonlinearity error	—	—	±3.0*	LSB
Offset error	—	—	±2.0*	LSB
Full-scale error	—	—	±2.0*	LSB
Quantization error	—	—	±0.5*	LSB
Absolute accuracy	—	—	±4.0	LSB

Note: * Reference values

chanhvinh.com

chanhvinh.com

Section 38 States and Handling of Pins

38.1 Pin States

Table 38.1 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 38.1 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	Bus Mastership Release
Clock	EXTAL*6	Clock operation mode	0, 2	I	I	I			I/Z*5	I	I
			1, 3	Z	Z	Z			Z	Z	Z
	XTAL*6			O	O	O			O/L*5	O/L*5	O
	CKIO			O	O	O	O/Z*7		O/Z*7	O/Z*7	O/Z*7
	AUDIO_CLK			I	—	—			Z	Z	I
	AUDIO_X1*6			I/Z*8	I	I			Z	Z	I/Z*8
	AUDIO_X2*6			O/L*8	O	O			L	L	O/L*8
	AUDIO_XOUT (640-Kbyte only)			O/L*8	—	O/Z*9*16			O/Z*9*16	L/Z*9	O
System control	$\overline{\text{RES}}$			I	I	I			I	I	I
	$\overline{\text{WDTOVF}}$			O	—	H			H	H	O
	$\overline{\text{BREQ}}$			I	—	—			Z	Z	I
	$\overline{\text{BACK}}$			O	—	Z			Z	Z	L
Operation mode control	MD_BOOT1, MD_BOOT0			—	I	—			—	—	—
	MD_CLK1, MD_CLK0			—	I	—			—	—	—
	$\overline{\text{ASEMD}}$			I	I	I			I	I	I

chanhvinh.com

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	Bus Mastership Release
Interrupt	NMI			I	I	I			I	I	I
	IRQ7 to IRQ0			I	—	—			Refer to table A.2	I	I
	PINT7 to PINT0			I	—	—			Z	Z	I
Address bus	A25 to A21, A0			O	—	O/Z*10			O/Z*10	O/Z*10	Z
	A20 to A1	Boot mode	0	O	Z	O	O/Z*10		O/Z*10	O/Z*10	Z
			1 to 3	O	—	O/Z*10			O/Z*10	O/Z*10	Z
Data bus	D15 to D0	Boot mode	0	I/Z	Z	I/Z		Z	Z	Z	Z
				O/Z	Z	O/Z	Z		Z	Z	Z
			1 to 3	I/Z	—	—			Z	Z	Z
				O/Z	—	Z			Z	Z	Z
Bus control	$\overline{CS0}$	Boot mode	0	O	Z	O	H/Z*10		H/Z*10	H/Z*10	Z
			1 to 3	O	—	H/Z*10			H/Z*10	H/Z*10	Z
	$\overline{CS6}$ to $\overline{CS1}$, $\overline{CE1A}$, $\overline{CE1B}$, $\overline{CE2A}$, $\overline{CE2B}$			O	—	H/Z*10			H/Z*10	H/Z*10	Z
	$\overline{RD}$	Boot mode	0	O	Z	O	H/Z*10		H/Z*10	H/Z*10	Z
			1 to 3	O	—	H/Z*10			H/Z*10	H/Z*10	Z
	RD/$\overline{WR}$			O	—	H/Z*10			H/Z*10	H/Z*10	Z
	$\overline{BS}$			O	—	H/Z*10			H/Z*10	H/Z*10	Z
	$\overline{WAIT}$			I	—	—			Z	Z	Z
	$\overline{ICIOWR}$/$\overline{AH}$, $\overline{ICIORD}$, $\overline{WE1}$/DQMLU/$\overline{WE}$, $\overline{WE0}$/DQMLL			O	—	H/Z*10			H/Z*10	H/Z*10	Z
	$\overline{RAS}$, $\overline{CAS}$			O	—	O/Z*11			O/Z*11	O/Z*11	O/Z*11
	CKE			O	—	O/Z*11			O/Z*11	O/Z**11	O/Z*11
	$\overline{IOIS16}$			I	—	—			Z	Z	I

chanhvinh.com

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	Bus Mastership Release
Direct memory access controller	DREQ1, DREQ0	I	—	—			Z	Z	I
	DACK1, DACK0	O	—	O/Z*9			O/Z*9	O/Z*9	O
	TEND1, TEND0	O	—	O/Z*9			O/Z*9	O/Z*9	O
Multi-function timer pulse unit 2	TCLKA to TCLKD	I	—	—			Z	Z	I
	TIOC0A to TIOC0D, TIOC1A, TIOC1B, TIOC2A, TIOC2B, TIOC3A to TIOC3D	I	—	—			Z	Z	I
		O/Z	—	O/Z*9			O/Z*9	O/Z*9	O/Z
	TIOC4A to TIOC4D	I	—	—			Refer to table A.2	I	I
		O/Z	—	O/Z*9			O/Z*9	O/Z*9	O/Z
Realtime clock	RTC_X1*6	I/Z*13	I	I/Z*13			I/Z*13	I/Z*13	I/Z*13
	RTC_X2*6	O/H*13	O	O/H*13			O/H*13	O/H*13	O/H*13
Serial communication interface with FIFO	TxD7 to TxD0	O/Z	—	O/Z*9			O/Z*9	O/Z*9	O/Z
	RxD7 to RxD0	I	—	—			Refer to table A.2	Z	I
	SCK3 to SCK0	I	—	—			Z	Z	I
		O/Z	—	O/Z*9			O/Z*9	O/Z*9	O/Z
	$\overline{\text{RTS3}}$, $\overline{\text{RTS1}}$	I	—	—			Z	Z	I
		O/Z	—	O/Z*9			O/Z*9	O/Z*9	O/Z
	$\overline{\text{CTS3}}$, $\overline{\text{CTS1}}$	I	—	—			Z	Z	I
		O/Z	—	O/Z*9			O/Z*9	O/Z*9	O/Z
Renesas serial peripheral interface	MISO1	I	—	—			Z	Z	I
		O/Z	—	O/Z*9			O/Z*9	O/Z*9	O/Z
	MISO0	I	—	—			Z	Z	I
	Boot mode 0, 2	O/Z	—	O/Z*9			O/Z*9	O/Z*9	O/Z
	Boot mode 1, 3	O/Z	—	—	O/Z*9		O/Z*9	O/Z*9	O/Z
	MOSI1	I	—	—			Z	Z	I
		O/Z	—	O/Z*9			O/Z*9	O/Z*9	O/Z

chanhvinh.com

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	Bus Mastership Release
Renesas serial peripheral interface	MOSI0			I	—	—			Z	Z	I
		Boot mode	0, 2	O/Z	—	O/Z*9			O/Z*9	O/Z*9	O/Z
			1, 3	O/Z	—	—	O/Z*9		O/Z*9	O/Z*9	O/Z
	RSPCK1			I	—	—			Z	Z	I
				O/Z	—	O/Z*9			O/Z*9	O/Z*9	O/Z
	RSPCK0			I	—	—			Z	Z	I
		Boot mode	0, 2	O/Z	—	O/Z*9			O/Z*9	O/Z*9	O/Z
			1, 3	O/Z	—	—	O/Z*9		O/Z*9	O/Z*9	O/Z
	SSL10			I	—	—			Z	Z	I
				O/Z	—	O/Z*9			O/Z*9	O/Z*9	O/Z
	SSL00			I	—	—			Z	Z	I
		Boot mode	0, 2	O/Z	—	O/Z*9			O/Z*9	O/Z*9	O/Z
			1, 3	O/Z	—	—	O/Z*9		O/Z*9	O/Z*9	O/Z
I^2C bus interface3	SCL2 to SCL0			I	—	—			Z	Z	I
				O/Z	—	Z			Z	Z	O/Z
	SDA2 to SDA0			I	—	—			Z	Z	I
				O/Z	—	Z			Z	Z	O/Z
Serial sound interface	SSITxD0			O	—	O/Z*9			O/Z*9	O/Z*9	O
	SSIRxD0			I	—	—			Refer to table A.2	Z	I
	SSIDATA3 to SSIDATA1			I	—	—			Z	Z	I
				O/Z	—	O/Z*9			O/Z*9	O/Z*9	O/Z
	SSISCK3 to SSISCK0			I	—	—			Z	Z	I
				O/Z	—	O/Z*9			O/Z*9	O/Z*9	O/Z
	SSIWS3 to SSIWS0			I	—	—			Z	Z	I
				O/Z	—	O/Z*9			O/Z*9	O/Z*9	O/Z

chanhvinh.com

Pin Function				Pin State							
Type	Pin Name			Normal State (Other than States at Right)	Power-On Reset*1	Pin State Retained*2			Power-Down State		Bus Mastership Release
						EBUSKEEPE*3 (Other than States at Right)		Power-On Reset*4	Deep Standby Mode	Software Standby Mode	
						0	1				
Serial I/O with FIFO	SIOFSCK			I	—	—			Z	Z	I
				O/Z	—	O/Z*9			O/Z*9	O/Z*9	O/Z
	SIOFSYNC			I	—	—			Z	Z	I
				O/Z	—	O/Z*9			O/Z*9	O/Z*9	O/Z
	SIOFTxD			O/Z	—	O/Z*9			O/Z*9	O/Z*9	O/Z
	SIOFRxD			I	—	—			Refer to table A.2	Z	I
Controller area network	CTx1, CTx0			O	—	O/Z*9			O/Z*9	O/Z*9	O
	CRx1, CRx0			I	—	—			I/Z*12	I	I
IEBus™ controller	IETxD			O	—	O/Z*9			O/Z*9	O/Z*9	O
	IERxD			I	—	—			Refer to table A.2	I	I
Renesas SPDIF interface	SPDIF_OUT			O	—	O/Z*9			O/Z*9	O/Z*9	O
	SPDIF_IN			I	—	—			Z	Z	I
A/D converter	AN7 to AN0			I	—	—			Z	Z	I
	$\overline{\text{ADTRG}}$			I	—	—			Z	Z	I
NAND flash memory controller	FRB			I	—	—			Z	Z	I
	$\overline{\text{FCE}}$			O	—	O/Z*9			O/Z*9	O/Z*9	O
	FALE	Boot mode	0, 1, 3	O	—	O/Z*9			O/Z*9	O/Z*9	Z
			2	O	—	—	O/Z*9		O/Z*9	O/Z*9	Z
	$\overline{\text{FRE}}$	Boot mode	0, 1, 3	O	—	O/Z*9			O/Z*9	O/Z*9	Z
			2	O	—	—	O/Z*9		O/Z*9	O/Z*9	Z
	FCLE	Boot mode	0, 1, 3	O	—	O/Z*9			O/Z*9	O/Z*9	Z
			2	O	—	—	O/Z*9		O/Z*9	O/Z*9	Z
	$\overline{\text{FWE}}$	Boot mode	0, 1, 3	O	—	O/Z*9			O/Z*9	O/Z*9	Z
			2	O	—	—	O/Z*9		O/Z*9	O/Z*9	Z
	NAF7 to NAF0			I	—	—			Z	Z	Z
		Boot mode	0, 1, 3	O/Z	—	O/Z*9			O/Z*9	O/Z*9	Z
			2	O/Z	—	—	O/Z*9		O/Z*9	O/Z*9	Z

chanhvinh.com

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	Bus Mastership Release
USB 2.0 host/ function module	DP, DM	I/Z	Z	I/Z			Z	I/Z	I/Z
		O/Z	Z	O/Z			Z	O/Z	O/Z
	VBUS	I	I	I			I	I	I
	REFRIN	I	I	I			I	I	I
	USB_X1*6	I	I	I			Z	Z	I
	USB_X2*6	O	O	O			L	L	O
Video display controller 3	LCD_DATA15 to LCD_DATA0	O	—	O/Z*9			O/Z*9	O/Z*9	O
	LCD_DE	O	—	O/Z*9			O/Z*9	O/Z*9	O
	LCD_CLK	O	—	O/Z*9			O/Z*9	O/Z*9	O
	LCD_VSYNC, LCD_HSYNC	O	—	O/Z*9			O/Z*9	O/Z*9	O
	LCD_M_DISP	O	—	O/Z*9			O/Z*9	O/Z*9	O
	LCD_EXTCLK	I	—	—			Z	Z	I
	DV_CLK	I	—	—			Z	Z	I
	DV_DATA7 to DV_DATA0	I	—	—			Z	Z	I
	DV_VSYNC, DV_HSYNC	I	—	—			Z	Z	I
SD host interface	SD_CLK	O	—	O/Z*9			O/Z*9	O/Z*9	O
	SD_CMD	I	—	—			Z	Z	I
		O/Z	—	O/Z*9			O/Z*9	O/Z*9	O/Z
	SD_D3 to SD_D0	I	—	—			Z	Z	I
		O/Z	—	O/Z*9			O/Z*9	O/Z*9	O/Z
	SD_CD	I	—	—			Z	Z	I
	SD_WP	I	—	—			Z	Z	I

chanhvinh.com

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	Bus Mastership Release
General purpose I/O ports	PA3, PA2		I	—	I			Z	Z	I
			O/Z	—	O/Z*9			O/Z*9	O/Z*9	O/Z
	PA1, PA0		I	—	I		Z	Z	Z	I
			O/Z	—	O/Z*9			O/Z*9	O/Z*9	O/Z
	PB22 to PB21, PC10, PC9, PC4 to PC2, PF8 to PF0, PG24 to PG12, PG9 to PG0, PJ11 to PJ4, PJ2, PJ0, PK11 to PK0		I	Z	I		Z	Z	Z	I
			O/Z	Z	O/Z*9			O/Z*9	O/Z*9	O/Z
	PB20 to PB1, PC1, PC0, PD3 to PD0 (Boot mode 1 to 3 only)		I	Z	I		Z	Z	Z	I
			O/Z	Z	O/Z*9			O/Z*9	O/Z*9	O/Z
	PC8 to PC5, PG11, PG10, PJ3, PJ1		I	Z	I		Refer to table A.2	Refer to table A.2	I	I
			O/Z	Z	O/Z*9			O/Z*9	O/Z*9	O/Z
	PD15 to PD4 (Boot mode 1 to 3 only)		I	Z	I		Z	Z	Z	I
		Boot mode 2	O/Z	Z	Z	O/Z*9		O/Z*9	O/Z*9	O/Z
		Boot mode 1, 3	O/Z	Z	O/Z*9			O/Z*9	O/Z*9	O/Z
	PE5 to PE0		I	Z	I		Z	Z	Z	I
			O/Z	Z	Z			Z	Z	O/Z
	PF12, PF11		I	Z	I		Z	Z	Z	I
		Boot mode 0, 2	O/Z	Z	O/Z*9			O/Z*9	O/Z*9	O/Z
		Boot mode 1, 3	O/Z	Z	Z	O/Z*9		O/Z*9	O/Z*9	O/Z
	PF10, PF9		I	Z	I		Z	Z	Z	I
		Boot mode 0	O/Z	Z	O/Z*9			O/Z*9	O/Z*9	O/Z
		Boot mode 1 to 3	O/Z	Z	Z	O/Z*9		O/Z*9	O/Z*9	O/Z
	PH7 to PH0		I	Z	I		Z	Z	Z	I

chanhvinh.com

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	Bus Mastership Release
Motor control PWM timer	PWM1A, PWM1B, PWM1C, PWM1D, PWM1E, PWM1F, PWM1G, PWM1H, PWM2A, PWM2B, PWM2C, PWM2D, PWM2E, PWM2F, PWM2G, PWM2H	O	—	O/Z*9			O/Z*9	O/Z*9	O
User debugging inter-face*15	$\overline{\text{TRST}}$	I	I	I			Z	I	I
	TCK	I	I	I			Z	I	I
	TDI	I	I	I			Z	I	I
	TDO	O/Z*14	O/Z*14	O/Z*14			O/Z*14	O/Z*14	O/Z*14
	TMS	I	I	I			Z	I	I
Emulator*15	$\overline{\text{AUDSYNC}}$	—	—	—			—	—	—
	AUDCK	—	—	—			—	—	—
	AUDATA3 to AUDATA0	—	—	—			—	—	—
	$\overline{\text{ASEBRKAK}}$/$\overline{\text{ASEBRK}}$	Z	Z	Z			Z	Z	Z

chanhvinh.com

Table 38.2 Input Buffer State in Deep Standby Mode

Pin Function			Pin State	
Type	**Pin Name**		**1-Mbyte Version**	**640-Kbyte Version**
Interrupt	IRQ7 to IRQ4	PC8 to PC5	I/Z*[1,2]	
		PG7 to PG4	Z	
	IRQ3, IRQ2	PE3, PE2	Z	
		PG11, PG10	Z	I/Z*[1,2]
	IRQ1, IRQ0	PE1, PE0	Z	
		PJ3, PJ1	I/Z*[1,2]	
Multi-function timer pulse unit 2	TIOC4A to TIOC4D	PB16 to PB19	Z	
		PC5 to PC8	I/Z*[1,2]	
Serial communication interface with FIFO	RxD7, RxD6, RxD4 to RxD1	—	Z	
	RxD5	—	Z	I/Z*[1,2]
	RxD0	—	I/Z*[1,2]	
Serial sound interface	SSIRxD0	PG10	Z	I/Z*[1,2]
		PK10	Z	
Serial I/O with FIFO	SIOFRxD	—	Z	I/Z*[1,2]
IEBus™ controller	IERxD	PJ1	I/Z*[1,2]	
		PJ5	Z	
General purpose I/O ports	PC8 to PC5	—	I/Z*[1,2]	
	PG11, PG10	—	Z	I/Z*[1,2]
	PJ3, PJ1	—	I/Z*[1,2]	

[Legend]

I: Input

O: Output

H: High-level output

L: Low-level output

Z: High-impedance

Notes: 1. Indicates the power-on reset by low-level input to the $\overline{\text{RES}}$ pin. The pin states after a power-on reset by the user debugging interface reset assert command or the watchdog timer overflow is the same as the initial pin states at normal operation (see section 32, General Purpose I/O Ports).

chanhvinh.com

2. After the chip has shifted to the power-on reset state from deep standby mode by the input on any of pins NMI, PC8 to PC5, PJ3, and PJ1, the pins retain the state until the IOKEEP bit in the deep standby cancel source flag register (DSFR) is cleared (see section 33, Power-Down Modes).
3. The EBUSKEEPE bit in deep standby control register (DSTCR) (see section 33, Power-Down Modes).
4. This LSI enters the power-on reset state for a certain period after recovery from deep standby control mode (see section 33, Power-Down Modes).
5. Depends on the setting of the RCKSEL bit in the realtime clock control register 5 (RCR5) (see section 14, Realtime Clock).
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.
7. Depends on the setting of the CKOEN bit in the frequency control register (FRQCR) of the clock pulse generator (see section 5, Clock Pulse Generator).
8. Depends on the setting of the AXTALE bit in the software reset control register (SWRSTCR) (see section 33, Power-Down Modes).
9. Depends on the setting of the HIZ bit in the standby control register 3 (STBCR3) (see section 33, Power-Down Modes).
10. Depends on the setting of the HIZMEM bit in the common control register (CMNCR) of the bus state controller (see section 9, Bus State Controller).
11. Depends on the setting of the HIZCNT bit in the common control register (CMNCR) of the bus state controller (see section 9, Bus State Controller).
12. Depends on the setting of the corresponding bit in the deep standby cancel source select register (DSSSR) (see section 33, Power-Down Modes).
13. Depends on the setting of the RTCEN bit in the realtime clock control register 2 (RCR2) (see section 14, Realtime Clock).
14. Z when the TAP controller of the user debugging interface is neither the Shift-DR nor Shift-IR state.
15. These are the pin states in product chip mode ($\overline{\text{ASEMD}}$ = H). See the Emulation Manual for the pin states in ASE mode ($\overline{\text{ASEMD}}$ = L).
16. When this is an output, the output is fixed to either the High or Low level. There is no oscillation.

chanhvinh.com

38.2 Treatment of Unused Pins

Table 38.3 Handling of Pins that are not in Use (Except for User Debugging Interface and Emulator Interface Pins)

Pin	Handling
NMI	Fix this pin at a high level (pull up or connect to a power-supply).
DP, DM, and VBUS	Connect these pins to USBDPVss.
REFRIN	Connect this pin, via a 5.6 kΩ ± 20 % resistor, to USBAPVcc.
1.2-V power dedicated to the USB (USBAVcc, USBDVcc, USBUVcc)	Supply power at 1.2 V
3.3-V power dedicated to the USB (USBAPVcc, USBDPVcc)	Supply power at 3.3 V
Dedicated USB ground (USBAPVss, USBDPVss, USBAVss, USBDVss, USBUVss)	Connect to ground
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).
Input/output pins other than those listed above	Make the input-pin settings and then fix the level (pull them up or down); alternatively, make the output-pin settings and leave the pins open-circuit.
Dedicated output pins other than those listed above	Open-circuit

Note: It is recommended that the values of pull-up or pull-down resistors are in the range from 4.7 kΩ to 100 kΩ.

chanhvinh.com

Table 38.4 Handling of Pins that are not in Use (When User Debugging Interface is Not Used in Product Chip Mode)

Pin	Handling
$\overline{\text{ASEMD}}$	Fix this pin at a high level (pull up or connect to the power-supply level).
$\overline{\text{TRST}}$	Fix this pin at a low level (pull down or connect to the ground level).
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, $\overline{\text{ASEBRKAK}}$/$\overline{\text{ASEBRK}}$	Open-circuit

Notes: 1. When using the user debugging interface, handle these pins as described in the manual for the emulator.
2. It is recommended that the values of pull-up or pull-down resistors are in the range from 4.7 kΩ to 100 kΩ.

chanhvinh.com

38.3 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, 38.1, Pin States. Handling of unused pins as described under section 38.2, Treatment of Unused Pins, also applies in deep standby mode.

Table 38.5 Handling of Pins in Deep Standby Mode

Pin	Handling
1.2-V power (Vcc, PLLVcc, USBDVcc, USBUVcc, USBAVcc)	Supply power at 1.2 V
3.3-V power (PVcc, AVcc, USBDPVcc, USBAPVcc)	Supply power at 3.3 V
Ground (Vss, PLLVss, USBDVss, USBUVss, USBAVss, AVss, USBDPVss, USBAPVss)	Connect to ground
VBUS	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 37.2, DC Characteristics (2) [Current Consumption] will be drawn by the pin fixed to the high level.
REFRIN	Connect this pin to USBAPVss via a 5.6 kΩ ± 1 % resistor
AVref	Fix the level on this pin (from 3.0 V to AVcc)
EXTAL, RTC_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, 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.

chanhvinh.com

Pin	Handling
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

38.4 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 5, Clock Pulse Generator.

Figures 38.1 and 38.2 are examples of externally allocated capacitors in the SH7262 Group and SH7264 Group, respectively.

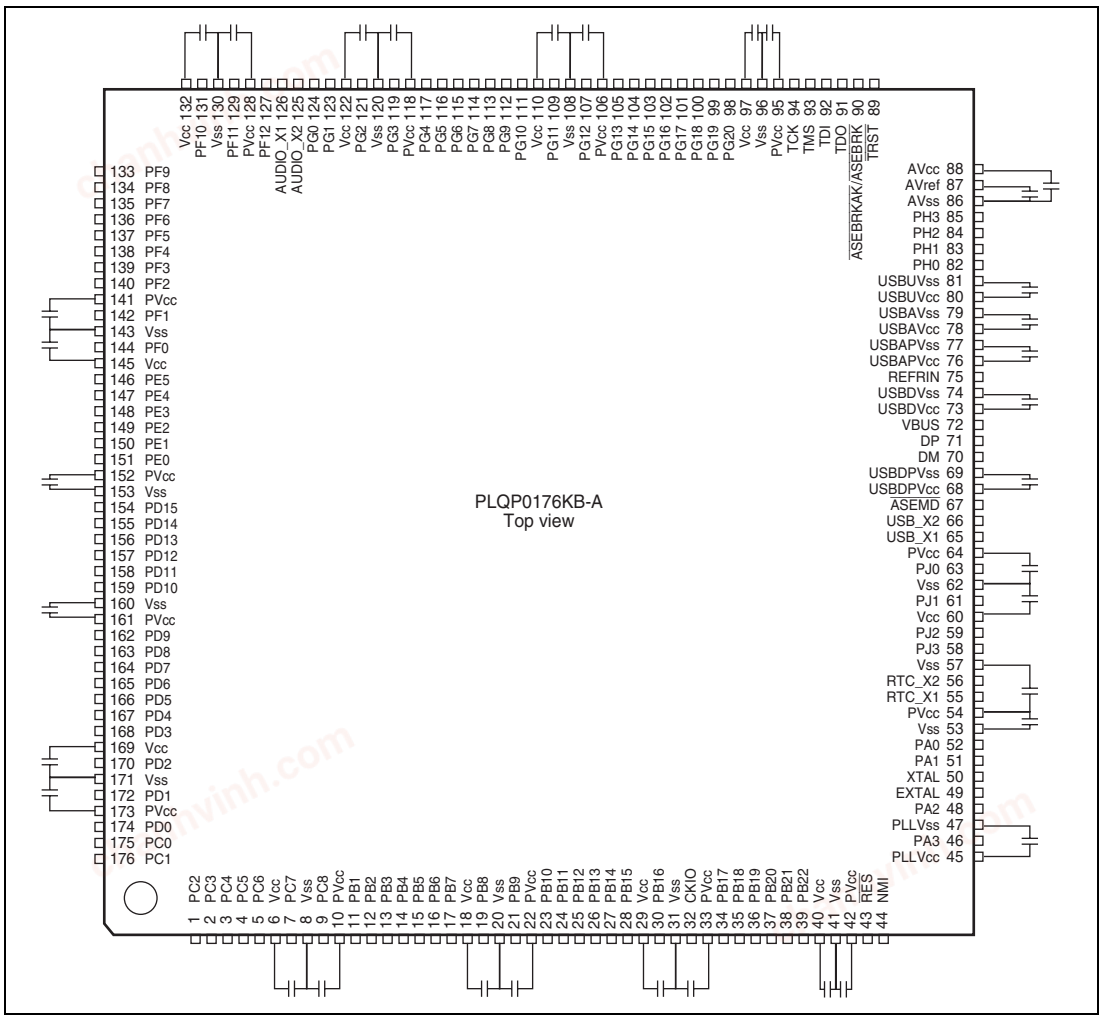


Figure 38.1 Example of Externally Allocated Capacitors in the SH7262 Group

chanhvinh.com

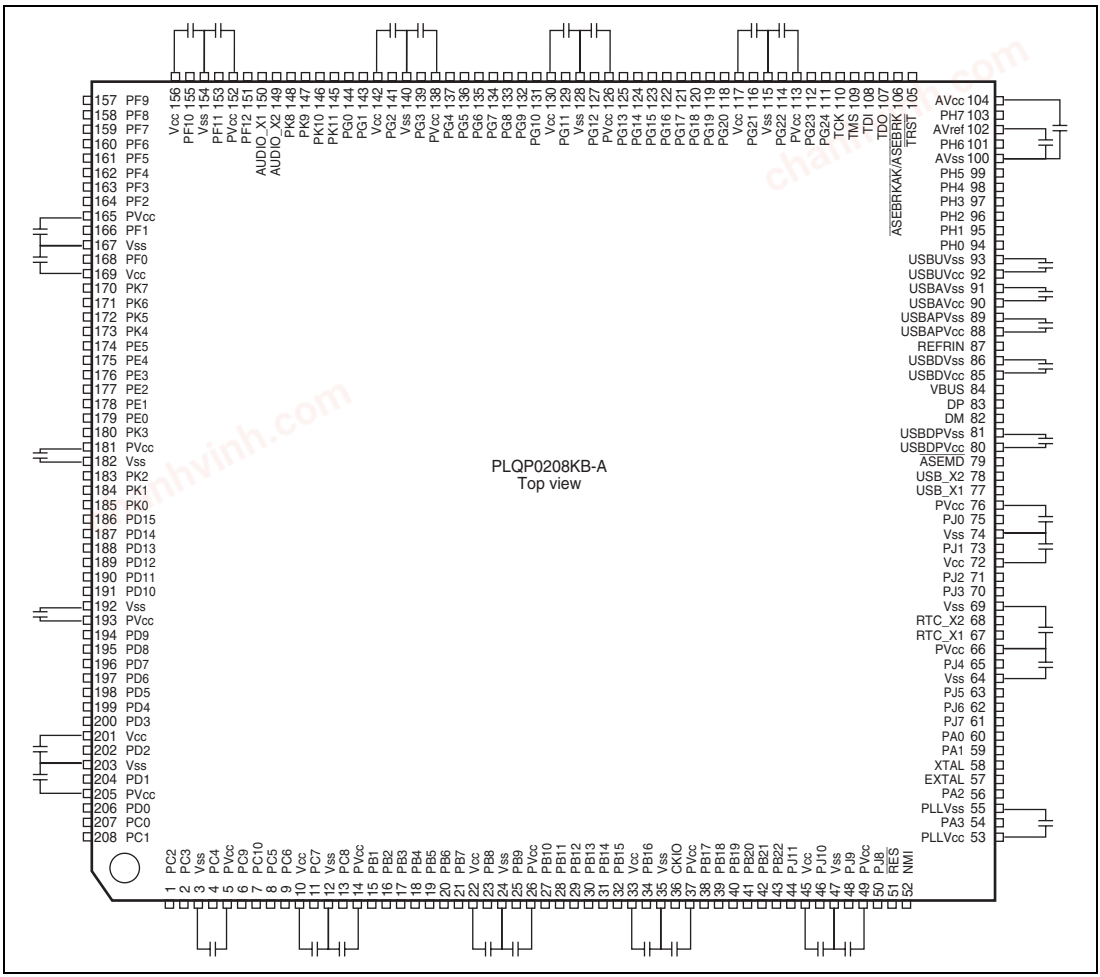
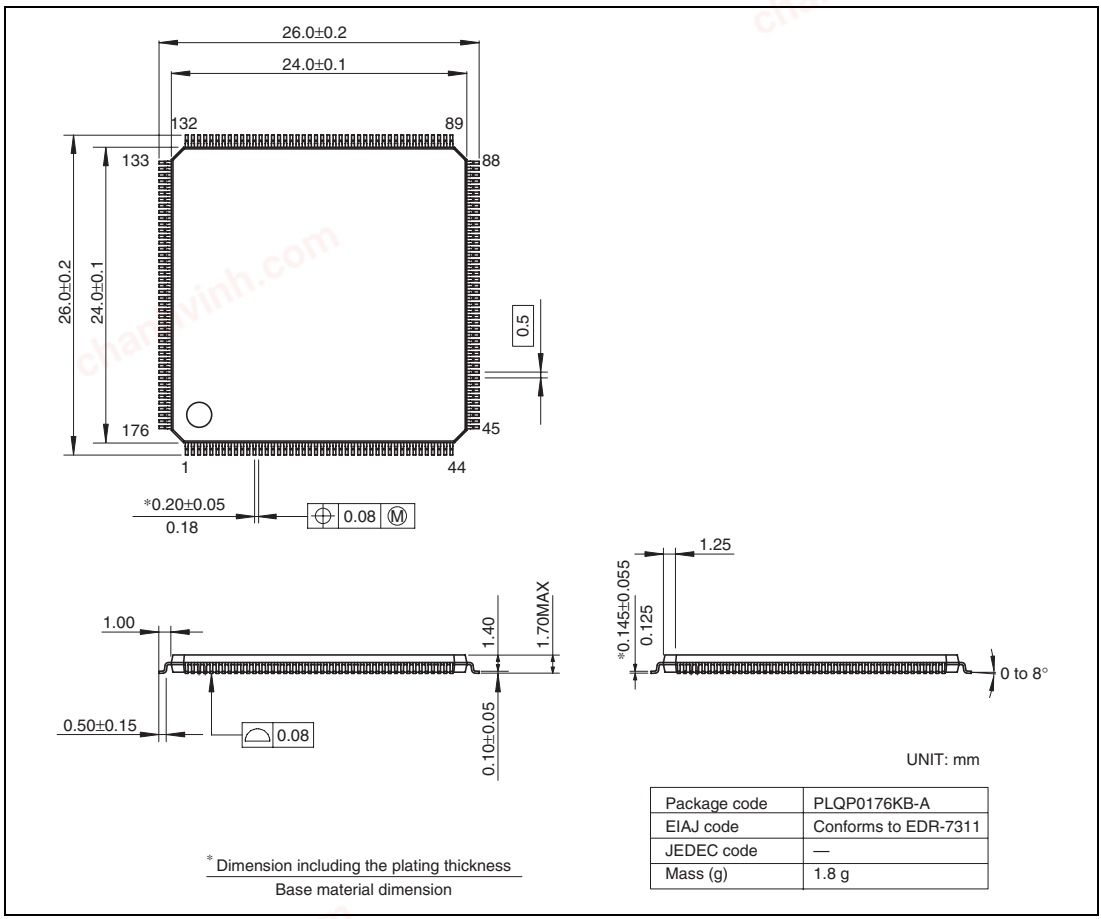


Figure 38.2 Example of Externally Allocated Capacitors in the SH7264 Group

chanhvinh.com

Appendix

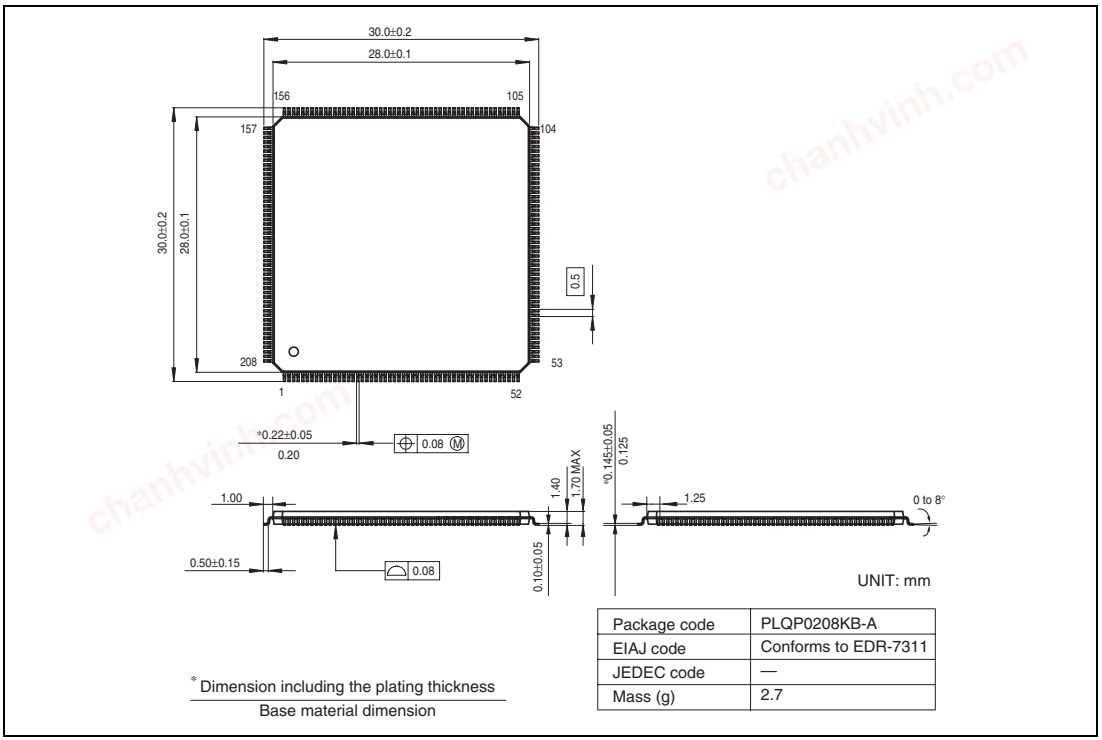
A. Package Dimensions



Package code	PLQP0176KB-A
EIAJ code	Conforms to EDR-7311
JEDEC code	—
Mass (g)	1.8 g

Figure A.1 Package Dimensions of the SH7262

chanhvinh.com

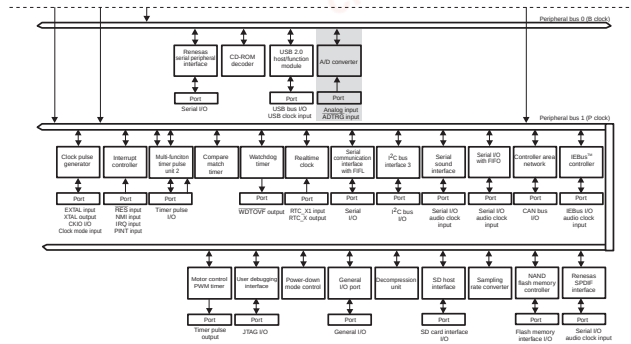
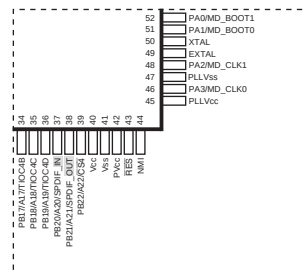
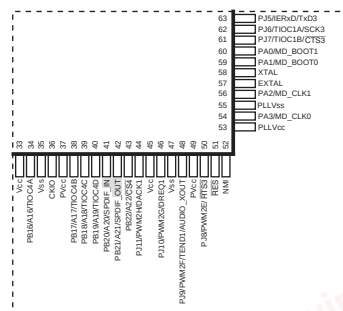


Package code	PLQP0208KB-A
EIAJ code	Conforms to EDR-7311
JEDEC code	—
Mass (g)	2.7

Figure A.2 Package Dimensions of the SH7264

chanhvinh.com

Main Revisions for This Edition

Item	Page	Revision (See Manual for Details)
1.3 Block Diagram Figure 1.1 Block Diagram	14	Figure amended 
1.4 Pin Assignment Figure 1.2 (2) Pin Assignment for the SH7262 Group (640-Kbyte Version)	16	Figure amended 
Figure 1.2 (4) Pin Assignment for the SH7264 Group (640-Kbyte Version)	18	Figure amended 
1.5 Pin Functions Table 1.3 Pin Functions	24	Table amended

Classification	Symbol	I/O	Name	Function
Renesas serial peripheral interface	SSL10, SSL00	I/O	Slave select	Slave select I/O pin.

chanhvinh.com

Item	Page	Revision (See Manual for Details)
1.6 List of Pins Table 1.4 List of Pins	31	Table amended

SH7262 Pin No.	SH7264 Pin No.	Function 1 Symbol	Function 1 I/O	Function 2 Symbol	Function 2 I/O	Function 3 Symbol	Function 3 I/O	Function 4 Symbol	Function 4 I/O
37	41	PB20	I(s)/O	A20	O	SPDIF_IN*	I(s)	—	—

	32	Table amended

SH7262 Pin No.	SH7264 Pin No.	Function 1 Symbol	Function 1 I/O	Function 2 Symbol	Function 2 I/O	Function 3 Symbol	Function 3 I/O	Function 4 Symbol	Function 4 I/O
38	42	PB21	I(s)/O	A21	O	SPDIF_OUT*	O	—	—

	36	Table amended

SH7262 Pin No.	SH7264 Pin No.	Function 1 Symbol	Function 1 I/O	Function 2 Symbol	Function 2 I/O	Function 3 Symbol	Function 3 I/O	Function 4 Symbol	Function 4 I/O
—	103	PH7	I	AN7	I(a)	—	—	—	—
88	104	AVcc							
89	105	$\overline{\text{TRST}}$	I(s)	—	—	—	—	—	—
90	106	ASEBRKAK/ASEBRK	I(s)/O	—	—	—	—	—	—
91	107	TDO	O	—	—	—	—	—	—
92	108	TDI	I	—	—	—	—	—	—
93	109	TMS	I	—	—	—	—	—	—
94	110	TCK	I	—	—	—	—	—	—

Table amended

SH7262 Pin No.	SH7264 Pin No.	Function 5 Symbol	Function 5 I/O	Function 6 Symbol	Function 6 I/O	Function 7 Symbol	Function 7 I/O	Figure 1.3
—	103	—	—	—	—	—	—	(4)
88	104							
89	105	—	—	—	—	—	—	(3)
90	106	—	—	—	—	—	—	(7)
91	107	—	—	—	—	—	—	(5)
92	108	—	—	—	—	—	—	(2)
93	109	—	—	—	—	—	—	(2)
94	110	—	—	—	—	—	—	(2)

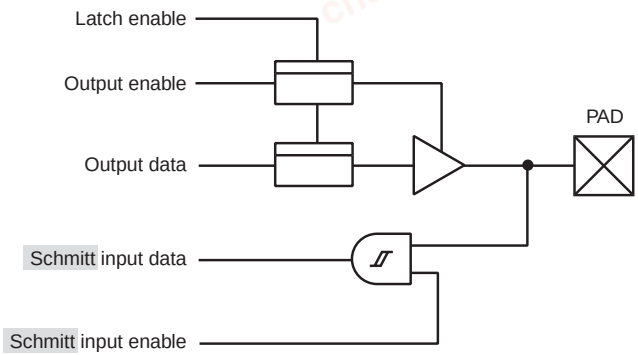
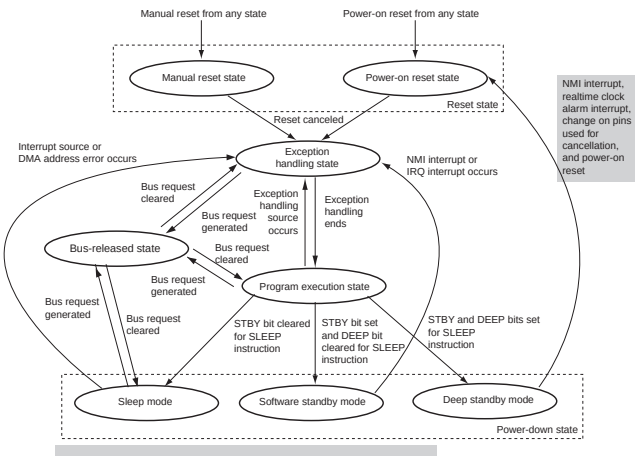
	38	Table amended

SH7262 Pin No.	SH7264 Pin No.	Function 5 Symbol	Function 5 I/O	Function 6 Symbol	Function 6 I/O	Function 7 Symbol	Function 7 I/O	Figure 1.3
114	134	IRQ7*	I(s)	—	—	—	—	(7)
115	135	IRQ6*	I(s)	—	—	—	—	(7)
116	136	IRQ5*	I(s)	—	—	—	—	(7)
117	137	IRQ4*	I(s)	—	—	—	—	(7)

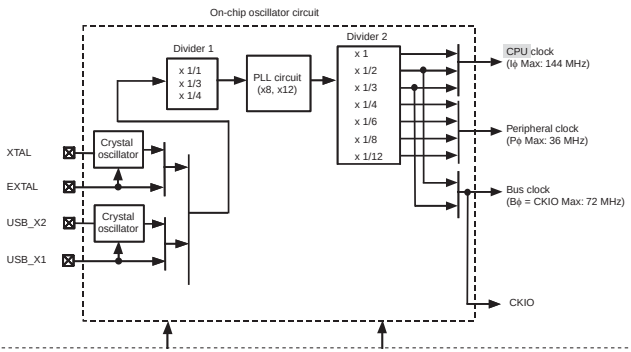
	40	Table amended

SH7262 Pin No.	SH7264 Pin No.	Function 5 Symbol	Function 5 I/O	Function 6 Symbol	Function 6 I/O	Function 7 Symbol	Function 7 I/O	Function 8 Symbol	Function 8 I/O	Figure 1.3
134	158	—	—	—	—	SD_CD*	I(s)	—	—	(7)
135	159	TCLKD	I(s)	—	—	SD_WP*	I(s)	—	—	(7)
136	160	TCLKB	I(s)	—	—	SD_D1*	I(s)/O	—	—	(7)
137	161	TCLKC	I(s)	—	—	SD_D0*	I(s)/O	AUDATA3	O	(7)
138	162	TxD3	O	—	—	SD_CLK*	O	AUDATA2	O	(7)
139	163	RxD3	I(s)	—	—	SD_CMD*	I(s)/O	AUDATA1	O	(7)
140	164	TxD2	O	DACK0	O	SD_D3*	I(s)/O	AUDATA0	O	(7)
141	165									
142	166	RxD2	I(s)	DREQ0	I(s)	SD_D2*	I(s)/O	$\overline{\text{AUDSYNC}}$	O	(7)

chanhvinh.com

Item	Page	Revision (See Manual for Details)
1.6 List of Pins Table 1.4 List of Pins	43	Note added Note: * Can be used in 640K-byte version only.
Figure 1.3 (7) Simplified Circuit Diagram (Bidirectional Buffer, Schmitt AND Input, with Latch)	45	Figure amended 
2.5 Processing States Figure 2.6 Transitions between Processing States	92	Figure amended 
Section 5 Clock Pulse Generator	115	Description amended This LSI has a clock pulse generator that generates a CPU clock (Iϕ), a peripheral clock (Pϕ), and a bus clock (Bϕ).

chanhvinh.com

Item	Page	Revision (See Manual for Details)
5.1 Features	115	Description amended • Three clocks generated independently A CPU clock (Iϕ) for the CPU and cache; a peripheral clock (Pϕ) for the on-chip peripheral modules; a bus clock (Bϕ = CKIO) for the external bus interface • Frequency change function CPU and peripheral clock frequencies can be changed independently using the PLL (phase locked loop) circuits and divider circuits within this module. Frequencies are changed by software using frequency control register (FRQCR) settings.
Figure 5.1 Block Diagram	116	Figure amended 
5.1 Features (4) Divider 2	117	Description amended Divider 2 generates a clock signal whose operating frequency can be used for the CPU clock, the peripheral clock, and the bus clock. The division ratio of the bus clock depends on the clock operating mode. The division ratio of the CPU clock and the peripheral clock is set by the frequency control register.
(7) 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 and the frequency division ratio of the CPU clock and the peripheral clock (Pϕ).

chanhvinh.com

Item	Page	Revision (See Manual for Details)
5.3 Clock Operating Modes Table 5.3 Relationship between Clock Operating Mode and Frequency Range	121	Table amended Clock Operating Mode / FRQCR Setting*1 / PLL Frequency Multiplier – PLL Circuit / Ratio of Internal Clock Frequencies (I:B:P)*2 / Selectable Frequency Range (MHz) – Input Clock*3 / Output Clock (CKIO Pin) / CPU Clock (Iϕ) / Bus Clock (Bϕ) / Peripheral Clock (Pϕ) Notes amended Notes: 1. x in the FRQCR register setting depends on the set value in bits 12, 13, and 14.
5.4.1 Frequency Control Register (FRQCR)	122	Description amended The register specifies the frequency division ratio for the CPU clock and peripheral clock (Pϕ). FRQCR is accessed by word.
	124	Table amended Bit / Bit Name / Initial Value / R/W / Description 5, 4 / IFC[1:0] / 01/10* / R/W / CPU Clock Frequency Division Ratio This bit specifies the frequency division ratio of the CPU clock with respect to the output frequency of PLL circuit. 00: 1 time 01: 1/2 times 10: 1/3 times 11: Reserved (setting prohibited)
5.5 Changing the Frequency	125	Description amended The frequency of the CPU clock (Iϕ) and peripheral clock (Pϕ) can be changed by changing the division rate of divider.
5.5.1 Changing the Division Ratio	125	Description amended The division rate of divider can be changed by the following operation. 1. In the initial state, IFC1 to IFC0 = B'01 or B'10 and PFC2 to PFC0 = B'100 or B'011. 2. Set the desired value in the IFC1 to IFC0 and PFC2 to PFC0 bits. The values that can be set are limited by the clock operating mode and the multiplication rate of PLL circuit. Note that if the wrong value is set, this LSI will malfunction. 3. After the register bits (IFC1 to IFC0 and PFC2 to PFC0) have been set, the clock is supplied of the new division ratio.
6.2.2 Types of Reset Table 6.6 Reset States	137, 138	Table replaced

chanhvinh.com

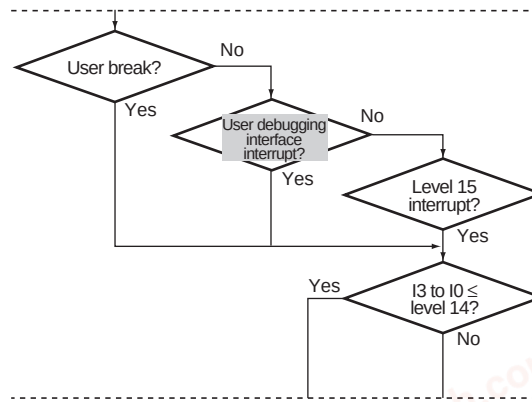
Item	Page	Revision (See Manual for Details)
6.9.4 Interrupt Control via Modification of Interrupt Mask Bits	153	Newly added
6.9.5 Note before Exception Handling Begins Running (1) Manual Reset	154	Description amended Before the settings necessary to access the external CS0 space are completed, the manual reset should not be generated. When a manual reset is issued, the CPU fetches the program execution start address from the manual reset vector table address offset (H'00000008), which is to say always from the external SC0 space. Additionally, in the case that no memory is connected to the external CS0 space in boot mode 1 to 3, the manual reset should not be generated.
(2) NMI Interrupt		Description amended Before the exception handling vector table is stored in a memory and the settings necessary to access the memory are completed, the NMI interrupt should not be generated. (Do not enable interrupts on the 640 KB version.) Specially in boot mode 1 to 3, the VBR is kept as the initial value H'00000000 in the period of the boot operation (before the transfer of the loader program is completed and the CPU jumps to the on-chip high-speed RAM). Before the VBR is changed or the settings necessary to access the external address space are completed in the loader program, the NMI interrupt should not be generated. (Do not enable interrupts on the 640 KB version.)
7.4 Interrupt Sources	173	Description amended There are five types of interrupt sources: NMI, user debugging interface, IRQ, PINT, and on-chip peripheral modules.

chanhvinh.com

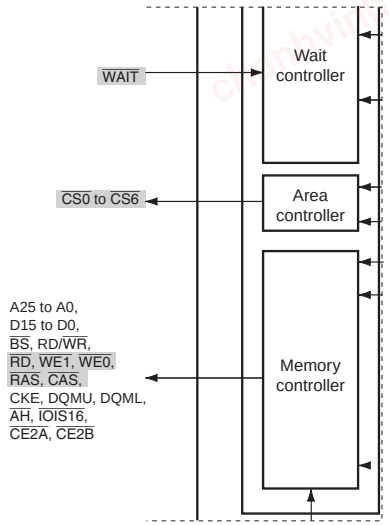
Item	Page	Revision (See Manual for Details)
7.5 Interrupt Exception Handling Vector Table and Priority Table 7.4 Interrupt Exception Handling Vectors and Priorities	189	Table amended

Interrupt Source		Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
CD-ROM decoder	ISY	282	H'00000468 to H'0000046B	0 to 15 (0)	IPR20 (3 to 0)	1	High
	IERR	283	H'0000046C to H'0000046F			2	
	ITARG	284	H'00000470 to H'00000473			3	
	ISEC	285	H'00000474 to H'00000477			4	
	IBUF	286	H'00000478 to H'0000047B			5	
	IREADY	287	H'0000047C to H'0000047F			6	
NAND flash memory controller	FLSTEI	288	H'00000480 to H'00000483	0 to 15 (0)	IPR21 (15 to 12)	1	
	FLTENDI	289	H'00000484 to H'00000487			2	
	FLTREQ0I	290	H'00000488 to H'0000048B			3	
	FLTREQ1I	291	H'0000048C to H'0000048F			4	
SD host interface	SDHI3	292	H'00000490 to H'00000493	0 to 15 (0)	IPR21 (11 to 8)	1	
	SDHI0	293	H'00000494 to H'00000497			2	
	SDHI1	294	H'00000498 to H'0000049B			3	Low

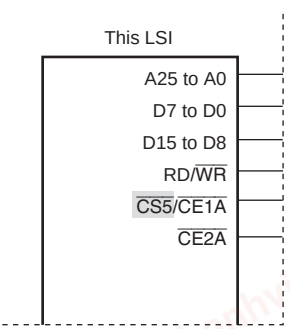
7.6.1 Interrupt Operation Sequence Figure 7.2 Interrupt Operation Flow	193	Figure amended



chanhvinh.com

Item	Page	Revision (See Manual for Details)
9.1 Features Figure 9.1 Block Diagram of Bus State Controller	231	Figure amended 
9.3.1 Address Map Table 9.2 Address Map	235	Table amended Internal Address: H'40000000 to H'FFFBFFFF Space: Other Memory to be Connected: On-chip RAM, reserved area* Cache: —
9.3.2 Data Bus Width and Endian Specification of Each Area and Related Pin Settings Depending on Boot Mode	235	Titlle and Description amended In boot mode 0, the state of area 0 is fixed to the state with bus width of 16 bits and big endian, because this LSI is started up by the program stored in the ROM connected to area 0. The initial state of areas 1 to 6 is the same as that of area 0, but the bus width and endian specification can be changed by a program. In this mode, settings required to read the ROM in area 0, such as settings for some of the addresses, the data bus, and the pin functions of $\overline{CS0}$ and $\overline{RD}$ only, are selected automatically immediately after a power-on reset, but the initial function settings of the other pins are as general ports. These pins cannot be used until their functions are set by a program. Do not perform other than read access to area 0 until the pin settings are completed. In boot modes 1 to 3, the state of areas 0 to 6 can be changed from the initial state by the program, because in these modes the LSI is started by the program stored in the NAND flash memory or the serial flash memory. Since pin functions related to this module are not set automatically, they need to be set by the user. Do not access the external address space until the pin settings are completed.

chanhvinh.com

Item	Page	Revision (See Manual for Details)
9.3.2 Data Bus Width and Endian Specification of Each Area and Related Pin Settings Depending on Boot Mode Table 9.3 Initial States of Areas in Boot Modes 0, 2, and 3	236	Table amended **Boot Mode** / **Item** / **Area 0** / **Areas 1 to 6** 0 / Data bus width / Fixed to 16 bits. Not changeable. / The initial state is 16 bits. Can be changed by program. Endian specification / Fixed to big endian. Not changeable. / The initial state is big endian. Can be changed by program. Settings of pins related to this module / Pin functions for A20 to A1, D15 to D0, $\overline{CS0}$, and $\overline{RD}$ are set automatically. Other pins need to be set by program. 1 to 3 / Data bus width / The initial state is 16 bits. Can be changed by program. Endian specification / The initial state is big endian. Can be changed by program. Note added Notes: 1. In order to connect boot ROM using address lines A21 and above in boot mode 0, pull-down processing must be provided on the board for address lines A21 and above. 2. Depending on the type of memory used, there may be limitations on the data bus width. For details, see 9.4.2, CSn Space Bus Control Register. 3. Since the $\overline{CS1}$ and A0 functions are assigned to the same pin, it is not possible to use area 1 and a device with an 8-bit bus connection at the same time. 4. Since the $\overline{CS4}$ and A22 functions are assigned to the same pin, it is not possible to use area 4 and a device using address lines A22 and above at the same time.
9.5.9 PCMCIA Interface Figure 9.38 Example of PCMCIA Interface Connection	346	Figure amended 
10.1 Features	367	Description amended • Transfer requests — Renesas SPDIF interface: two sources

chanhvinh.com

Item	Page	Revision (See Manual for Details)
10.3.4 DMA Channel Control Registers (CHCR)	382	Table amended

Bit	Bit Name	Initial Value	R/W	Description
31	TC	0	R/W	Transfer Count Mode Specifies whether to transmit data once or for the count specified in DMATCR by one transfer request. This function is valid only in on-chip peripheral module request mode. Note that when this bit is set to 0, the TB bit must not be set to 1 (burst mode). Do not set TC to 1 when a module other than multi-function timer pulse unit 2, compare-match timer, controller area network, CD-ROM decoder, or A/D converter is set as the transfer request source. 0: Transmits data once by one transfer request 1: Transmits data for the count specified in DMATCR by one transfer request

	383	Table amended		

Bit	Bit Name	Initial Value	R/W	Description
26	DAF	0	R/W	Fixed Destination Address 16-Byte Transfer Enabled when the transfer size (set in TS[1:0]) is 16 bytes and the destination address mode (set in DM[1:0]) is fixed address. 0: 16 bytes of data are transferred to the address set in the DAR register. The write destination address is the address set in the DAR register + H'0, + H'4, + H'8, or + H'C. 1: Four bytes of data are transferred four times to the address set in the DAR register. The write destination address is fixed at the address set in the DAR register. This function is exclusively for use with the CD-ROM decoder, USB 2.0 host/function module, and sampling rate converter.
25	SAF	0	R/W	Fixed Source Address 16-Byte Transfer Enabled when the transfer size (set in TS[1:0]) is 16 bytes and the source address mode (set in SM[1:0]) is fixed address. 0: 16 bytes of data are transferred from the address specified in SAR. The read address is the address set in the SAR register + H'0, + H'4, + H'8, or + H'C. 1: Four bytes of data are transferred four times from the address specified in SAR. The read address is fixed at the address set in the SAR register. This function is exclusively for use with the CD-ROM decoder, USB 2.0 host/function module, and sampling rate converter.

10.3.9 DMA Extension Resource Selectors 0 to 7 (DMARS0 to DMARS7)	398	Description amended • Renesas SPDIF interface: two sources

chanhvinh.com

Item	Page	Revision (See Manual for Details)
11.3.29 Timer Waveform Control Register (TWCR)	502	Table amended

Bit	Bit Name	Initial Value	R/W	Description
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 11.40. 0: Outputs the initial value specified in TOCR 1: Suppresses initial output [Setting condition] When 1 is written to WRE after reading WRE = 0

11.4.8 Complementary PWM Mode (2) Outline of Complementary PWM Mode Operation (n) Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode	558	Description amended Even in the Tb interval at the trough, if synchronous clearing occurs in the initial value output period (indicated by (1) in figure 11.56) 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 11.7.23, Notes on Output Waveform Control During Synchronous Counter Clearing in Complementary PWM Mode.
11.7.23 Notes on Output Waveform Control During Synchronous Counter Clearing in Complementary PWM Mode	610, 611	Newly added

chanhvinh.com

Item	Page	Revision (See Manual for Details)
14.3.17 Control Register 2 (RCR2)	692	Table amended

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	PES[2:0]	000	R/W	Interrupt Enable Flags These bits specify 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

14.3.19 Control Register 5 (RCR5)	694	Figure amended

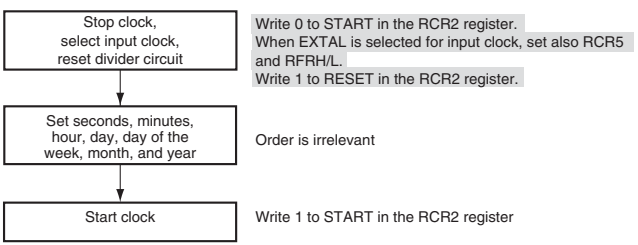
Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	RCKSEL
Initial value:	0	0	0	0	0	0	0	Undefined
R/W:	R	R	R	R	R	R	R	R/W

Table amended

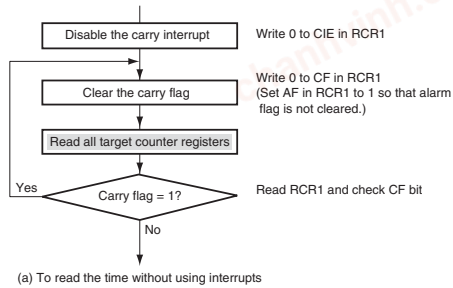
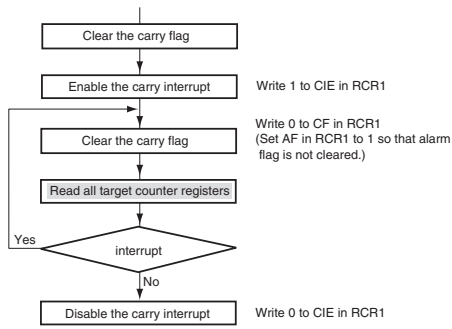
Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	RCKSEL	Undefined	R/W	Operation clock select Operation clock can be selected from either RTC_X1 or EXTAL. The setting of this bit should not be switched during operation. 0: Selects RTC_X1. 1: Selects EXTAL.

14.4.2 Setting Time	697	Figure amended

Figure 14.2 Setting Time



chanhvinh.com

Item	Page	Revision (See Manual for Details)
14.4.3 Reading Time Figure 14.3 Reading Time	698	Figure amended  (a) To read the time without using interrupts  (b) To read the time using interrupts
14.5.3 Transition to Standby Mode after Setting Register	700	Description amended When a transition to standby mode is made after registers in this module are set, sometimes counting is not performed correctly. After making register settings, do not fail to perform a dummy read of the registers to which settings were made before entering the standby state.
14.5.4 Usage Notes when Writing to and Reading the Register	701	Description amended • After writing to a count register such as the second counter or to the RCR2 register, perform two dummy reads before reading from the register. The register value before the write is returned by the two dummy reads. The value newly written to the register takes effect only on the third read operation.
15.2 Input/Output Pins Table 15.1 Pin Configuration	706	Table amended Channel: 0 to 3 / Pin Name: Serial clock pins / Symbol: SCK0 to SCK3* / I/O: I/O / Function: Clock I/O Note amended Note: * Pins SCK3, RTS3 and CTS3 cannot be used in the SH7262 Group.

chanhvinh.com

Item	Page	Revision (See Manual for Details)
15.3.6 Serial Control Register (SCSCR)	718	Table amended

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, BR 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.

15.3.8 Bit Rate Register (SCBRR) — 731 — Table amended

Table 15.5 Bit Rates and SCBRR Settings (Clock Synchronous Mode)

Bit Rate (bits/s)	Pϕ (MHz) 24 n	24 N	28 n	28 N	32 n	32 N	36 n	36 N
500	3	187	3	218	3	249	—	—
1000	3	93	3	108	3	124	3	140
2500	2	149	2	174	2	199	2	224
5000	2	74	2	87	2	99	2	112
10000	1	149	1	174	1	199	1	224
25000	1	59	1	69	1	79	1	89
50000	1	29	0	139	0	159	0	179
100000	0	59	0	69	0	79	0	89
250000	0	23	0	27	0	31	0	35
500000	0	11	0	13	0	15	0	17
1000000	0	5	0	6	0	7	0	8
2000000	0	2	—	—	0	3	—	—

Note deleted

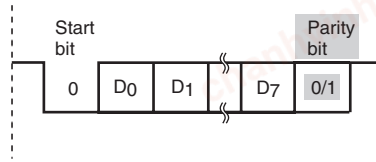
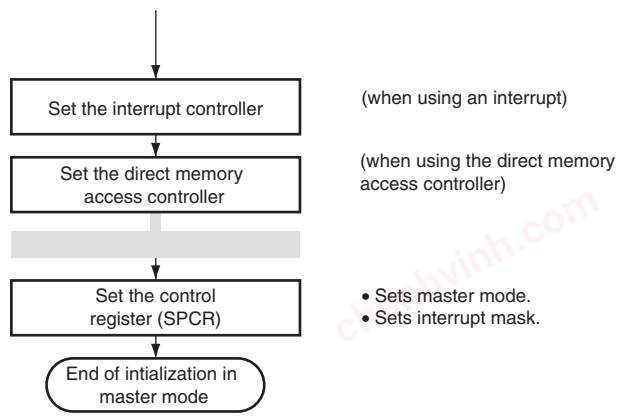
[Legend]

—: Setting possible, but error occurs

Table 15.8 Maximum Bit Rates with External Clock Input (Clock Synchronous Mode, $t_{Scyc} = 12\ t_{pcyc}$) — 733 — Table amended

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
24	2.0000	2000000.0
28	2.3333	2333333.3
32	2.6667	2666666.7
36	3.0000	3000000.0

chanhvinh.com

Item	Page	Revision (See Manual for Details)
15.4.2 Operation in Asynchronous Mode (3) Transmitting and Receiving Data Figure 15.10 Example of Operation Using Modem Control ($\overline{RTS}$)	756	Figure amended 
15.5 Interrupts	766	Description amended It is possible to generate only the ERI or BRI interrupt without generating the RXI interrupt by setting the RIE bit to 0 and the REIE bit to 1 in SCSCR.
16.3.8 Bit Rate Register (SPBR) Table 16.3 Relationship between SPBR and BRDV1 and BRDV0 Settings	790	Table amended Bit Rate SPBR (n) / BRDV[1:0] (N) / Division Ratio / Bφ = 40 MHz / Bφ = 48 MHz / Bφ = 72 MHz 255 / 3 / 4096 / 9.77 kbps / 11.72 kbps / 17.58 kbps
16.4.8 SPI Operation (1) Multi-Master Mode Operation (d) Burst Transfer	832	Description amended 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.
Figure 16.17 Example of Initialization Flowchart in Master Mode	835	Figure amended 

chanhvinh.com

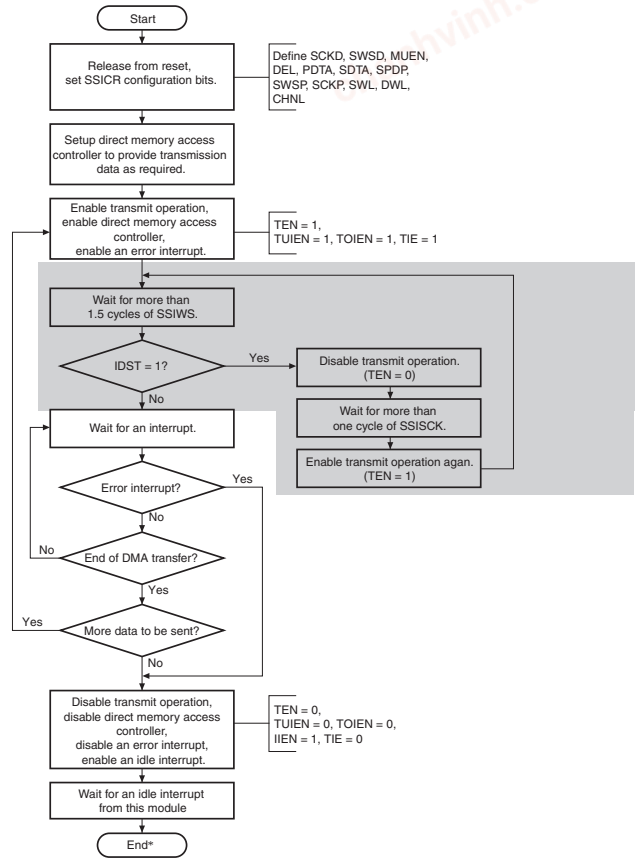
Item	Page	Revision (See Manual for Details)
17.3.1 I²C Bus Control Register 1 (ICCR1)	849	Table amended Bit 7 — Bit Name: ICE — Initial Value: 0 — R/W: R/W — Description: I²C Bus Interface 3 Enable 0: SCL and SDA output is disabled. (Input to SCL and SDA is enabled.) 1: This bit is enabled for transfer operations. (SCL and SDA pins are bus drive state.)
17.3.4 I²C Bus Interrupt Enable Register (ICIER)	856	Table amended Bit 5 — Bit Name: RIE — Initial Value: 0 — R/W: R/W — Description: Receive Interrupt Enable Enables or disables the receive data full interrupt (RXI) when receive data is transferred from ICDRS to ICDRR while RDRF in ICSR is set to 1. RXI can be canceled by clearing the RDRF or RIE bit to 0. 0: Receive data full interrupt request (RXI) are disabled. 1: Receive data full interrupt request (RXI) are enabled.
	857	Table amended Bit 4 — Bit Name: NAKIE — Initial Value: 0 — R/W: R/W — Description: NACK Receive Interrupt Enable Enables or disables the NACK detection and arbitration lost/overrun error interrupt request (NAKI) when NACKF or AL/OVE in ICSR is set to 1. NAKI can be canceled by clearing the NACKF, AL/OVE, or NAKIE bit to 0. 0: NACK receive interrupt request (NAKI) is disabled. 1: NACK receive interrupt request (NAKI) is enabled.
17.4.5 Slave Receive Operation Figure 17.12 Slave Receive Mode Operation Timing (2)	873	Figure replaced
17.7.5 Note on I²C-bus Interface Master Receive Mode	886	Newly added
17.7.6 Note on IICRST and BBSY bits	886	Newly added
Section 18 Serial Sound Interface	887	Description amended The serial sound interface is a module designed to send or receive audio data interface with various devices offering I²S bus compatibility.
18.4.2 Non-Compressed Modes	913	Description amended The non-compressed modes support all serial audio streams split into channels. It supports I²S compatible format as well as many more variants on these modes.

chanhvinh.com

Item	Page	Revision (See Manual for Details)
18.4.2 Non-Compressed Modes (7) Operating Setting Related to Word Length	914	Description amended All bits related to the SSICR's word length are valid in non-compressed modes. There are many configurations this module supports, but the formats described below are I²S compatible, MSB-first left-aligned, and MSB-first right-aligned.
• I²S Compatible Format		Title and description amended Figures 18.3 and 18.4 demonstrate the supported I²S compatible format both with and without padding. Padding occurs when the data word length is smaller than the system word length.
Figure 18.3 I²S Compatible Format (without Padding)		Figure title amended
Figure 18.4 I²S Compatible Format (with Padding)		Figure title amended
	915	Description amended Figure 18.5 shows MSB-first left-aligned format, and figure 18.6 shows MSB-first right-aligned format. Padding is assumed in both cases, but may not be present in a final implementation if the system word length equals the data word length.
(7) Operating Setting Related to Word Length • MSB-First Left-Aligned Format		Title amended
Figure 18.5 MSB-First Left-Aligned Format (Transmitted and Received in the order of Serial Data and Padding Bits)		Figure title amended
• MSB-First Right-Aligned Format		Title amended
Figure 18.6 MSB-First Right-Aligned Format (Transmitted and Received in the Order of Padding Bits and Serial Data)		Figure title amended
(8) Multi-channel Formats	916	Description amended Some devices extend the definition of the specification by I²S bus and allow more than 2 channels to be transferred within two system words.

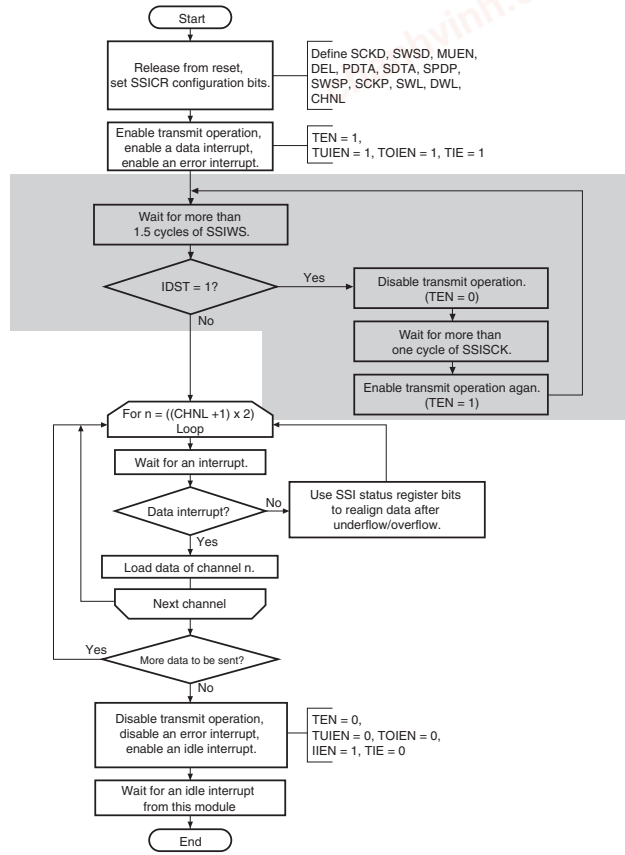
chanhvinh.com

Item	Page	Revision (See Manual for Details)
18.4.4 Transmit Operation (1) Transmission Using Direct Memory Access Controller Figure 18.20 Transmission Using Direct Memory Access Controller	925	Figure amended

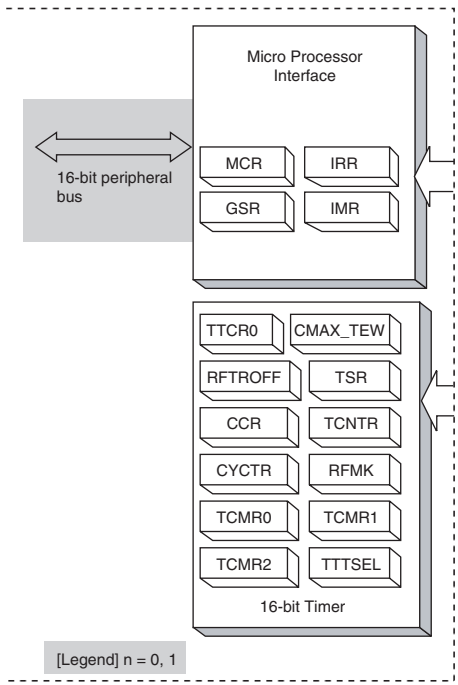


Note: Transmission may not start if the specified procedure is not followed.
* If an error interrupt (underflow/overflow) occurs, go back to the start in the flowchart again.

chanhvinh.com

Item	Page	Revision (See Manual for Details)
18.4.4 Transmit Operation (2) Transmission Using Interrupt-Driven Data Flow Control Figure 18.21 Transmission Using Interrupt-Driven Data Flow Control	926	Figure amended  Note: Transmission may not start if the specified procedure is not followed.
18.5.1 Limitations from Underflow or Overflow during DMA Operation	931	Description amended 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).
19.3 Register Descriptions Table 19.2 Register Configuration	936	Title added and table amended **Register Name** / **Abbreviation** / **R/W** / **Initial Value** / **Address** / **Access Size** Clock select register / SISCR / R/W / H'8000 / H'FFFF4802 / 16 FIFO control register / SIFCTR / R/W* / H'1000 / H'FFFF4810 / 16

chanhvinh.com

Item	Page	Revision (See Manual for Details)
20.1.1 Overview	975	Description amended It serves to facilitate the hardware/software interface so that engineers involved in this module implementation can ensure the design is successful. Deep standby mode can be canceled by change on CRxn (PJ3, PJ1) pin. For details, refer to section 33, Power-Down Modes.
20.2 Architecture	977	Figure amended 
	978	Description amended **Important:** LongWord (32-bit) accesses are converted into two consecutive word (16-bit) accesses by the bus interface.
20.3.3 Control Registers	998	Table amended **Register Name** / **Address** / **Abbreviation** / **Access Size (bits)** Interrupt Request Register / 008 / IRR / 16
(3) Bit Configuration Register (BCR0, BCR1)	1006	Description amended Where: BRP (Baud Rate Pre-scaler) is the value stored in BCR0 incremented by 1 and fclk is the used peripheral clock frequency.

chanhvinh.com

Item	Page	Revision (See Manual for Details)
20.3.3 Control Registers • BCR0 (Address = H'006)	1008	Description amended **Bits 7 to 0—Baud Rate Pre-scale (BRP[7:0] = BCR0 [7:0]):** These bits are used to define the peripheral clock periods contained in a Time Quantum. Table amended

Bit 7: BRP[7]	Bit 6: BRP[6]	Bit 5: BRP[5]	Bit 4: BRP[4]	Bit 3: BRP[3]	Bit 2: BRP[2]	Bit 1: BRP[1]	Bit 0: BRP[0]	Description
0	0	0	0	0	0	0	0	2 X peripheral clock (Initial value)
0	0	0	0	0	0	0	1	4 X peripheral clock
0	0	0	0	0	0	1	0	6 X peripheral clock
:	:	:	:	:	:	:	:	2*(register value + 1) X peripheral clock
1	1	1	1	1	1	1	1	512 X peripheral clock

Item	Page	Revision (See Manual for Details)
(4) Interrupt Request Register (IRR)	1011	Description amended The interrupt request register (IRR) is a 16-bit read/write-clearable register containing status flags for the various interrupt sources.
20.5 Interrupt Sources Table 20.2 Interrupt Sources	1071	Table amended

Interrupt	Description	Interrupt Flag	DMAC Activation
RM0n*1*2, RM1n*1*2	Data frame reception	IRR1*3	Possible*4
	Remote frame reception	IRR2*3	

Notes amended

Notes 4. The direct memory access controller is activated only by an RM0n interrupt.

Item	Page	Revision (See Manual for Details)
20.7 CAN Bus Interface	1073	Description replaced
24.3.2 A/D Control/Status Register (ADCSR)	1260	Table amended

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: AD conversion on 1 to 8 channels*3 110: Scan mode: A/D conversion on 1 to 4 channels 111: Scan mode: AD conversion on 1 to 8 channels*3

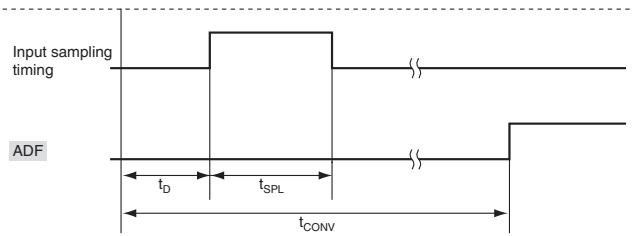
chanhvinh.com

Item	Page	Revision (See Manual for Details)
24.3.2 A/D Control/Status Register (ADCSR)	1261	Table amended

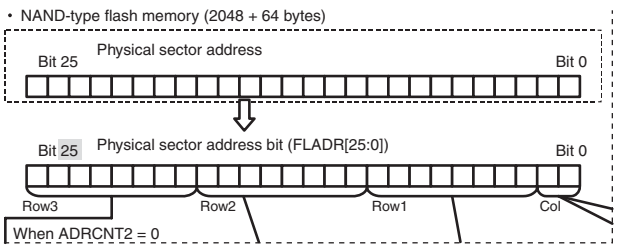
Bit	Bit Name	Initial Value	R/W	Description
2 to 0	CH[2:0]	000	R/W	Channel Select These bits and the MDS bits in ADCSR select the analog input channels.

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*3	100: AN4*3	100: AN0 to AN4*3
101: AN5*3	101: AN4, AN5*3	101: AN0 to AN5*3
110: AN6*3	110: AN4 to AN6*3	110: AN0 to AN6*3
111: AN7*3	111: AN4 to AN7*3	111: AN0 to AN7*3

24.4.5 Input Sampling and A/D Conversion Time Figure 24.5 A/D Conversion Timing	1271	Figure amended

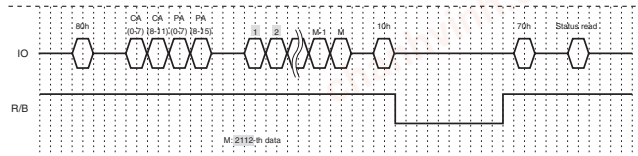
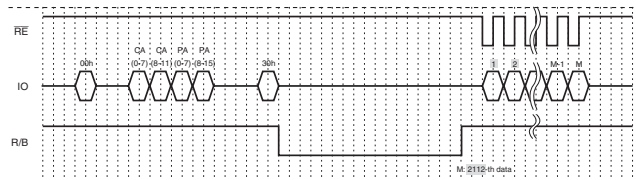
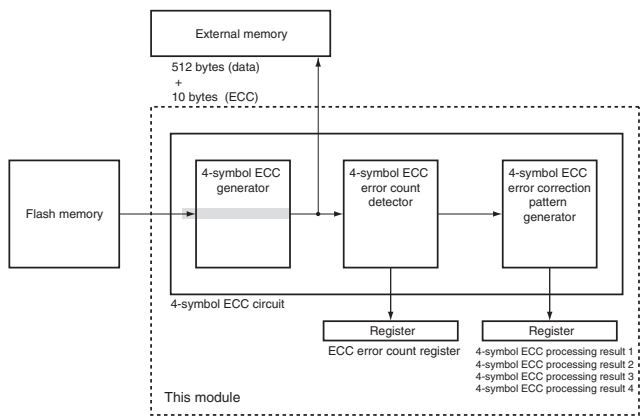


25.4.5 Sector Access Mode Figure 25.11 Relationship between Sector Number and Address Expansion of NAND-Type Flash Memory	1323	Figure amended



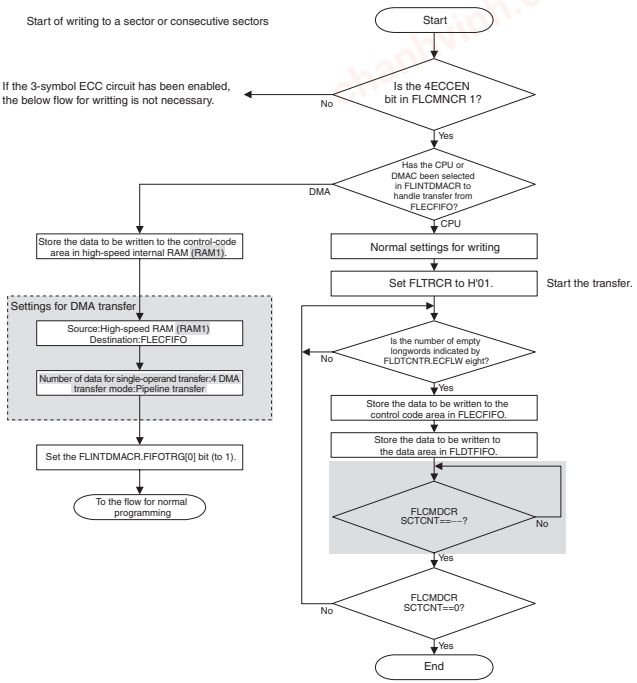
(3) Flash Memory Access in Sector Access Mode	1325	Description amended During the execution of sequential sector access spanning multiple pages, data are written to the flash memory with the timing shown in the figure for every page (2048 + 64 bytes).

chanhvinh.com

Item	Page	Revision (See Manual for Details)
25.4.5 Sector Access Mode Figure 25.13 Programming Operation Timing for NAND-Type Flash Memory	1325	Figure amended 
	1326	Description amended During the execution of sequential sector access spanning multiple pages, data are read from the flash memory with the timing shown in the figure for every page (2048 + 64 bytes).
Figure 25.14 Read Timing from NAND-Type Flash Memory (Sector Access Mode)		Figure amended 
25.4.6 ECC Error Correction Figure 25.15 4-Symbol ECC Circuit	1327	Figure amended 
25.4.7 Status Read	1331	Description amended If a program error or erase error is detected when the status register value is stored in the bits STAT[7:0] in FLBSYCNT, the STERB bit in FLINTDMACR is set to 1 and generates an interrupt to the CPU if the STERINTE bit in FLINTDMACR is enabled. If a status error occurs during continuous sector access, the TREND bit in FLTRCR is set to 1 and the procedure stops.

chanhvinh.com

Item	Page	Revision (See Manual for Details)
25.7.2 Writing to the Control-Code Area when 4-Symbol ECC Circuit is in Use Figure 25.19 Writing Procedure to the Control-Code Area when 4-Symbol ECC is Used	1335	Figure amended



25.7.3 Usage Notes for the SNAND Bit	1336	Newly added
26.3.1 System Configuration Control Register (SYSCFG)	1346	Table amended

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 function 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 function is selected allows this module to pull up the D+ line to 3.3 V, 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 function is selected, and should be set to 0 if the host controller function is selected. Note: Set this bit to 0 when the USB is disconnected. Include the following processing when this bit is changed from 1 to 0. (1) Set the DPRPU bit to 0. (2) Wait for at least 1 µs. (3) Set the DCFM bit to 1. (4) Wait for at least 200 ns. (5) Set the DCFM bit to 0.

chanhvinh.com

Item	Page	Revision (See Manual for Details)
26.3.4 Device State Control Register (DVSTCTR)	1354	Table amended

Bit	Bit Name	Initial Value	R/W	Description
4	UACT	0	R/W	USB Bus Enable . . . 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 function is selected.

26.3.5 Test Mode Register (TESTMODE)	1356	Figure amended

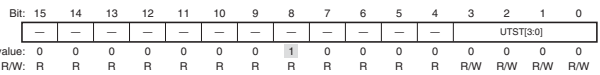


Table amended

Bit	Bit Name	Initial Value	R/W	Description
15 to 9	—	ALL 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
7 to 4	—	ALL 0	R	Reserved These bits are always read as 0. The write value should always be 0.

26.3.6 DMA-FIFO Bus Configuration Registers (D0FBCFG, D1FBCFG)	1359	Table amended

Bit	Bit Name	Initial Value	R/W	Description
4	TENDE	0	R/W	DMA Transfer End Sampling Enable Controls acceptance of DMA transfer end signal output from the direct memory access controller on completion of a DMA transfer. For details, see section 26.4.4 (3), DMA Transfers (D0FIFO/D1FIFO Port). 0: DMA transfer end signal is not sampled. 1: DMA transfer end signal is sampled. For a DMA transfer size of 16 bytes, clear the TENDE bit to 0.

chanhvinh.com

Item	Page	Revision (See Manual for Details)
26.3.8 FIFO Port Select Registers (CFIFOSEL, D0FIFOSEL, D1FIFOSEL) (2) D0FIFOSEL, D1FIFOSEL	1367	Table amended

Bit	Bit Name	Initial Value	R/W	Description
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 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 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.

26.3.9 FIFO Port Control Registers (CFIFOCTR, D0FIFOCTR, D1FIFOCTR)	1371	Table amended		

Bit	Bit Name	Initial Value	R/W	Description
15	BVAL	0	R/W*2	Buffer Memory Valid Flag . . . 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. When the selected pipe is in the transmitting direction, if 1 is written to BVAL and BCLR bits simultaneously, this module clears the data that has been written before it, enabling transmission of a zero-length packet. Writing 1 to this bit should be done while FRDY indicates 1 (set by this module). When the selected pipe is in the receiving direction, do not set this bit to 1.

	1372	Table amended		

Bit	Bit Name	Initial Value	R/W	Description
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. To clear 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 (set by this module).

chanhvinh.com

Item	Page	Revision (See Manual for Details)
26.3.15 SOF Output Configuration Register (SOFCFG)	1382	Title amended
	1383	Table amended

Bit	Bit Name	Initial Value	R/W	Description
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. Set the BRDYM bit during the initial settings of the USB 2.0 host/function module (before performing data communication). Do not change the setting of the BRDYM bit after data communication starts. 0: Writing 0 clears the status. 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.

Item	Page	Revision (See Manual for Details)
26.3.21 Frame Number Register (FRMNUM)	1401	Table amended

Bit	Bit Name	Initial Value	R/W	Description
15	OVRN	0	R/W*	Overrun/Underrun Detection Status Indicates whether an overrun/underrun error has been detected in the pipe during isochronous transfer. 0: No error 1: An error occurred This bit can be cleared to 0 by writing 0 to the bit. (1) When the host controller function is selected This module sets this bit to 1 on any of the following conditions. .

Item	Page	Revision (See Manual for Details)
26.3.30 DCP Control Register (DCPCTR)	1416	Table amended

Bit	Bit Name	Initial Value	R/W	Description
5	PBUSY	0	R	Pipe Busy This bit indicates whether or not a transaction using the pertinent pipe is currently in progress. 0: The pertinent pipe is not in use by a transaction. 1: The pertinent pipe is in use by a transaction. The USB 2.0 host/function module changes the setting of the PBUSY bit from 0 to 1 when a USB transaction using the DCP starts. It clears the PBUSY bit from 1 to 0 when one transaction completes successfully. Reading this bit after setting PID to NAK allows checking that modification of the pipe settings is possible. For details, refer to section 26.4.3 (1), Pipe Control Register Switching Procedures.

chanhvinh.com

Item	Page	Revision (See Manual for Details)
26.3.30 DCP Control Register (DCPCTR)	1417	Table amended

Bit	Bit Name	Initial Value	R/W	Description
4	PINGE	0	R/W	PING Token Issue Enable . . . These bits should be modified while CSSTS is 0 and PID is NAK. Before changing the value of this bit after changing the PID bits for the DCP from BUF to NAK, confirm that CSSTS and PBUSY are both cleared to 0. However, if the PID bits have been modified to NAK by this module, checking PBUSY is not necessary. When the function controller function is selected, be sure to write 0 to this bit.

26.3.31 Pipe Window Select Register (PIPESEL)	1425	Table amended

Bit	Bit Name	Initial Value	R/W	Description
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 are selected. When this bit has been 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 these bits while CSSTS is 0 and PID is NAK and before the pipe is selected by the CURPIPE bits. To modify these bits 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 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, checking PBUSY is not necessary.

chanhvinh.com

Item	Page	Revision (See Manual for Details)
26.3.31 Pipe Window Select Register (PIPESEL)	1428	Table amended

Bit	Bit Name	Initial Value	R/W	Description
4	DIR	0	R/W	Transfer Direction Specifies the transfer direction for the selected pipe. 0: Receiving direction 1: Sending direction When this bit has been set to 0, this module uses the selected pipe in the receiving direction, and when this bit has been set to 1, this module uses the selected pipe in the transmitting direction. Modify these bits when the value of CSSTS is 0, the PID bits are set to NAK, and no pipe is specified by the CURPIPE bits. To modify these bits 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 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, checking PBUSY is not necessary.

26.3.33 Pipe Buffer Setting Register (PIPEBUF)	1430	Table amended		

Bit	Bit Name	Initial Value	R/W	Description
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 has been 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: Any value from H'00 to H'1F is valid. PIPE6 to PIPE9: H'00 should be set. When used with CNTMD = 1, set an integer multiple of the maximum packet size to the BUFSIZE bits. Modify these bits when the value of CSSTS is 0, the PID bits are set to NAK, and no pipe is specified 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, checking PBUSY is not necessary.

chanhvinh.com

Item	Page	Revision (See Manual for Details)
26.3.33 Pipe Buffer Setting Register (PIPEBUF)	1431	Table amended

Bit	Bit Name	Initial Value	R/W	Description
6 to 0	BUFNMB[6:0]	H'00	R/W	Buffer Number These bits specify the start block number of the FIFO buffers to be assigned to the selected pipe. The following blocks of FIFO buffers are assigned to the selected pipe. Block number BUFNMB to block number BUFNMB + (BUFSIZE + 1) × (DBLB + 1) - 1 Specify a value from H'04 to H'7F. When the selected pipe is one of PIPE1 to PIPE5, any value can be set to these bits according to the user system. BUFNUMB = H'00 to H'03 are 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. BUFNUMB = 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. BUFNUMB = 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 invalid and H'07 is automatically assigned by this module. Modify these bits when the value of CSSTS is 0. the PID bits are set to NAK, and no pipe is specified 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, checking PBUSY is not necessary.

chanhvinh.com

Item	Page	Revision (See Manual for Details)
26.3.34 Pipe Maximum Packet Size Register (PIPEMAXP)	1433	Table amended

Bit	Bit Name	Initial Value	R/W	Description
10 to 0	MXPS[10:0]	*	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 to 0 are not provided.) PIPE6 to PIPE9: 1 byte (H'001) to 64 bytes (H'040) These bits should be set to the appropriate 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 when the value of CSSTS is 0, the PID bits are set to NAK, and no pipe is specified 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, checking PBUSY is not necessary. While MXPS is 0, do not write to the FIFO buffer or set PID to BUF.

26.3.35 Pipe Timing Control Register (PIPEPERI)	1435	Table amended

Bit	Bit Name	Initial Value	R/W	Description
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 when the value of CSSTS is 0, the PID bits are set to NAK, and no pipe is specified 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, checking PBUSY is not necessary. 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; set these bits to 000 for these pipes.

26.3.36 PIPEn Control Registers (PIPEnCTR) (n = 1 to 9) (1) PIPEnCTR (n = 1 to 5)	1436	Figure amended

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	0	0	0	0
R/W:	R	R	R/W*	R	R	R/W	R/W	R/W*	R/W*	R	R	R	R	R	R/W	R/W

chanhvinh.com

Item	Page	Revision (See Manual for Details)
26.3.36 PIPEn Control Registers (PIPEnCTR) (n = 1 to 9) (1) PIPEnCTR (n = 1 to 5)	1438	Table amended

Bit	Bit Name	Initial Value	R/W	Description
13	CSCLR	0	R/W*	C-SPLIT Status Clear Bit When the host controller function 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 function is selected, be sure to write 0 to this bit.
12	CSSTS	0	R	CSSTS Status Bit Indicates the C-SPLIT status of the split transaction when the host controller function 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. If the USB device is detached when processing of a C-SPLIT transaction is in progress, the CSSTS bit may remain set to 1. In such a case (DTCH = 1 detected), use the CSCLR bit to clear the CSSTS bit to 0. Indicates the valid value only when the host controller function is selected.

	1440	Table amended		

Bit	Bit Name	Initial Value	R/W	Description
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 all the information assigned to the pertinent pipe from the FIFO buffer, write 1 and 0 in succession to the ACLRM bit. Table 26.11 (1) lists the information related to the USB 2.0 host/function module that is cleared by setting the ACLRM bit to 1 and then to 0 in succession. Table 26.11 (2) lists the cases in which it is necessary to do this. Modify this bit while CSSTS is 0, PID is NAK, and the pertinent pipe is not specified in 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, checking PBUSY is not necessary.

chanhvinh.com

Item	Page	Revision (See Manual for Details)
26.3.36 PIPEn Control Registers (PIPEnCTR) (n = 1 to 9) (1) PIPEnCTR (n = 1 to 5)	1441	Table amended

Bit	Bit Name	Initial Value	R/W	Description
8	SQCLR	0	R/W*	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. . . .

1442 Table amended

Bit	Bit Name	Initial Value	R/W	Description
7	SQSET	0	R/W*	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. . . .

1443 Table amended

Bit	Bit Name	Initial Value	R/W	Description
5	PBUSY	0	R	Pipe Busy This bit indicates whether the relevant pipe is used or not for the transaction. 0: The relevant pipe is not used for the transaction. 1: The relevant pipe is used for the transaction. The USB 2.0 host/function module switches the PBUSY bit from 0 to 1 when a USB transaction using the pertinent pipe starts. It clears the PBUSY bit from 1 to 0 when one transaction completes successfully. Reading this bit after PID has been set to NAK allows checking that modification of the pipe settings is possible. For details, refer to section 26.4.3 (1), Pipe Control Register Switching Procedures.

1444 Table amended

Bit	Bit Name	Initial Value	R/W	Description
1, 0	PID[1:0]	00	R/W	Response PID . . . 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, checking PBUSY is not necessary. . . .

1445 Note amended

Note: * Only 0 can be read and 1 can be written.

chanhvinh.com

Item	Page	Revision (See Manual for Details)
26.3.36 PIPEn Control Registers (PIPEnCTR) (n = 1 to 9) (1) PIPEnCTR (n = 1 to 5) Table 26.11(1) Information Cleared by this Module by Setting ACLRM = 1	1446	Table amended

No.	Information Cleared by ACLRM Bit Manipulation
1	All the information 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

Table 26.11(2) Cases That Require Setting ACLRM to 1		Newly added
Table 26.13 Operation of This Module depending on PID Setting (when Function Controller Function is Selected)	1448	Table amended

PID	Transfer Type	Transfer Direction (DIR Bit)	Operation of This Module
00 (NAK)	Bulk or interrupt	Operation does not depend on the setting.	Returns NAK in response to the token from the USB host. For the operation when ATREPM is 1, refer to the description of the ATREPM bit.
	Isochronous	Receiving direction (DIR = 0)	Returns nothing in response to a token from the USB host.
		Transmitting direction (DIR = 1)	Returns a zero-length packet in response to a token from the USB host.

26.3.36 PIPEn Control Registers (PIPEnCTR) (n = 1 to 9) (2) PIPEnCTR (n = 6 to 9)	1449	Table amended

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BSTS	—	CSCLR	CSSTS	—	—	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[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/W*	R	R	R	R	R	R/W	R/W

	1450	Table amended

Bit	Bit Name	Initial Value	R/W	Description
13	CSCLR	0	R/W*	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. . . .

chanhvinh.com

Item	Page	Revision (See Manual for Details)
26.3.36 PIPEn Control Registers (PIPEnCTR) (n = 1 to 9) (2) PIPEnCTR (n = 6 to 9)	1451	Table amended

Bit	Bit Name	Initial Value	R/W	Description
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 all the information assigned to the pertinent pipe from the FIFO buffer, write 1 and 0 in succession to the ACLRM bit. Table 26.14 (1) lists the information related to the USB 2.0 host/function module that is cleared by setting the ACLRM bit to 1 and then to 0 in succession. Table 26.14 (2) lists the cases in which it is necessary to do this. Modify this bit while CSSTS is 0 and PID is NAK and before the pipe is 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, checking PBUSY is not necessary.

	1452	Table amended

Bit	Bit Name	Initial Value	R/W	Description
8	SQCLR	0	R/W*	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. . . .

	1453	Table amended

Bit	Bit Name	Initial Value	R/W	Description
7	SQSET	0	R/W*	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. . . .

chanhvinh.com

Item	Page	Revision (See Manual for Details)
26.3.36 PIPEn Control Registers (PIPEnCTR) (n = 1 to 9) (2) PIPEnCTR (n = 6 to 9)	1454	Table amended

Bit	Bit Name	Initial Value	R/W	Description
5	PBUSY	0	R	Pipe Busy This bit indicates whether the relevant pipe is used or not for the transaction. 0: The relevant pipe is not used for the transaction. 1: The relevant pipe is used for the transaction. This module modifies this bit from 0 to 1 upon start of the USB transaction for the pertinent pipe, and clears it from 1 to 0 when one transaction completes successfully. Reading this bit after PID has been set to NAK allows checking that modification of the pipe settings is possible. For details, refer to section 26.4.3 (1), Pipe Control Register Switching Procedures.

	1455	Table amended

Bit	Bit Name	Initial Value	R/W	Description
1, 0	PID[1:0]	00	R/W	Response PID . . . 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, checking PBUSY is not necessary. After S-SPLIT is issued (CSSTS = 1) for split transaction in the pertinent pipe, the transaction continues until C-SPLIT is completed even when the PID bits are set to NAK. . . .

	1456	Note amended Note: * Only 0 can be read and 1 can be written.
Table 26.14 (1) Information Cleared by this Module by Setting ACLRM = 1	1457	Table amended

No.	Information Cleared by ACLRM Bit Manipulation
1	All the information in the FIFO buffer assigned to the pertinent pipe
2	When the host controller function is selected, the interval count value when the pertinent pipe is for isochronous transfer

Table 26.14 (2) Cases That Require Setting ACLRM to 1		Newly added

chanhvinh.com

Item	Page	Revision (See Manual for Details)
26.3.38 PIPEn Transaction Counter Registers (PIPEnTRN) (n = 1 to 5)	1460	Table amended

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	TRNCNT[15:0]	All 0	R/W	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 set value 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 set value = 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. The TRCLR bit has been set to 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, checking PBUSY is not necessary. To modify the value of these bits, set TRNCNT to 1 before setting TRENB to 1.

26.4.1 System Control and Oscillation Control (5) Register Access Wait Control	1468	Description amended Wait-related restriction: The cycle time for successive accesses to the USB 2.0 host/function module must be a duration of at least four USB clock (48 MHz) cycles (83.33 ns). To fulfill the above restriction, a register access wait control is necessary using the BWAIT[3:0] bits in BUSWAIT. . . Calculation: (2 cycles (access cycles for the registers of this module) + 1 cycle (interval between successive accesses) + BWAIT) × 1/72 MHz ≥ 83.33 ns . . Calculation: (2 cycles (access cycles for the registers of this module) + 2 cycles (access cycles for the on-chip high-speed RAM) + BWAIT) × 1/72 MHz ≥ 83.33 ns

chanhvinh.com

Item	Page	Revision (See Manual for Details)
26.4.2 Interrupt Functions Table 26.16 Interrupt Generation Conditions	1471	Table amended

Bit	Interrupt Name	Cause of Interrupt	Function That Generates the Interrupt	Related Status
NRDY	Buffer not ready interrupt	When the host controller function 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 function is selected: • When a token is received while the PID bits are set to BUF and transmission is not enabled for the buffer memory • 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	NRDYSTS.PIPENRDY
BCHG	Bus change interrupt	When a change of USB bus state is detected	Host	

(1) BRDY Interrupt (b) When the BRDYM bit is 0 and the BFRE bit is 1	1476	Description amended . . . 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 while the FIFO buffer is empty, the USB 2.0 host/function module determines that all the data for a single transfer has been read at the point at which the FRDY bit is set to 1 and the DTLN bit cleared to 0 in the FIFO port control register. In this case, to start the next transfer, write 1 to the BCLR bit in the corresponding FIFOCTR register.
(c) When the BRDYM bit is 1 and the BFRE bit is 0 Figure 26.3 Timing at which a BRDY Interrupt is Generated	1478	Figure replaced

chanhvinh.com

Item	Page	Revision (See Manual for Details)
26.4.2 Interrupt Functions (2) NRDY Interrupt Figure 26.4 Timing at which NRDY Interrupt is Generated when Function Controller Function is Selected	1484	Figure replaced
(3) BEMP Interrupt Figure 26.5 Timing at which BEMP Interrupt is Generated when Function Controller Function is Selected	1486	Figure replaced
(5) Control Transfer Stage Transition Interrupt (Function Controller Function)	1488, 1489	Description amended When a CTRT interrupt occurs in response to a sequence error, 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 held, the CTRT interrupt that ends the setup stage will not be generated even if a new USB request is received. (The USB 2.0 host/function module retains the setting of the CTSQ bits when the setup stage ends, and it generates a CTRT interrupt after the interrupt status is cleared.)
26.4.3 Pipe Control (1) Pipe Control Register Switching Procedures	1495	Description amended • Bits in DCPMAXP • The SQCLR, SQSET, and PINGE bits in DCPCTR • Bits in PIPECFG, PIPEBUF, PIPEMAXP and PIPEPERI • The ATREPM, ACLRM, SQCLR and SQSET bits in PIPEnCTR • Bits in PIPEnTRE and PIPEnTRN • Bits in DEVADDn Note: In addition to the above, observe the setting procedures described in the register descriptions regarding the settings of the CSCLR bit and DEVADDn register.
	1496	Description amended • Bits in DCPCFG and DCPMAXP • Bits in PIPECFG, PIPEBUF, PIPEMAXP and PIPEPERI • ACLRM bit in PIPEnCTR

chanhvinh.com

<table>
<tr><th>Item</th><th>Page</th><th>Revision (See Manual for Details)</th></tr>
<tr><td>26.4.3 Pipe Control
(6) Response PID
(a) When the host controller function has been selected and the response PID is set by hardware</td><td>1498</td><td>Description amended
• NAK setting: In the following cases, PID = NAK is set and issuing of tokens is automatically stopped:
— When, during a transfer other than isochronous transfer, three receive errors such as no response, bit stuffing error, or CRC error are returned in succession after token transmission.</td></tr>
<tr><td>(7) Data PID Sequence Bit</td><td>1500</td><td>Description amended
When the function controller function has been selected and control transfer is used, this module automatically sets the sequence bit when a stage transition is made. The bit is set to DATA1 when the setup stage ends. In the status stage, DATA1 is returned without referencing the sequence bit. Therefore, settings are not required. However, when the host controller function has been selected and control transfer is used, the sequence bit should be set at the stage transition.</td></tr>
<tr><td>26.4.4 FIFO Buffer Memory
(1) FIFO Buffer Memory Allocation
(f) Buffer Memory Operation (Continuous Transfer Setting)
Table 26.22 Relationship between Transfer Mode Settings by CNTMD Bit and Timings at which Reading Data or Transmitting Data from FIFO Buffer is Enabled</td><td>1508</td><td>Table amended
<table><tr><th>Continuous or Non-Continuous Transfer Mode</th><th>Method of Determining if Reading or Transmitting Data is Enabled</th></tr><tr><td>Continuous transfer (CNTMD = 1)</td><td>In the transmitting direction (DIR = 1), transmitting data from the FIFO buffer is enabled when:
• 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.
or
• The number of data bytes less than the size of a single FIFO buffer plane (including 0-byte data) assigned to the selected pipe is written to the FIFO buffer and then 1 is written to BVAL.
• A number of data bytes less than the size of a single FIFO buffer plane assigned to the selected pipe is written to the FIFO buffer and the DMA transfer end signal is received when the last byte is written.</td></tr></table></td></tr>
<tr><td>(2) FIFO Port Functions
(a) FIFO Port Selection</td><td>1510</td><td>Table amended
<table><tr><th>Pipe</th><th>Access Method</th><th>Port that can be Used</th></tr><tr><td>DCP</td><td>CPU access</td><td>CFIFO port register</td></tr><tr><td>PIPE1 to PIPE9</td><td>CPU access</td><td>CFIFO port register</td></tr><tr><td></td><td>DMA access</td><td>D0FIFO/D1FIFO port register</td></tr></table></td></tr>
<tr><td>(3) DMA Transfers (D0FIFO/D1FIFO Port)</td><td>1515</td><td>Note added
Note that this function cannot be used when the DMA transfer size is set to 16 bytes.</td></tr>
</table>

chanhvinh.com

Item	Page	Revision (See Manual for Details)
26.4.5 Control Transfers (DCP) (1) Control Transfers when the Host Controller Function is Selected (b) Data Stage	1516	Description amended The access direction of the DCP buffer memory should be specified using the ISEL bit in CFIFOSEL. Also specify the transfer direction using the DIR bit in the DCPCFG register. For the first data packet of the data stage, the data PID must be transferred as DATA1. Transaction is done by setting the data PID = DATA1 and the PID bit = BUF using the SQSET bit in DCPCTR. Completion of data transfer is detected using the BRDY or BEMP interrupts.
(2) Control Transfers when the Function Controller Function is Selected (b) Data Stage	1518	Description amended 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. A transaction is executed by setting the PID bits in the DCPCTR register to BUF. The BRDY interrupt or the BEMP interrupt can be used to detect the end of data transfer. Use the BRDY interrupt to detect the end of control write transfers and the BEMP interrupt to detect the end of control read transfers.
(d) Control Transfer Auto Response Function	1519	Description amended (i) bmRequestType ≠ H'00 (ii) wIndex ≠ H'00 (ii) wLength ≠ H'00 (iv) wValue > H'7F (v) DVSQ = 011 (Configured)
26.4.6 Bulk Transfers (PIPE1 to PIPE5) (1) PING Packet Control when the Host Controller Function is Selected	1520	Description amended 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. The module returns to PING packet sending mode when a NAK or NYET is received during an OUT transaction. . . . This module is returned to PING packet sending mode by a power-on reset, receiving a NYET/NAK handshake, clearing the sequence toggle bits (SQCLR), and setting the buffer clear bit (ACLRM) in PIPEnCTR.

chanhvinh.com

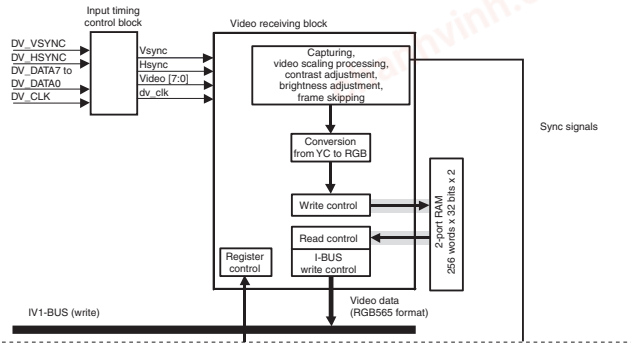
Item	Page	Revision (See Manual for Details)
26.4.6 Bulk Transfers (PIPE1 to PIPE5) (2) NYET Handshake Control when the Function Controller Function is Selected	1521	Description amended Table 26.26 lists the responses to received tokens during bulk transmission and control transmission. The USB 2.0 host/function module returns a NYET when an OUT token is received while there is only one packet of empty space in the buffer memory, during both bulk transmission and control transmission. However, an ACK is returned instead of a NYET even under the above conditions when a short packet is received.
Table 26.26 List of Responses to Received Tokens		Table Title amended
26.4.7 Interrupt Transfers (PIPE6 to PIPE9) (3) Interval Counter (a) Interval Counter Initialization when the Function Controller Function is Selected	1527	Description amended • Buffer memory initialization using the ACLRM bit The IITV bits are not initialized but the count value is. • USB bus reset
(b) Interval Counting and Transfer Control when the Host Controller Function is Selected Figure 26.16 Token Issuance when IITV = 0	1528	Figure replaced
Figure 26.17 Token Issuance when IITV = 1	1529	Figure replaced
(c) Interval Counting and Transfer Control when the Function Controller Function is Selected Figure 26.18 Relationship between (μ) Frames and Expected Token Reception when IITV = 0	1530	Figure replaced
Figure 26.19 Relationship between (μ) Frames and Expected Token Reception when IITV = 1	1531	Figure replaced

chanhvinh.com

Item	Page	Revision (See Manual for Details)
26.4.7 Interrupt Transfers (PIPE6 to PIPE9) (3) Interval Counter (c) Interval Counting and Transfer Control when the Function Controller Function is Selected	1531	Description amended The interval counting starts at the different timing depending on the IITV bit setting (similar to the timing during OUT transfers). The interval count clearing condition is any of the following in function controller mode. — When a power-on reset is applied. — 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 Function is Selected	1532	Description amended Figure 26.20 shows an example of transmission using the isochronous transfer transmission data setup function with this module, when IITV = 0 (every frame) has been set.
Figure 26.20 Example of Data Setup Function Operation	1533	Figure replaced
(5) Isochronous Transfer Transmission Buffer Flush when the Function Controller Function is Selected Figure 26.21 Example of Buffer Flush Function Operation	1534	Figure replaced
Figure 26.22 Example of an Interval Error Being Generated when IITV = 1	1535	Figure replaced
26.4.9 SOF Interpolation Function	1536	Description amended • Refreshing of the frame number and the micro-frame number • SOFR interrupt and μSOF lock • Isochronous transfer interval count

chanhvinh.com

Item	Page	Revision (See Manual for Details)
27.4 Configuration Figure 27.1 Block Diagram	1543	Figure amended



27.6.3 Panel Control Signal Output Function (4) Sync Signal Output Timing	1557	Newly added
27.7.1 Video Operating Mode Register (VIDEO_MODE)	1564	Table amended

Bit	Bit Name	Initial Value	R/W	Description
0	VIDEO_MAIN_EXE	0	R	Enables the video receiving block operation. Setting to 1 starts video storing in the large-capacity on-chip RAM or SDRAM. 0: Disabled*2 1: Enabled

Notes added

Notes: 1. Operation of the video receiving and supplying blocks proceeds after bit VIDEO_DISP_EXE or VIDEO_MAIN_EXE, respectively, is set to 1, and detection of VSYCN. Operation stops when the protocol for the internal bus is ended after the corresponding bit has been cleared to 0.

2. To disable the video receiving block operation, please follow the procedure shown in section 27.10.1, The Procedure of Disabling the Video Receiving Block Operation.

chanhvinh.com

Item	Page	Revision (See Manual for Details)
27.7.25 α Control Registers (GROPEDPA1 and GROPEDPA2)	1599	Description amended If AMOD[1:0] = [0 1], the ACOEF value is added to the DEFA value according to the frame rate (ARATE) and the result is applied to the α control area as the α value. When the α value becomes 255 or larger, processing stops (fade-in). If AMOD[1:0] = [1 0], the ACOEF value is subtracted from the DEFA value according to the frame rate (ARATE) and the result is applied to the α control area as the α value. When the α value becomes 0 or smaller, processing stops (fade-out).
27.10 Usage Note	1628	Newly added
31.1 Features	1665	Description amended • Number of access cycles On-chip high-speed RAM: the number of cycles for access to read or write from buses F and I is one cycle of Iϕ. Number of cycles for access from the ID bus depend on the ratio of the CPU clock (Iϕ) to the bus clock (Bϕ). Table 31.6 indicates number of cycles for access from the ID bus.
31.2.3 Data Retention	1668	Newly added
32.1 Features Table 32.4 Multiplexed Pins (Port C)	1672	Table amended

	Setting Mode Bit (PCnMD[1:0])			
	00	01	10	11
Setting Register	Function 1	Function 2	Function 3	Function 4
PCCR1	PC4	$\overline{WE1}$/DQMU/$\overline{WE}$	—	—
PCCR0	PC3	$\overline{WE0}$/DQML	—	—

32.2.8 Port C Control Registers 0 to 2 (PCCR0 to PCCR2)	1700	Table amended

(2) Port C Control Register 1 (PCCR1)

Bit	Bit Name	Initial Value	R/W	Description
0	PC4MD0	0	R/W	PC4 Mode Select the function of the PC4 0: PC4 1: $\overline{WE1}$/DQMU/$\overline{WE}$

(3) Port C Control Register 0 (PCCR0)

Table amended

Bit	Bit Name	Initial Value	R/W	Description
12	PC3MD0	0	R/W	PC3 Mode Select the function of the PC3 0: PC3 1: $\overline{WE0}$/DQML

chanhvinh.com

Item	Page	Revision (See Manual for Details)
32.2.12 Port D Control Register 0 to 3 (PDCR0 to PDCR3) (1) Port D Control Register 3 (PDCR3)	1705	Table amended

Bit	Bit Name	Initial Value	R/W	Description
13, 12	PD15MD[1:0]	00/01	R/W	PD15 Mode Select the function of the PD15. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PD15 (initial value) 01: D15/NAF7 (initial value) / 01: D15/NAF7 10: Setting prohibited / 10: PWM2H 11: Setting prohibited / 11: Setting prohibited
9, 8	PD14MD[1:0]	00/01	R/W	PD14 Mode Select the function of the PD14. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PD14 (initial value) 01: D14/NAF6 (initial value) / 01: D14/NAF6 10: Setting prohibited / 10: PWM2G 11: Setting prohibited / 11: Setting prohibited

	1706	Table amended		

Bit	Bit Name	Initial Value	R/W	Description
5, 4	PD13MD[1:0]	00/01	R/W	PD13 Mode Select the function of the PD13. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PD13 (initial value) 01: D13/NAF5 (initial value) / 01: D13/NAF5 10: Setting prohibited / 10: PWM2F 11: Setting prohibited / 11: Setting prohibited
1, 0	PD12MD[1:0]	00/01	R/W	PD12 Mode Select the function of the PD12. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PD12 (initial value) 01: D12/NAF4 (initial value) / 01: D12/NAF4 10: Setting prohibited / 10: PWM2E 11: Setting prohibited / 11: Setting prohibited

32.2.12 Port D Control Register 0 to 3 (PDCR0 to PDCR3) (2) Port D Control Register 2 (PDCR2)	1707	Table amended		

Bit	Bit Name	Initial Value	R/W	Description
13, 12	PD11MD[1:0]	00/01	R/W	PD11 Mode Select the function of the PD11. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PD11 (initial value) 01: D11/NAF3 (initial value) / 01: D11/NAF3 10: Setting prohibited / 10: PWM2D 11: Setting prohibited / 11: Setting prohibited
9, 8	PD10MD[1:0]	00/01	R/W	PD10 Mode Select the function of the PD10. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PD10 (initial value) 01: D10/NAF2 (initial value) / 01: D10/NAF2 10: Setting prohibited / 10: PWM2C 11: Setting prohibited / 11: Setting prohibited
5, 4	PD9MD[1:0]	00/01	R/W	PD9 Mode Select the function of the PD9. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PD9 (initial value) 01: D9/NAF1 (initial value) / 01: D10/NAF1 10: Setting prohibited / 10: PWM2B 11: Setting prohibited / 11: Setting prohibited

chanhvinh.com

Item	Page	Revision (See Manual for Details)
32.2.12 Port D Control Register 0 to 3 (PDCR0 to PDCR3) (2) Port D Control Register 2 (PDCR2)	1708	Table amended

Bit	Bit Name	Initial Value	R/W	Description
1, 0	PD8MD[1:0]	00/01	R/W	PD8 Mode Select the function of the PD8. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PD8 (initial value) 01: D8/NAF0 (initial value) / 01: D8/NAF0 10: Setting prohibited / 10: PWM2A 11: Setting prohibited / 11: Setting prohibited

Item	Page	Revision (See Manual for Details)
(3) Port D Control Register 1 (PDCR1)	1708	Table amended

Bit	Bit Name	Initial Value	R/W	Description
13, 12	PD7MD[1:0]	00/01	R/W	PD7 Mode Select the function of the PD7. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PD7 (initial value) 01: D7/FWE (initial value) / 01: D7/FWE 10: Setting prohibited / 10: PWM1H 11: Setting prohibited / 11: Setting prohibited

	1709	Table amended

Bit	Bit Name	Initial Value	R/W	Description
9, 8	PD6MD[1:0]	00/01	R/W	PD6 Mode Select the function of the PD6. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PD6 (initial value) 01: D6/FALE (initial value) / 01: D6/FALE 10: Setting prohibited / 10: PWM1G 11: Setting prohibited / 11: Setting prohibited
5, 4	PD5MD[1:0]	00/01	R/W	PD5 Mode Select the function of the PD5. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PD5 (initial value) 01: D5/FCLE (initial value) / 01: D5/FCLE 10: Setting prohibited / 10: PWM1F 11: Setting prohibited / 11: Setting prohibited
1, 0	PD4MD[1:0]	00/01	R/W	PD4 Mode Select the function of the PD4. Boot mode 0 / Boot mode 1 to 3 00: Setting prohibited / 00: PD4 (initial value) 01: D4/FRE (initial value) / 01: D4/FRE 10: Setting prohibited / 10: PWM1E 11: Setting prohibited / 11: Setting prohibited

Item	Page	Revision (See Manual for Details)
32.2.24 Port G Control Register 0 to 7 (PGCR0 to PGCR7) (1) Port G Control Register 7 (PGCR7)	1729	Table amended

Bit	Bit Name	Initial Value	R/W	Description
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should be H'5A for bits 15 to 8 and 0 for bits 7 to 3.

32.3 Usage Notes	1760	Newly added

chanhvinh.com

Item	Page	Revision (See Manual for Details)
33.2 Register Descriptions Table 33.2 Register Configuration	1764	Table amended

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
On-chip data-retention RAM area setting register	RRAMKP	R/W	H'00	H'FFFE6800	8
Deep standby control register	DSCTR	R/W	H'00	H'FFFE6802	8
Deep standby cancel source select register	DSSSR	R/W	H'0000	H'FFFE6804	16
Deep standby cancel edge select register	DSESR	R/W	H'0000	H'FFFE6806	16
Deep standby cancel source flag register	DSFR	R/W	H'0000	H'FFFE6808	16
XTAL crystal oscillator gain control register	XTALCTR	R/W	H'00	H'FFFE6810	8

33.2.2 Standby Control Register 2 (STBCR2)	1766	Description amended STBCR2 is an 8-bit readable/writable register that controls the operation of modules .
33.2.3 Standby Control Register 3 (STBCR3)	1767	Description amended STBCR3 is an 8-bit readable/writable register that controls the operation of modules .
	1769	Table amended

Bit	Bit Name	Initial Value	R/W	Description
1	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.

33.2.4 Standby Control Register 4 (STBCR4)	1769	Description amended STBCR4 is an 8-bit readable/writable register that controls the operation of modules .
33.2.5 Standby Control Register 5 (STBCR5)	1772	Description amended STBCR5 is an 8-bit readable/writable register that controls the operation of modules .
33.2.6 Standby Control Register 6 (STBCR6)	1774	Description amended STBCR6 is an 8-bit readable/writable register that controls the operation of each module .
33.2.7 Standby Control Register 7 (STBCR7)	1776	Description amended STBCR7 is an 8-bit readable/writable register that controls the operation of each module .
33.2.8 Standby Control Register 8 (STBCR8)	1778	Description amended STBCR8 is an 8-bit readable/writable register that controls the operation of each module .

chanhvinh.com

Item	Page	Revision (See Manual for Details)
33.3.6 Adjustment of XTAL Crystal Oscillator Gain	1809	Description amended 4. 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 36.3, Register States in Each Operating Mode.
33.4.1 Usage Notes on Setting Registers	1810	Title added
33.4.2 Usage Notes when the Realtime Clock is not Used	1810	Newly added
34.1 Features	1811	Description amended The user debugging interface is a serial input/output interface that supports JTAG-standard, IEEE Std.1149.1 .
Section 36 List of Registers	1837	Description amended 1. Register Addresses (by functional module, in order of the corresponding section numbers) • Registers are described by functional module, in order of the corresponding section numbers. • Access to reserved addresses which are not described in this register address list is prohibited. • When registers consist of 16 or 32 bits, the addresses of the MSBs are given when big endian mode is selected. An asterisk (*) in the column "Access Size" indicates that the unit of access in reading differs from that in writing for the given register. For details, see the register descriptions in the relevant section.
36.1 Register Addresses (by functional module, in order of the corresponding section numbers)	1846	Table amended

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
Watchdog timer	Watchdog timer control/status register	WTCSR	8	H'FFFE0000	8, 16*
	Watchdog timer counter	WTCNT	8	H'FFFE0002	8, 16*
	Watchdog reset control/status register	WRCSR	8	H'FFFE0004	8, 16

chanhvinh.com

Item	Page	Revision (See Manual for Details)
36.1 Register Addresses (by functional module, in order of the corresponding section numbers)	1855	Table amended

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
Controller area network	Interrupt Request Register_1	IRR_1	16	H'FFFE5808	16

1861 Table amended

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
CD-ROM decoder	CD-ROM decoder stream data input register	STRMDIN0	16	H'FFFF9200	16, 32*
	CD-ROM decoder stream data input register	STRMDIN2	16	H'FFFF9202	16
	CD-ROM decoder stream data output register	STRMDOUT0	16	H'FFFF9204	16, 32

1868 Table amended

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
General purpose I/O ports	Port A I/O register 0	PAIOR0	16	H'FFFE3812	8, 16
	Port A data register 1	PADR1	16	H'FFFE3814	8, 16, 32*
	Port A data register 0	PADR0	16	H'FFFE3816	8, 16*

1869 Table amended

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
General purpose I/O ports	Port G control register 7	PGCR7	16	H'FFFE38C0	8, 16, 32*

36.2 Register Bits	1941	Table amended

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
General purpose I/O ports	PADR1	—	—	—	—	—	—	—	PA3DR
		—	—	—	—	—	—	—	PA2DR

1942 Table amended

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
General purpose I/O ports	PBPR0	PB15PR	PB14PR	PB13PR	PB12PR	PB11PR	PB10PR	PB9PR	PB8PR
		PB7PR	PB6PR	PB5PR	PB4PR	PB3PR	PB2PR	PB1PR	—

36.3 Register States in Each Operating Mode	1949	Table amended

Module Name	Register Abbreviation	Power-On Reset	Manual Reset	Deep Standby	Software Standby	Module Standby	Sleep
Serial communication interface with FIFO	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained
Renesas serial peripheral interface	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained
I²C bus interface 3	ICMR_0 to 2	Initialized	Retained	Initialized	Retained*6	Retained*6	Retained
	Other than above	Initialized	Retained	Initialized	Retained	Retained	Retained

chanhvinh.com

Item	Page	Revision (See Manual for Details)
36.3 Register States in Each Operating Mode	1950	Table amended

Module Name	Register Abbreviation	Power-On Reset	Manual Reset	Deep Standby	Software Standby	Module Standby	Sleep
Power-down modes	DSFR	Initialized	Retained	Retained	Retained	—	Retained
	XTALCTR	Initialized*10	Retained	Retained*9	Retained*9	—	Retained
	Other than above	Initialized	Retained	Initialized	Retained	—	Retained

	1951	Note added 10. Retains the previous value after an internal power-on reset by the watchdog timer or a user debugging interface reset.
37.3 DC Characteristics Table 37.2 DC Characteristics (2) 1-Mbyte Version [Current Consumption]	1957	Table amended

Item		Power Supply	Symbol	Typ.	Max.	Unit	Test Conditions
Current consumption in deep standby mode	Ta > 50 C	PVcc+ Avcc+ Avref+ USBAPVcc+ USBDPVcc	PIdstby	1	11	µA	RTC is not operating
				3.5	21	µA	RTC_X1 selected
				1		mA	EXTAL selected, small gain*
		VBUS	VIdstby	6.5	7	A	

Table 37.2 DC Characteristics (3) [Except I2C Bus Interface 3, and USB 2.0 Host/Function Module-Related Pins]	1962	Table amended

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input characteristics	IRQ7 to IRQ0, PINT7 to PINT0, WAIT, $\overline{\text{BREQ}}$, DREQ1, DREQ0, TIOC0A to TIOC0D, TIOC1A, TIOC1B, TIOC2A, TIOC2B, TIOC3A to TIOC3D, TIOC4A to TIOC4D, TCLKB to TCLKD, SCK3 to SCK0, RxD7 to RxD0, $\overline{\text{CTS3}}$, $\overline{\text{RTS3}}$, $\overline{\text{CTS1}}$, $\overline{\text{RTS1}}$, RSPCK1, RSPCK0, MOSI1, MOSI0, MISO1, MISO0, SSL10, SSL00,	V_T^+	$PV_{CC} \times 0.75$	—	—	V	
		V_T^-	—	—	0.5	V	
		$V_T^+ - V_T^-$	0.2	—	—	V	

37.4 AC Characteristics Table 37.4 Maximum Operating Frequency	1966	Table amended

Item		Symbol	Min.	Max.	Unit	Remarks
Operating frequency	CPU clock (Iφ)	f	40.00	144.00	MHz	
	Bus clock (Bφ)		40.00	72.00	MHz	
	Peripheral clock (Pφ)		10.00	36.00	MHz	

37.4.3 Bus Timing Figure 37.38 PCMCIA I/O Card Bus Cycle (TED = 0 Cycle, TEH = 0 Cycle, No Wait)	2005	Figure amended

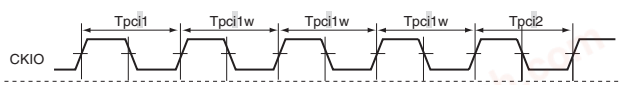


Figure 37.39 PCMCIA I/O Card Bus Cycle (TED = 2 Cycles, TEH = 1 Cycle, Software Wait Cycle 0, Hardware Wait Cycle 1)	2006	Figure replaced

chanhvinh.com

Item	Page	Revision (See Manual for Details)
37.6 Usage Note	—	Deleted
Section 38 States and Handling of Pins	2039 to 2054	Newly added

chanhvinh.com

Index

Numerics

(Potential) time master 1059
16-bit/32-bit displacement 59

A

A/D conversion time (multi mode and scan mode) 1272
A/D conversion time (single mode) 1272
A/D conversion timing 1271
A/D converter 1253
A/D converter activation 581
A/D converter characteristics 2037
A/D converter start request delaying function 574
A/D converter timing 2020
Absolute address 59
Absolute address accessing 59
Absolute maximum ratings 1953
AC characteristics 1966
AC characteristics measurement conditions 2036
Access size and data alignment 290
Access wait control 297
Address array 210, 224
Address array read 224
Address errors 142
Address map 234
Address multiplexing 304
Address spaces of on-chip data retention RAM 1664
Address spaces of on-chip high-speed RAM 1663
Address spaces of on-chip large capacity RAM 1664
Address-array write (associative operation) 225
Addressing modes 60
Analog input pin ratings 1278
Arithmetic operation instructions 79
Automatic decoding stop function 1243
Auto-refreshing 326
Auto-request mode 406

B

Bank active 319
Banked register and input/output of banks 202
Baud rate generator 957
Bit manipulation instructions 90
Bit synchronous circuit 883
Block diagram of this LSI 14
Boot mode 109
Branch instructions 84
Break detection and processing 768
Buffering format 1244
Burst mode 419
Burst read 311
Burst ROM (clocked asynchronous) interface 339
Burst ROM (clocked synchronous) interface 352
Burst write 316
Bus arbitration 361
Bus state controller 229
Bus timing 1973
Bus-released state 93

C

Cache 209
Cache operations 222

chanhvinh.com

Calculating exception handling vector table addresses 136
CAN bus interface 1073
CAN interface 979
Canceling software standby mode (watchdog timer) 666
Cascaded operation 515
Caution on period setting 595
CD-ROM decoder 1185
Changing the division ratio 125
Changing the frequency 125
Clock frequency control circuit 117
Clock operating modes 119
Clock pulse generator 115
Clock timing 1966
Clocked synchronous serial format 873
CMCNT count timing 651
Coherency of cache and external memory or large-capacity on-chip RAM 223
Command access mode 1318
Communications protocol 1082
Compare match timer 645
Complementary PWM mode 535
Conditions for determining number of idle cycles 354
Configuration mode 923
Configuration of controller area network 1048
Conflict between byte-write and count-up processes of CMCNT 656
Conflict between word-write and count-up processes of CMCNT 655
Conflict between write and compare-match processes of CMCNT 654
Control signal timing 1971
Controller area network 975
Controller area network control registers 998
Controller area network mailbox registers 1019
Controller area network memory map 981
Controller area network timer registers 1033
CPU 49
Crystal oscillator 117
$\overline{\text{CS}}$n assert period expansion 299
Cycle steal mode 417

D

Data array 210, 225
Data array read 225
Data array write 226
Data format 811
Data format in registers 54
Data formats in memory 54
Data transfer instructions 75
Data transfer with interrupt request signals 206
DC characteristics 1955
Decompression unit 1661
Deep power-down mode 338
Deep standby mode 1803
Definitions of A/D conversion accuracy 1275
Delayed branch instructions 57
Denormalized numbers 100
Direct memory access controller 367
Direct memory access controller interface 1072
Direct memory access controller timing 2007
Displacement accessing 59
Divider 1 117
Divider 2 117
DMA transfer flowchart 405
DREQ pin sampling timing 422
Dual address mode 414

chanhvinh.com

E

ECC correction 1242
ECC error check 1326
EDC checking 1243
Effective address calculation 60
Electrical characteristics 1953
Endian 290
Endian conversion for data in the input stream 1236
Equation for getting SCBRR value 728
Error detection function 823
Example of time triggered system 1063
Exception handling 131
Exception handling state 93
Exception handling vector table 134
Exception source generation immediately after delayed branch instruction 151
Exceptions triggered by instructions 147
External request mode 406
External trigger input timing 1273

F

Features of this LSI 1
Floating point operation instructions 150
Floating-point operation instructions 87
Floating-point ranges 98
Floating-point registers 101
Floating-point unit (FPU) 95
Format of double-precision floating-point number 96
Format of single-precision foating-point number 96
FPU exception sources 106
FPU-related CPU instructions 89
Full-scale error 1275

G

General illegal instructions 149
General purpose I/O ports timing 2032
General registers 49
Global base register (GBR) 51

H

Halt mode 1049

I

I^2C bus format 864
I^2C bus interface 3 845
I^2C bus interface 3 timing 2015
IBUF interrupt 1249
ID reorder 999
IEBus bit format 1093
IEBus communications protocol 1078
IEBus™ controller 1077
IERR interrupt 1249
Immediate data 58
Immediate data accessing 58
Immediate data format 55
Influences on absolute precision 1279
Initial values of control registers 53
Initial values of general registers 53
Initial values of system registers 53
Instruction features 56
Instruction format 65
Instruction set 69
Integer division instructions 149
Internal arbitration for transmission 1053
Interrupt controller 157
Interrupt exception handling 146
Interrupt exception handling vectors and priorities 178
Interrupt priority level 145
Interrupt response time 195
Interrupt sources 766

chanhvinh.com

IREADY interrupt 1249
IRQ interrupts 174
ISEC interrupt 1248
ISY interrupt 1249
ITARG interrupt 1248

J

Jump table base register (TBR) 51

L

List of pins of this LSI 29
Load-store architecture 56
Local acceptance filter mask (LAFM) 991
Logic operation instructions 82
Loopback mode 843
Low-frequency mode 330
Low-power SDRAM 336
LRU 211

M

Mailbox 978, 982
Mailbox configuration 990
Mailbox control 978
Manual reset 140
Master receive operation 867
Master transmit operation 865
Memory-mapped cache 224
Message control field 987
Message data fields 992
Message receive sequence 1067
Message transmission request 1052, 1062
Micro processor interface (MPI) 978
Module enabled mode 923
Module standby function 1809
MOSI signal value determination during SSL negate period 805
Motor control PWM timer 1821
Motor control PWM timer timing 2035
MPX-I/O interface 300
Multi mode 1265
Multi-function timer pulse unit 2 427
Multi-function timer pulse unit 2 timing 2008
Multi-master mode operation 829
Multiply and accumulate register high (MACH) 52
Multiply and accumulate register low (MACL) 52
Multiply/multiply-and-accumulate operations 57

N

NAND flash memory controller 1281
NAND flash memory controller interrupt requests 1332
NAND type flash memory controller timing 2021
NMI interrupt 173
Noise filter 877
Non-compressed modes 913
Nonlinearity error 1275
Non-numbers (NaN) 99
Normal space interface 293
Note on using a PLL oscillation circuit ... 129

O

Offset error 1275
On-chip peripheral module interrupts 176
On-chip peripheral module request 408
On-chip RAM 1663
Operation in asynchronous mode 746
Operation in clocked synchronous mode 757
Output load circuit 2036

chanhvinh.com

Output pin initialization for multi-function timer pulse unit 2 612

P

Package dimensions of this LSI..... 2055
Page conflict 1667
PCMCIA interface 345
Permissible signal source impedance.... 1278
Phase counting mode 525
Pin assignment of this LSI 16
Pin functions of this LSI 19
Pin states of this LSI 2039
PINT interrupts 175
PLL circuit 117
Power-down mode 332
Power-down modes 1761
Power-down state..... 93
Power-on reset 139
Power-on sequence 333
Power-on/power-off sequence 1954
Prefetch operation (only for operand cache)..... 220
Procedure register (PR)..... 52
Processing of analog input pins 1277
Product lineup of this LSI 12
Program counter (PC)..... 52
Program execution state..... 93
PWM Modes..... 520
PWM operation..... 1834

Q

Quantization error..... 1275

R

Realtime clock 671
Receive data sampling timing and receive margin (asynchronous mode) 768
Reconfiguration of mailbox 1069
Register addresses (by functional module, in order of the corresponding section numbers) 1838
Register bank error exception handling 144, 205
Register bank errors 143
Register bank exception 205
Register banks..... 53, 201
Register bits 1872
Register states in each operating mode..... 1949
Registers
ABACK0..... 1027
ABACK1..... 1027
ACKEYR 288
ACSWR 287
ADCSR 1258
ADDRA to ADDRH 1257
BCR0 1008
BCR1 1006
BEMPENB..... 1381
BEMPSTS..... 1399
BRDYENB 1378
BRDYSTS..... 1395
BUSWAIT 1348
CBUFCTL0..... 1224
CBUFCTL1..... 1226
CBUFCTL2..... 1226
CBUFCTL3..... 1227
CBUFST0 1210
CBUFST1 1211
CBUFST2 1212
CCR 1041
CCR1 212
CCR2 214
CFIFO 1360
CFIFOCTR 1370
CFIFOSEL 1362
CHCR..... 381

chanhvinh.com

CMAX_TEW 1036
CMCNT 650
CMCOR 650
CMCSR 648
CMNCR 239
CMSTR 647
CROMCTL0 1196
CROMCTL1 1198
CROMCTL3 1199
CROMCTL4 1200
CROMCTL5 1202
CROMEN 1194
CROMST0 1203
CROMST0M 1227
CROMST1 1204
CROMST3 1205
CROMST4 1206
CROMST5 1207
CROMST6 1208
CROMSY0 1195
CS0WCR 247, 263, 277
CS1WCR 250
CS2WCR 253, 268
CS3WCR 253, 269
CS4WCR 255, 265
CS5WCR 258, 272
CS6WCR 261, 272
CSnBCR (n = 0 to 6) 242
CTRL 1150
CYCTR 1042
D0FBCFG 1359
D0FIFO 1360
D0FIFOCTR 1370
D0FIFOSEL 1362
D1FBCFG 1359
D1FIFO 1360
D1FIFOCTR 1370
D1FIFOSEL 1362
DAR 380
DCPCFG 1409
DCPCTR 1411
DCPMAXP 1410
DEVADDn 1461
DMAOR 394
DMARS0 to DMARS7 398
DMATCR 381
DSESR 1793
DSFR 1795
DSSSR 1791
DVSTCTR 1350
FL4ECCCNT 1314
FL4ECCCR 1312
FL4ECCRESn 1311
FLADR 1295
FLADR2 1297
FLBSYCNT 1306
FLBSYTMR 1305
FLCMCDR 1294
FLCMDCR 1291
FLCMNCR 1287
FLDATAR 1299
FLDTCNTR 1298
FLDTFIFO 1307
FLECFIFO 1308
FLHOLDCR 1310
FLINTDMACR 1300
FLTRCR 1309
FPSCR 102
FPUL 104
FRMNUM 1400
FRQCR 122
GSR 1003
HEAD00 1212
HEAD01 1213
HEAD02 1213
HEAD03 1214
HEAD20 1218
HEAD21 1219
HEAD22 1219
HEAD23 1220

chanhvinh.com

IBCR 171
IBNR 172
ICCR1 849
ICCR2 852
ICDRR 862
ICDRS 862
ICDRT 861
ICIER 856
ICMR 854
ICR0 164
ICR1 166
ICR2 167
ICSR 858
IEAR1 1103
IEAR2 1104
IECKSR 1127
IECMR 1099
IECTR 1098
IEFLG 1112
IEIER 1125
IEIET 1119
IELA1 1110
IELA2 1111
IEMA1 1107
IEMA2 1108
IEMCR 1101
IERB 1129
IERBFL 1110
IERCTL 1109
IERSR 1121
IESA1 1104
IESA2 1105
IETB 1128
IETBFL 1106
IETSR 1115
IMR 1018
INHINT 1233
INTENB0 1374
INTENB1 1376
INTHOLD 1232
INTSTS0 1384
INTSTS1 1389
IPR01, IPR02, IPR05 to IPR22 162
IRQRR 168
IRR 1011
MBIMR0 1031
MBIMR1 1031
MCR 998
NF2CYC 863
NRDYENB 1379
NRDYSTS 1397
PADR0 1682
PADR1 1681
PAIOR0 1681
PAPR0 1683
PBCR0 1692
PBCR1 1690
PBCR2 1689
PBCR3 1687
PBCR4 1685
PBCR5 1683
PBDR0 1695
PBDR1 1694
PBIOR0 1693
PBIOR1 1693
PBPR0 1697
PBPR1 1696
PCCR0 1700
PCCR1 1699
PCCR2 1698
PCDR0 1702
PCIOR0 1701
PCPR0 1704
PDCR0 1710
PDCR1 1708
PDCR2 1706
PDCR3 1705
PDDR0 1712
PDIOR0 1711
PDPR0 1714

chanhvinh.com

PECR0 1716
PECR1 1715
PEDR0 1717
PEIOR0 1717
PEPR0 1719
PFCR0 1724
PFCR1 1722
PFCR2 1720
PFCR3 1720
PFDR0 1726
PFIOR0 1726
PFPR0 1728
PGCR0 1738
PGCR1 1736
PGCR2 1734
PGCR3 1733
PGCR4 1731
PGCR5 1730
PGCR6 1729
PGCR7 1729
PGDR0 1741
PGDR1 1740
PGIOR0 1739
PGIOR1 1739
PGPR0 1743
PGPR1 1742
PHCR0 1745
PHCR1 1744
PHPR0 1747
PINTER 169
PIPEBUF 1429
PIPECFG 1422
PIPEMAXP 1432
PIPEnCTR 1436
PIPEnTRE 1457
PIPEnTRN 1459
PIPEPERI 1434
PIPESEL 1421
PIRR 170
PJCR0 1750
PJCR1 1749
PJCR2 1748
PJDR0 1752
PJIOR0 1751
PJPR0 1753
PKCR0 1757
PKCR1 1756
PKCR2 1754
PKDR0 1758
PKIOR0 1758
PKPR0 1760
PWBFR_n 1831
PWBTCR 1832
PWCNT_n 1827
PWCR_n 1825
PWCYR_n 1827
PWDTR_n 1828
PWPR_n 1826
R64CNT 674
RCR1 689
RCR2 691
RCR3 693
RCR5 694
RDAD 1169
RDAR 392
RDAYAR 686
RDAYCNT 679
RDMATCR 393
REC 1018
RFMK 1043
RFPR0 1030
RFPR1 1029
RFRH/L 695
RFTROFF 1038
RHRAR 684
RHRCNT 677
RLCA 1167
RLCS 1171
RMINAR 683
RMINCNT 676

chanhvinh.com

RMONAR 687
RMONCNT 680
ROMDECRST 1228
RRCA 1168
RRCS 1173
RSAR 392
RSECAR 682
RSECCNT 675
RSTSTAT 1229
RTCNT 285
RTCOR 286
RTCSR 283
RUI 1170
RWKAR 685
RWKCNT 678
RXPR0 1029
RXPR1 1028
RYRAR 688
RYRCNT 681
SAR
(Direct memory access controller) 380
SAR (I^2C bus interface 3) 861
SCBRR 728
SCEMR 742
SCFCR 734
SCFDR 737
SCFRDR 711
SCFSR 720
SCFTDR 712
SCLSR 741
SCRSR 711
SCSCR 716
SCSMR 713
SCSPTR 738
SCTSR 712
SDBPR 1813
SDCR 279
SDENR 1815
SDIR 1813
SHEAD00 1214
SHEAD01 1215
SHEAD02 1215
SHEAD03 1216
SHEAD04 1216
SHEAD05 1217
SHEAD06 1217
SHEAD07 1218
SHEAD20 1220
SHEAD21 1221
SHEAD22 1221
SHEAD23 1222
SHEAD24 1222
SHEAD25 1223
SHEAD26 1223
SHEAD27 1224
SICTR 939
SIFCTR 951
SIIER 949
SIMDR 937
SIRDAR 956
SIRDR 943
SISCR 953
SISTR 944
SITDAR 954
SITDR 942
SOFCFG 1382
SPBR 789
SPCKD 793
SPCMD 796
SPCR 777
SPDR 785
SPND 795
SPPCR 780
SPSCR 786
SPSR 782
SPSSR 788
SRCCTRL 1639
SRCID 1632
SRCIDCTRL 1635
SRCOD 1633

chanhvinh.com

SRCODCTRL 1637
SRCSTAT 1645
SSI 1229
SSICR 893
SSIFCR 905
SSIFRDR 911
SSIFSR 908
SSIFTDR 911
SSIRDR 904
SSISR 900
SSITDR 904
SSLND 794
SSLP 779
STAT 1155
STBCR1 1765
STBCR2 1766
STBCR3 1767
STBCR4 1769
STBCR5 1772
STBCR6 1774
STBCR7 1776
STBCR8 1778
STRMDIN0 1234
STRMDIN2 1234
STRMDOUT0 1235
SWRSTCR 1779
SYSCFG 1344
SYSCR1 1781
SYSCR2 1782
SYSCR3 1783
SYSCR4 1785
SYSCR5 1786
SYSSTS 1349
TADCOBRA_4 476
TADCOBRB_4 476
TADCORA_4 476
TADCORB_4 476
TADCR 473
TBTER 498
TBTM 470
TCBR 495
TCDR 494
TCMR0 to TCMR2 1043
TCNT 477
TCNTR 1042
TCNTS 493
TCR 437
TDAD 1161
TDDR 494
TDER 500
TEC 1018
TESTMODE 1356
TGCR 491
TGR 477
TICCR 471
TIER 462
TIOR 444
TITCNT 497
TITCR 495
TLCA 1159
TLCS 1163
TMDR 441
TOCR1 484
TOCR2 487
TOER 482
TOLBR 490
TRCA 1160
TRCS 1165
TRWER 481
TSR 465, 1039
TSTR 478
TSYR 479
TTCR0 1034
TTTSEL 1045
TUI 1162
TWCR 501
TXACK0 1026
TXACK1 1025
TXCR0 1025
TXCR1 1024

chanhvinh.com

TXPR0 1023
TXPR1 1022
UFRMNUM 1403
UMSR0 1032
UMSR1 1032
USBACSWR1 1464
USBADDR 1404
USBINDX 1407
USBLENG 1408
USBREQ 1405
USBVAL 1406
WRCSR 663
WTCNT 659
WTCSR 660
XTALCTR 1797
Relationship between access size and number of bursts 311
Relationship between refresh requests and bus cycles 330
Renesas serial peripheral interface 771
Renesas serial peripheral interface timing 2011
Renesas SPDIF interface 1145
Reset sequence 1048
Reset state 93
Reset-synchronized PWM mode 532
Restoration from bank 203
Restoration from stack 204
Restriction on direct memory access controller usage 768
RISC-type instruction set 56
Roles of mailboxes 984
Rounding 105

S

Sampling rate converter 1629
Saving to bank 202
Saving to stack 204
Scan mode 1267
SD host interface 1659
SD host interface timing 2031
SDRAM interface 303
Searching cache 218
Sector access mode 1321
Self-refreshing 328
Sending a break signal 768
Serial bit clock control 930
Serial communication interface with FIFO 703
Serial communication interface with FIFO timing 2010
Serial I/O with FIFO 933
Serial I/O with FIFO timing 2018
Serial sound interface 887
Serial sound interface timing 2016
Setting analog input voltage 1276
Setting I/O ports for controller area network 1074
Shift instructions 83
Sign extension of word data 56
Single address mode 416
Single mode 1262
Single read 315
Single write 318
Slave mode operation 837
Slave receive operation 872
Slave transmit operation 869
Sleep mode 1049, 1798
Slot illegal instructions 148
Software standby mode 1799
SRAM interface with byte selection 341
Stack after interrupt exception handling 194
Stack status after exception handling ends 151
Standby control circuit 117
Status register (SR) 50
Stopping and resuming CD-DSP operation 1251

chanhvinh.com

Supported DMA transfers 413
Syndrome calculation 1242
System configuration example 806
System control instructions 85
System matrix 997

T

T bit 57
TAP controller 1816
Target-sector buffering function 1246
TDO output timing 1817
Test mode settings 1046
Time slave 1060
Time trigger control (TT control) 994
Time triggered transmission 1055
Timestamp 993
Timing to clear an interrupt source 208
Transfer format 809, 810
Transfer rate 851
Trap instructions 148
TTW[1:0] (time trigger window) 995
Tx-trigger control field 994
Tx-trigger time (TTT) 994
Types of exception handling and priority order 131

U

Unconditional branch instructions with no delay slot 57
USB 2.0 host/function module 1337
USB 2.0 host/function module timing 2026
User debugging interface 1811
User debugging interface interrupt 174
User debugging interface interrupt 1818
User debugging interface reset 1818
User debugging interface timing 2033
Using alarm function 699
Using interval timer mode 668
Using watchdog timer mode 666

V

Vector base register (VBR) 51
Video display controller 3 1539
Video display controller 3 timing 2029

W

Wait between access cycles 353
Watchdog timer 657
Watchdog timer timing 2009
Write-back buffer (only for operand cache) 221

chanhvinh.com

Renesas 32-Bit RISC Microcomputer
SH7262 Group, SH7264 Group
User's Manual: Hardware

Publication Date: Rev.1.00, March 5, 2009
Rev.3.00, March 17, 2011
Published by: Renesas Electronics Corporation

chanhvinh.com

RENESAS

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
7F, No. 363 Fu Shing North Road Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

© 2011 Renesas Electronics Corporation. All rights reserved.
Colophon 1.0

chanhvinh.com
chanhvinh.com
chanhvinh.com
chanhvinh.com

chanhvinh.com

SH7262 Group, SH7264 Group
User's Manual: Hardware

Renesas Electronics Corporation

R01UH0134EJ0300
(Previous Number: REJ09B0445-0200)