chanhvinh.com

User's Manual

16

The revision list summarizes the locations of revisions and additions. Details should always be checked by referring to the relevant text.

# H8S/20103, H8S/20203, H8S/20223, H8S/20115, H8S/20215, H8S/20235 Group

## User's Manual: Hardware

## Renesas 16-Bit Single-Chip Microcomputer

## H8S Family / H8S/Tiny Series

H8S/20103 R4F20103

H8S/20203 R4F20203

H8S/20223 R4F20223

H8S/20115 R4F20115

H8S/20215 R4F20215

H8S/20235 R4F20235

Renesas Electronics
www.renesas.com

Rev.3.00 Sep 2010

chanhvinh.com
chanhvinh.com
chanhvinh.com
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 part number, confirm that the change will not lead to problems.

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

chanhvinh.com

# How to Use This Manual

1. Objective and Target Users

   This manual was written to explain the hardware functions and electrical characteristics of this LSI to the target users, i.e. those who will be using this LSI in the design of application systems. Target users are expected to understand the fundamentals of electrical circuits, logic circuits, and microcomputers.

   This manual is organized in the following items: an overview of the product, descriptions of the CPU, system control functions, and peripheral functions, electrical characteristics of the device, and usage notes.

> When designing an application system that includes this LSI, take all points to note into account. Points to note are given in their contexts and at the final part of each section, and in the section giving usage notes.

> The list of revisions is a summary of major points of revision or addition for earlier versions. It does not cover all revised items. For details on the revised points, see the actual locations in the manual.

The following documents have been prepared for the H8S/20103, H8S/20203, H8S/20223, H8S/20115, H8S/20215, and H8S/20235 Groups. Before using any of the documents, please visit our web site to verify that you have the most up-to-date available version of the document.

| Document Type | Contents | Document Title | Document No. |
|---|---|---|---|
| Data Sheet | Overview of hardware and electrical characteristics | — | — |
| User's manual for Hardware | Hardware specifications (pin assignments, memory maps, peripheral specifications, electrical characteristics, and timing charts) and descriptions of operation | H8S/20103, H8S/20203, H8S/20223, H8S/20115, H8S/20215, H8S/20235 Group User's Manual: Hardware | This User's manual |
| User's manual for Software | Detailed descriptions of the CPU and instruction set | H8S/2600 Series, H8S/2000 Series Software Manual | REJ09B0139 |
| Application Note | Examples of applications and sample programs | The latest versions are available from our web site. | |
| Renesas Technical Update | Preliminary report on the specifications of a product, document, etc. | | |

chanhvinh.com

## 2. 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

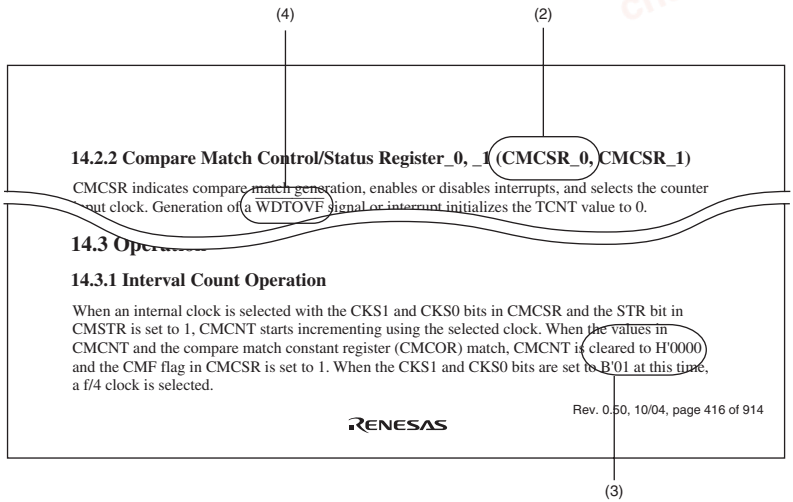
The style "register name"_"instance number" is used in cases where there is more than one instance of the same function or similar functions.
[Example] CMCSR_0: Indicates the CMCSR register for the compare-match timer of channel 0.

(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] $\overline{\text{WDTOVF}}$

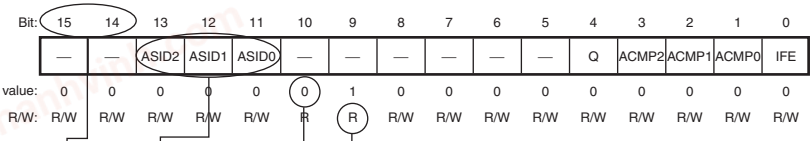



Note: The bit names and sentences in the above figure are examples and have nothing to do with the contents of this manual.

3. 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<br>These bits enable or disable the pin function. |
| 10 | – | 0 | R | Reserved<br>This bit is always read as 0. |
| 9 | – | 1 | R | Reserved<br>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.

4. Description of Abbreviations

The abbreviations used in this manual are listed below.

- Abbreviations specific to this product

| Abbreviation | Description |
|---|---|
| BSC | Bus controller |
| CPG | Clock pulse generator |
| INT | Interrupt controller |
| SCI3 | Serial communications interface 3 |
| PMC | Peripheral I/O mapping controller |
| WDT | Watchdog timer |
| DTC | Data transfer controller |
| ELC | Event link controller |
| IIC2 | Inter IC bus interface 2 |
| SSU | Synchronous serial communication unit |
| LVD | Low-voltage detection circuits |

- Abbreviations other than those listed above

| Abbreviation | Description |
|---|---|
| Bps | Bits per second |
| CRC | Cyclic redundancy check |
| DMA | Direct memory access |
| Hi-Z | High impedance |
| I/O | Input/output |
| IrDA | Infrared Data Association |
| LSB | Least significant bit |
| MSB | Most significant bit |
| NC | No connection |
| PLL | Phase-locked loop |
| PWM | Pulse width modulation |
| SFR | Special function register |
| UART | Universal asynchronous receiver/transmitter |
| VCO | Voltage-controlled oscillator |

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

chanhvinh.com

# Contents

Section 1 Overview ..... 1
1.1 Features ..... 1
1.1.1 Applications ..... 1
1.1.2 Overview of Functions ..... 1
1.2 List of Products ..... 6
1.3 Block Diagram ..... 8
1.4 Pin Assignments ..... 11
1.4.1 Pin Functions ..... 14

Section 2 CPU ..... 21
2.1 Features ..... 21
2.1.1 Differences between H8S/2600 CPU and H8S/2000 CPU ..... 22
2.1.2 Differences from H8/300 CPU ..... 23
2.1.3 Differences from H8/300H CPU ..... 23
2.2 CPU Operating Modes ..... 24
2.2.1 Advanced Mode ..... 24
2.3 Address Space ..... 26
2.4 Register Configuration ..... 33
2.4.1 General Registers ..... 34
2.4.2 Program Counter (PC) ..... 35
2.4.3 Extended Control Register (EXR) ..... 35
2.4.4 Condition-Code Register (CCR) ..... 36
2.4.5 Initial Register Values ..... 38
2.5 Data Formats ..... 39
2.5.1 General Register Data Formats ..... 39
2.5.2 Memory Data Formats ..... 41
2.6 Instruction Set ..... 42
2.6.1 Table of Instructions Classified by Function ..... 43
2.6.2 Basic Instruction Formats ..... 53
2.7 Addressing Modes and Effective Address Calculation ..... 55
2.7.1 Register Direct—Rn ..... 55
2.7.2 Register Indirect—@ERn ..... 55
2.7.3 Register Indirect with Displacement—@(d:16, ERn) or @(d:32, ERn) ..... 56
2.7.4 Register Indirect with Post-Increment or Pre-Decrement—@ERn+ or @–ERn ..... 56
2.7.5 Absolute Address—@aa:8, @aa:16, @aa:24, or @aa:32 ..... 56
2.7.6 Immediate—#xx:8, #xx:16, or #xx:32 ..... 57

chanhvinh.com

2.7.7 Program-Counter Relative—@(d:8, PC) or @(d:16, PC) ..... 57
2.7.8 Memory Indirect—@@aa:8 ..... 58
2.7.9 Effective Address Calculation ..... 59
2.8 Processing States ..... 61
2.9 Usage Notes ..... 63
2.9.1 TAS Instruction ..... 63
2.9.2 STM and LDM Instructions ..... 63
2.9.3 Note on Bit Manipulation Instructions ..... 63
2.9.4 EEPMOV Instruction ..... 64

Section 3 Exception Handling ..... 65
3.1 Exception Handling Types and Priority ..... 65
3.2 Exception Handling Sources and Vector Table ..... 65
3.3 Reset ..... 66
3.3.1 Reset Sources ..... 66
3.3.2 Reset Exception Handling ..... 69
3.3.3 Interrupts immediately after Reset ..... 70
3.3.4 On-Chip Peripheral Functions after Reset Release ..... 70
3.4 Trace Exception Handling ..... 71
3.5 Interrupt Exception Handling ..... 72
3.6 Trap Instruction Exception Handling ..... 72
3.7 Stack Status after Exception Handling ..... 73
3.8 Usage Note ..... 74

Section 4 Interrupt Controller ..... 75
4.1 Features ..... 75
4.2 Register Descriptions ..... 77
4.2.1 Interrupt Control Register (INTCR) ..... 78
4.2.2 Interrupt Priority Registers A to I (IPRA to IPRI) ..... 79
4.2.3 IRQ Enable Register (IER) ..... 81
4.2.4 IRQ Sense Control Register H and L (ISCRH and ISCRL) ..... 82
4.2.5 IRQ Status Register (ISR) ..... 85
4.2.6 IRQ Noise Canceler Control Register (INCCR) ..... 86
4.2.7 Interrupt Vector Offset Register (VOFR) ..... 87
4.2.8 Event Link Interrupt Control Status Register (ELCSR) ..... 88
4.3 Interrupt Sources ..... 89
4.3.1 External Interrupt sources ..... 89
4.3.2 Internal Interrupts ..... 90
4.4 Interrupt Exception Handling Vector Table ..... 91

chanhvinh.com

4.5 Interrupt Control Modes and Interrupt Operation ..... 98
4.5.1 Interrupt Control Mode 0 ..... 98
4.5.2 Interrupt Control Mode 2 ..... 100
4.5.3 Interrupt Exception Handling Sequence ..... 102
4.5.4 Interrupt Response Time ..... 104
4.5.5 DTC Activation by Interrupt ..... 104
4.6 Usage Notes ..... 105
4.6.1 Conflict between Interrupt Generation and Disabling ..... 105
4.6.2 Instructions that Disable Interrupts ..... 106
4.6.3 Time when Interrupts are Disabled ..... 106
4.6.4 Interrupts during Execution of EEPMOV Instruction ..... 106
4.6.5 Changing PMR, ISCRH, ISCRL and INCCR ..... 107
4.6.6 IRQ Status Register (ISR) ..... 107
4.6.7 $\overline{\text{NMI}}$ Pin ..... 108

Section 5 Clock Pulse Generator ..... 109
5.1 Overview ..... 110
5.2 Register Descriptions ..... 111
5.2.1 Backup Control Register (BAKCR) ..... 112
5.2.2 System Clock Control Register (SYSCCR) ..... 114
5.2.3 Power-Down Control Register 1 (LPCR1) ..... 116
5.2.4 Power-Down Control Register 2 (LPCR2) ..... 118
5.2.5 Power-Down Control Register 3 (LPCR3) ..... 119
5.2.6 OSC Oscillation Settling Control Status Register (OSCCSR) ..... 121
5.3 Operation of Selection of System Reference Clock ..... 123
5.3.1 Switching System Reference Clock to $\phi$osc ..... 126
5.3.2 Clock Change Timing ..... 128
5.3.3 Backup Operation ..... 131
5.4 Main Clock Oscillator ..... 135
5.4.1 Connecting Crystal Resonator ..... 135
5.4.2 Connecting Ceramic Resonator ..... 136
5.4.3 External Clock Input Method ..... 136
5.5 Subclock Oscillator ..... 137
5.5.1 Connecting 32.768-kHz Crystal Resonator ..... 137
5.5.2 Pin Connection when not Using Subclock ..... 137
5.6 Prescaler ..... 138
5.7 Usage Notes ..... 139
5.7.1 Note on Resonators ..... 139
5.7.2 Notes on Board Design ..... 139

chanhvinh.com

Section 6 Power-Down Modes ..... 141
6.1 Register Descriptions ..... 142
6.1.1 Power-Down Control Registers 1, 2, and 3 (LPCR1, LPCR2, LPCR3) ..... 142
6.1.2 Module Standby Control Register 1 (MSTCR1) ..... 142
6.1.3 Module Standby Control Register 2 (MSTCR2) ..... 144
6.1.4 Module Standby Control Register 3 (MSTCR3) ..... 145
6.2 Mode Transitions and States of LSI ..... 147
6.2.1 Active Mode ..... 149
6.2.2 Sleep Mode ..... 149
6.2.3 Standby Mode ..... 150
6.3 Bus Master Clock Division Function ..... 150
6.3.1 Reset States ..... 150
6.4 Module Standby Function ..... 151
6.5 PSC Divider Stop Function ..... 151

Section 7 ROM ..... 153
7.1 Overview ..... 153
7.2 Block Configuration ..... 154
7.3 CPU Reprogramming Mode ..... 159
7.3.1 EW0 Mode ..... 161
7.3.2 EW1 Mode ..... 161
7.4 Register Descriptions ..... 162
7.4.1 Flash Memory Control Register 1 (FLMCR1) ..... 162
7.4.2 Flash Memory Control Register 2 (FLMCR2) ..... 164
7.4.3 Flash Memory Data Flash Protect Register (DFPR) ..... 166
7.4.4 Flash Memory Status Register (FLMSTR) ..... 167
7.5 On-Board Programming ..... 170
7.5.1 Boot Mode ..... 170
7.5.2 Specifications of Standard Serial Communication Interface in Boot Mode ..... 175
7.5.3 Programming/Erasing in User Mode ..... 210
7.6 Programming/Erasing ..... 212
7.6.1 Software Commands ..... 212
7.7 Protection ..... 228
7.7.1 Software Protection ..... 228
7.7.2 Lock-Bit Protection ..... 228
7.7.3 PROM Programmer Protection/Boot Mode Protection ..... 229
7.8 Programmer Mode ..... 230
7.9 Usage Notes ..... 231

Section 8 RAM ..... 235

chanhvinh.com

Section 9 Peripheral I/O Mapping Controller ..... 237
9.1 Register Descriptions ..... 239
9.1.1 Peripheral Function Mapping Register Write-Protect Register (PMCWPR) ..... 240
9.1.2 Port Group 1 Peripheral Function Mapping Registers 1 to 4 (PMCRn1 to PMCRn4 (n = 1, 2, 3, 5, and 6)) ..... 241
9.1.3 Port Group 2 Peripheral Function Mapping Registers 1 to 4 (PMCRn1 to PMCRn4 (n = 8, 9, and A) ..... 262
9.2 Usage Notes ..... 270
9.2.1 Procedures for Setting Multiplexed Port Functions ..... 270
9.2.2 Notes on Setting PMC Registers ..... 270

Section 10 I/O Ports ..... 271
10.1 Port 1 ..... 271
10.1.1 Port Mode Register 1 (PMR1) ..... 272
10.1.2 Port Control Register 1 (PCR1) ..... 273
10.1.3 Port Data Register 1 (PDR1) ..... 274
10.1.4 Port Pull-Up Control Register 1 (PUCR1) ..... 275
10.1.5 Port Drive Control Register 1 (PDVR1) ..... 276
10.2 Port 2 ..... 277
10.2.1 Port Mode Register 2 (PMR2) ..... 278
10.2.2 Port Control Register 2 (PCR2) ..... 279
10.2.3 Port Data Register 2 (PDR2) ..... 280
10.2.4 Port Pull-Up Control Register 2 (PUCR2) ..... 281
10.2.5 Port Drive Control Register 2 (PDVR2) ..... 282
10.3 Port 3 ..... 283
10.3.1 Port Mode Register 3 (PMR3) ..... 284
10.3.2 Port Control Register 3 (PCR3) ..... 285
10.3.3 Port Data Register 3 (PDR3) ..... 286
10.3.4 Port Pull-Up Control Register 3 (PUCR3) ..... 287
10.3.5 Port Drive Control Register 3 (PDVR3) ..... 288
10.4 Port 5 ..... 289
10.4.1 Port Mode Register 5 (PMR5) ..... 290
10.4.2 Port Control Register 5 (PCR5) ..... 291
10.4.3 Port Data Register 5 (PDR5) ..... 292
10.4.4 Port Pull-Up Control Register 5 (PUCR5) ..... 293
10.4.5 Port Drive Control Register 5 (PDVR5) ..... 294
10.5 Port 6 ..... 295
10.5.1 Port Mode Register 6 (PMR6) ..... 296
10.5.2 Port Control Register 6 (PCR6) ..... 297
10.5.3 Port Data Register 6 (PDR6) ..... 298

chanhvinh.com

10.5.4 Port Pull-Up Control Register 6 (PUCR6) ..... 299
10.5.5 Port Drive Control Register 6 (PDVR6) ..... 300
10.6 Port 8 ..... 301
10.6.1 Port Mode Register 8 (PMR8) ..... 302
10.6.2 Port Control Register 8 (PCR8) ..... 303
10.6.3 Port Data Register 8 (PDR8) ..... 304
10.6.4 Port Pull-Up Control Register 8 (PUCR8) ..... 305
10.6.5 Port Drive Control Register 8 (PDVR8) ..... 306
10.6.6 Notes on Using Port 8 ..... 306
10.7 Port 9 ..... 307
10.7.1 Port Mode Register 9 (PMR9) ..... 308
10.7.2 Port Control Register 9 (PCR9) ..... 309
10.7.3 Port Data Register 9 (PDR9) ..... 310
10.7.4 Port Pull-Up Control Register 9 (PUCR9) ..... 311
10.7.5 Port Drive Control Register 9 (PDVR9) ..... 312
10.8 Port A ..... 313
10.8.1 Port Mode Register A (PMRA) ..... 314
10.8.2 Port Control Register A (PCRA) ..... 315
10.8.3 Port Data Register A (PDRA) ..... 316
10.8.4 Port Pull-Up Control Register A (PUCRA) ..... 317
10.8.5 Port Mode Register A (PMRA) ..... 318
10.8.6 Port Control Register A (PCRA) ..... 319
10.8.7 Port Data Register A (PDRA) ..... 320
10.8.8 Port Pull-Up Control Register A (PUCRA) ..... 321
10.8.9 Port Mode Register A (PMRA) ..... 322
10.8.10 Port Control Register A (PCRA) ..... 323
10.8.11 Port Data Register A (PDRA) ..... 324
10.8.12 Port Pull-Up Control Register A (PUCRA) ..... 325
10.8.13 Notes on Using Port A ..... 325
10.9 Port B ..... 326
10.9.1 Port Control Register B (PCRB) ..... 327
10.9.2 Port Data Register B (PDRB) ..... 328
10.9.3 Port Pull-Up Control Register B (PUCRB) ..... 329
10.9.4 Notes on Using Port B ..... 329
10.10 Port J ..... 330
10.10.1 Port Mode Register J (PMRJ) ..... 331
10.10.2 Port Control Register J (PCRJ) ..... 332
10.10.3 Port Data Register J (PDRJ) ..... 333
10.10.4 Port Pull-Up Control Register J (PUCRJ) ..... 334

chanhvinh.com

Section 11 Data Transfer Controller (DTC) ..... 335
11.1 Features ..... 335
11.2 Register Descriptions ..... 337
11.2.1 DTC Mode Register A (MRA) ..... 338
11.2.2 DTC Mode Register B (MRB) ..... 340
11.2.3 DTC Source Address Register (SAR) ..... 341
11.2.4 DTC Destination Address Register (DAR) ..... 341
11.2.5 DTC Transfer Count Register A (CRA) ..... 342
11.2.6 DTC Transfer Count Register B (CRB) ..... 342
11.2.7 DTC Enable Registers A to H (DTCERA to DTCERH) ..... 343
11.2.8 DTC Vector Register (DTVECR) ..... 345
11.3 Activation Sources ..... 346
11.4 Location of Register Information and DTC Vector Table ..... 348
11.5 Operation ..... 353
11.5.1 Normal Mode ..... 355
11.5.2 Repeat Mode ..... 356
11.5.3 Block Transfer Mode ..... 357
11.5.4 Chain Transfer ..... 358
11.5.5 Interrupt Sources ..... 359
11.5.6 Operation Timing ..... 360
11.5.7 Number of DTC Execution States ..... 361
11.6 Procedures for Using DTC ..... 363
11.6.1 Activation by Interrupt ..... 363
11.6.2 Activation by Software ..... 363
11.7 Examples of Use of the DTC ..... 364
11.7.1 Normal Mode ..... 364
11.7.2 Chain Transfer when Transfer Counter = 0 ..... 365
11.7.3 Software Activation ..... 367
11.8 Usage Notes ..... 368
11.8.1 Module Standby Mode Setting ..... 368
11.8.2 DTCE Bit Setting ..... 368
11.8.3 DTC Activation by SCI3, IIC2/SSU and A/D Converter Interrupt Sources ..... 368
11.8.4 Limitation on Usage of the Interrupt Vector Offset Register (VOFR) ..... 368

Section 12 Event Link Controller ..... 369
12.1 Overview ..... 369
12.2 Register Descriptions ..... 371
12.2.1 Event Link Control Register (ELCR) ..... 371
12.2.2 Event Link Setting Registers 0 to 32 (ELSR0 to ELSR32) ..... 372
12.2.3 Event Link Option Setting Register A (ELOPA) ..... 376

chanhvinh.com

12.2.4 Event Link Option Setting Register B (ELOPB) ..... 377
12.2.5 Event Link Option Setting Register C (ELOPC) ..... 377
12.2.6 Port-Group Setting Registers 1 and 2 (PGR1 and PGR2) ..... 378
12.2.7 Port-Group Control Registers 1 and 2 (PGC1 and PGC2) ..... 379
12.2.8 Port Buffer Registers 1 and 2 (PDBF1 and PDBF2) ..... 380
12.2.9 Event Link Port Setting Registers 0 to 3 (PEL0 to PEL3) ..... 381
12.2.10 Event-Generation Timer Control Register (ELTMCR) ..... 382
12.2.11 Event-Generation Timer Interval Setting Register A (ELTMSA) ..... 383
12.2.12 Event-Generation Timer Interval Setting Register B (ELTMSB) ..... 385
12.2.13 Event-Generation Timer Delay Selection Register (ELTMDR) ..... 387
12.2.14 ELC Timer Counter (ELTMCNT) ..... 388
12.3 Operation ..... 389
12.3.1 Relation between Interrupt Processing and Event Linking ..... 389
12.3.2 Event Linkage ..... 389
12.3.3 Operation of Peripheral Timer Modules When Event is Input ..... 391
12.3.4 Operation of A/D and D/A Converters When Event is Input ..... 391
12.3.5 Port Operation upon Event Input and Event Generation ..... 392
12.3.6 Event-Generation Timer ..... 398
12.3.7 Procedure for Linking Events ..... 400

Section 13 Timer RA ..... 401
13.1 Overview ..... 401
13.2 Register Descriptions ..... 402
13.2.1 Timer RA Control Register (TRACR) ..... 403
13.2.2 Timer RA I/O Control Register (TRAIOC) ..... 405
13.2.3 Timer RA Mode Register (TRAMR) ..... 407
13.2.4 Timer RA Interrupt Enable Status Register (TRAIR) ..... 408
13.2.5 Timer RA Prescaler Register (TRAPRE) ..... 409
13.2.6 Timer RA Timer Register (TRATR) ..... 410
13.3 Operation ..... 411
13.3.1 Operations Common to Various Modes ..... 411
13.3.2 Timer Mode ..... 412
13.3.3 Pulse Output Mode ..... 412
13.3.4 Event Counter Mode ..... 413
13.3.5 Pulse Width Measurement Mode ..... 413
13.3.6 Pulse Cycle Measurement Mode ..... 415
13.3.7 Operation through an Event Link ..... 416
13.4 Usage Notes ..... 418

chanhvinh.com

Section 14 Timer RB .....419
14.1 Overview .....419
14.2 Register Descriptions .....420
14.2.1 Timer RB Control Register (TRBCR) .....421
14.2.2 Timer RB One-Shot Control Register (TRBOCR) .....422
14.2.3 Timer RB I/O Control Register (TRBIOC) .....423
14.2.4 Timer RB Mode Register (TRBMR) .....425
14.2.5 Timer RB Interrupt Request Status Register (TRBIR) .....426
14.2.6 Timer RB Prescaler Register (TRBPRE) .....427
14.2.7 Timer RB Secondary Register (TRBSC) .....427
14.2.8 Timer RB Primary Register (TRBPR) .....428
14.3 Operation .....429
14.3.1 Timer Mode .....429
14.3.2 Programmable Waveform Generation Mode .....430
14.3.3 Programmable One-Shot Generation Mode .....432
14.3.4 Programmable Wait One-Shot Generation Mode .....434
14.3.5 Timing at Which Values Take Effect in Prescaler or Counter Depending on TWRC Bit .....436
14.3.6 TOCNT Settings and Pin State Update Conditions .....438
14.3.7 Operation through an Event Link .....439
14.4 Interrupt Request .....439
14.5 Usage Notes .....440

Section 15 Timer RC .....441
15.1 Features .....441
15.2 Register Descriptions .....444
15.2.1 Timer RC Mode Register (TRCMR) .....445
15.2.2 Timer RC Control Register 1 (TRCCR1) .....446
15.2.3 Timer RC Control Register 2 (TRCCR2) .....448
15.2.4 Timer RC Interrupt Enable Register (TRCIER) .....449
15.2.5 Timer RC Status Register (TRCSR) .....450
15.2.6 Timer RC I/O Control Register 0 (TRCIOR0) .....453
15.2.7 Timer RC I/O Control Register 1 (TRCIOR1) .....455
15.2.8 Timer RC Output Enable Register (TRCOER) .....457
15.2.9 Timer RC Digital Filtering Function Select Register (TRCDF) .....458
15.2.10 Timer RC A/D Conversion Start Trigger Control Register (TRCADCR) .....459
15.2.11 Timer RC Counter (TRCCNT) .....460
15.2.12 General Registers A, B, C, and D (GRA, GRB, GRC, and GRD) .....461

chanhvinh.com

15.3 Operation ..... 463
15.3.1 Timer Mode Operation ..... 465
15.3.2 PWM Mode Operation ..... 470
15.3.3 PWM2 Mode Operation ..... 475
15.3.4 Digital Filtering Function for Input Capture Inputs ..... 481
15.3.5 A/D Conversion Start Trigger Setting Function ..... 482
15.3.6 Function of Changing Output Pins for GR ..... 484
15.3.7 Operation through an Event Link ..... 486
15.4 Operation Timing ..... 487
15.4.1 TRCCNT Counting Timing ..... 487
15.4.2 Output Compare Output Timing ..... 488
15.4.3 Input Capture Timing ..... 489
15.4.4 Timing of Counter Clearing by Compare Match ..... 489
15.4.5 Buffer Operation Timing ..... 490
15.4.6 Timing of IMFA to IMFD Flag Setting at Compare Match ..... 491
15.4.7 Timing of IMFA to IMFD Setting at Input Capture ..... 492
15.4.8 Timing of Status Flag Clearing ..... 493
15.4.9 Timing of A/D Conversion Start Trigger Generation on Compare Match ..... 494
15.5 Usage Notes ..... 495

Section 16 Timer RD ..... 499
16.1 Features ..... 499
16.2 Register Descriptions ..... 507
16.2.1 Timer RD Start Register (TRDSTR) ..... 509
16.2.2 Timer RD Mode Register (TRDMDR) ..... 511
16.2.3 Timer RD PWM Mode Register (TRDPMR) ..... 512
16.2.4 Timer RD Function Control Register (TRDFCR) ..... 513
16.2.5 Timer RD Output Master Enable Register 1 (TRDOER1) ..... 515
16.2.6 Timer RD Output Master Enable Register 2 (TRDOER2) ..... 517
16.2.7 Timer RD Output Control Register (TRDOCR) ..... 517
16.2.8 Timer RD A/D Conversion Start Trigger Control Register (TRDADCR) ..... 519
16.2.9 Timer RD Counter (TRDCNT) ..... 520
16.2.10 General Registers A, B, C, and D (GRA, GRB, GRC, and GRD) ..... 521
16.2.11 Timer RD Control Register (TRDCR) ..... 523
16.2.12 Timer RD I/O Control Registers (TRDIORA and TRDIORC) ..... 525
16.2.13 Timer RD Status Register (TRDSR) ..... 529
16.2.14 Timer RD Interrupt Enable Register (TRDIER) ..... 532
16.2.15 PWM Mode Output Level Control Register (POCR) ..... 533
16.2.16 Timer RD Digital Filtering Function Select Register (TRDDF) ..... 534
16.2.17 Interface with CPU ..... 535

chanhvinh.com

16.3 Operation ..... 536
16.3.1 Counter Operation ..... 545
16.3.2 Waveform Output by Compare Match ..... 548
16.3.3 Input Capture Function ..... 551
16.3.4 Synchronous Operation ..... 554
16.3.5 PWM Mode ..... 555
16.3.6 Reset Synchronous PWM Mode ..... 561
16.3.7 Complementary PWM Mode ..... 565
16.3.8 PWM3 Mode Operation ..... 571
16.3.9 Buffer Operation ..... 577
16.3.10 Timer RD Output Timing ..... 585
16.3.11 Digital Filtering Function for Input Capture Inputs ..... 588
16.3.12 Function of Changing Output Pins for GR ..... 589
16.3.13 A/D Conversion Start Trigger Setting Function ..... 591
16.3.14 Operation by Event Clear ..... 593
16.4 Interrupt Sources ..... 594
16.4.1 Status Flag Set Timing ..... 594
16.4.2 Status Flag Clearing Timing ..... 596
16.5 Usage Notes ..... 596

Section 17 Timer RE ..... 605
17.1 Features ..... 605
17.2 Register Descriptions ..... 607
17.2.1 Timer RE Second Data Register/Counter Data Register (TRESEC) ..... 608
17.2.2 Timer RE Minute Data Register/Compare Data Register (TREMIN) ..... 609
17.2.3 Timer RE Hour Data Register (TREHR) ..... 610
17.2.4 Timer RE Day-of-Week Data Register (TREWK) ..... 611
17.2.5 Timer RE Control Register 1 (TRECR1) ..... 612
17.2.6 Timer RE Control Register 2 (TRECR2) ..... 615
17.2.7 Timer RE Interrupt Flag Register (TREIFR) ..... 616
17.2.8 Timer RE Clock Source Select Register (TRECSR) ..... 618
17.3 Operation of Realtime Clock Mode ..... 619
17.3.1 Initial Settings of Registers after Power-On ..... 619
17.3.2 Initial Setting Procedure ..... 619
17.3.3 Data Reading Procedure in Realtime Clock Mode ..... 621
17.3.4 Operation in Realtime Clock Mode ..... 622
17.4 Operation of Output Compare Mode ..... 623
17.5 Interrupt Sources ..... 626
17.6 Usage Notes ..... 627

chanhvinh.com

Section 18 Timer RG ..... 629
18.1 Features ..... 629
18.2 Register Descriptions ..... 632
18.2.1 Timer RG Mode Register (TRGMDR) ..... 633
18.2.2 Timer RG Counter Control Register (TRGCNTCR) ..... 634
18.2.3 Timer RG Control Register (TRGCR) ..... 635
18.2.4 Timer RG I/O Control Register (TRGIOR) ..... 636
18.2.5 Timer RG Status Register (TRGSR) ..... 638
18.2.6 Timer RG Interrupt Enable Register (TRGIER) ..... 639
18.2.7 Timer RG Counter (TRGCNT) ..... 640
18.2.8 General Registers A and B (GRA, GRB), GRA and GRB Buffer Registers (BRA, BRB) ..... 641
18.3 Operation ..... 643
18.3.1 Timer Mode ..... 644
18.3.2 PWM Mode ..... 650
18.3.3 Phase Counting Mode ..... 655
18.3.4 Buffer Operation ..... 660
18.3.5 Operation through an Event Link ..... 663
18.3.6 Digital Filtering Function for Input Capture Inputs ..... 664

Section 19 Watchdog Timer (WDT) ..... 665
19.1 Features ..... 666
19.2 Register Descriptions ..... 667
19.2.1 Timer Control/Status Register WD (TCSRWD) ..... 667
19.2.2 Timer Counter WD (TCWD) ..... 668
19.2.3 Timer Mode Register WD (TMWD) ..... 669
19.2.4 Timer Interrupt Control Register WD (TICRWD) ..... 670
19.2.5 Timer Interrupt Flag Register WD (TIFRWD) ..... 671
19.3 Operation ..... 672
19.3.1 Watchdog Timer Overflow Reset ..... 672
19.3.2 Watchdog Timer Setting Flow ..... 673
19.3.3 Watchdog Timer Periodic Interrupt ..... 674
19.4 Usage Notes ..... 675
19.4.1 Notes on System Design ..... 675
19.4.2 Notes on Stopping the Watchdog Timer or Switching the Count Clock ..... 675

Section 20 Serial Communication Interface 3 (SCI3, IrDA) ..... 677
20.1 Features ..... 677
20.2 Register Descriptions ..... 682
20.2.1 Receive Shift Register (RSR) ..... 683

chanhvinh.com

20.2.2 Receive Data Register (RDR) ..... 683
20.2.3 Transmit Shift Register (TSR) ..... 683
20.2.4 Transmit Data Register (TDR) ..... 684
20.2.5 Serial Mode Register (SMR) ..... 684
20.2.6 Serial Control Register 3 (SCR3) ..... 686
20.2.7 Serial Status Register (SSR) ..... 688
20.2.8 Bit Rate Register (BRR) ..... 690
20.2.9 Sampling Mode Register (SPMR) ..... 695
20.2.10 IrDA Control Register (IrCR) ..... 696
20.3 Operation in Asynchronous Mode ..... 698
20.3.1 Clock ..... 698
20.3.2 SCI3 Initialization ..... 699
20.3.3 Data Transmission ..... 700
20.3.4 Data Reception ..... 702
20.4 Operation in Clocked Synchronous Mode ..... 706
20.4.1 Clock ..... 706
20.4.2 SCI3 Initialization ..... 706
20.4.3 Data Transmission ..... 707
20.4.4 Data Reception (Clocked Synchronous Mode) ..... 709
20.4.5 Simultaneous Data Transmission and Reception ..... 711
20.5 Multiprocessor Communication Function ..... 713
20.5.1 Multiprocessor Data Transmission ..... 714
20.5.2 Multiprocessor Data Reception ..... 716
20.6 IrDA Operation ..... 720
20.6.1 Transmission ..... 721
20.6.2 Reception ..... 721
20.6.3 High-Level Pulse Width Selection ..... 722
20.7 Noise Canceler ..... 723
20.8 Interrupt Requests ..... 724
20.9 Usage Notes ..... 725
20.9.1 Break Detection and Processing ..... 725
20.9.2 Mark State and Break Sending ..... 725
20.9.3 Receive Error Flags and Transmit Operations (Clocked Synchronous Mode Only) ..... 725
20.9.4 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode ..... 726
20.9.5 Relation between Writes to TDR and TDRE Flag ..... 727
20.9.6 Restrictions on Using DTC ..... 727

chanhvinh.com

Section 21 I²C Bus Interface 2 (IIC2) ..... 729
21.1 Features ..... 729
21.2 Register Descriptions ..... 732
21.2.1 IIC2/SSU Select Register (ICSUSR) ..... 732
21.2.2 I²C Bus Control Register 1 (ICCR1) ..... 733
21.2.3 I²C Bus Control Register 2 (ICCR2) ..... 735
21.2.4 I²C Bus Mode Register (ICMR) ..... 737
21.2.5 I²C Bus Interrupt Enable Register (ICIER) ..... 739
21.2.6 I²C Bus Status Register (ICSR) ..... 741
21.2.7 Slave Address Register (SAR) ..... 744
21.2.8 I²C Bus Transmit Data Register (ICDRT) ..... 745
21.2.9 I²C Bus Receive Data Register (ICDRR) ..... 745
21.2.10 I²C Bus Shift Register (ICDRS) ..... 745
21.3 Operation ..... 746
21.3.1 I²C Bus Format ..... 746
21.3.2 Master Transmit Operation ..... 747
21.3.3 Master Receive Operation ..... 749
21.3.4 Slave Transmit Operation ..... 751
21.3.5 Slave Receive Operation ..... 754
21.3.6 Clock Synchronous Serial Format ..... 755
21.3.7 Noise Filter Circuit ..... 758
21.3.8 Example of Use ..... 759
21.4 Interrupt Request ..... 763
21.5 Bit Synchronous Circuit ..... 764
21.6 Usage Notes ..... 765
21.6.1 SCL and SDA pins selected by PMC ..... 765
21.6.2 Restriction on Use of Bit Manipulation Instructions to Set MST and TRS in Multi-Master Usage ..... 765
21.6.3 Note regarding Master Receive Mode of I²C-Bus Interface Mode ..... 765
21.6.4 Note on Access to ICE in ICCR1 and IICRST in ICCR2 during I²C Bus Operation ..... 766

Section 22 Synchronous Serial Communication Unit (SSU) ..... 767
22.1 Features ..... 767
22.2 Register Descriptions ..... 769
22.2.1 IIC2/SSU Select Register (ICSUSR) ..... 769
22.2.2 SS Control Register H (SSCRH) ..... 770
22.2.3 SS Control Register L (SSCRL) ..... 771
22.2.4 SS Mode Register (SSMR) ..... 773

chanhvinh.com

22.2.5 SS Mode Register 2 (SSMR2) ..... 774
22.2.6 SS Enable Register (SSER) ..... 777
22.2.7 SS Status Register (SSSR) ..... 778
22.2.8 SS Receive Data Register (SSRDR) ..... 780
22.2.9 SS Transmit Data Register (SSTDR) ..... 780
22.2.10 SS Shift Register (SSTRSR) ..... 780
22.3 Operation ..... 781
22.3.1 Transfer Clock ..... 781
22.3.2 Relationship between Clock Polarity and Phase, and Data ..... 781
22.3.3 Relationship between Data Input/Output Pin and Shift Register ..... 782
22.3.4 Communication Modes and Pin Functions ..... 783
22.3.5 Operation in Clocked Synchronous Communication Mode ..... 784
22.3.6 Operation in Four-Line Bus Communication Mode ..... 791
22.3.7 $\overline{SCS}$ Pin Control and Arbitration ..... 797
22.4 Interrupt Requests ..... 798
22.5 Usage Notes ..... 798

Section 23 Hardware LIN ..... 799
23.1 Overview ..... 799
23.2 Register Configuration ..... 800
23.2.1 LIN Control Register (LINCR) ..... 800
23.2.2 LIN Status Register (LINST) ..... 802
23.3 Operation ..... 803
23.3.1 Master Mode ..... 803
23.3.2 Slave Mode ..... 806
23.3.3 Bus Conflict Detection Function ..... 811
23.3.4 Terminating Hardware LIN ..... 812
23.4 Interrupt Requests ..... 813
23.5 Usage Note ..... 813

Section 24 A/D Converter ..... 815
24.1 Features ..... 815
24.2 Register Description ..... 819
24.2.1 A/D Data Registers 0 to 7 (ADDR0 to ADDR7) ..... 820
24.2.2 A/D Control/Status Register (ADCSR) ..... 821
24.2.3 A/D Control Register (ADCR) ..... 823
24.2.4 A/D Mode Register (ADMR) ..... 825
24.2.5 Compare Data Register (CMPR) ..... 826
24.2.6 Compare Control/Status Register (CMPCSR) ..... 828
24.2.7 Compare Voltage Registers H and L (CMPVALH and CMPVALL) ..... 830

chanhvinh.com

24.3 Operation ..... 832
24.4 A/D Conversion Mode Operation ..... 833
24.4.1 Single Mode in A/D Conversion Mode ..... 833
24.4.2 Scan Mode in A/D Conversion Mode ..... 835
24.5 Compare Mode Operation ..... 837
24.5.1 Single Mode in Compare Mode ..... 837
24.5.2 Scan Mode in Comparison Mode ..... 838
24.5.3 Input Sampling and A/D Conversion Time ..... 839
24.5.4 External Trigger Input Timing ..... 841
24.6 Interrupt Source ..... 842
24.7 A/D Conversion Accuracy Definitions ..... 843
24.8 Usage Notes ..... 845
24.8.1 Module Standby Mode Setting ..... 845
24.8.2 Permissible Signal Source Impedance ..... 845
24.8.3 Influences on Absolute Precision ..... 846
24.8.4 Setting Range of Analog Power Supply and Other Pins ..... 846
24.8.5 Notes on Board Design ..... 846
24.8.6 Notes on Noise Countermeasures ..... 847
24.8.7 Notes on Analog Input Pins ..... 848

Section 25 D/A Converter ..... 849
25.1 Features ..... 849
25.2 Register Descriptions ..... 850
25.2.1 D/A Data Registers 0 and 1 (DADR0 and DADR1) ..... 850
25.2.2 D/A Control Register (DACR) ..... 851
25.3 Operation ..... 852
25.4 Usage Notes ..... 854
25.4.1 Setting for Module Stop Mode ..... 854
25.4.2 Operation in Standby Mode ..... 854

Section 26 Low-Voltage Detection Circuits ..... 855
26.1 Features ..... 856
26.2 Register Descriptions ..... 858
26.2.1 Low-Voltage Detection Circuit Control Protect Register (VDCPR) ..... 859
26.2.2 Low-Voltage Detection Circuit 2 Control Register H (LD2CRH) ..... 860
26.2.3 Low-Voltage Detection Circuit 2 Control Register L (LD2CRL) ..... 862
26.2.4 Low-Voltage Detection Circuit 1 Control Register H (LD1CRH) ..... 863
26.2.5 Low-Voltage Detection Circuit 1 Control Register L (LD1CRL) ..... 865
26.2.6 Low-Voltage Detection Circuit 0 Control Register H (LD0CRH) ..... 866
26.2.7 Low-Voltage Detection Circuit 0 Control Register L (LD0CRL) ..... 867

chanhvinh.com

26.3 Operation ..... 868
26.3.1 Power-On Reset Function ..... 868
26.3.2 Low-Voltage Detection Circuit ..... 869

Section 27 List of Registers ..... 881
27.1 Register Addresses (Address Order) ..... 882
27.2 Register Bits ..... 898

Section 28 Electrical Characteristics ..... 913
28.1 Absolute Maximum Ratings ..... 913
28.2 Electrical Characteristics ..... 914
28.2.1 Power Supply Voltage and Operating Ranges ..... 914
28.3 DC Characteristics ..... 916
28.4 AC Characteristics ..... 926
28.5 A/D Converter Characteristics ..... 933
28.6 D/A Converter Characteristics ..... 934
28.7 Flash Memory Characteristics ..... 935
28.8 Low-Voltage Detection Circuits Characteristics ..... 937
28.9 Power-On Reset Function Characteristics ..... 940
28.10 Timing Charts ..... 941
28.11 Output Load Circuit ..... 949

Appendix ..... 951
A. Package Dimensions ..... 951
B. Handling of Unused Pins ..... 954

Main Revisions and Additions in this Edition ..... 955

Index ..... 979

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

chanhvinh.com



# Section 1 Overview

## 1.1 Features

The H8S/Tiny series is a 16-bit CISC (complex instruction set computer) microcontroller, each member of the H8S/Tiny series has the powerful H8S/2000 CPU with an internal 32-bit architecture as its core. The H8S/2000 CPU provides upwards-compatibility with the other members of the Renesas H8 Family: H8/300, H8/300H Tiny and H8/300H.

The on-chip peripheral function modules include a data transfer controller, event link controller, serial communication interface 3, $I^2C$ bus interface 2, synchronous serial communication unit, hardware LIN communication interface, A/D and D/A converters, low-voltage detection circuit, and versatile timers. These modules realize low-cost systems. The power consumption of these modules can be controlled dynamically using power-down modes.

### 1.1.1 Applications

Examples of the applications include home appliances, office automation equipment, consumer equipment, and industrial equipment.

### 1.1.2 Overview of Functions

Table 1.1 lists the specifications of the products of this series.

**Table 1.1 Overview of Functions**

| Classification | Module/ Function | Description |
|---|---|---|
| Memory | ROM | • Flash memory version<br>Program memory: 256 Kbytes, 192 Kbytes, 128 Kbytes, or 96 Kbytes<br>Number of program/erase times: 1000 times<br>Data flash: 4 Kbytes × two blocks<br>Number of program/erase times: 10000 times |
| | RAM | • Capacity: 12 Kbytes, 8 Kbytes |

chanhvinh.com



| Classification | Module/ Function | Description |
|---|---|---|
| CPU | CPU | • 16-bit high-speed H8S/2000 CPU (CISC type)<br>Upwardly compatible with H8/300 and H8/300H CPUs at object level<br>• General-register architecture (sixteen 16-bit general registers)<br>• Eight addressing modes<br>• 16-Mbyte address space<br>— Program: 16 Mbytes available<br>— Data: 16 Mbytes available<br>• 65 basic instructions including bit operation instructions, multiply and divide instructions, bit manipulation instructions, and others<br>• Minimum instruction execution time: 50 ns (for an ADD instruction) while system clock $\phi$ = 20 MHz and $V_{CC}$ = 2.7 to 5.5 V |
| | Operating mode | Advanced single-chip mode |
| Interrupt (source) | Interrupt controller (INTC) | • Nine external interrupt pins (NMI, and $\overline{IRQ7}$ to $\overline{IRQ0}$)<br>• Internal interrupt sources<br>— 55 (H8S/20103 and H8S/20115 Groups)<br>— 61 (H8S/20203 and H8S/20215 Groups)<br>— 63 (H8S/20223 and H8S/20235 Groups)<br>• Two interrupt control modes (specified by the interrupt control register)<br>• Four interrupt priority orders specifiable (by setting the interrupt priority register)<br>• Independent vector addresses |
| Clock | Clock pulse generator (CPG) | • Two clock generation circuits: main and sub-clock oscillators<br>• One on-chip oscillator<br>— Low speed: 125 kHz<br>• Three power-down modes: sleep mode, software standby mode, and module standby mode |

chanhvinh.com



| Classification | Module/ Function | Description |
|---|---|---|
| Voltage detection | Low-voltage detection circuit (LVD) | Voltage drop detected |
| DMA | Data transfer controller (DTC) | • Transfer via any number of channels possible<br>• Three transfer modes |
| A/D converter | A/D converter (ADC) | • 10-bit resolution × eight to sixteen input channels<br>• Sample and hold function included<br>• Conversion time: 2 µs per channel<br>• Two operating modes: single mode and scan mode<br>• Three ways to start A/D conversion: software, timer trigger, and external pin trigger. |
| D/A converter | D/A converter (DAC) | • 8-bit resolution × two input channels |
| Timers | Timer RA | 8 bits × one channel (with 8-bit prescaler) |
| | Timer RB | 8 bits × one channel (with 8-bit prescaler) |
| | Timer RC | 16 bits × one channel (only available with H8S/20103 and H8S/20115 Groups) |
| | Timer RD | 16 bits × two channels (× two units in H8S/20203, H8S/20223, H8S/20215, and H8S/20235 Groups) |
| | Timer RE | 8 bits × one channel with real-time clock function |
| | Timer RG | 16 bits × one channel with phase-counting mode |
| | Watchdog timer (WDT) | 8 bit × one channel |
| Serial interfaces | Serial communi-cation interface 3 (SCI3) | • Three channels (either for asynchronous or clock-synchronous communication)<br>• Full-duplex communication capability<br>• Any desired bit rate selectable |
| | | IrDA (only available with channel 2) |
| | Synchro-nous serial communi-cation unit (SSU) | • One channel (IIC2 and selection format)<br>• Clock-synchronous communication with chip-select function |

chanhvinh.com



| Classification | Module/ Function | Description |
|---|---|---|
| Serial interfaces | $I^2C$ bus interface 2 (IIC2) | • One channel (SSU and selection format)<br>• Continuous transmission and reception possible<br>• Two transmission/reception formats<br>— $I^2C$ bus format: generates start and stop conditions in master mode automatically, acknowledge bit, master or slave operation<br>— Clock-synchronous serial format: no acknowledge bit, master operation only |
| | Hardware LIN interface | One channel (timer RA and SCI3 used) |
| Event link controller (ELC) | | Events (interrupts) generated by peripheral modules can be interconnected between modules, enabling cooperation between the modules without CPU intervention. |
| I/O ports | | • I/O pins<br>— 55 (H8S/20103 and H8S/20115 Groups)<br>— 69 (H8S/20203, H8S/20223, H8S/20215, and H8S/20235 Groups)<br>• Pull-up resistors settable for all ports<br>• LED driving capability |

chanhvinh.com



| Classification | Module/ Function | Description |
|---|---|---|
| Packages | | • 64-pin QFP package (PLQP0064KB-A)<br>— Former code: 64P6Q-A<br>— Body size: 10 × 10 mm<br>— Pin pitch: 0.50 mm<br>• 64-pin QFP package (PLQP0064GA-A)<br>— Former code: 64P6U-A<br>— Body size: 14 × 14 mm<br>— Pin pitch: 0.80 mm<br>• 80-pin QFP package (PLQP0080JA-A)<br>— Former code: FP-80W<br>— Body size: 14 × 14 mm<br>— Pin pitch: 0.65 mm<br>• 80-pin QFP package (PLQP0080KB-A) (in planning)<br>— Former code: 80P6Q-A<br>— Body size: 12 × 12 mm<br>— Pin pitch: 0.5 mm |
| Operating frequency/ Power supply voltage | | • Operating frequency: 4 to 20 MHz<br>• Power supply voltage: Vcc = 2.7 to 5.5 V, Avcc = 2.7 to 5.5 V |
| Operating ambient temperature (°C) | | • –20 to +85°C (version N)<br>• –40 to +85°C (version D) |

chanhvinh.com



## 1.2 List of Products

Table 1.2 lists products of this series, and figure 1.1 shows how to read the part number.

**Table 1.2 List of Products**

| Group | Part No. | ROM Capacity | RAM Capacity | Package | Remarks |
|---|---|---|---|---|---|
| H8S/20103 | R4F20103NFA | 128 Kbytes | 8 Kbytes | PLQP0064KB-A | Version N |
| | R4F20102NFA | 96 Kbytes | 8 Kbytes | (LQFP1010-64) | |
| | R4F20103NFB | 128 Kbytes | 8 Kbytes | PLQP0064GA-A | |
| | R4F20102NFB | 96 Kbytes | 8 Kbytes | (LQFP1414-64) | |
| | R4F20103DFA | 128 Kbytes | 8 Kbytes | PLQP0064KB-A | Version D |
| | R4F20102DFA | 96 Kbytes | 8 Kbytes | (LQFP1010-64) | |
| | R4F20103DFB | 128 Kbytes | 8 Kbytes | PLQP0064GA-A | |
| | R4F20102DFB | 96 Kbytes | 8 Kbytes | (LQFP1414-64) | |
| H8S/20203 | R4F20203NFD | 128 Kbytes | 8 Kbytes | PLQP0080JA-A | Version N |
| | R4F20202NFD | 96 Kbytes | 8 Kbytes | (LQFP1414-80) | |
| | R4F20203DFD | 128 Kbytes | 8 Kbytes | | Version D |
| | R4F20202DFD | 96 Kbytes | 8 Kbytes | | |
| | R4F20203NFC | 128 Kbytes | 8 Kbytes | PLQP0080KB-A | Version N in planning |
| | R4F20202NFC | 96 Kbytes | 8 Kbytes | (LQFP1212-80) | |
| H8S/20223 | R4F20223NFD | 128 Kbytes | 8 Kbytes | PLQP0080JA-A | Version N |
| | R4F20222NFD | 96 Kbytes | 8 Kbytes | (LQFP1414-80) | |
| | R4F20223DFD | 128 Kbytes | 8 Kbytes | | Version D |
| | R4F20222DFD | 96 Kbytes | 8 Kbytes | | |
| | R4F20223NFC | 128 Kbytes | 8 Kbytes | PLQP0080KB-A | Version N in planning |
| | R4F20222NFC | 96 Kbytes | 8 Kbytes | (LQFP1212-80) | |

chanhvinh.com



| Group | Part No. | ROM Capacity | RAM Capacity | Package | Remarks |
|---|---|---|---|---|---|
| H8S/20115 | R4F20115NFA | 256 Kbytes | 12 Kbytes | PLQP0064KB-A | Version N |
| | R4F20114NFA | 192 Kbytes | 12 Kbytes | (LQFP1010-64) | |
| | R4F20115NFB | 256 Kbytes | 12 Kbytes | PLQP0064GA-A | |
| | R4F20114NFB | 192 Kbytes | 12 Kbytes | (LQFP1414-64) | |
| H8S/20215 | R4F20215NFD | 256 Kbytes | 12 Kbytes | PLQP0080JA-A | |
| | R4F20214NFD | 192 Kbytes | 12 Kbytes | (LQFP1414-80) | |
| H8S/20235 | R4F20235NFD | 256 Kbytes | 12 Kbytes | | |
| | R4F20234NFD | 192 Kbytes | 12 Kbytes | | |

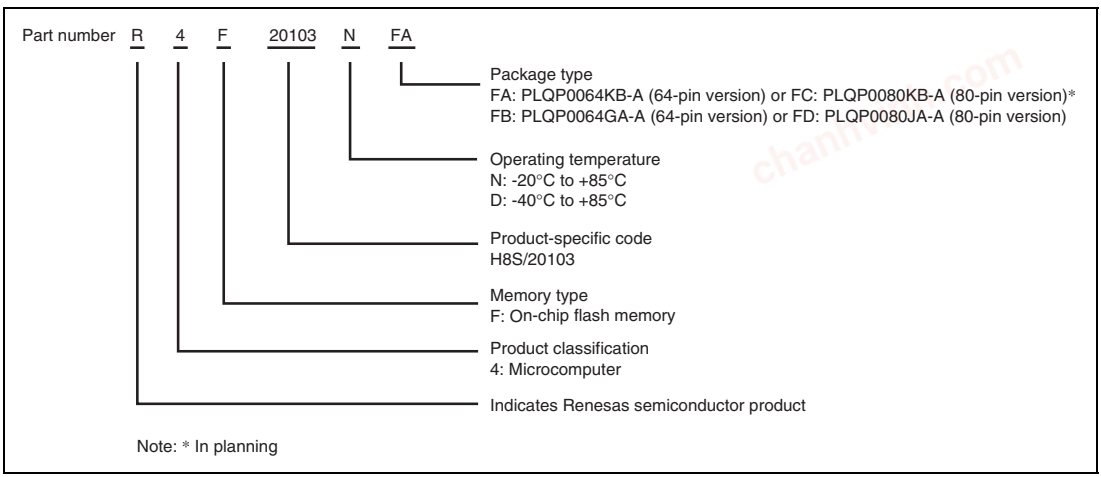



**Figure 1.1 How to Read the Part Number**

chanhvinh.com



## 1.3 Block Diagram

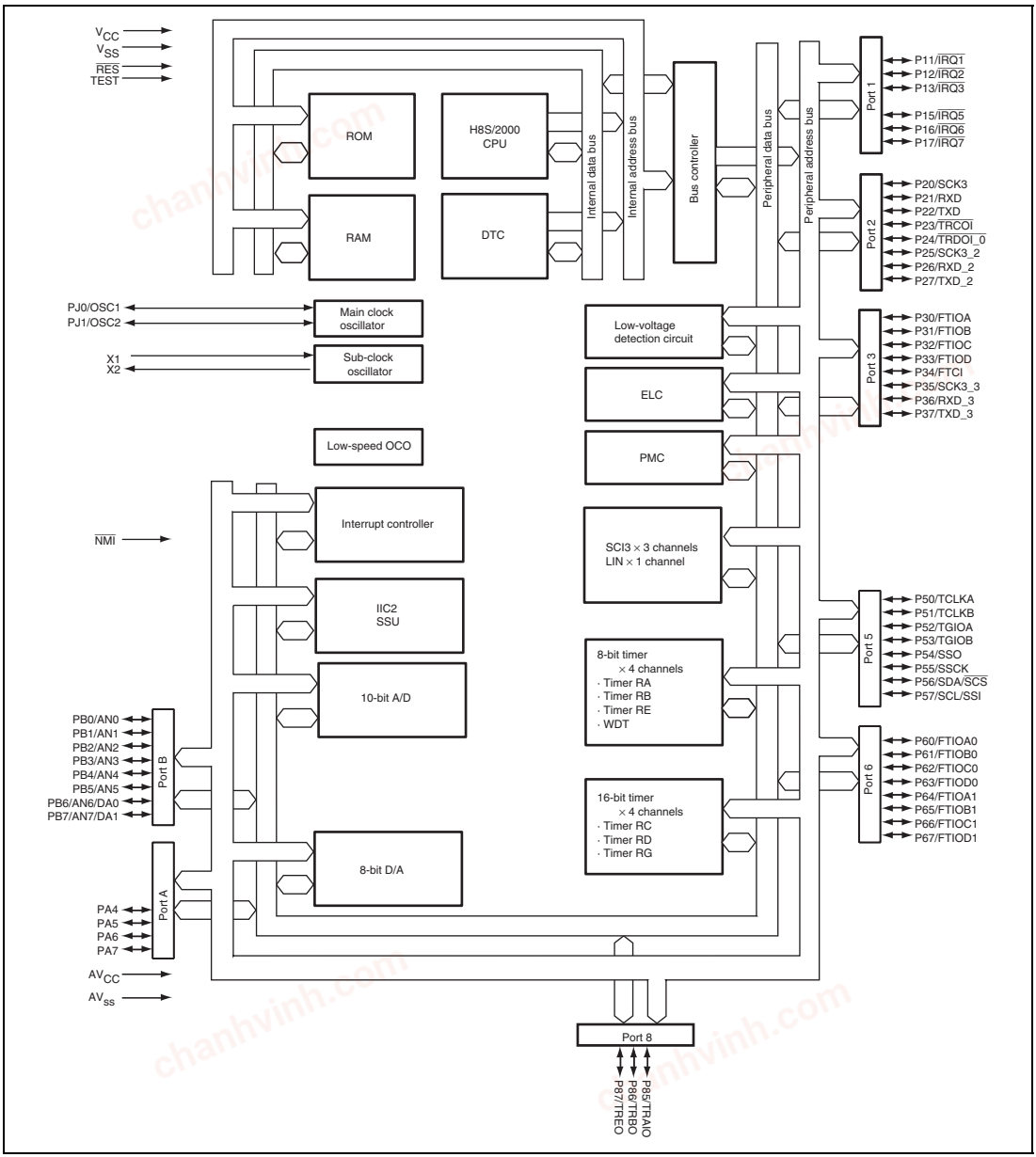



**Figure 1.2 Block Diagram of the H8S/20103 and H8S/20115 Groups**

chanhvinh.com



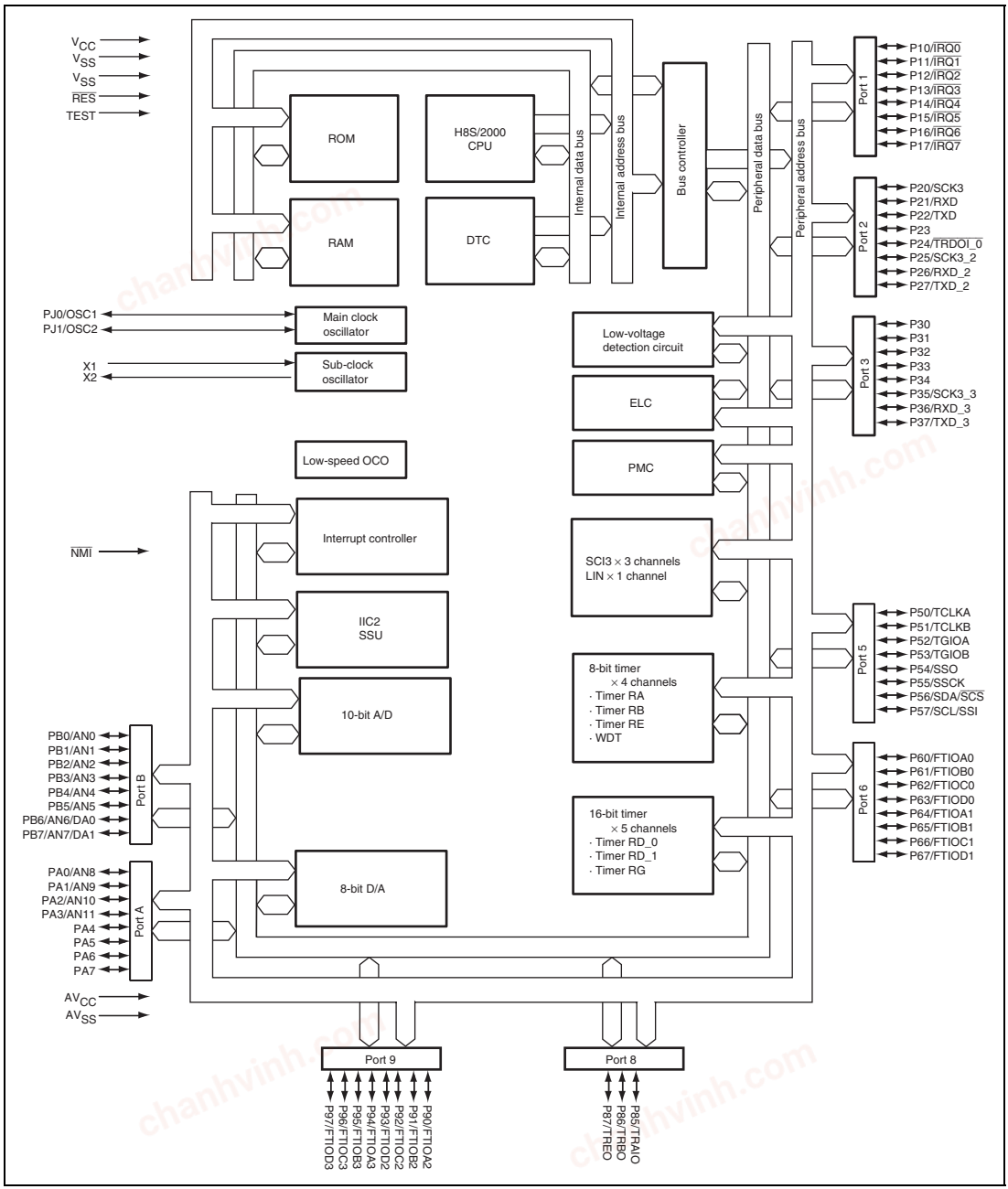



**Figure 1.3 Block Diagram of the H8S/20203 and H8S/20215 Groups**

chanhvinh.com



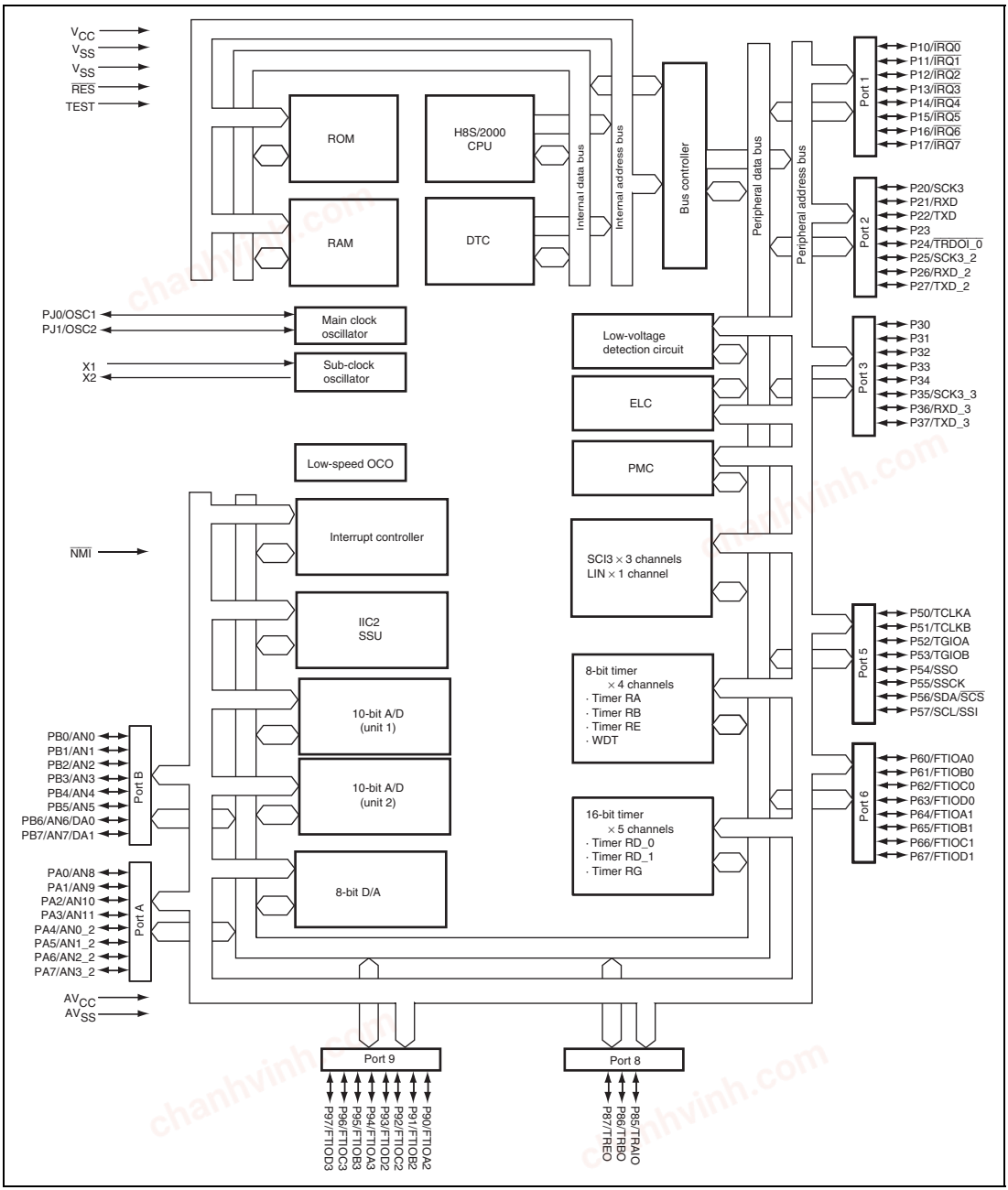



**Figure 1.4 Block Diagram of the H8S/20223 and H8S/20235 Groups**

chanhvinh.com



## 1.4 Pin Assignments

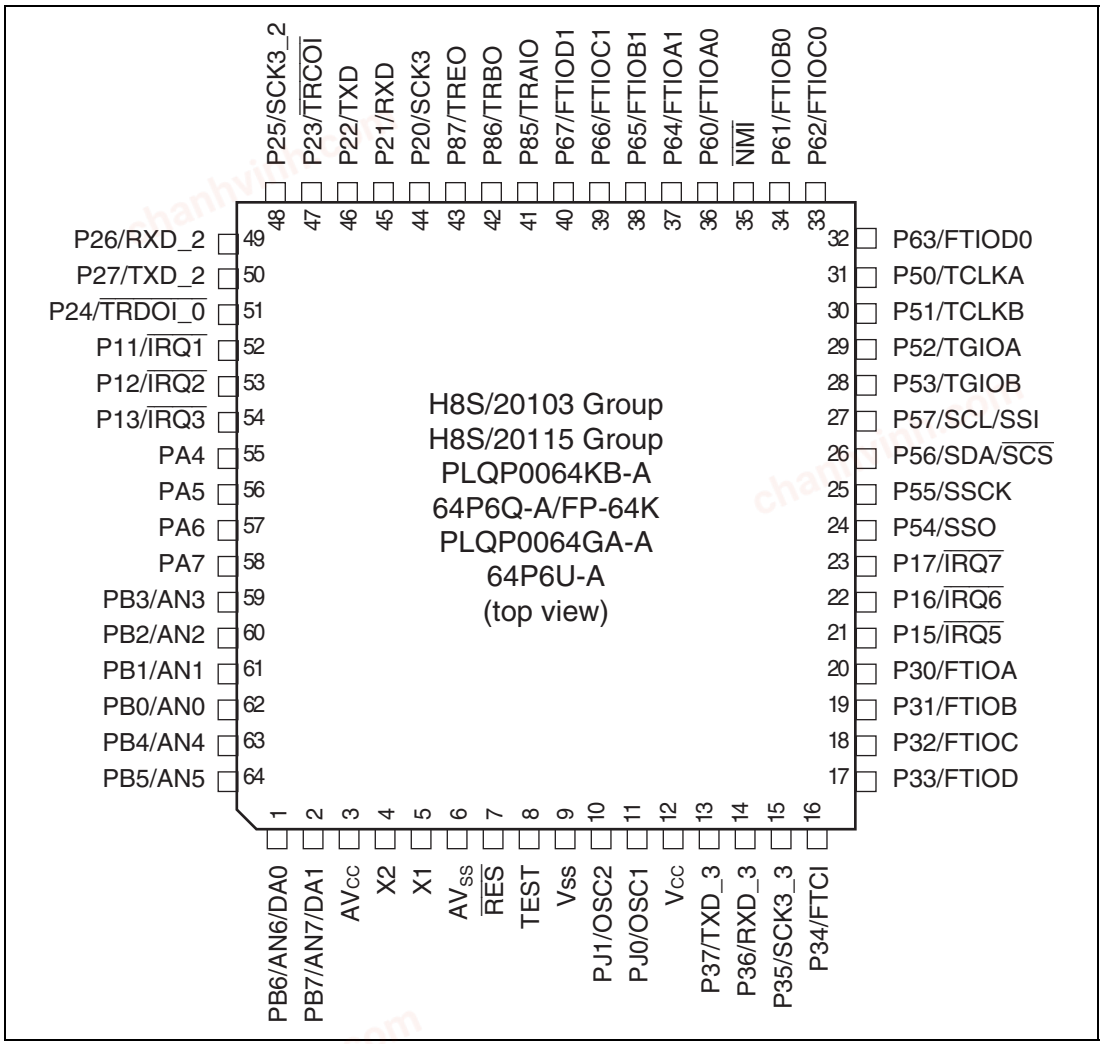
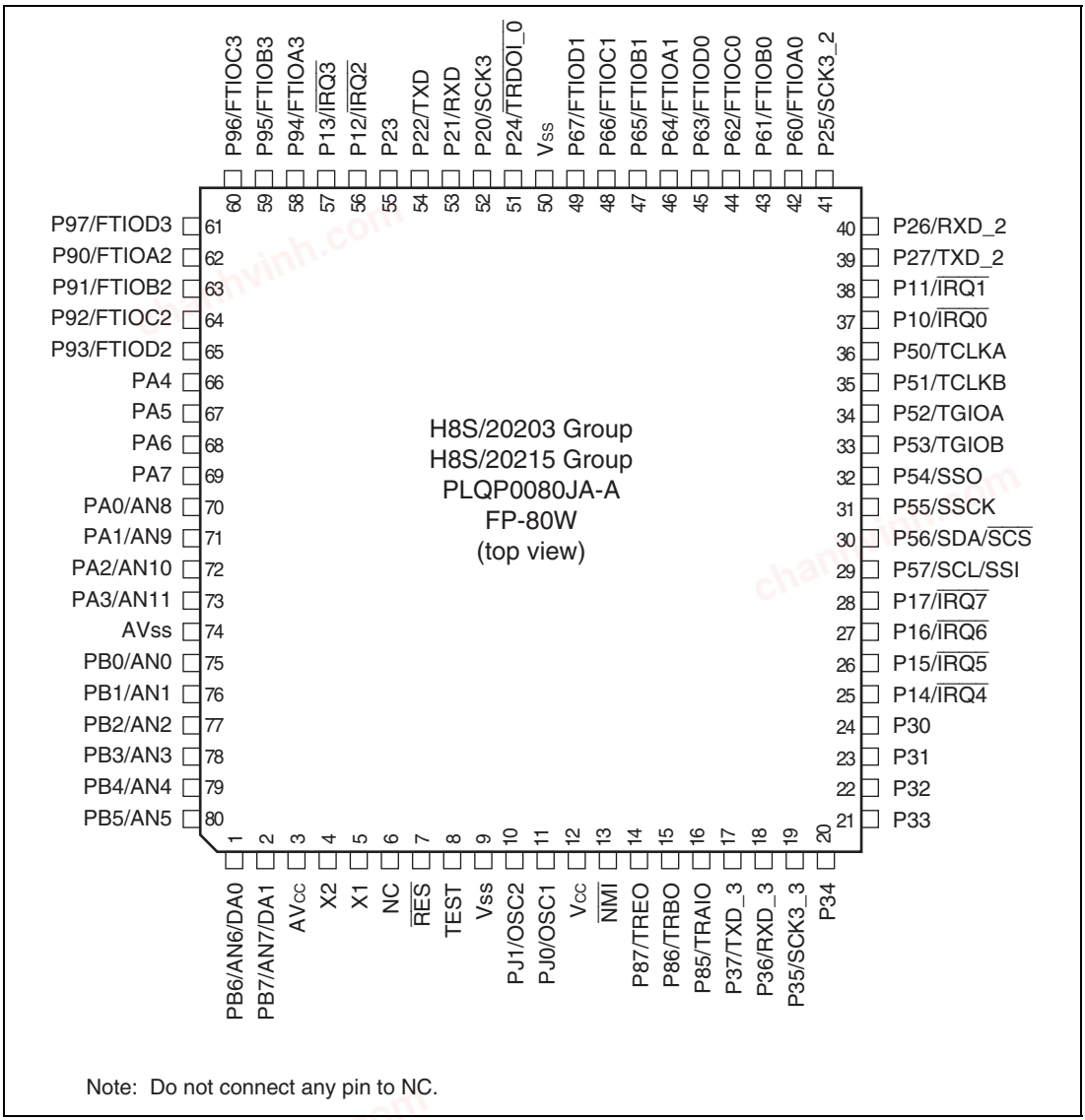



**Figure 1.5 Pin Assignment of the H8S/20103 and H8S/20115 Groups**

chanhvinh.com



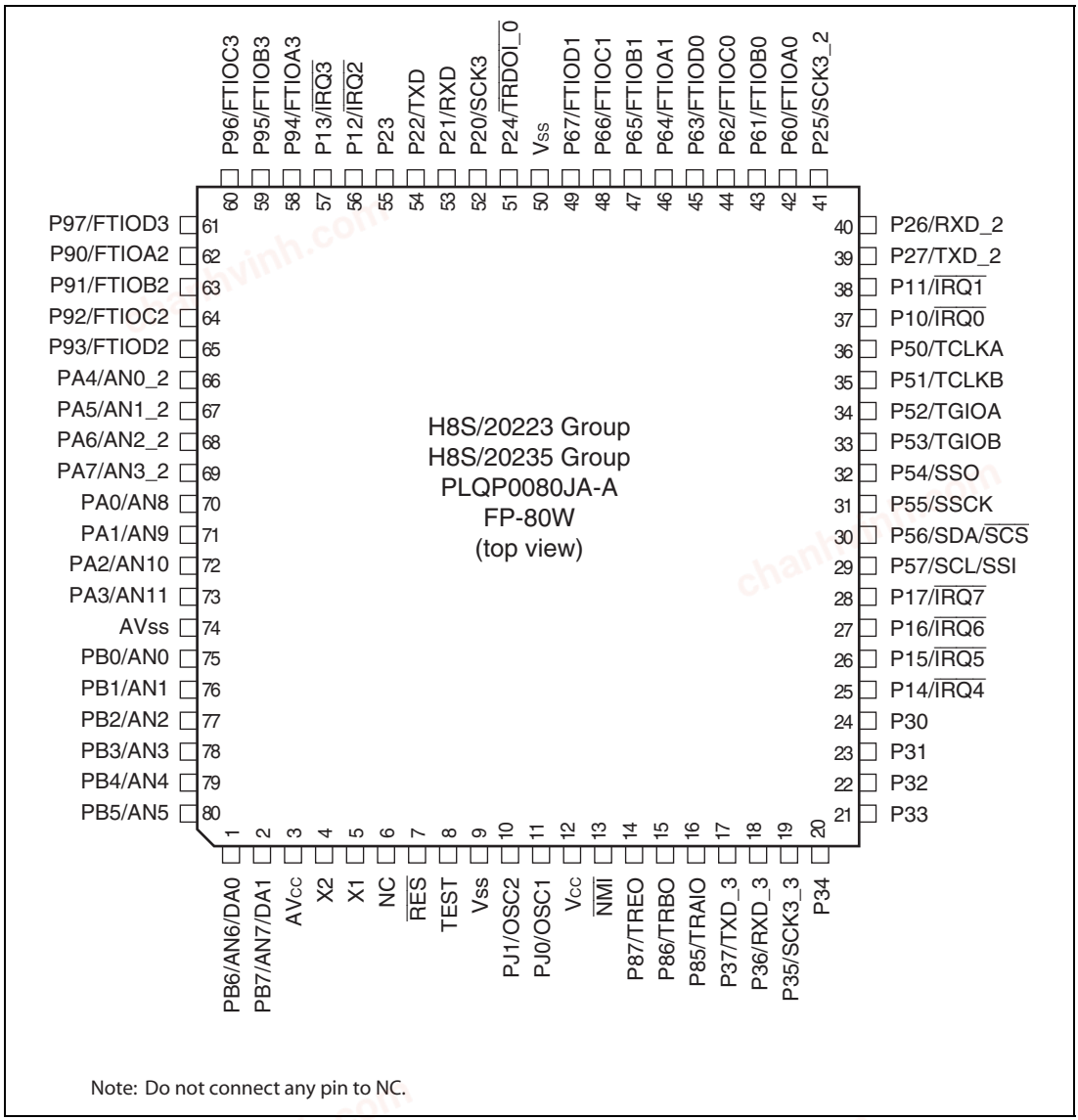



Note: Do not connect any pin to NC.

**Figure 1.6 Pin Assignment of the H8S/20203 and H8S/20215 Groups**

chanhvinh.com






Note: Do not connect any pin to NC.

**Figure 1.7 Pin Assignment of the H8S/20223 and H8S/20235 Groups**

chanhvinh.com



### 1.4.1 Pin Functions

**Table 1.3 Pin Functions**

| Classification | Symbol | Pin No. H8S/20103 and H8S/20115 Groups | Pin No. H8S/20203, H8S/20223, H8S/20215, and H8S/20235 Groups | I/O | Description |
|---|---|---|---|---|---|
| Power supply | $V_{CC}$ | 12 | 12 | Input | Power supply pin. Connect this pin to the system power supply. |
| | $V_{SS}$ | 9 | 9, 50 | Input | Ground pin. Connect this pin to the system power supply (0 V). |
| | $AV_{CC}$ | 3 | 3 | Input | Analog power supply pin for A/D and D/A converters. When A/D and D/A converters are not used, connect this pin to the system power supply. |
| | $AV_{SS}$ | 6 | 74 | Input | Analog ground pin for A/D and D/A converters. Connect this pin to the system power supply (0 V). |
| Clock | OSC1 | 11 | 11 | Input | Pins to be connected to a crystal or ceramic resonator for the system clock. An external clock can also be input to OSC1. When the on-chip oscillator is not used, the system clock signal can be output from OSC2. For connection examples, see section 5, Clock Pulse Generator. |
| | OSC2/CLKOUT | 10 | 10 | Output | |

chanhvinh.com



| Classification | Symbol | Pin No. H8S/20103 and H8S/20115 Groups | Pin No. H8S/20203, H8S/20223, H8S/20215, and H8S/20235 Groups | I/O | Description |
|---|---|---|---|---|---|
| Clock | X1 | 5 | 5 | Input | Pins to be connected to a crystal resonator for the 32.768-kHz sub-clock. For connection examples, see section 5, Clock Pulse Generator. |
| | X2 | 4 | 4 | Output | |
| System control | $\overline{RES}$ | 7 | 7 | Input | Reset pin. Applying a low level signal to this pin resets this LSI. |
| | TEST | 8 | 8 | Input | Test pin. Connect this pin to $V_{SS}$. |
| External interrupt | $\overline{NMI}$ | 35 | 13 | Input | Non-maskable interrupt request input pin. Be sure to pull up this pin with a resistor. |
| | $\overline{IRQ0}$ to $\overline{IRQ7}$ | 52 to 54*1<br>21 to 23 | 37, 38<br>56, 57<br>25 to 28 | Input | External interrupt request input pins. Either rising, falling, or rising/falling edge of these pins can be detected. |
| Timer RA | TRAIO | 41 | 16 | I/O | Pin for pulse output, count source input, and input of pulses to be measured. |
| | TRAO | *2 | *2 | Output | Pin for inverted pulse output. |
| Timer RB | TRGB | *2 | *2 | Input | Pin for trigger input. |
| | TRBO | 42 | 15 | Output | Pin for pulse output and PWM output. |
| Timer RC*3 | FTCI | 16 | — | Input | Pin for external event input. |
| | FTIOA to FTIOD | 20 to 17 | — | I/O | Pins for output-compare output, input-capture input, and PWM output. |

chanhvinh.com



| Classification | Symbol | Pin No. H8S/20103 and H8S/20115 Groups | Pin No. H8S/20203, H8S/20223, H8S/20215, and H8S/20235 Groups | I/O | Description |
|---|---|---|---|---|---|
| Timer RC*[3] | TRGC | 20 | — | Input | Pin for external trigger input. |
| | $\overline{\text{TRCOI}}$ | 47 | — | Input | Pin for inputting the timer-output enable or disable signal. |
| Timer RD_0 | FTIOA0 | 36 | 42 | I/O | Pin for output-compare output, input-capture input, and external clock input. |
| | FTIOB0 | 34 | 43 | I/O | Pin for output-compare output, input-capture input, and PWM output. |
| | FTIOC0 | 33 | 44 | I/O | Pin for output-compare output, input-capture input, and PWM synchronous output (at reset or in complementary PWM mode). |
| | FTIOD0 | 32 | 45 | I/O | Pin for output-compare output, input-capture input, and PWM output. |
| | FTIOA1 | 37 | 46 | I/O | Pin for output-compare output, input-capture input, and PWM output (at reset or in complementary PWM mode). |
| | FTIOB1 to FTIOD1 | 38 to 40 | 47 to 49 | I/O | Pins for output-compare output, input-capture input, and PWM output. |
| | $\overline{\text{TRDOI_0}}$ | 51 | 51 | Input | Pin for inputting the timer-output enable or disable signal. |

chanhvinh.com



| Classification | Symbol | Pin No. H8S/20103 and H8S/20115 Groups | Pin No. H8S/20203, H8S/20223, H8S/20215, and H8S/20235 Groups | I/O | Description |
|---|---|---|---|---|---|
| Timer RD_1*[5] | FTIOA2 | — | 62 | I/O | Pin for output-compare output, input-capture input, and external clock input. |
| | FTIOB2 | — | 63 | I/O | Pin for output-compare output, input-capture input, and PWM output. |
| | FTIOC2 | — | 64 | I/O | Pin for output-compare output, input-capture input, and PWM synchronous output (at reset or in complementary PWM mode). |
| | FTIOD2 | — | 65 | I/O | Pin for output-compare output, input-capture input, and PWM output. |
| | FTIOA3 | — | 58 | I/O | Pin for output-compare output, input-capture input, and PWM output (at reset or in complementary PWM mode). |
| | FTIOB3 to FTIOD3 | — | 59 to 61 | I/O | Pins for output-compare output, input-capture input, and PWM output. |
| | $\overline{\text{TRDOI_1}}$ | — | *[4] | Input | Pin for inputting the timer-output enable or disable signal. |
| Timer RE | TREO | 43 | 14 | Output | Pin for clock signal output. |
| Timer RG | TCLKA | 31 | 36 | Input | Pins for external clock input. |
| | TCLKB | 30 | 35 | | |
| | TGIOA | 29 | 34 | I/O | Pins for output-compare output, input-capture input, and PWM output. |
| | TGIOB | 28 | 33 | | |

chanhvinh.com



| Classification | Symbol | Pin No. H8S/20103 and H8S/20115 Groups | Pin No. H8S/20203, H8S/20223, H8S/20215, and H8S/20235 Groups | I/O | Description |
|---|---|---|---|---|---|
| Serial communication interface 3 (SCI3) | TXD | 46 | 54 | Output | Output pins for data transmission. |
| | TXD_2 | 50 | 39 | | |
| | TXD_3 | 13 | 17 | | |
| | RXD | 45 | 53 | Input | Input pins for data reception. |
| | RXD_2 | 49 | 40 | | |
| | RXD_3 | 14 | 18 | | |
| | SCK3 | 44 | 52 | I/O | Input/output pins for clock signals. |
| | SCK3_2 | 48 | 41 | | |
| | SCK3_3 | 15 | 19 | | |
| $I^2C$ bus interface 2 (IIC2) | SDA | 26 | 30 | I/O | Input/output pin for $I^2C$ data. Bus can be directly driven by the NMOS open-drain output. When this pin is used, an external pull-up resistor is required. |
| | SCL | 27 | 29 | I/O | Input/output pin for $I^2C$ clock signal. Bus can be directly driven by the NMOS open-drain output. When this pin is used, an external pull-up resistor is required. |
| Synchronous serial communication unit (SSU) | $\overline{SCS}$ | 26 | 30 | I/O | Input/output pin for the chip select signal. |
| | SSCK | 25 | 31 | I/O | Input/output pin for the clock signal. |
| | SSI | 27 | 29 | I/O | Input/output pin for data transmission/reception. |
| | SSO | 24 | 32 | I/O | Input/output pin for data transmission/reception. |

chanhvinh.com



| Classification | Symbol | Pin No. H8S/20103 and H8S/20115 Groups | Pin No. H8S/20203, H8S/20223, H8S/20215, and H8S/20235 Groups | I/O | Description |
|---|---|---|---|---|---|
| AD converter_1 | AN11 to AN0*[6] | 2, 1, 64, 63, 59 to 62 | 73 to 70, 2, 1 80 to 75 | Input | Analog input pins. |
| | $\overline{\text{ADTRG1}}$ | *[7] | *[7] | Input | Input pin for the conversion-start trigger signal. |
| AD converter_2*[8] | AN3_2 to AN0_2 | — | 69 to 66 | Input | Analog input pins. |
| | $\overline{\text{ADTRG2}}$ | — | *[7] | Input | Input pin for the conversion-start trigger signal. |
| DA converter | DA1 | 2 | 2 | Output | Analog output pins. |
| | DA0 | 1 | 1 | | |
| I/O ports | P17 to P10*[9] | 23 to 21, 54 to 52 | 28 to 25, 57, 56, 38, 37 | I/O | 8-bit input/output port pins. |
| | P27 to P20 | 50 to 48, 51, 47 to 44 | 39 to 41, 51, 55 to 52 | I/O | 8-bit input/output port pins. |
| | P37 to P30 | 13 to 20 | 17 to 24 | I/O | 8-bit input/output port pins. |
| | P57 to P50 | 27 to 24 28 to 31 | 29 to 36 | I/O | 8-bit input/output port pins. |
| | P67 to P60 | 40 to 37, 32 to 34, 36 | 49 to 42 | I/O | 8-bit input/output port pins. |
| | P87 to P85 | 43 to 41 | 14 to 16 | I/O | 3-bit input/output port pins. |
| | P97 to P90*[10] | — | 61 to 58 65 to 62 | I/O | 8-bit input/output port pins. |
| | PA7 to PA0*[11] | 58 to 55 | 69 to 66 73 to 70 | I/O | 8-bit input/output port pins. |
| | PB7 to PB0 | 2, 1, 64, 63, 59 to 62 | 2, 1, 80 to 75 | I/O | 8-bit input/output port pins. |
| | PJ1 and PJ0 | 10, 11 | 10, 11 | I/O | 2-bit input/output port pins. |

chanhvinh.com



Notes:
1. In the H8S/20103 and H8S/20115 Groups, the $\overline{\text{IRQ0}}$ and $\overline{\text{IRQ4}}$ pins are not available with the initial setting of the PMC.
2. The TRAO and TRGB pins are not available with the initial setting of the PMC.
3. The H8S/20203, H8S/20223, H8S/20215, and H8S/20235 Groups do not incorporate timer RC.
4. The $\overline{\text{TRDOI_1}}$ pin is not available with the initial setting of the PMC.
5. The H8S/20103 and H8S/20115 Groups do not incorporate timer RD_1.
6. In the H8S/20103 and H8S/20115 Groups, AN8 to AN11 are not available.
7. The $\overline{\text{ADTRG1}}$ and $\overline{\text{ADTRG2}}$ functions are not available due to the initial setting of the PMC.
8. The H8S/20103, H8S/20203, H8S/20115, and H8S/20215 Groups do not incorporate A/D converter_2.
9. The H8S/20103 and H8S/20115 Groups do not provide P14 or P10.
10. The H8S/20103 and H8S/20115 Groups do not provide P97 to P90.
11. The H8S/20103 and H8S/20115 Groups do not provide PA3 to PA0.

chanhvinh.com



# Section 2 CPU

The H8S/2000 CPU is a high-speed central processing unit with an internal 32-bit architecture that is upward-compatible with the H8/300 and H8/300H CPUs. The H8S/2000 CPU has sixteen 16-bit general registers, can address a 16-Mbyte linear address space, and is ideal for realtime control.

This section describes the H8S/2000 CPU.

## 2.1 Features

- Upward-compatibility with H8/300 and H8/300H CPUs
  Can execute H8/300 and H8/300H CPU object programs
- General-register architecture
  Sixteen 16-bit general registers also usable as sixteen 8-bit registers or eight 32-bit registers
- Sixty-five basic instructions
  - 8/16/32-bit arithmetic and logic instructions
  - Multiply and divide instructions
  - Powerful bit-manipulation instructions
- Eight addressing modes
  - Register direct [Rn]
  - Register indirect [@ERn]
  - Register indirect with displacement [@(d:16,ERn) or @(d:32,ERn)]
  - Register indirect with post-increment or pre-decrement [@ERn+ or @–ERn]
  - Absolute address [@aa:8, @aa:16, @aa:24, or @aa:32]
  - Immediate [#xx:8, #xx:16, or #xx:32]
  - Program-counter relative [@(d:8,PC) or @(d:16,PC)]
  - Memory indirect [@@aa:8]
- 16-Mbyte address space
  - Program: 16 Mbytes
  - Data: 16 Mbytes

chanhvinh.com



- High-speed operation
  All frequently-used instructions are executed in one or two states
  — 8/16/32-bit register-register add/subtract: 1 state
  — 8 × 8-bit register-register multiply: 12 states (MULXU.B), 13 states (MULXS.B)
  — 16 ÷ 8-bit register-register divide: 12 states (DIVXU.B)
  — 16 × 16-bit register-register multiply: 20 states (MULXU.W), 21 states (MULXS.W)
  — 32 ÷ 16-bit register-register divide: 20 states (DIVXU.W)
- Two CPU operating modes
  — Normal mode
  — Advanced mode
- Power-down state
  Transition to power-down state by SLEEP instruction
  Selectable CPU clock speed

### 2.1.1 Differences between H8S/2600 CPU and H8S/2000 CPU

The differences between the H8S/2600 CPU and the H8S/2000 CPU are as shown below.

- Register configuration
  The MAC register is supported only by the H8S/2600 CPU.
- Basic instructions
  The four instructions MAC, CLRMAC, LDMAC, and STMAC are supported only by the H8S/2600 CPU.
- The number of execution states of the MULXU and MULXS instructions

| | | Execution States | |
|---|---|---|---|
| **Instruction** | **Mnemonic** | **H8S/2600** | **H8S/2000** |
| MULXU | MULXU.B Rs, Rd | 3 | 12 |
| | MULXU.W Rs, ERd | 4 | 20 |
| MULXS | MULXS.B Rs, Rd | 4 | 13 |
| | MULXS.W Rs, ERd | 5 | 21 |

In addition, there are differences in address space, CCR and EXR register functions, power-down modes, etc., depending on the model.

chanhvinh.com



### 2.1.2 Differences from H8/300 CPU

In comparison to the H8/300 CPU, the H8S/2000 CPU has the following enhancements.

- More general registers and control registers
  Eight 16-bit extended registers, and one 8-bit and two 32-bit control registers, have been added.
- Expanded address space
  Normal mode supports the same 64-Kbyte address space as the H8/300 CPU.
  Advanced mode supports a maximum 16-Mbyte address space.
- Enhanced addressing
  The addressing modes have been enhanced to make effective use of the 16-Mbyte address space.
- Enhanced instructions
  Addressing modes of bit-manipulation instructions have been enhanced.
  Signed multiply and divide instructions have been added.
  Two-bit shift and two-bit rotate instructions have been added.
  Instructions for saving and restoring multiple registers have been added.
  A test and set instruction has been added.
- Higher speed
  Basic instructions are executed twice as fast.

### 2.1.3 Differences from H8/300H CPU

In comparison to the H8/300H CPU, the H8S/2000 CPU has the following enhancements.

- Additional control register
  One 8-bit control register has been added.
- Enhanced instructions
  Addressing modes of bit-manipulation instructions have been enhanced.
  Two-bit shift and two-bit rotate instructions have been added.
  Instructions for saving and restoring multiple registers have been added.
  A test and set instruction has been added.
- Higher speed
  Basic instructions are executed twice as fast.

chanhvinh.com



## 2.2 CPU Operating Modes

The H8S/2000 CPU has two operating modes: normal and advanced. Note that this LSI supports only advanced mode. Advanced mode supports a maximum 16-Mbyte address space.

### 2.2.1 Advanced Mode

- Address space

  Linear access to a maximum address space of 16 Mbytes is possible.
- Extended registers (En)

  The extended registers (E0 to E7) can be used as 16-bit registers. They can also be used as the upper 16-bit segments of 32-bit registers or address registers.
- Instruction set

  All instructions and addressing modes can be used.
- Exception vector table and memory indirect branch addresses

  In advanced mode, the top area starting at H'00000000 is allocated to the exception vector table in 32-bit units. In each 32 bits, the upper 8 bits are ignored and a branch address is stored in the lower 24 bits (see figure 2.1). For details of the exception vector table, see section 3, Exception Handling.

chanhvinh.com



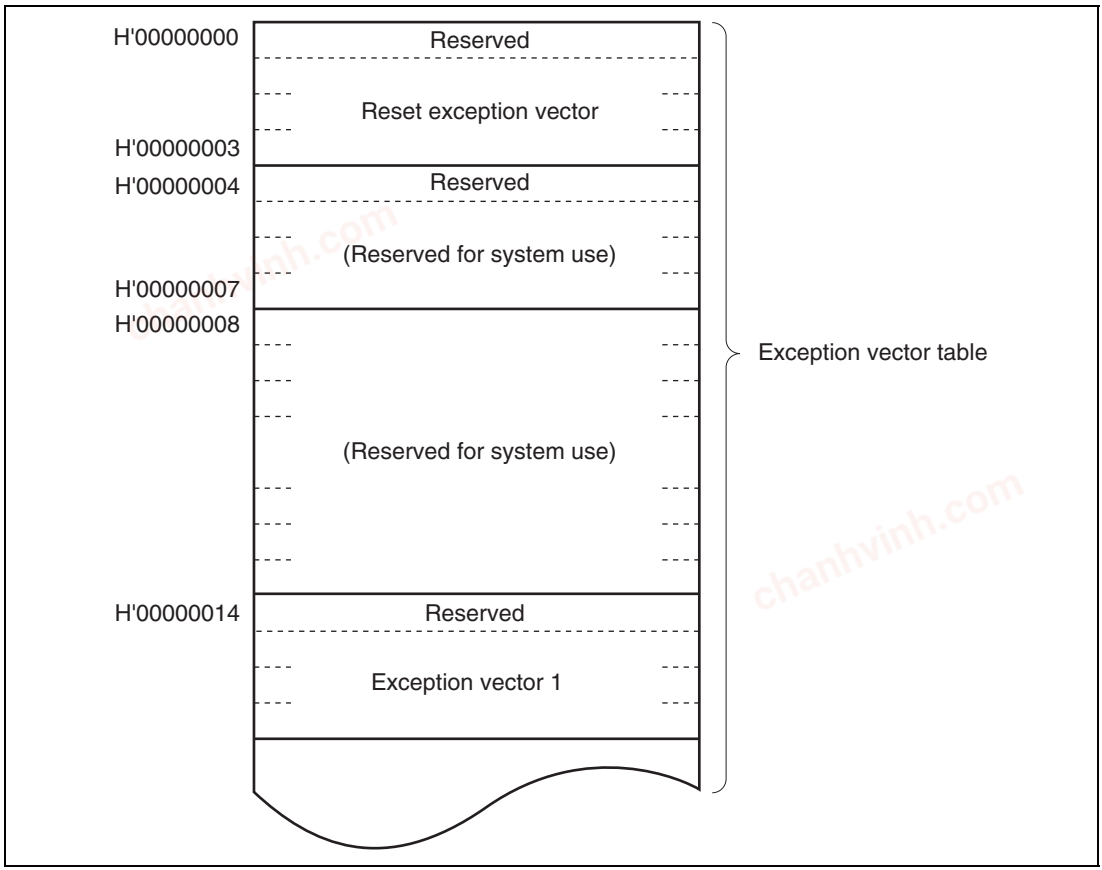



**Figure 2.1 Exception Vector Table (Advanced Mode)**

The memory indirect addressing mode (@@aa:8) employed in the JMP and JSR instructions uses an 8-bit absolute address included in the instruction code to specify a memory operand that contains a branch address. In advanced mode, the operand is a 32-bit longword operand, providing a 32-bit branch address. The upper 8 bits of these 32 bits are a reserved area that is regarded as H'00. Branch addresses can be stored in the area from H'00000000 to H'000000FF. Note that the top area of this range is also used for the exception vector table.

chanhvinh.com



- Stack structure

  In advanced mode, the program counter (PC) is pushed onto the stack in a subroutine call, and the PC and condition-code register (CCR) are pushed onto the stack in exception handling. They are stored as shown in figure 2.2. EXR is not pushed onto the stack in interrupt control mode 0. For details, see section 3, Exception Handling.

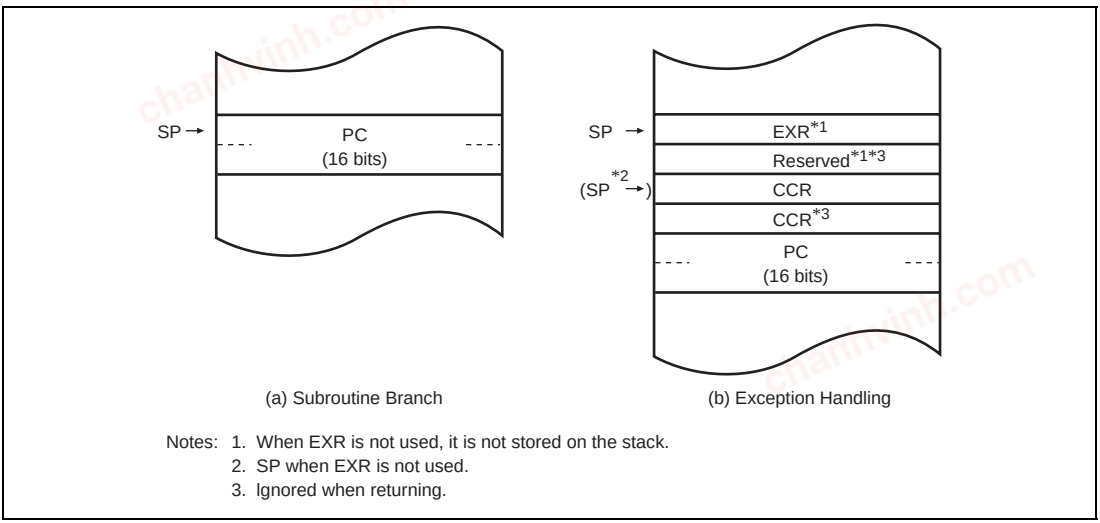



**Figure 2.2 Stack Structure in Advanced Mode**

## 2.3 Address Space

Figure 2.3 shows a memory map of the H8S/2000 CPU. The H8S/2000 CPU provides linear access to a maximum 16-Mbyte (architecturally 4-Gbyte) address space in advanced mode. The usable modes and address spaces differ depending on the product.

chanhvinh.com



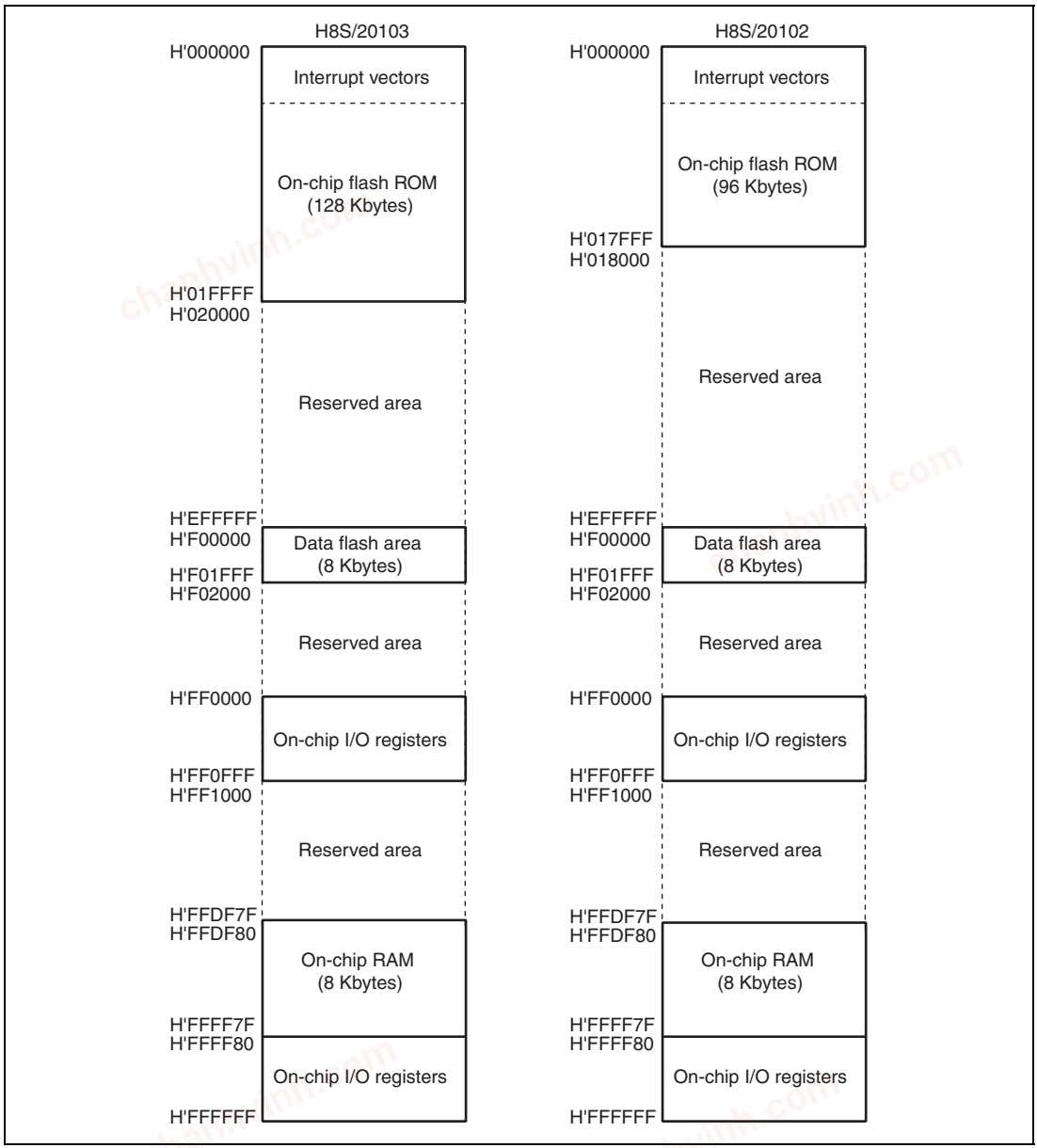



**Figure 2.3 Memory Map (1) (H8S/20103 Group)**

chanhvinh.com



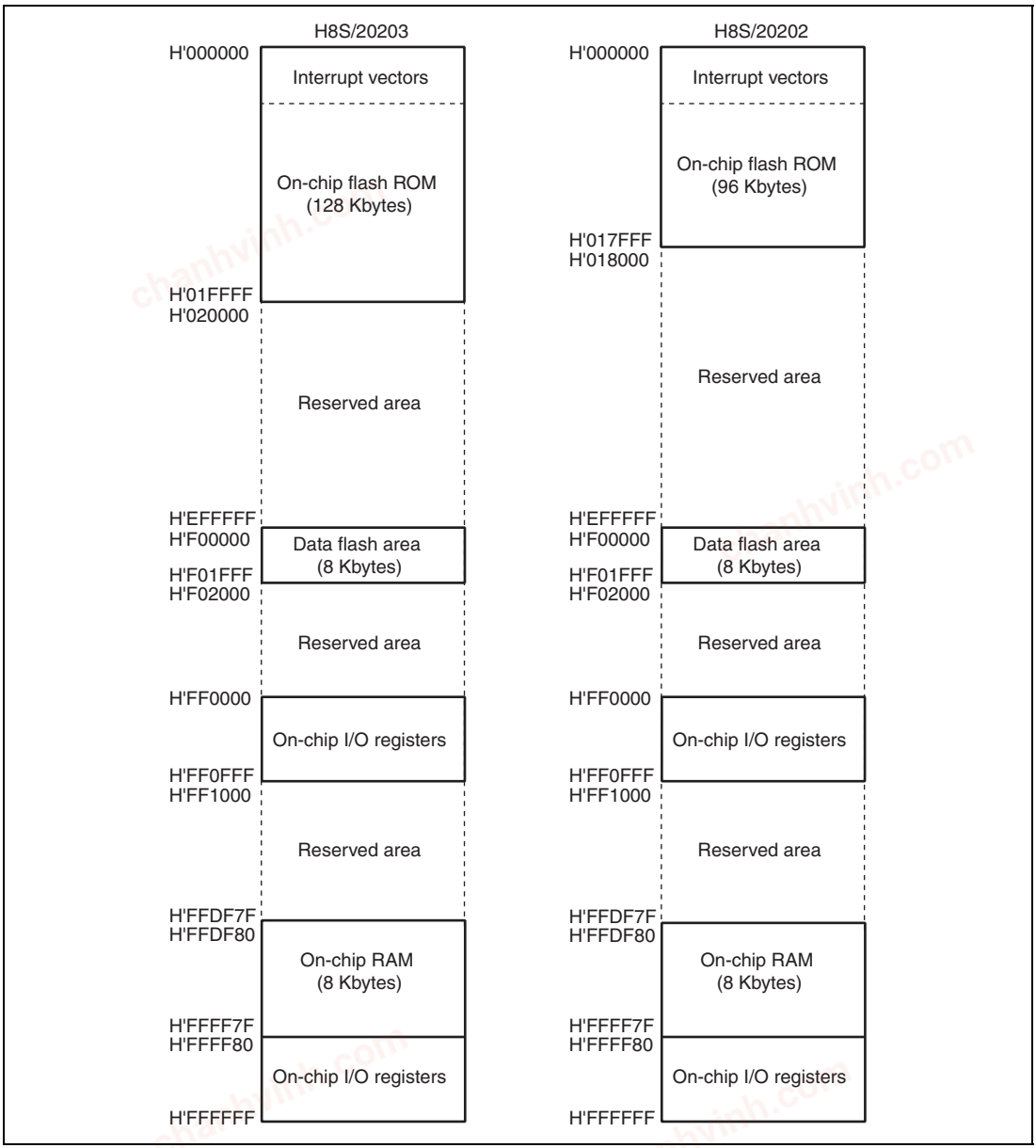



**Figure 2.3 Memory Map (2) (H8S/20203 Group)**

chanhvinh.com



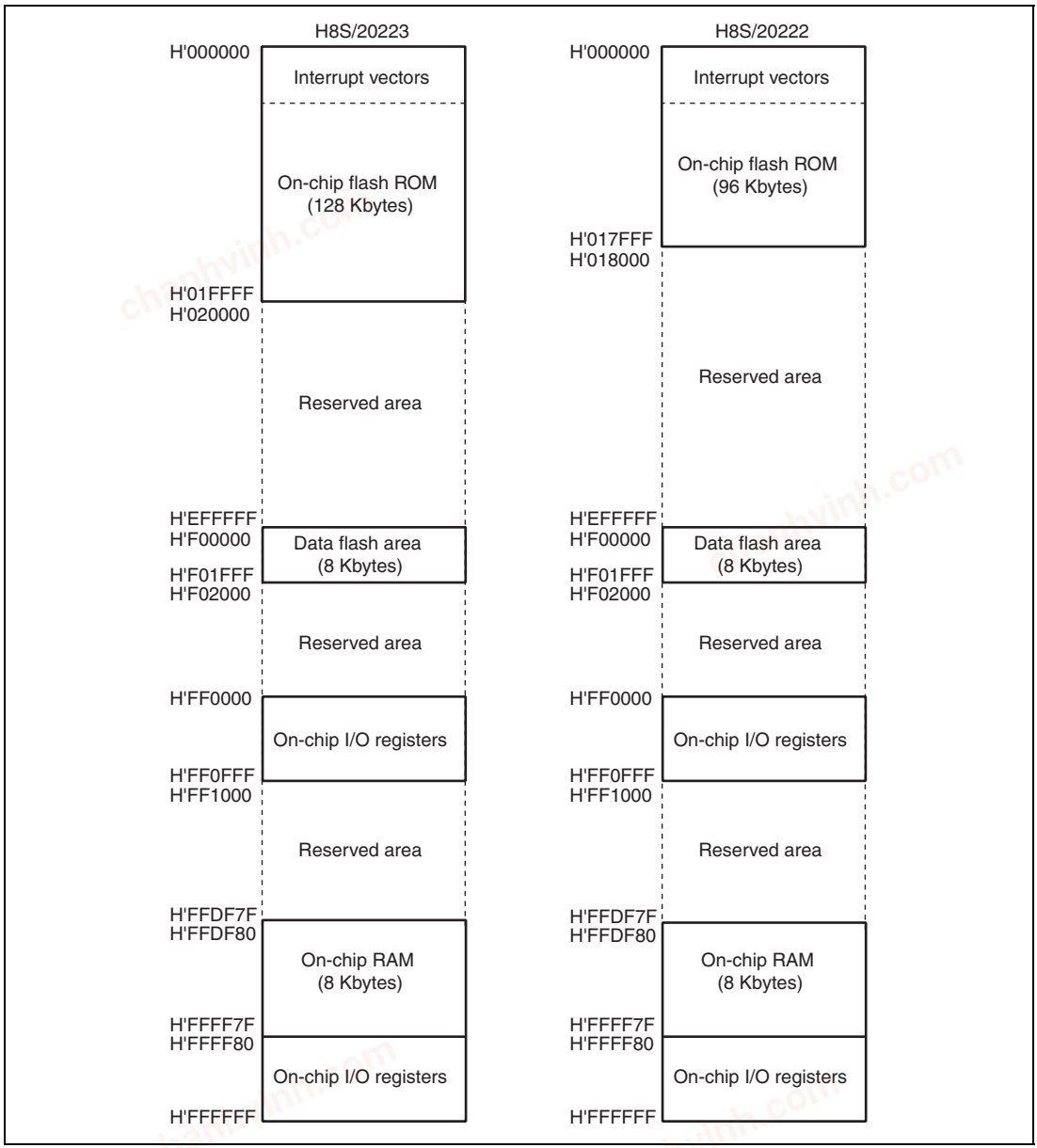



**Figure 2.3 Memory Map (3) (H8S/20223 Group)**

chanhvinh.com



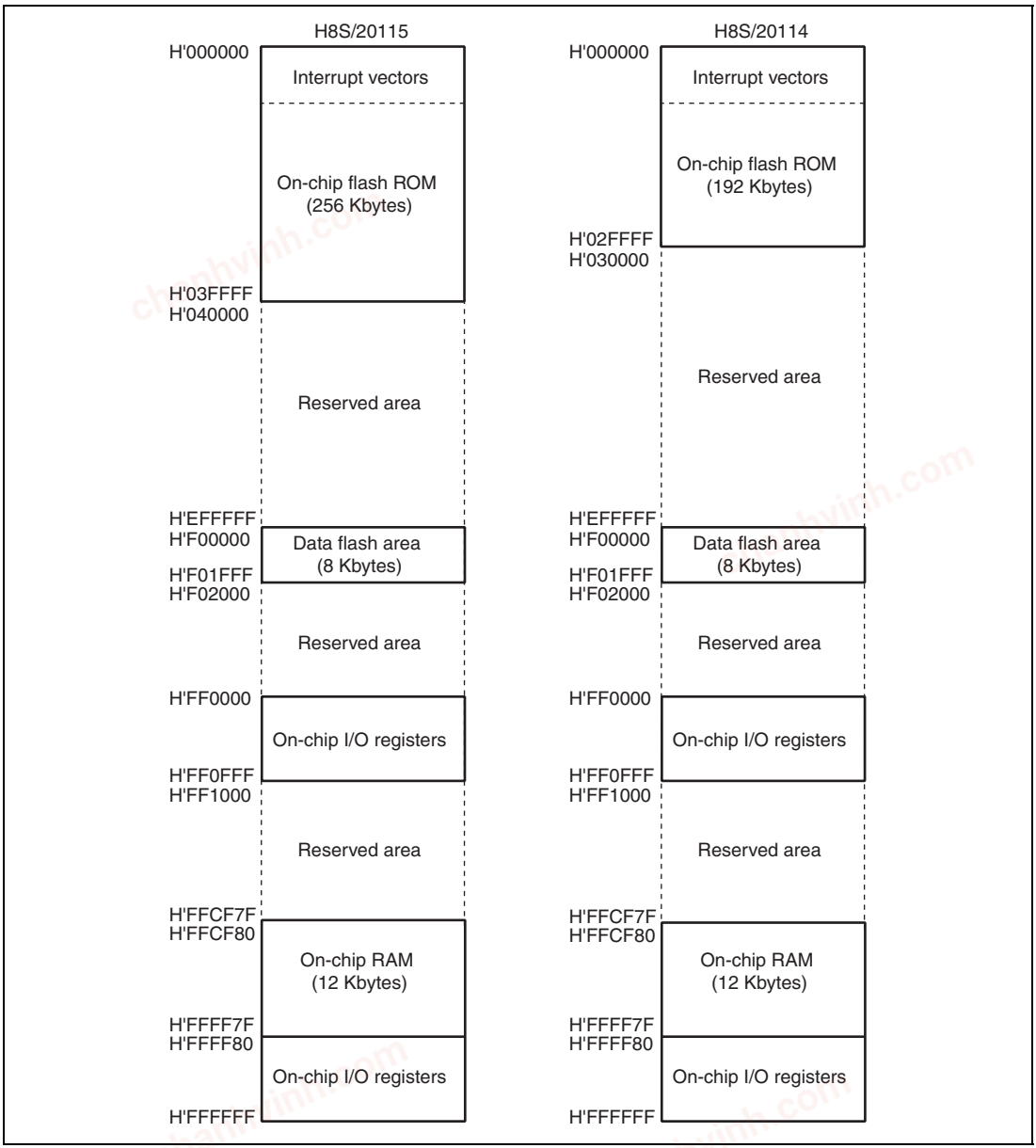



**Figure 2.3 Memory Map (4) (H8S/20115 Group)**

chanhvinh.com



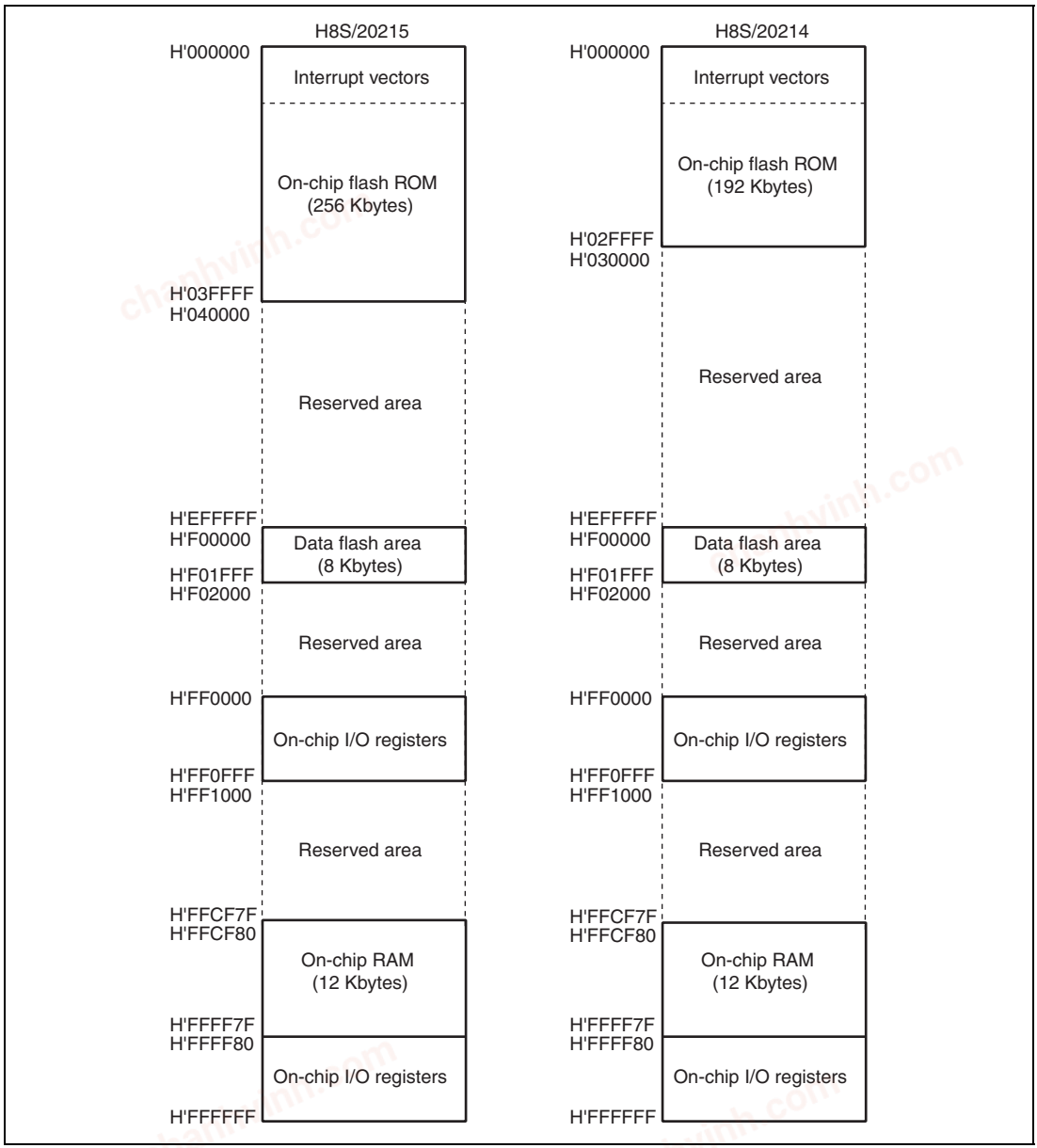



**Figure 2.3 Memory Map (5) (H8S/20215 Group)**

chanhvinh.com



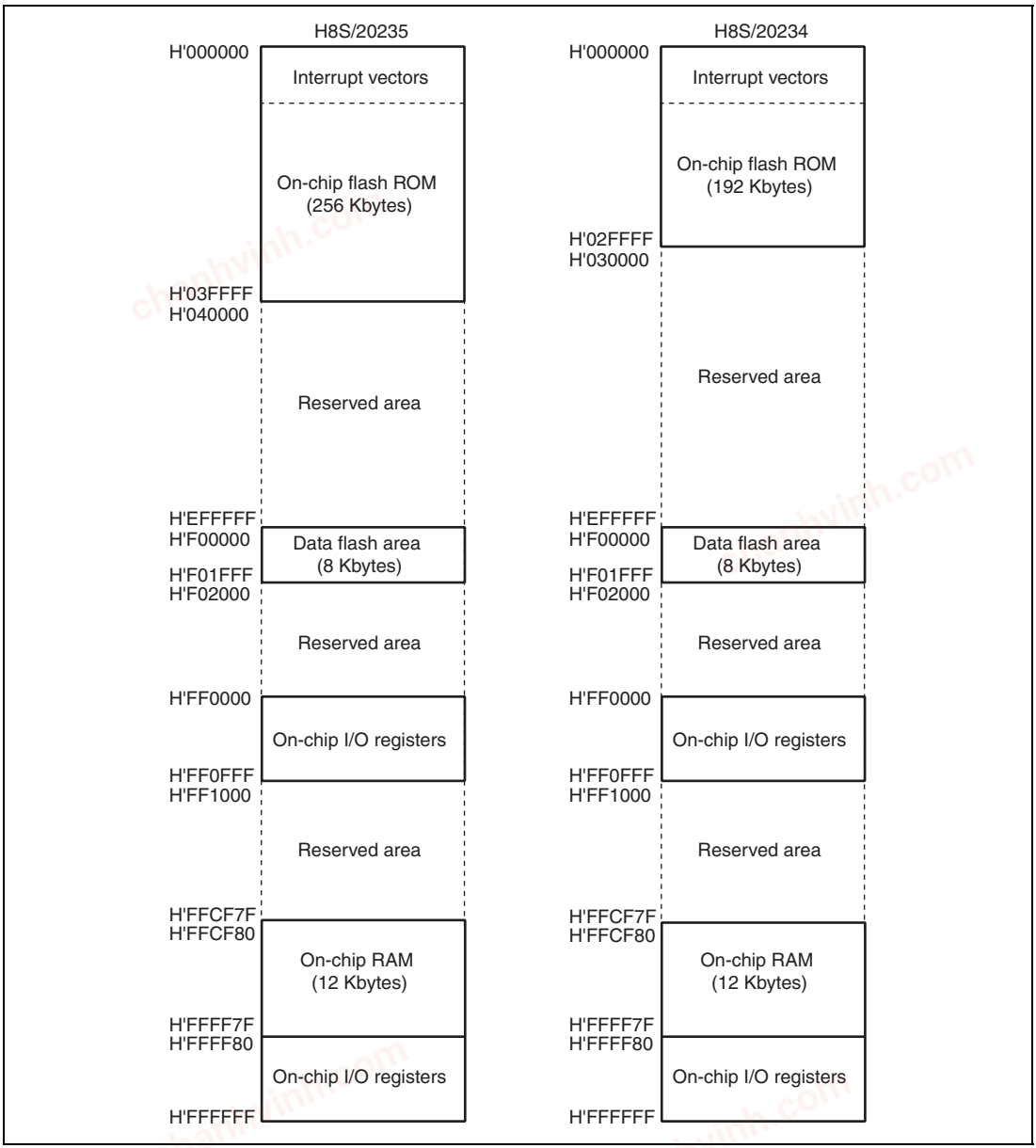



**Figure 2.3 Memory Map (6) (H8S/20235 Group)**

chanhvinh.com



## 2.4 Register Configuration

The H8S/2000 CPU has the internal registers shown in figure 2.4. There are two types of registers: general registers and control registers. Control registers are a 24-bit program counter (PC), an 8-bit extended control register (EXR), and an 8-bit condition code register (CCR).

General Registers (Rn) and Extended Registers (En)

| | 15 0 | 7 0 | 7 0 |
|---|---|---|---|
| ER0 | E0 | R0H | R0L |
| ER1 | E1 | R1H | R1L |
| ER2 | E2 | R2H | R2L |
| ER3 | E3 | R3H | R3L |
| ER4 | E4 | R4H | R4L |
| ER5 | E5 | R5H | R5L |
| ER6 | E6 | R6H | R6L |
| ER7 (SP) | E7 | R7H | R7L |

Control Registers

| 23 0 |
|---|
| PC |

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| EXR | T | - | - | - | - | I2 | I1 | I0 |

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| CCR | I | UI | H | U | N | Z | V | C |

Legend:

SP : Stack pointer
PC : Program counter
EXR : Extended control register
T : Trace bit
I2 to I0 : Interrupt mask bits
CCR : Condition-code register
I : Interrupt mask bit
UI : User bit or interrupt mask bit*
H : Half-carry flag
U : User bit
N : Negative flag
Z : Zero flag
V : Overflow flag
C : Carry flag

Note: * For this LSI, the interrupt mask bit is not available.

**Figure 2.4 CPU Internal Registers**

chanhvinh.com



### 2.4.1 General Registers

The H8S/2000 CPU has eight 32-bit general registers. These general registers are all functionally alike and can be used as both address registers and data registers. When a general register is used as a data register, it can be accessed as a 32-bit, 16-bit, or 8-bit register. Figure 2.5 illustrates the usage of the general registers. When the general registers are used as 32-bit registers or address registers, they are designated by the letters ER (ER0 to ER7).

When the general registers are used as 16-bit registers, the ER registers are divided into 16-bit general registers designated by the letters E (E0 to E7) and R (R0 to R7). These registers are functionally equivalent, providing a maximum sixteen 16-bit registers. The E registers (E0 to E7) are also referred to as extended registers.

When the general registers are used as 8-bit registers, the R registers are divided into 8-bit general registers designated by the letters RH (R0H to R7H) and RL (R0L to R7L). These registers are functionally equivalent, providing a maximum sixteen 8-bit registers.

The usage of each register can be selected independently.

General register ER7 has the function of the stack pointer (SP) in addition to its general-register function, and is used implicitly in exception handling and subroutine calls. Figure 2.6 shows the stack.

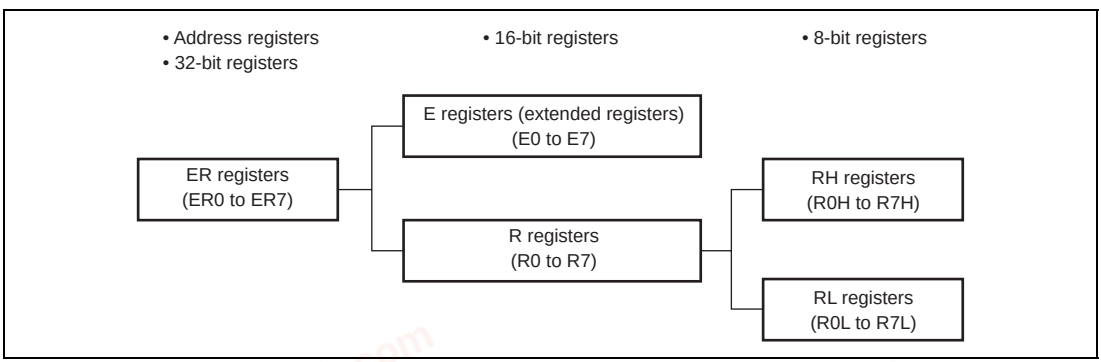



**Figure 2.5 Usage of General Registers**

chanhvinh.com



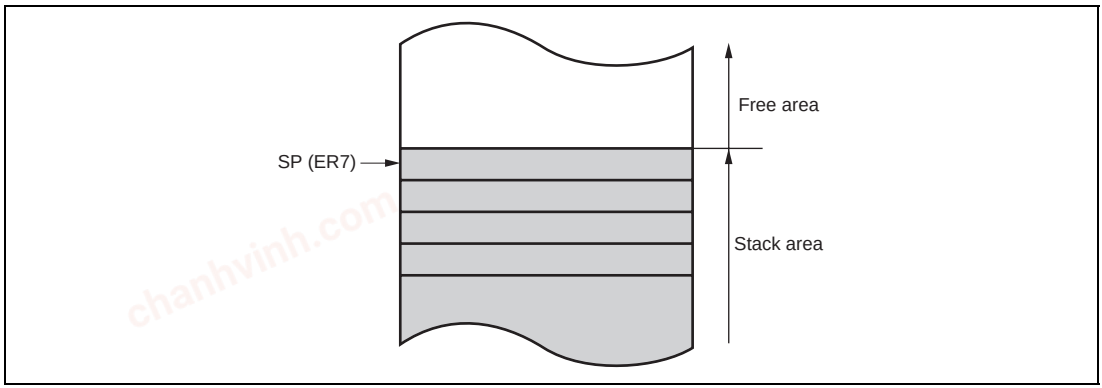



**Figure 2.6 Stack**

### 2.4.2 Program Counter (PC)

This 24-bit counter indicates the address of the next instruction the CPU will execute. The length of all CPU instructions is 2 bytes (one word), so the least significant PC bit is ignored. (When an instruction is fetched for read, the least significant PC bit is regarded as 0.)

### 2.4.3 Extended Control Register (EXR)

EXR is an 8-bit register that can be operated by the LDC, STC, ANDC, ORC, and XORC instructions. When an instruction other than STC is executed, all interrupts including NMI are masked in three states after the instruction is completed.

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | T | Trace bit | 0: Consecutively executes instructions.<br>1: Starts trace exception processing each time an instruction is executed. | R/W |
| 6 to 3 | — | Reserved | These bits are always read as 1. | — |
| 2 to 0 | I2*<br>I1<br>I0 | Interrupt request mask level 2 to 0 | These bits specify interrupt request mask levels (0 to 3). For details, see section 4, Interrupt Controller. | R/W |

Note: * The I2-bit is reserved in this product. The I2 bit is set to 1 if an interrupt is accepted, but this does not affect the mask level for interrupt requests.

chanhvinh.com



### 2.4.4 Condition-Code Register (CCR)

This 8-bit register contains internal CPU status information, including an interrupt mask bit (I) and half-carry (H), negative (N), zero (Z), overflow (V), and carry (C) flags.

Operations can be performed on the CCR bits by the LDC, STC, ANDC, ORC, and XORC instructions. The N, Z, V, and C flags are used as branching conditions for conditional branch (Bcc) instructions.

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | I | Interrupt mask bit | 0: Does not mask interrupts.<br>1: Masks interrupts. | R/W |
| 6 | UI | User bit or interrupt mask bit | This bit does not affect this LSI operation. | R/W |
| 5 | H | Half-carry flag | [Setting conditions]<br>• If there is a carry or borrow bit 3 when the ADD.B, ADDX.B, SUB.B, SUBX.B, CMP.B or NEG.B instruction is executed.<br>• If there is a carry or borrow at bit 11 when the ADD.W, SUB.W, CMP.W, or NEG.W instruction is executed.<br>• If there is a carry or borrow at bit 27 when the ADD.L, SUB.L, CMP.L, or NEG.L instruction is executed.<br>[Clearing condition]<br>When none of the above setting conditions are satisfied. | R/W |
| 4 | U | User bit | This bit does not affect the LSI operation. | R/W |
| 3 | N | Negative flag | [Setting condition]<br>When the execution result is negative.<br>[Clearing condition]<br>When the execution result is not negative. | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 2 | Z | Zero flag | [Setting condition]<br>When data is zero.<br>[Clearing condition]<br>When data is not zero. | R/W |
| 1 | V | Overflow flag | [Setting condition]<br>When an overflow occurs after an arithmetic instruction has been executed.<br>[Clearing condition]<br>When no overflow occurs after an arithmetic instruction has been executed. | R/W |
| 0 | C | Carry flag | [Setting condition]<br>When a carry occurs after an instruction has been executed.<br>[Clearing condition]<br>When no carry occurs after an instruction has been executed. | R/W |

- I (interrupt mask bit)

  This bit masks interrupts other than NMI when set to 1. NMI is accepted regardless of the I bit setting. The I bit is set to 1 at the start of an exception-handling sequence. For details, see section 4, Interrupt Controller.

- UI (user bit/interrupt mask bit)

  This bit can be written to and read from by software using the LDC, STC, ANDC, ORC, and XORC instructions.

  For this LSI, interrupt mask bit is not available.

- H (half carry flag)

  When the ADD.B, ADDX.B, SUB.B, SUBX.B, CMP.B or NEG.B instruction is executed, this flag is set to 1 if there is a carry or borrow at bit 3, and cleared to 0 otherwise. When the ADD.W, SUB.W, CMP.W, or NEG.W instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 11, and cleared to 0 otherwise. When the ADD.L, SUB.L, CMP.L, or NEG.L instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 27, and cleared to 0 otherwise.

chanhvinh.com



- U (user bit)

  This bit can be written to and read from by software using the LDC, STC, ANDC, ORC, and XORC instructions.

- N (negative bit)

  This bit stores the value of the most significant bit of data as a sign bit.

- C (carry flag)

  This flag is set to 1 when a carry occurs, and cleared to 0 otherwise. Used by:

  — Add instructions, to indicate a carry

  — Subtract instructions, to indicate a borrow

  — Shift and rotate instructions, to indicate a carry

  The carry flag is also used as a bit accumulator by bit manipulation instructions.

### 2.4.5 Initial Register Values

Reset exception handling loads the CPU's program counter (PC) from the vector table, clears the trace (T) bit in EXR to 0, and sets the interrupt mask (I) bits in CCR and EXR to 1. The other CCR bits and the general registers are not initialized. Note that the stack pointer (ER7) is undefined. The stack pointer should therefore be initialized by an MOV.L instruction executed immediately after a reset.

chanhvinh.com



## 2.5 Data Formats

The H8S/2000 CPU can process 1-bit, 4-bit BCD, 8-bit (byte), 16-bit (word), and 32-bit (longword) data. Bit-manipulation instructions operate on 1-bit data by accessing bit n (n = 0, 1, 2, …, 7) of byte operand data. The DAA and DAS decimal-adjust instructions treat byte data as two digits of 4-bit BCD data.

### 2.5.1 General Register Data Formats

Figure 2.7 shows the data formats of general registers.

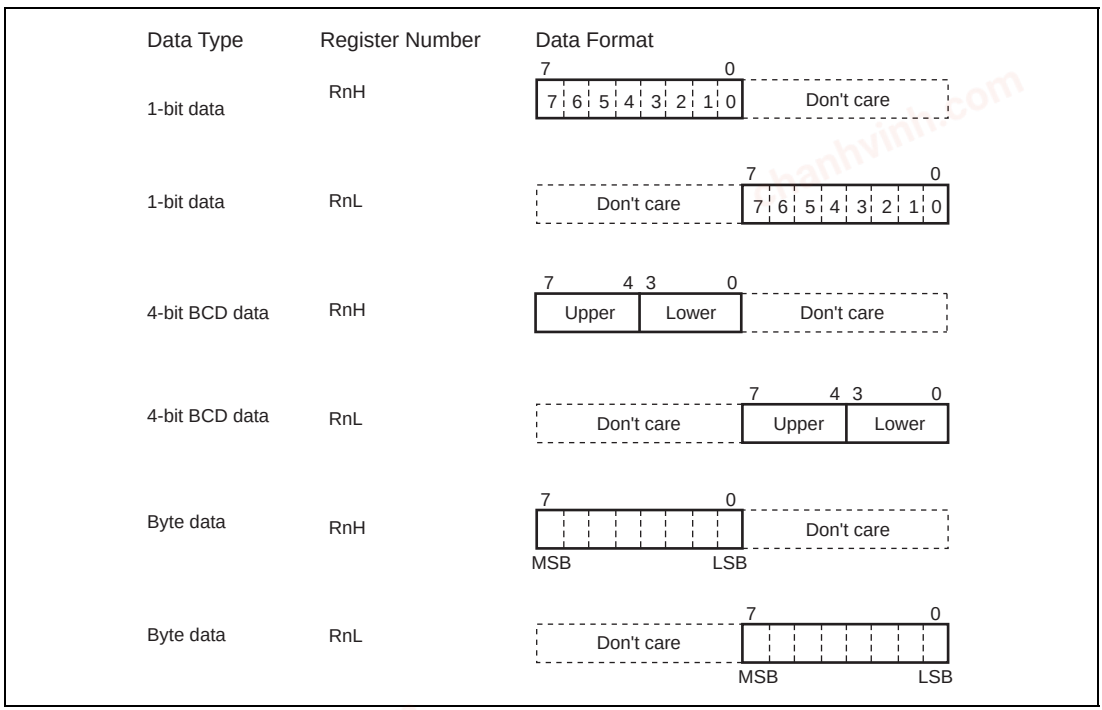



**Figure 2.7 General Register Data Formats (1)**

chanhvinh.com



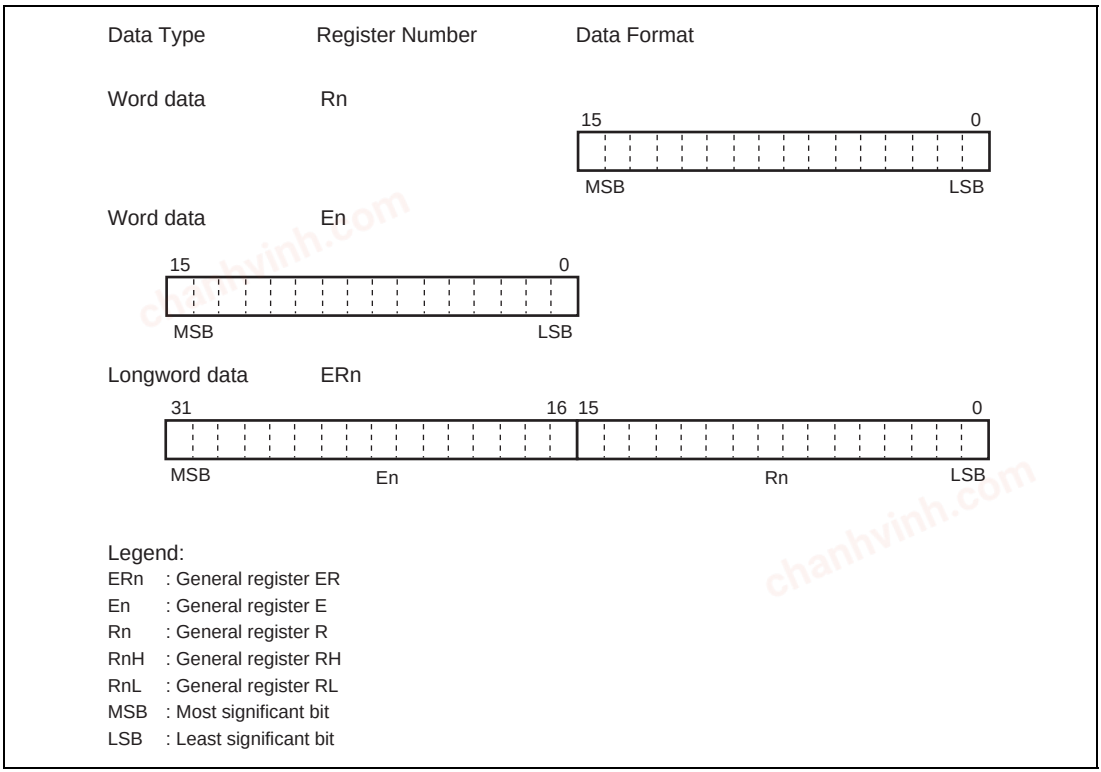



**Figure 2.7 General Register Data Formats (2)**

chanhvinh.com



### 2.5.2 Memory Data Formats

Figure 2.8 shows the data formats in memory. The H8S/2000 CPU can access word data and longword data in memory, but word or longword data must begin at an even address. If an attempt is made to access word or longword data at an odd address, no address error occurs but the least significant bit of the address is regarded as 0, so the access starts at the preceding address. This also applies to instruction fetches.

When SP (ER7) is used as an address register to access the stack, the operand size should be word size or longword size.

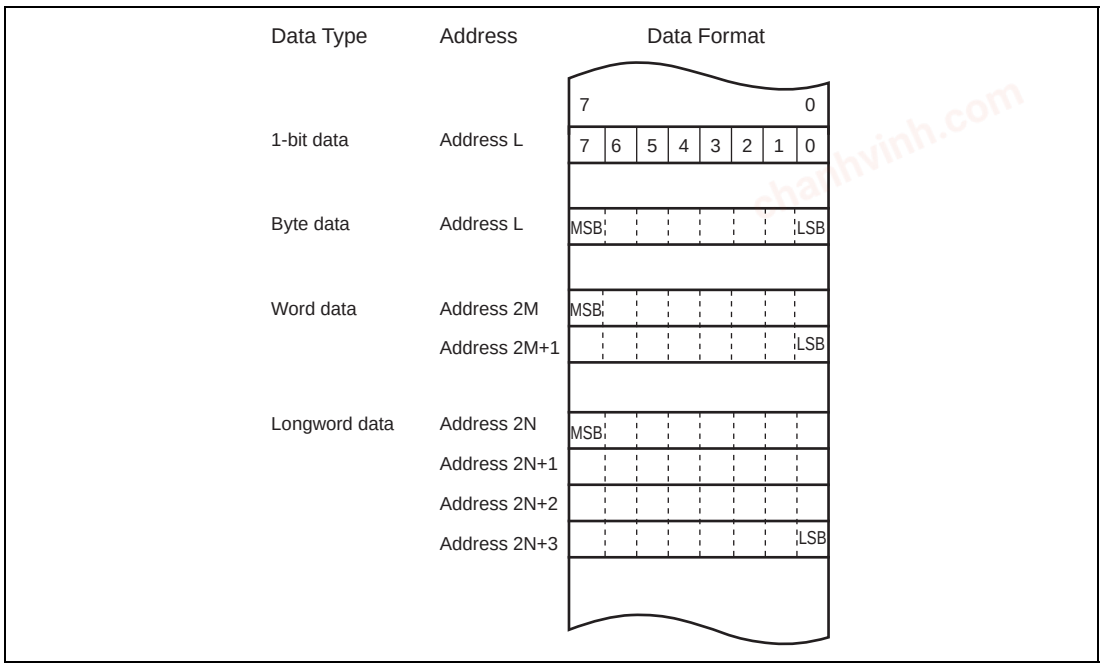



**Figure 2.8 Memory Data Formats**

chanhvinh.com



## 2.6 Instruction Set

The H8S/2000 CPU has 65 types of instructions. The instructions are classified by function as shown in table 2.1.

**Table 2.1 Instruction Classification**

| Function | Instructions | Size | Types |
|---|---|---|---|
| Data transfer | MOV | B/W/L | 5 |
| | POP*1, PUSH*1 | W/L | |
| | LDM*5, STM*5 | L | |
| | MOVFPE*3, MOVTPE*3 | B | |
| Arithmetic operations | ADD, SUB, CMP, NEG | B/W/L | 19 |
| | ADDX, SUBX, DAA, DAS | B | |
| | INC, DEC | B/W/L | |
| | ADDS, SUBS | L | |
| | MULXU, DIVXU, MULXS, DIVXS | B/W | |
| | EXTU, EXTS | W/L | |
| | TAS*4 | B | |
| Logic operations | AND, OR, XOR, NOT | B/W/L | 4 |
| Shift | SHAL, SHAR, SHLL, SHLR, ROTL, ROTR, ROTXL, ROTXR | B/W/L | 8 |
| Bit manipulation | BSET, BCLR, BNOT, BTST, BLD, BILD, BST, BIST, BAND, BIAND, BOR, BIOR, BXOR, BIXOR | B | 14 |
| Branch | $B_{CC}$*2, JMP, BSR, JSR, RTS | — | 5 |
| System control | TRAPA, RTE, SLEEP, LDC, STC, ANDC, ORC, XORC, NOP | — | 9 |
| Block data transfer | EEPMOV | — | 1 |
| | | | Total: 65 |

Notes: B: Byte size; W: Word size; L: Longword size.

1. POP.W Rn and PUSH.W Rn are identical to MOV.W @SP+, Rn and MOV.W Rn, @-SP. POP.L ERn and PUSH.L ERn are identical to MOV.L @SP+, ERn and MOV.L ERn, @-SP.
2. $B_{CC}$ is the general name for conditional branch instructions.
3. Cannot be used in this LSI.
4. Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction.
5. The ER7 register is used as a stack pointer in an STM and LDM instructions. Accordingly, ER7 cannot be stored by STM or loaded by LDM.

chanhvinh.com



### 2.6.1 Table of Instructions Classified by Function

Tables 2.3 to 2.10 summarize the instructions in each functional category. The notation used in tables 2.3 to 2.10 is defined below.

**Table 2.2 Operation Notation**

| Symbol | Description |
|---|---|
| Rd | General register (destination)* |
| Rs | General register (source)* |
| Rn | General register* |
| ERn | General register (32-bit register) |
| (EAd) | Destination operand |
| (EAs) | Source operand |
| EXR | Extended control register |
| CCR | Condition-code register |
| N | N (negative) flag in CCR |
| Z | Z (zero) flag in CCR |
| V | V (overflow) flag in CCR |
| C | C (carry) flag in CCR |
| PC | Program counter |
| SP | Stack pointer |
| #IMM | Immediate data |
| disp | Displacement |
| + | Addition |
| – | Subtraction |
| × | Multiplication |
| ÷ | Division |
| ∧ | Logical AND |
| ∨ | Logical OR |
| ⊕ | Logical exclusive OR |
| → | Move |
| ~ | NOT (logical complement) |
| :8/:16/:24/:32 | 8-, 16-, 24-, or 32-bit length |

Note: * General registers include 8-bit registers (R0H to R7H, R0L to R7L), 16-bit registers (R0 to R7, E0 to E7), and 32-bit registers (ER0 to ER7).

chanhvinh.com



**Table 2.3 Data Transfer Instructions**

| Instruction | Size*1 | Function |
|---|---|---|
| MOV | B/W/L | (EAs) → Rd, Rs → (EAd)<br>Moves data between two general registers or between a general register and memory, or moves immediate data to a general register. |
| MOVFPE | B | Cannot be used in this LSI. |
| MOVTPE | B | Cannot be used in this LSI. |
| POP | W/L | @SP+ → Rn<br>Pops a general register from the stack. POP.W Rn is identical to MOV.W @SP+, Rn. POP.L ERn is identical to MOV.L @SP+, ERn |
| PUSH | W/L | Rn → @-SP<br>Pushes a general register onto the stack. PUSH.W Rn is identical to MOV.W Rn, @-SP. PUSH.L ERn is identical to MOV.L ERn, @-SP. |
| LDM*2 | L | @SP+ → Rn (register list)<br>Pops two or more general registers from the stack. |
| STM*2 | L | Rn (register list) → @-SP<br>Pushes two or more general registers onto the stack. |

Notes: 1. Size refers to the operand size.
B: Byte
W: Word
L: Longword
2. The ER7 register is used as a stack pointer in the STM and LDM instructions. Accordingly, ER7 cannot be stored by STM or loaded by LDM.

chanhvinh.com



**Table 2.4 Arithmetic Operations Instructions (1)**

| Instruction | Size* | Function |
|---|---|---|
| ADD | B/W/L | Rd ± Rs → Rd, Rd ± #IMM → Rd |
| SUB | | Performs addition or subtraction on data in two general registers, or on immediate data and data in a general register. (Subtraction on immediate data and data in a general register cannot be performed in bytes. Use the SUBX or ADD instruction.) |
| ADDX | B | Rd ± Rs ± C → Rd, Rd ± #IMM ± C → Rd |
| SUBX | | Performs addition or subtraction with carry on data in two general registers, or on immediate data and data in a general register. |
| INC | B/W/L | Rd ± 1 → Rd, Rd ± 2 → Rd |
| DEC | | Adds or subtracts the value 1 or 2 to or from data in a general register. (Only the value 1 can be added to or subtracted from byte operands.) |
| ADDS | L | Rd ± 1 → Rd, Rd ± 2 → Rd, Rd ± 4 → Rd |
| SUBS | | Adds or subtracts the value 1, 2, or 4 to or from data in a 32-bit register. |
| DAA | B | Rd (decimal adjust) → Rd |
| DAS | | Decimal-adjusts an addition or subtraction result in a general register by referring to CCR to produce 4-bit BCD data. |
| MULXU | B/W | Rd × Rs → Rd |
| | | Performs unsigned multiplication on data in two general registers: either 8 bits × 8 bits → 16 bits or 16 bits × 16 bits → 32 bits. |
| MULXS | B/W | Rd × Rs → Rd |
| | | Performs signed multiplication on data in two general registers: either 8 bits × 8 bits → 16 bits or 16 bits × 16 bits → 32 bits. |
| DIVXU | B/W | Rd ÷ Rs → Rd |
| | | Performs unsigned division on data in two general registers: either 16 bits ÷ 8 bits → 8-bit quotient and 8-bit remainder or 32 bits ÷ 16 bits → 16-bit quotient and 16-bit remainder. |

Note: * Size refers to the operand size.
B: Byte
W: Word
L: Longword

chanhvinh.com



**Table 2.4 Arithmetic Operations Instructions (2)**

| Instruction | Size*1 | Function |
|---|---|---|
| DIVXS | B/W | Rd ÷ Rs → Rd<br>Performs signed division on data in two general registers: either 16 bits ÷ 8 bits → 8-bit quotient and 8-bit remainder or 32 bits ÷ 16 bits → 16-bit quotient and 16-bit remainder. |
| CMP | B/W/L | Rd – Rs, Rd – #IMM<br>Compares data in a general register with data in another general register or with immediate data, and sets the CCR bits according to the result. |
| NEG | B/W/L | 0 – Rd → Rd<br>Takes the two's complement (arithmetic complement) of data in a general register. |
| EXTU | W/L | Rd (zero extension) → Rd<br>Extends the lower 8 bits of a 16-bit register to word size, or the lower 16 bits of a 32-bit register to longword size, by padding with zeros on the left. |
| EXTS | W/L | Rd (sign extension) → Rd<br>Extends the lower 8 bits of a 16-bit register to word size, or the lower 16 bits of a 32-bit register to longword size, by extending the sign bit. |
| TAS*2 | B | @ERd – 0, 1 → (<bit 7> of @ERd)<br>Tests memory contents, and sets the most significant bit (bit 7) to 1. |

Notes: 1. Size refers to the operand size.
B: Byte
W: Word
L: Longword
2. Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction.

chanhvinh.com



**Table 2.5 Logic Operations Instructions**

| Instruction | Size* | Function |
|---|---|---|
| AND | B/W/L | Rd ∧ Rs → Rd, Rd ∧ #IMM → Rd<br>Performs a logical AND operation on a general register and another general register or immediate data. |
| OR | B/W/L | Rd ∨ Rs → Rd, Rd ∨ #IMM → Rd<br>Performs a logical OR operation on a general register and another general register or immediate data. |
| XOR | B/W/L | Rd ⊕ Rs → Rd, Rd ⊕ #IMM → Rd<br>Performs a logical exclusive OR operation on a general register and another general register or immediate data. |
| NOT | B/W/L | ~ Rd → Rd<br>Takes the one's complement (logical complement) of data in a general register. |

Note: * Size refers to the operand size.
B: Byte
W: Word
L: Longword

chanhvinh.com



**Table 2.6 Shift Instructions**

| Instruction | Size* | Function |
|---|---|---|
| SHAL | B/W/L | Rd (shift) → Rd |
| SHAR | | Performs an arithmetic shift on data in a general register. 1-bit or 2 bit shift is possible. |
| SHLL | B/W/L | Rd (shift) → Rd |
| SHLR | | Performs a logical shift on data in a general register. 1-bit or 2 bit shift is possible. |
| ROTL | B/W/L | Rd (rotate) → Rd |
| ROTR | | Rotates data in a general register. 1-bit or 2 bit rotation is possible. |
| ROTXL | B/W/L | Rd (rotate) → Rd |
| ROTXR | | Rotates data including the carry flag in a general register. 1-bit or 2 bit rotation is possible. |

Note: * Size refers to the operand size.
B: Byte
W: Word
L: Longword

chanhvinh.com



**Table 2.7 Bit Manipulation Instructions (1)**

| Instruction | Size* | Function |
|---|---|---|
| BSET | B | 1 → (<bit-No.> of <EAd>)<br>Sets a specified bit in a general register or memory operand to 1. The bit number is specified by 3-bit immediate data or the lower three bits of a general register. |
| BCLR | B | 0 → (<bit-No.> of <EAd>)<br>Clears a specified bit in a general register or memory operand to 0. The bit number is specified by 3-bit immediate data or the lower three bits of a general register. |
| BNOT | B | ~ (<bit-No.> of <EAd>) → (<bit-No.> of <EAd>)<br>Inverts a specified bit in a general register or memory operand. The bit number is specified by 3-bit immediate data or the lower three bits of a general register. |
| BTST | B | ~ (<bit-No.> of <EAd>) → Z<br>Tests a specified bit in a general register or memory operand and sets or clears the Z flag accordingly. The bit number is specified by 3-bit immediate data or the lower three bits of a general register. |
| BAND | B | C ∧ (<bit-No.> of <EAd>) → C<br>Logically ANDs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag. |
| BIAND | B | C ∧ (<bit-No.> of <EAd>) → C<br>Logically ANDs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag.<br>The bit number is specified by 3-bit immediate data. |
| BOR | B | C ∨ (<bit-No.> of <EAd>) → C<br>Logically ORs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag. |
| BIOR | B | C ∨ (~ <bit-No.> of <EAd>) → C<br>Logically ORs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag.<br>The bit number is specified by 3-bit immediate data. |

Note: * Size refers to the operand size.
B: Byte

chanhvinh.com



**Table 2.7 Bit Manipulation Instructions (2)**

| Instruction | Size* | Function |
|---|---|---|
| BXOR | B | C ⊕ (<bit-No.> of <EAd>) → C<br>Logically exclusive-ORs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag. |
| BIXOR | B | C ⊕ ~ (<bit-No.> of <EAd>) → C<br>Logically exclusive-ORs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag.<br>The bit number is specified by 3-bit immediate data. |
| BLD | B | (<bit-No.> of <EAd>) → C<br>Transfers a specified bit in a general register or memory operand to the carry flag. |
| BILD | B | ~ (<bit-No.> of <EAd>) → C<br>Transfers the inverse of a specified bit in a general register or memory operand to the carry flag.<br>The bit number is specified by 3-bit immediate data. |
| BST | B | C → (<bit-No.> of <EAd>)<br>Transfers the carry flag value to a specified bit in a general register or memory operand. |
| BIST | B | ~ C → (<bit-No.>. of <EAd>)<br>Transfers the inverse of the carry flag value to a specified bit in a general register or memory operand.<br>The bit number is specified by 3-bit immediate data. |

Note: * Size refers to the operand size.
B: Byte

chanhvinh.com



**Table 2.8 Branch Instructions**

| Instruction | Size | Function |
|---|---|---|
| Bcc | — | Branches to a specified address if a specified condition is true. The branching conditions are listed below. |

| Mnemonic | Description | Condition |
|---|---|---|
| BRA (BT) | Always (true) | Always |
| BRN (BF) | Never (false) | Never |
| BHI | High | C ∨ Z = 0 |
| BLS | Low or same | C ∨ Z = 1 |
| BCC (BHS) | Carry clear (high or same) | C = 0 |
| BCS (BLO) | Carry set (low) | C = 1 |
| BNE | Not equal | Z = 0 |
| BEQ | Equal | Z = 1 |
| BVC | Overflow clear | V = 0 |
| BVS | Overflow set | V = 1 |
| BPL | Plus | N = 0 |
| BMI | Minus | N = 1 |
| BGE | Greater or equal | N ⊕ V = 0 |
| BLT | Less than | N ⊕ V = 1 |
| BGT | Greater than | Z ∨ (N ⊕ V) = 0 |
| BLE | Less or equal | Z ∨ (N ⊕ V) = 1 |

| | | |
|---|---|---|
| JMP | — | Branches unconditionally to a specified address |
| BSR | — | Branches to a subroutine at a specified address |
| JSR | — | Branches to a subroutine at a specified address |
| RTS | — | Returns from a subroutine |

chanhvinh.com



**Table 2.9 System Control Instructions**

| Instruction | Size* | Function |
|---|---|---|
| TRAPA | — | Starts trap-instruction exception handling. |
| RTE | — | Returns from an exception-handling routine. |
| SLEEP | — | Causes a transition to a power-down state. |
| LDC | B/W | (EAs) → CCR, (EAs) → EXR<br>Moves the memory operand contents or immediate data to CCR or EXR. Although CCR and EXR are 8-bit registers, word-size transfers are performed between them and memory. The upper 8 bits are valid. |
| STC | B/W | CCR → (EAd), EXR → (EAd)<br>Transfers CCR or EXR contents to a general register or memory operand. Although CCR and EXR are 8-bit registers, word-size transfers are performed between them and memory. The upper 8 bits are valid. |
| ANDC | B | CCR ∧ #IMM → CCR, EXR ∧ #IMM → EXR<br>Logically ANDs the CCR or EXR contents with immediate data. |
| ORC | B | CCR ∨ #IMM → CCR, EXR ∨ #IMM → EXR<br>Logically ORs the CCR or EXR contents with immediate data. |
| XORC | B | CCR ⊕ #IMM → CCR, EXR ⊕ #IMM → EXR<br>Logically exclusive-ORs the CCR or EXR contents with immediate data. |
| NOP | — | PC + 2 → PC<br>Only increments the program counter. |

Note: * Size refers to the operand size.
B: Byte
W: Word

chanhvinh.com



**Table 2.10 Block Data Transfer Instructions**

| Instruction | Size | Function |
|---|---|---|
| EEPMOV.B | — | if R4L ≠ 0 then<br>Repeat @ER5+ → @ER6+<br>R4L–1 → R4L<br>Until R4L = 0<br>else next: |
| EEPMOV.W | — | if R4 ≠ 0 then<br>Repeat @ER5+ → @ER6+<br>R4–1 → R4<br>Until R4 = 0<br>else next:<br>Transfers a data block. Starting from the address set in ER5, transfers data for the number of bytes set in R4L or R4 to the address location set in ER6.<br>Execution of the next instruction begins as soon as the transfer is completed. |

### 2.6.2 Basic Instruction Formats

The H8S/2000 CPU instructions consist of 2-byte (1-word) units. An instruction consists of an operation field (op), a register field (r), an effective address extension (EA), and a condition field (cc).

Figure 2.9 shows examples of instruction formats.

- Operation field

  Indicates the function of the instruction, the addressing mode, and the operation to be carried out on the operand. The operation field always includes the first four bits of the instruction. Some instructions have two operation fields.
- Register field

  Specifies a general register. Address registers are specified by 3 bits, and data registers by 3 bits or 4 bits. Some instructions have two register fields, and some have no register field.
- Effective address extension

  8, 16, or 32 bits specifying immediate data, an absolute address, or a displacement.
- Condition field

  Specifies the branching condition of Bcc instructions.

chanhvinh.com



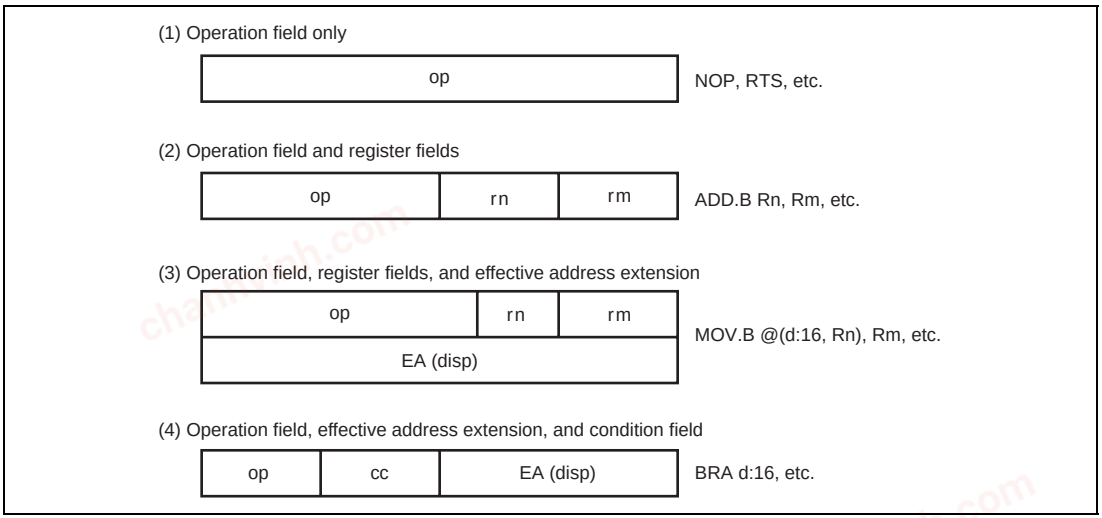



**Figure 2.9 Instruction Formats (Examples)**

chanhvinh.com



## 2.7 Addressing Modes and Effective Address Calculation

The H8S/2000 CPU supports the eight addressing modes listed in table 2.11. Each instruction uses a subset of these addressing modes.

Arithmetic and logic operations instructions can use the register direct and immediate addressing modes. Data transfer instructions can use all addressing modes except program-counter relative and memory indirect. Bit manipulation instructions can use register direct, register indirect, or absolute addressing mode to specify an operand, and register direct (BSET, BCLR, BNOT, and BTST instructions) or immediate (3-bit) addressing mode to specify a bit number in the operand.

**Table 2.11 Addressing Modes**

| No. | Addressing Mode | Symbol |
|---|---|---|
| 1 | Register direct | Rn |
| 2 | Register indirect | @ERn |
| 3 | Register indirect with displacement | @(d:16,ERn)/@(d:32,ERn) |
| 4 | Register indirect with post-increment<br>Register indirect with pre-decrement | @ERn+<br>@–ERn |
| 5 | Absolute address | @aa:8/@aa:16/@aa:24/@aa:32 |
| 6 | Immediate | #xx:8/#xx:16/#xx:32 |
| 7 | Program-counter relative | @(d:8,PC)/@(d:16,PC) |
| 8 | Memory indirect | @@aa:8 |

### 2.7.1 Register Direct—Rn

The register field of the instruction code specifies an 8-, 16-, or 32-bit general register which contains the operand. R0H to R7H and R0L to R7L can be specified as 8-bit registers. R0 to R7 and E0 to E7 can be specified as 16-bit registers. ER0 to ER7 can be specified as 32-bit registers.

### 2.7.2 Register Indirect—@ERn

The register field of the instruction code specifies an address register (ERn) which contains the address of a memory operand. If the address is a program instruction address, the lower 24 bits are valid and the upper 8 bits are all assumed to be 0 (H'00).

chanhvinh.com



### 2.7.3 Register Indirect with Displacement—@(d:16, ERn) or @(d:32, ERn)

A 16-bit or 32-bit displacement contained in the instruction code is added to an address register (ERn) specified by the register field of the instruction, and the sum gives the address of a memory operand. A 16-bit displacement is sign-extended when added.

### 2.7.4 Register Indirect with Post-Increment or Pre-Decrement—@ERn+ or @–ERn

#### (1) Register Indirect with Post-Increment—@ERn+

The register field of the instruction code specifies an address register (ERn) which contains the address of a memory operand. After the operand is accessed, 1, 2, or 4 is added to the address register contents and the sum is stored in the address register. The value added is 1 for byte access, 2 for word access, and 4 for longword access. For word or longword transfer instructions, the register value should be even.

#### (2) Register Indirect with Pre-Decrement—@–ERn

The value 1, 2, or 4 is subtracted from an address register (ERn) specified by the register field in the instruction code, and the result becomes the address of a memory operand. The result is also stored in the address register. The value subtracted is 1 for byte access, 2 for word access, and 4 for longword access. For word or longword transfer instructions, the register value should be even.

### 2.7.5 Absolute Address—@aa:8, @aa:16, @aa:24, or @aa:32

The instruction code contains the absolute address of a memory operand. The absolute address may be 8 bits long (@aa:8), 16 bits long (@aa:16), 24 bits long (@aa:24), or 32 bits long (@aa:32). Table 2.12 indicates the accessible absolute address ranges.

To access data, the absolute address should be 8 bits (@aa:8), 16 bits (@aa:16), or 32 bits (@aa:32) long. For an 8-bit absolute address, the upper 24 bits are all assumed to be 1 (H'FFFF). For a 16-bit absolute address, the upper 16 bits are a sign extension. For a 32-bit absolute address, the entire address space is accessed.

A 24-bit absolute address (@aa:24) indicates the address of a program instruction. The upper 8 bits are all assumed to be 0 (H'00).

chanhvinh.com



**Table 2.12 Absolute Address Access Ranges**

| Absolute Address | | Advanced Mode |
|---|---|---|
| Data address | 8 bits (@aa:8) | H'FFFF00 to H'FFFFFF |
| | 16 bits (@aa:16) | H'000000 to H'007FFF, H'FF8000 to H'FFFFFF |
| | 32 bits (@aa:32) | H'000000 to H'FFFFFF |
| Program instruction address | 24 bits (@aa:24) | |

### 2.7.6 Immediate—#xx:8, #xx:16, or #xx:32

The 8-bit (#xx:8), 16-bit (#xx:16), or 32-bit (#xx:32) immediate data contained in an instruction code can be used directly as an operand.

The ADDS, SUBS, INC, and DEC instructions implicitly contain immediate data in their instruction codes. Some bit manipulation instructions contain 3-bit immediate data in the instruction code, specifying a bit number. The TRAPA instruction contains 2-bit immediate data in its instruction code, specifying a vector address.

### 2.7.7 Program-Counter Relative—@(d:8, PC) or @(d:16, PC)

This mode can be used by the Bcc and BSR instructions. An 8-bit or 16-bit displacement contained in the instruction code is sign-extended to 24 bits and added to the 24-bit address indicated by the PC value to generate a 24-bit branch address. Only the lower 24 bits of this branch address are valid; the upper 8 bits are all assumed to be 0 (H'00). The PC value to which the displacement is added is the address of the first byte of the next instruction, so the possible branching range is –126 to +128 bytes (–63 to +64 words) or –32766 to +32768 bytes (–16383 to +16384 words) from the branch instruction. The resulting value should be an even number.

chanhvinh.com



### 2.7.8 Memory Indirect—@@aa:8

This mode can be used by the JMP and JSR instructions. The instruction code contains an 8-bit absolute address specifying a memory operand which contains a branch address. The upper bits of the 8-bit absolute address are all assumed to be 0, so the address range is 0 to 255 (H'000000 to H'0000FF in advanced mode).

In advanced mode, the memory operand is a longword operand, the first byte of which is assumed to be 0 (H'00).

Note that the top area of the address range in which the branch address is stored is also used for the exception vector area. For further details, see section 3, Exception Handling.

If an odd address is specified in word or longword memory access, or as a branch address, the least significant bit is regarded as 0, causing data to be accessed or the instruction code to be fetched at the address preceding the specified address. (For further information, see section 2.5.2, Memory Data Formats.)

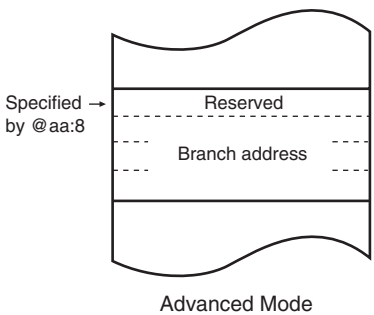



**Figure 2.10 Branch Address Specification in Memory Indirect Addressing Mode**

chanhvinh.com



### 2.7.9 Effective Address Calculation

Table 2.13 indicates how effective addresses are calculated in each addressing mode.

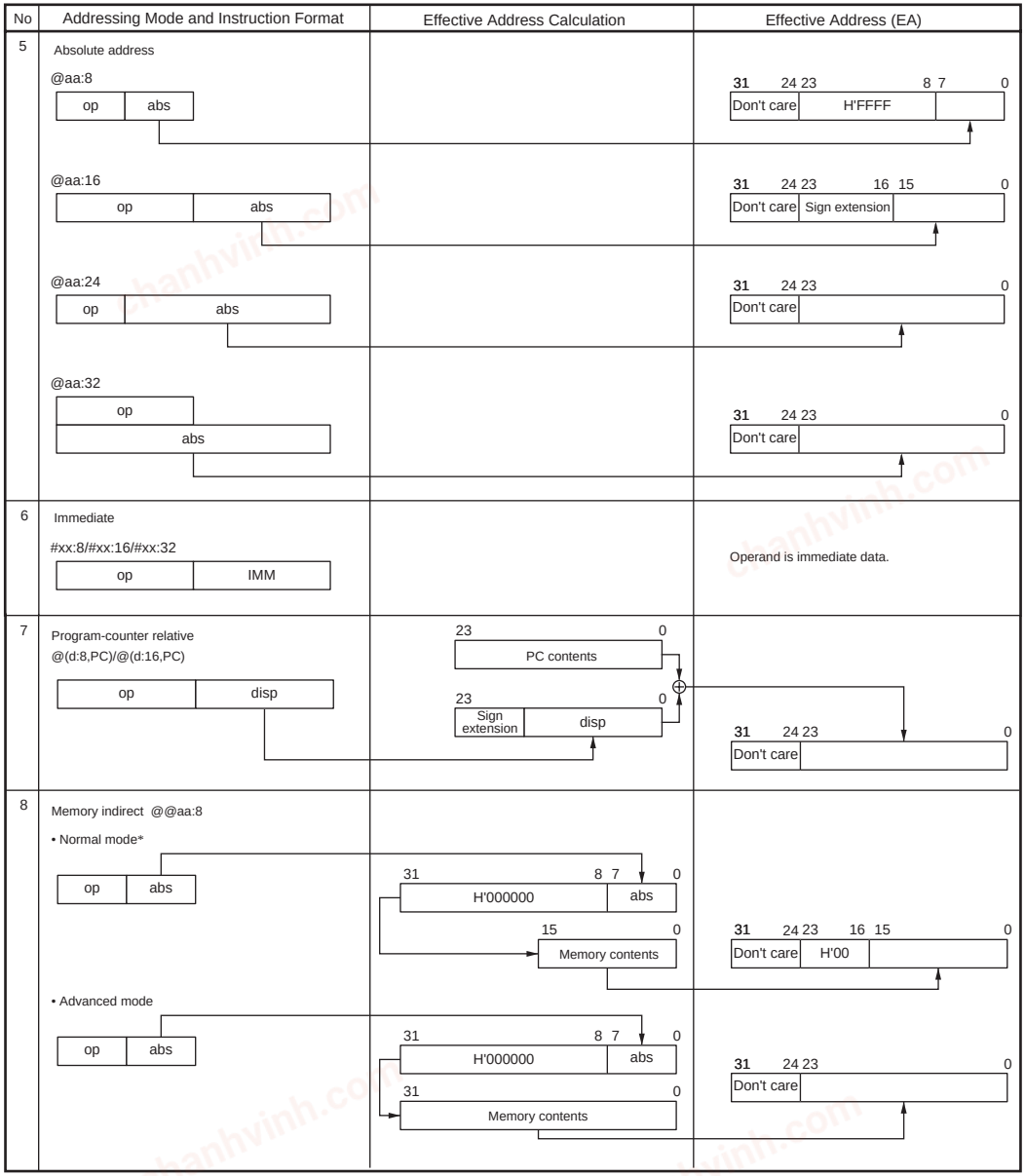
**Table 2.13 Effective Address Calculation**

| No | Addressing Mode and Instruction Format | Effective Address Calculation | Effective Address (EA) |
|---|---|---|---|
| 1 | Register direct (Rn)<br>op \| rm \| rn | | Operand is general register contents. |
| 2 | Register indirect (@ERn)<br>op \| r | 31 … 0: General register contents | 31 … 24 23 … 0: Don't care |
| 3 | Register indirect with displacement @(d:16,ERn) or @(d:32,ERn)<br>op \| r \| disp | 31 … 0: General register contents<br>31 … 0: Sign extension \| disp | 31 … 24 23 … 0: Don't care |
| 4 | Register indirect with post-increment or pre-decrement<br>• Register indirect with post-increment @ERn+<br>op \| r | 31 … 0: General register contents<br>1, 2, or 4 | 31 … 24 23 … 0: Don't care |
| | • Register indirect with pre-decrement @-ERn<br>op \| r | 31 … 0: General register contents<br>1, 2, or 4 | 31 … 24 23 … 0: Don't care |

| Operand Size | Offset |
|---|---|
| Byte | 1 |
| Word | 2 |
| Longword | 4 |

chanhvinh.com






Note: * For this LSI, normal mode is not available.

chanhvinh.com



## 2.8 Processing States

The H8S/2000 CPU has five main processing states: the reset state, exception handling state, program execution state, bus-released state, and program stop state. Figure 2.11 indicates the state transitions.

- Reset state

  In this state the CPU and internal peripheral modules are all initialized and stopped. When the $\overline{\text{RES}}$ input goes low, all current processing stops and the CPU enters the reset state. All interrupts are masked in the reset state. Reset exception handling starts when the $\overline{\text{RES}}$ signal changes from low to high. For details, see section 3, Exception Handling.

  The reset state can also be entered by a watchdog timer overflow.

- Exception-handling state

  The exception-handling state is a transient state that occurs when the CPU alters the normal processing flow due to an exception source, such as, a reset, trace, interrupt, or trap instruction. The CPU fetches a start address (vector) from the exception vector table and branches to that address. For further details, see section 3, Exception Handling.

- Program execution state

  In this state the CPU executes program instructions in sequence.

- Bus-released state

  The bus-released state occurs when the bus has been released in response to a bus request* from a bus master (DTC) other than the CPU. While the bus is released, the CPU halts operations.

- Program stop state

  This is a power-down state in which the CPU stops operating. The program stop state occurs when a SLEEP instruction is executed. For details, see section 6, Power-Down Modes.

Notes: * The DTC requests bus mastership when an activation request for the DTC is generated. Bus mastership is transferred from the CPU to the DTC with the following timing.

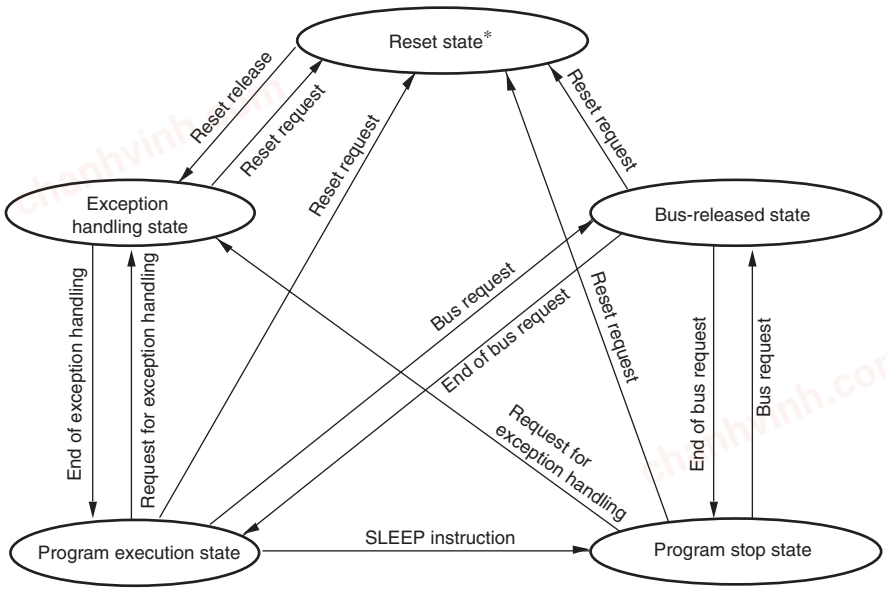
1. Bus mastership is transferred at the end of the current bus cycle.

   However, when the bus activity for a single instruction is divided up into multiple bus cycles due to longword access, etc., the bus mastership will not necessarily be transferred at the end of the current bus cycle. For details, see section 2.7, Bus States During Instruction Execution, in the H8S/2600 Series, H8S/2000 Series Software Manual.

chanhvinh.com



2. If the CPU is in sleep mode at the time of the request, bus mastership is transferred immediately.




Note: * A transition to the reset state occurs in any of the following cases.
1. When $\overline{RES}$ goes low in any state
2. When the watchdog timer overflows
3. When an LVD reset is caused by a low-voltage detection

**Figure 2.11 State Transitions**

chanhvinh.com



## 2.9 Usage Notes

### 2.9.1 TAS Instruction

Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction. The H8S and H8/300 Series C/C++ Compiler of Renesas Electronics Corp. does not generate a TAS instruction. Accordingly, when a TAS instruction is used as a user-defined embedded function, register ER0, ER1, ER4, or ER5 should be used.

### 2.9.2 STM and LDM Instructions

The ER7 register is used as a stack pointer in an STM and LDM instructions. Accordingly, ER7 cannot be stored by STM or loaded by LDM. Two, three, or four registers can be stored or loaded by a single STM or LDM instruction. The combination of registers that can be stored or loaded are as follows.

- Two registers: ER0 and ER1, ER2 and ER3, or ER4 and ER5
- Three registers: ER0 to ER2 or ER4 to ER6
- Four registers: ER0 to ER3

The H8S and H8/300 Series C/C++ Compiler of Renesas Electronics Corp. does not generate an STM or LDM instruction that uses ER7.

### 2.9.3 Note on Bit Manipulation Instructions

Bit manipulation instructions such as BSET, BCLR, BNOT, BST, and BIST read data in byte units, perform bit manipulation, and write data in byte units. Thus, care must be taken when these bit manipulation instructions are executed for a register or port including write-only bits.

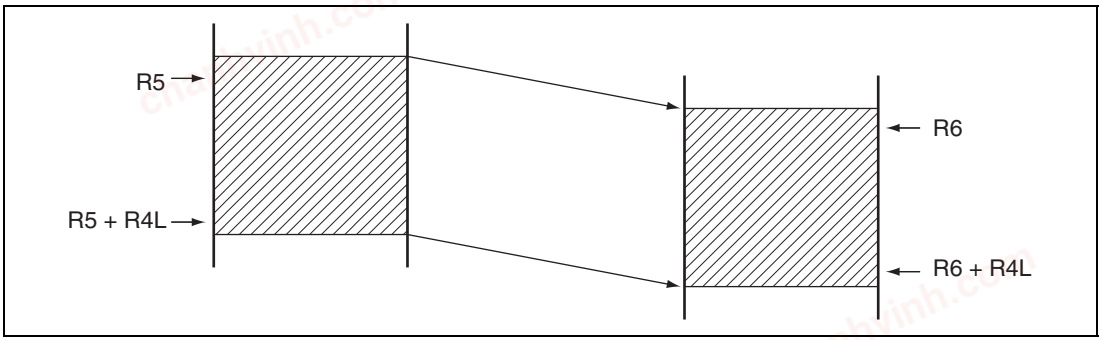
In addition, the BCLR instruction can be used to clear the flag of an internal I/O register. In this case, if the flag to be cleared has been set by an interrupt processing routine, the flag need not be read before executing the BCLR instruction.

chanhvinh.com

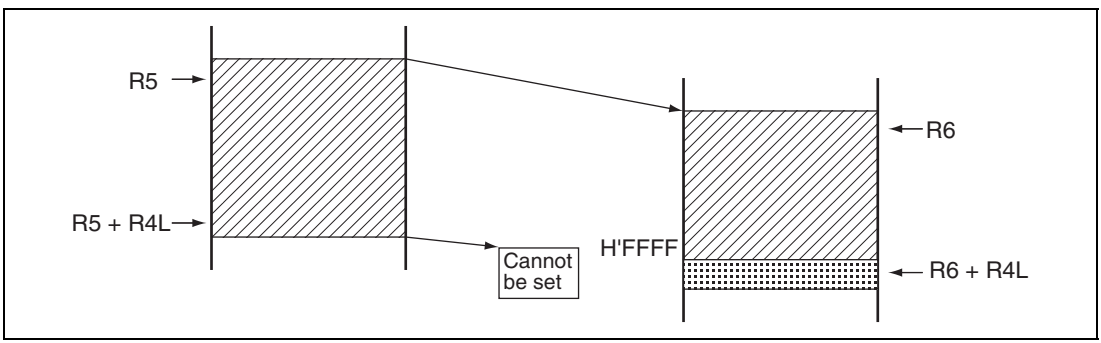


### 2.9.4 EEPMOV Instruction

1. The EEPMOV instruction performs a block transfer. As shown in the following figure, EEPMOV transfers data whose start address is indicated by R5 for the number of bytes indicated by R4L to the address indicated by R6.




2. R4L and R6 should be set so that the end address (R6 + R4L) of the transfer destination does not exceed H'FFFF (R6 should not change from H'FFFF to H'0000 during EEPMOV instruction execution).



chanhvinh.com



# Section 3 Exception Handling

## 3.1 Exception Handling Types and Priority

As table 3.1 indicates, exception handling is caused by a reset, trace, NMI interrupt, trap instruction, or interrupt. Exception handling is prioritized as shown in table 3.1. If two or more exceptions occur simultaneously, they are accepted and processed in order of priority. Exception sources, the stack structure, and operation of the CPU vary depending on the interrupt control mode. For details on the interrupt control mode, see section 4, Interrupt Controller.

**Table 3.1 Exception Handling Types and Priority**

| Priority | Exception Type | Start Timing of Exception Handling |
|---|---|---|
| High | Reset | Started immediately after a low-to-high transition at the $\overline{\text{RES}}$ pin, or by other reset sources. The CPU enters the reset state when the $\overline{\text{RES}}$ pin is low. |
| | Trace*1 | Started when execution of the current instruction or exception handling ends, if the trace (T) bit in EXR is set to 1. |
| | NMI | Generated when an edge of the $\overline{\text{NMI}}$ pin is input. An NMI interrupt request has the highest priority among interrupt requests. It is always accepted regardless of the value of the I bit in CCR. |
| | Trap instruction*3 | Started by execution of a trap instruction (TRAPA). |
| Low | Interrupt | Started when execution of the current instruction or exception handling ends, if an interrupt request has been issued.*2 |

Notes: 1. Traces are enabled only in interrupt control mode 2. Trace exception handling is not executed after execution of an RTE instruction.
2. Interrupt detection is not performed on completion of ANDC, ORC, XORC, or LDC instruction execution, or on completion of reset exception handling.
3. Trap instruction exception handling requests are accepted at all times in program execution state.

## 3.2 Exception Handling Sources and Vector Table

Different vector addresses are assigned to different exception sources. For details on the exception sources and their vector addresses, see section 4, Interrupt Controller.

chanhvinh.com



## 3.3 Reset

A reset has the highest exception handling priority. When the $\overline{\text{RES}}$ pin goes low, all processing halts and this LSI enters the reset. To ensure that this LSI is reset, hold the $\overline{\text{RES}}$ pin low for the specified time at power-on and during operation, hold the $\overline{\text{RES}}$ pin low for the specified time. A reset initializes the internal state of the CPU and the registers of on-chip peripheral modules, and selects low-speed on-chip oscillator as a system clock. The chip can also be reset by detection of the low-voltage, overflow of the watchdog timer, or software.

The interrupt control mode is 0 immediately after reset.

### 3.3.1 Reset Sources

This LSI enters the reset state by reset sources listed in table 3.2. If multiple reset sources occur simultaneously, a reset source having the highest priority will be accepted. A reset source can be identified by reading the reset source flag register (RSTFR).

For details on a low-voltage detection reset, see section 26, Low-Voltage Detection Circuits. For details on a watchdog timer overflow reset, see section 19, Watchdog Timer (WDT).

**Table 3.2 List of Reset Sources**

| Reset Source | Description | Priority |
|---|---|---|
| Reset by $\overline{\text{RES}}$ pin | This LSI enters the reset state if the $\overline{\text{RES}}$ pin is held low for at least a specified period. | High |
| Low-voltage detection reset | This LSI enters the reset state if the power voltage becomes the specified voltage or lower. | ↕ |
| Watchdog timer overflow reset | This LSI enters the reset state if the counter in the watchdog timer overflows. | ↕ |
| Software reset | This LSI enters the reset state if the SRST bit in RSTCR is set to 1. | Low |

chanhvinh.com



### (1) Reset Source Flag Register (RSTFR)

Address: H'FF0620

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | SWRST | PRST | LVD2RST | LVD1RST | PORRST | WRST |
| Value after reset: | 0 | 0 | (0) | (0) | (0) | (0) | (0) | (0) |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is read as 0. The write value should be 0. | — |
| 6 | — | | | |
| 5 | SWRST | Software reset detection flag | 1: Indicates that a reset by a software reset occurs.<br>0: Indicates that a reset by a software reset does not occur. | R/W |
| 4 | PRST | $\overline{\text{RES}}$ pin reset detection flag | 1: Indicates that a reset by a $\overline{\text{RES}}$ pin reset occurs.<br>0: Indicates that a reset by a $\overline{\text{RES}}$ pin reset does not occur. | R/W |
| 3 | LVD2RST | LVD2 reset detection flag | 1: Indicates that a reset by an LVD2 reset occurs.<br>0: Indicates that a reset by an LVD2 reset does not occur. | R/W |
| 2 | LVD1RST | LVD1 reset detection flag | 1: Indicates that a reset by an LVD1 reset occurs.<br>0: Indicates that a reset by an LVD1 reset does not occur. | R/W |
| 1 | PORRST | LVD0 reset detection flag | 1: Indicates that a reset by an LVD0 reset occurs.<br>0: Indicates that a reset by an LVD0 reset does not occur. | R/W |
| 0 | WRST | Watchdog timer reset detection flag | 1: Indicates that a reset by a watchdog timer overflows.<br>0: Indicates that a reset by a watchdog timer does not occur. | R/W |

Note: Each flag in this register can be cleared by writing 0 to it. The write value to the reserved bits should always be 0.

chanhvinh.com



### (2) Reset Control Register (RSTCR)

Address: H'FF06DA

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | WI | WE | — | — | — | — | — | SRST |
| Value after reset: | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | WI | Write inhibit | 0: Writing is permitted.<br>1: Writing is inhibited. | W |
| 6 | WE | Write enable | 0: Writing is disabled.<br>1: Writing is enabled.<br>[Setting condition]<br>When 0 is written to WI and 1 is written to WE.<br>[Clearing condition]<br>When 0 is written to WI and WE. | R/W |
| 5 to 1 | — | Reserved | These bits are read as 0. The write value should be 0. | — |
| 0 | SRST | Software reset | 0: Normal operation<br>1: A software reset is generated. | R/W |

Note: A MOV instruction should be used to write to this register.

- WI bit (write inhibit)
  This register can be written to only when this bit is 0. This bit is always read as 1.
- WE bit (write enable)
  Bit 0 in this register can be written to when this bit is 1.
- SRST bit (software reset)
  A software reset is generated when this bit is 1.

chanhvinh.com



### 3.3.2 Reset Exception Handling

When the $\overline{RES}$ pin goes high after being held low for the necessary time, this LSI starts reset exception handling as follows:

1. The internal state of the CPU and the registers of the on-chip peripheral modules are initialized, VOFR is cleared to H'0000, the T bit in EXR is cleared to 0, and the I bit in EXR and CCR is set to 1.
2. The low-speed on-chip oscillator is selected as a system clock.
3. After the reset exception handling vector address is read and transferred to the PC, program execution starts from the address indicated by the PC.

Figure 3.1 shows an example of the reset sequence.

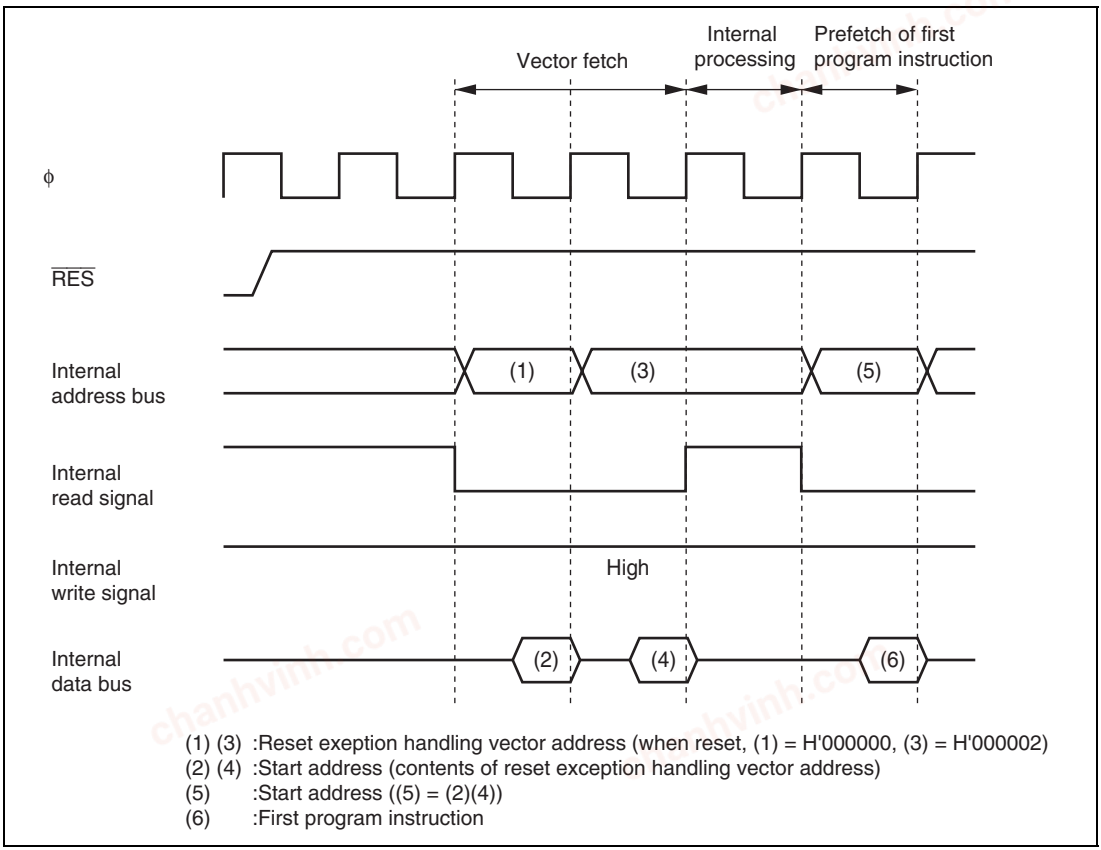



**Figure 3.1 Reset Sequence**

chanhvinh.com



### 3.3.3 Interrupts immediately after Reset

If an interrupt is accepted after a reset but before the stack pointer (SP) is initialized, the PC and CCR will not be saved correctly, leading to a program crash. To prevent this, all interrupt requests, including NMI, are disabled immediately after a reset. Since the first instruction of a program is always executed immediately after the reset state ends, make sure that this instruction initializes the stack pointer (example: MOV.L #xx: 32, SP).

### 3.3.4 On-Chip Peripheral Functions after Reset Release

After release from a reset, MSTCR is initialized, and the DTC and all modules other than timer RE enter module standby mode.

Consequently, on-chip peripheral module registers cannot be read or written to. Register reading and writing is enabled when module standby mode is exited.

chanhvinh.com



## 3.4 Trace Exception Handling

Traces are enabled in interrupt control mode 2. Trace mode is not activated in interrupt control mode 0, irrespective of the state of the T bit. For details on interrupt control modes, see section 4, Interrupt Controller.

If the T bit in EXR is set to 1, trace mode is activated. In trace mode, a trace exception occurs on completion of each instruction. Trace mode is not affected by the interrupt masking bit in CCR. Table 3.3 shows the state of CCR and EXR after execution of trace exception handling. Trace mode is canceled by clearing the T bit in EXR to 0. The T bit saved on the stack retains its value of 1, and trace mode resumes when control is returned from the trace exception handling routine by the RTE instruction. Trace exception handling is not carried out after execution of the RTE instruction.

Interrupts are accepted even within the trace exception handling routine.

**Table 3.3 Status of CCR and EXR after Trace Exception Handling**

| | CCR | | | EXR |
|---|---|---|---|---|
| **Interrupt Control Mode** | **I** | **UI** | **I2 to I0** | **T** |
| 0 | Trace exception handling cannot be used. | | | |
| 2 | 1 | — | — | 0 |

[Legend]

1: Set to 1

0: Cleared to 0

—: Retains value prior to execution.

chanhvinh.com



## 3.5 Interrupt Exception Handling

Interrupts are controlled by the interrupt controller. The interrupt controller has two interrupt control modes and can assign interrupts other than NMI to four priority/mask levels to enable multiplexed interrupt control. The source to start interrupt exception handling and the vector address differ depending on the product. For details, see section 4, Interrupt Controller.

The interrupt exception handling is as follows:

1. The values in the program counter (PC), condition code register (CCR), and extended register (EXR) are saved in the stack.
2. The interrupt mask bit is updated and the T bit is cleared to 0.
3. A vector address corresponding to the interrupt source is generated, the start address is loaded from the vector table to the PC, and program execution starts from that address.

## 3.6 Trap Instruction Exception Handling

Trap instruction exception handling starts when a TRAPA instruction is executed. Trap instruction exception handling can be executed at all times in the program execution state.

The trap instruction exception handling is as follows:

1. The values in the program counter (PC), condition code register (CCR), and extended register (EXR) are saved in the stack.
2. The interrupt mask bit is updated and the T bit is cleared to 0.
3. A vector address corresponding to the interrupt source is generated, the start address is loaded from the vector table to the PC, and program execution starts from that address.

The TRAPA instruction fetches a start address from a vector table entry corresponding to a vector number from 0 to 3, as specified in the instruction code.

Table 3.4 shows the status of CCR and EXR after execution of trap instruction exception handling.

chanhvinh.com



**Table 3.4 Status of CCR and EXR after Trap Instruction Exception Handling**

| | CCR | | | EXR |
|---|---|---|---|---|
| **Interrupt Control Mode** | **I** | **UI** | **I2 to I0** | **T** |
| 0 | 1 | — | — | — |
| 2 | 1 | — | — | 0 |

[Legend]
1: Set to 1
0: Cleared to 0
—: Retains value prior to execution.

## 3.7 Stack Status after Exception Handling

Figure 3.2 shows the stack after completion of trap instruction exception handling and interrupt exception handling.

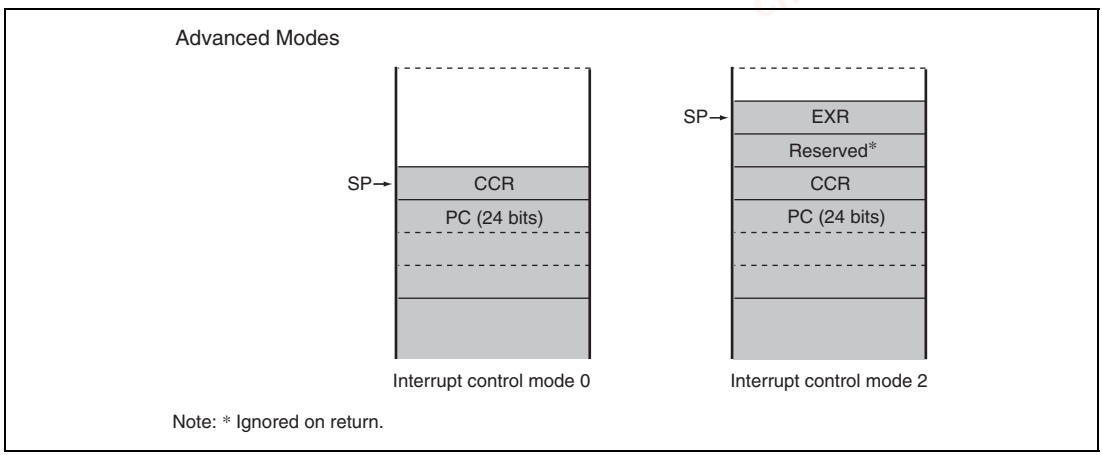



**Figure 3.2 Stack Status after Exception Handling**

chanhvinh.com



## 3.8 Usage Note

When accessing word data or longword data, this LSI assumes that the lowest address bit is 0. The stack should always be accessed by word transfer instruction or longword transfer instruction, and the value of the stack pointer (SP, ER7) should always be kept even.
Use the following instructions to save registers:

```
PUSH.W   Rn    (or MOV.W Rn, @-SP)
PUSH.L   ERn   (or MOV.L ERn, @-SP)
```

Use the following instructions to restore registers:

```
POP.W    Rn    (or MOV.W @SP+, Rn)
POP.L    ERn   (or MOV.L @SP+, ERn)
```

Setting SP to an odd value may lead to a malfunction. Figure 3.3 shows an example of operation when the SP value is odd.

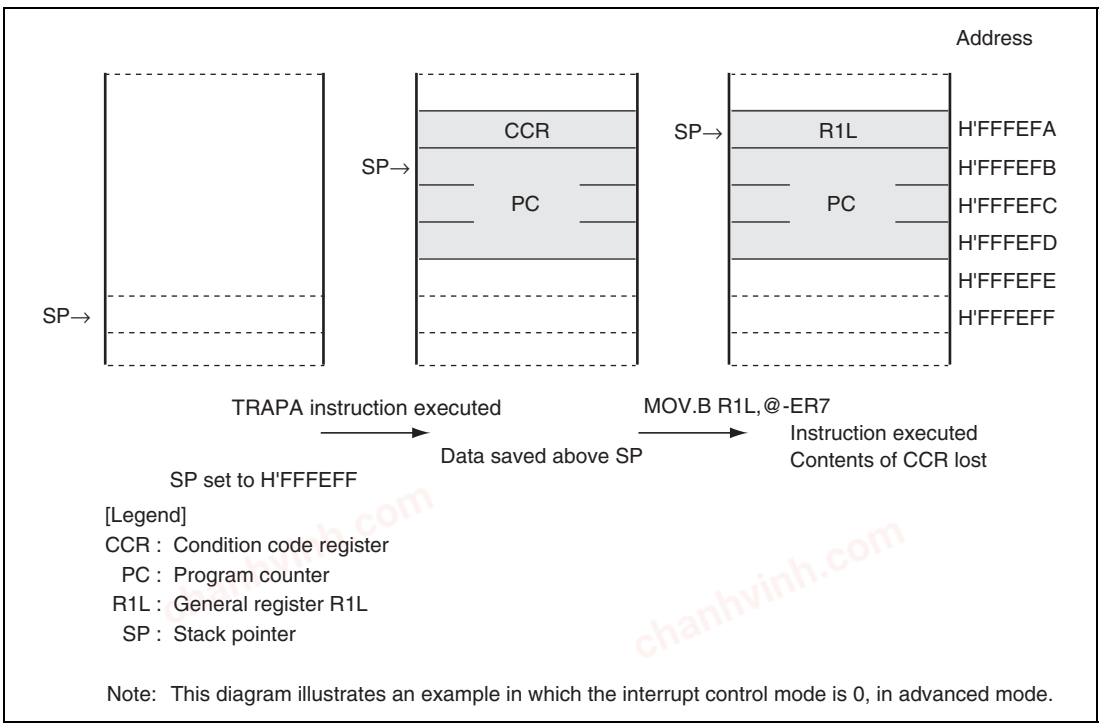



**Figure 3.3 Operation when SP Value Is Odd**

chanhvinh.com



# Section 4 Interrupt Controller

## 4.1 Features

- Two interrupt control modes

  Either of the two interrupt control modes can be selected by means of the INTM1 and INTM0 bits in the interrupt control register (INTCR).
- Priorities settable with IPR

  An interrupt priority register (IPR) is provided for setting interrupt priorities. Four priority levels can be set for each module for all interrupts except NMI. NMI and some flash memory interrupts are assigned the highest priority level of 3, and can be accepted at all times.
- Independent vector addresses

  Most interrupt sources are assigned independent vector addresses, making it unnecessary for the source to be identified in the interrupt handling routine.
- Nine external interrupts

  NMI is the highest-priority interrupt, and is accepted at all times. Rising edge or falling edge can be selected for NMI. Falling edge, rising edge, or both edges can be selected independently for $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$.
- DTC control

  DTC activation is performed by means of interrupt requests.

chanhvinh.com



A block diagram of the interrupt controller is shown in figure 4.1.

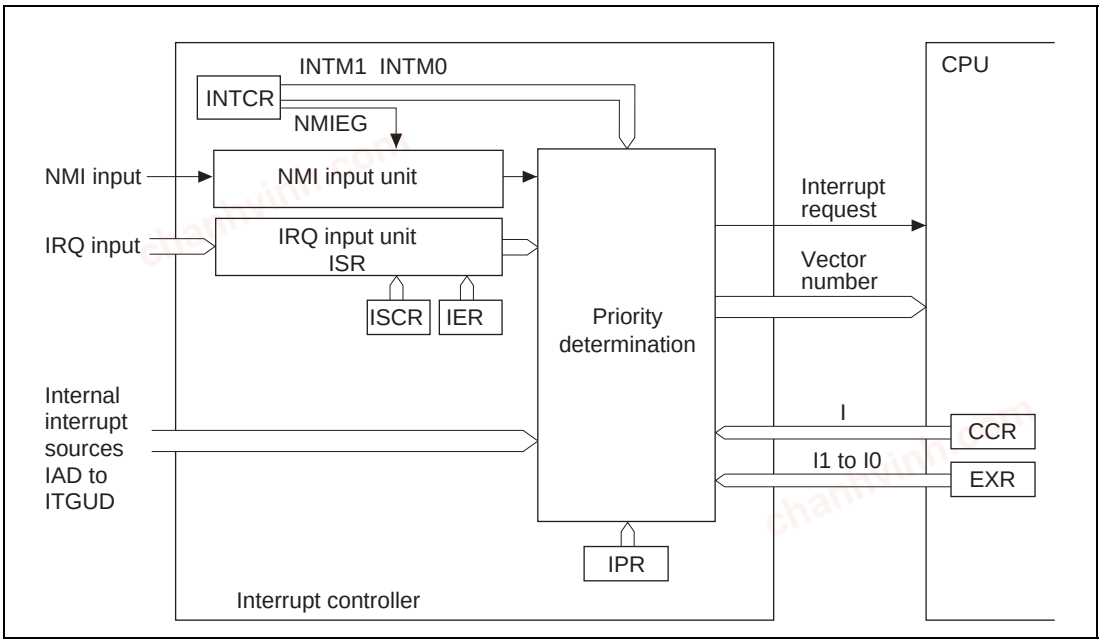



**Figure 4.1 Block Diagram of Interrupt Controller**

Table 4.1 shows the pin configuration of the interrupt controller.

**Table 4.1 Pin Configuration**

| Name | I/O | Function |
|---|---|---|
| NMI | Input | Nonmaskable external interrupt<br>Rising or falling edge can be selected. |
| $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ | Input | Maskable external interrupts<br>Rising, falling, or both edges can be selected independently. |

chanhvinh.com



## 4.2 Register Descriptions

- Interrupt control register (INTCR)
- IRQ sense control register H (ISCRH)
- IRQ sense control register L (ISCRL)
- IRQ enable register (IER)
- IRQ status register (ISR)
- Interrupt priority register A (IPRA)
- Interrupt priority register B (IPRB)
- Interrupt priority register C (IPRC)
- Interrupt priority register D (IPRD)
- Interrupt priority register E (IPRE)
- Interrupt priority register F (IPRF)
- Interrupt priority register G (IPRG)
- Interrupt priority register H (IPRH)
- Interrupt priority register I (IPRI)
- Interrupt vector offset register (VOFR)
- IRQ noise canceler control register (INCCR)
- Event link interrupt control status register (ELCSR)

chanhvinh.com



### 4.2.1 Interrupt Control Register (INTCR)

Address: H'FF0520

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | INTM[1:0] | | NMIEG | ADTRG1 | ADTRG0 | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7<br>6 | — | Reserved | These bits are read as 0. The write value should be 0. | — |
| 5<br>4 | INTM[1:0] | Interrupt control select mode 1 and 0 | 00: Interrupt control mode 0<br>Interrupts are controlled by the I bit.<br>01: Setting prohibited<br>10: Interrupt control mode 2<br>Interrupts are controlled by bits I1 and I0, and IPR.<br>11: Setting prohibited | R/W |
| 3 | NMIEG | NMI edge select | 0: Interrupt request is generated at falling edge of NMI input.<br>1: Interrupt request is generated at rising edge of NMI input. | R/W |
| 2 | ADTRG1 | $\overline{\text{ADTRG2}}$ edge select | 0: AD2 conversion is started at falling edge of $\overline{\text{ADTRG2}}$ input.<br>1: AD2 conversion is started at rising edge of $\overline{\text{ADTRG2}}$ input. | R/W |
| 1 | ADTRG0 | $\overline{\text{ADTRG1}}$ edge select | 0: AD1 or AD2 conversion is started at falling edge of $\overline{\text{ADTRG1}}$ input.<br>1: AD1 or AD2 conversion is started at rising edge of $\overline{\text{ADTRG1}}$ input. | R/W |
| 0 | — | Reserved | This bit is read as 0. The write value should be 0. | — |

- INTM1 and INTM0 bits (interrupt control select mode 1 and 0)
  These bits select the interrupt control mode for the interrupt controller.
- NMIEG bit (NMI edge select)
  Selects the input edge for the $\overline{\text{NMI}}$ pin.

chanhvinh.com



- ADTRG1 and ADTRG0 bits ($\overline{\text{ADTRG2}}$ and $\overline{\text{ADTRG1}}$ edge select)
  These bits select the input edge for the $\overline{\text{ADTRG2}}$ and $\overline{\text{ADTRG1}}$ pins.

### 4.2.2 Interrupt Priority Registers A to I (IPRA to IPRI)

- IPRA to IPRK

Address: H'FF0529 to H'FF0531

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | IPRn[7:6] | | IPRn[5:4] | | IPRn[3:2] | | IPRn[1:0] | |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

(n = A to I)

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7<br>6 | IPRn[7:6] | Interrupt priority 7 and 6 | 00: Priority level 0 (lowest)<br>01: Priority level 1<br>10: Priority level 2<br>11: Priority level 3 (highest) | R/W |
| 5<br>4 | IPRn[5:4] | Interrupt priority 5 and 4 | 00: Priority level 0 (lowest)<br>01: Priority level 1<br>10: Priority level 2<br>11: Priority level 3 (highest) | R/W |
| 3<br>2 | IPRn[3:2] | Interrupt priority 3 and 2 | 00: Priority level 0 (lowest)<br>01: Priority level 1<br>10: Priority level 2<br>11: Priority level 3 (highest) | R/W |
| 1<br>0 | IPRn[1:0] | Interrupt priority 1 and 0 | 00: Priority level 0 (lowest)<br>01: Priority level 1<br>10: Priority level 2<br>11: Priority level 3 (highest) | R/W |

[Legend]
n = A to I

- IPR7 to IPR0 bits (Interrupt priority 7 to 0)

IPR are nine 8-bit readable/writable registers that set priorities (levels 3 to 0) for interrupt sources other than Nonmaskable interrupt request (NMI). The correspondence between interrupt sources and IPR settings is shown in table 4.2. Setting a value in the range from H'0 to H'3 in the 2-bit groups of bits 7 and 6, 5 and 4, 3 and 2, and, 1 and 0 determines the priority of the corresponding interrupt requests.

chanhvinh.com



**Table 4.2 Correspondence between Interrupt Sources and IPR Settings**

| Register | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| IPRA | Flash memory | | WDT | | LVD | | CPG | |
| IPRB | IRQ0 | | IRQ1 | | IRQ2 | | IRQ3 | |
| IPRC | IRQ4 | | IRQ5 | | IRQ6 | | IRQ7 | |
| IPRD | A/D converter unit 1 | | A/D converter unit 2*[1] | | DTC | | ELC | |
| IPRE | SCI3 channel 1 | | SCI3 channel 2 | | SCI3 channel 3 | | — | |
| IPRF | — | | — | | IIC2/SSU | | — | |
| IPRG | — | | Timer RA | | Timer RB | | Timer RC*[2] | |
| IPRH | Timer RD unit 0 channel 0 | | Timer RD unit 0 channel 1 | | Timer RD unit 1 channel 2*[3] | | Timer RD unit 1 channel 3*[3] | |
| IPRI | Timer RE | | — | | Timer RG | | — | |

Notes: —: Reserved

1. Provided for the H8S/20223 and H8S/20235 Groups only.
2. Provided for the H8S/20103 and H8S/20115 Groups only.
3. Not provided for the H8S/20103 and H8S/20115 Groups.

chanhvinh.com



### 4.2.3 IRQ Enable Register (IER)

Address: H'FF0521

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | IRQ7E | IRQ6E | IRQ5E | IRQ4E | IRQ3E | IRQ2E | IRQ1E | IRQ0E |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | IRQ7E | IRQ7 enable | 0: IRQ7 interrupts are disabled.<br>1: IRQ7 interrupts are enabled. | R/W |
| 6 | IRQ6E | IRQ6 enable | 0: IRQ6 interrupts are disabled.<br>1: IRQ6 interrupts are enabled. | R/W |
| 5 | IRQ5E | IRQ5 enable | 0: IRQ5 interrupts are disabled.<br>1: IRQ5 interrupts are enabled. | R/W |
| 4 | IRQ4E | IRQ4 enable | 0: IRQ4 interrupts are disabled.<br>1: IRQ4 interrupts are enabled. | R/W |
| 3 | IRQ3E | IRQ3 enable | 0: IRQ3 interrupts are disabled.<br>1: IRQ3 interrupts are enabled. | R/W |
| 2 | IRQ2E | IRQ2 enable | 0: IRQ2 interrupts are disabled.<br>1: IRQ2 interrupts are enabled. | R/W |
| 1 | IRQ1E | IRQ1 enable | 0: IRQ1 interrupts are disabled.<br>1: IRQ1 interrupts are enabled. | R/W |
| 0 | IRQ0E | IRQ0 enable | 0: IRQ0 interrupts are disabled.<br>1: IRQ0 interrupts are enabled. | R/W |

chanhvinh.com



### 4.2.4 IRQ Sense Control Register H and L (ISCRH and ISCRL)

- ISCRH

Address: H'FF0522

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | [IRQ7SCB : IRQ7SCA] | | [IRQ6SCB : IRQ6SCA] | | [IRQ5SCB : IRQ5SCA] | | [IRQ4SCB : IRQ4SCA] | |
| Value after reset: | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

- ISCRL

Address: H'FF0523

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | [IRQ3SCB : IRQ3SCA] | | [IRQ2SCB : IRQ2SCA] | | [IRQ1SCB : IRQ1SCA] | | [IRQ0SCB : IRQ0SCA] | |
| Value after reset: | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

- ISCRH

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7<br>6 | [IRQ7SCB :<br>IRQ7SCA] | IRQ7 sense control B and A | 00: Reserved (setting prohibited)<br>01: Interrupt request is generated at falling edge of $\overline{\text{IRQ7}}$ input.<br>10: Interrupt request is generated at rising edge of $\overline{\text{IRQ7}}$ input.<br>11: Interrupt request is generated at both falling and rising edges of $\overline{\text{IRQ7}}$ input. | R/W |
| 5<br>4 | [IRQ6SCB :<br>IRQ6SCA] | IRQ6 sense control B and A | 00: Reserved (setting prohibited)<br>01: Interrupt request is generated at falling edge of $\overline{\text{IRQ6}}$ input.<br>10: Interrupt request is generated at rising edge of $\overline{\text{IRQ6}}$ input.<br>11: Interrupt request is generated at both falling and rising edges of $\overline{\text{IRQ6}}$ input. | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 3<br>2 | [IRQ5SCB :<br>IRQ5SCA] | IRQ5 sense control B and A | 00: Reserved (setting prohibited)<br>01: Interrupt request is generated at falling edge of $\overline{\text{IRQ5}}$ input.<br>10: Interrupt request is generated at rising edge of $\overline{\text{IRQ5}}$ input.<br>11: Interrupt request is generated at both falling and rising edges of $\overline{\text{IRQ5}}$ input. | R/W |
| 1<br>0 | [IRQ4SCB :<br>IRQ4SCA] | IRQ4 sense control B and A | 00: Reserved (setting prohibited)<br>01: Interrupt request is generated at falling edge of $\overline{\text{IRQ4}}$ input.<br>10: Interrupt request is generated at rising edge of $\overline{\text{IRQ4}}$ input.<br>11: Interrupt request is generated at both falling and rising edges of $\overline{\text{IRQ4}}$ input. | R/W |

chanhvinh.com



- ISCRL

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7<br>6 | [IRQ3SCB :<br>IRQ3SCA] | IRQ3 sense control B and A | 00: Reserved (setting prohibited)<br>01: Interrupt request is generated at falling edge of $\overline{\text{IRQ3}}$ input.<br>10: Interrupt request is generated at rising edge of $\overline{\text{IRQ3}}$ input.<br>11: Interrupt request is generated at both falling and rising edges of $\overline{\text{IRQ3}}$ input. | R/W |
| 5<br>4 | [IRQ2SCB :<br>IRQ2SCA] | IRQ2 sense control B and A | 00: Reserved (setting prohibited)<br>01: Interrupt request is generated at falling edge of $\overline{\text{IRQ2}}$ input.<br>10: Interrupt request is generated at rising edge of $\overline{\text{IRQ2}}$ input.<br>11: Interrupt request is generated at both falling and rising edges of $\overline{\text{IRQ2}}$ input. | R/W |
| 3<br>2 | [IRQ1SCB :<br>IRQ1SCA] | IRQ1 sense control B and A | 00: Reserved (setting prohibited)<br>01: Interrupt request is generated at falling edge of $\overline{\text{IRQ1}}$ input.<br>10: Interrupt request is generated at rising edge of $\overline{\text{IRQ1}}$ input.<br>11: Interrupt request is generated at both falling and rising edges of $\overline{\text{IRQ1}}$ input. | R/W |
| 1<br>0 | [IRQ0SCB :<br>IRQ0SCA] | IRQ0 sense control B and A | 00: Reserved (setting prohibited)<br>01: Interrupt request is generated at falling edge of $\overline{\text{IRQ0}}$ input.<br>10: Interrupt request is generated at rising edge of $\overline{\text{IRQ0}}$ input.<br>11: Interrupt request is generated at both falling and rising edges of $\overline{\text{IRQ0}}$ input. | R/W |

chanhvinh.com



### 4.2.5 IRQ Status Register (ISR)

Address: H'FF0524

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | IRQ7F | IRQ6F | IRQ5F | IRQ4F | IRQ3F | IRQ2F | IRQ1F | IRQ0F |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | IRQ7F | IRQ7 flag | [Setting condition]<br>• When the interrupt edge selected by ISCR occurs.<br>[Clearing conditions]<br>• When 1 is read from the bit and then 0 is written to the same bit.<br>• When IRQn interrupt exception handling is executed while falling, rising, or both-edge detection is set.<br>• When the DTC is activated by an IRQn interrupt and the DISEL bit in MRB of the DTC is 0. | R/W |
| 6 | IRQ6F | IRQ6 flag | | R/W |
| 5 | IRQ5F | IRQ5 flag | | R/W |
| 4 | IRQ4F | IRQ4 flag | | R/W |
| 3 | IRQ3F | IRQ3 flag | | R/W |
| 2 | IRQ2F | IRQ2 flag | | R/W |
| 1 | IRQ1F | IRQ1 flag | | R/W |
| 0 | IRQ0F | IRQ0 flag | | R/W |

chanhvinh.com



### 4.2.6 IRQ Noise Canceler Control Register (INCCR)

Address: H'FF0525

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | INCCR[5:4] | | INCCR[3:2] | | INCCR[1:0] | |
| Value after reset: | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | Bit Name | Initial Value | Description | R/W |
|---|---|---|---|---|
| 7<br>6 | — | Reserved | These bits are always read as 0. The write value should always be 0. | — |
| 5<br>4 | INCCR[5:4] | Noise canceling ability setting 5 and 4 for $\overline{\text{NMI}}$ pin | 00: Noise canceling ability 1<br>01: Noise canceling ability 2 (approximately 1.5 times as capable as noise canceling ability 1)<br>10: Noise canceling ability 3 (approximately 2.7 times as capable as noise canceling ability 1)<br>11: Noise canceling ability 4 (approximately 5 times as capable as noise canceling ability 1) | R/W |
| 3<br>2 | INCCR[3:2] | Noise canceling ability setting 3 and 2 for $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ4}}$ pins | 00: Noise canceling ability 1<br>01: Noise canceling ability 2 (approximately 1.5 times as capable as noise canceling ability 1)<br>10: Noise canceling ability 3 (approximately 2.7 times as capable as noise canceling ability 1)<br>11: Noise canceling ability 4 (approximately 5 times as capable as noise canceling ability 1) | R/W |
| 1<br>0 | INCCR[1:0] | Noise canceling ability setting 1 and 0 for $\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$ pins | 00: Noise canceling ability 1<br>01: Noise canceling ability 2 (approximately 1.5 times as capable as noise canceling ability 1)<br>10: Noise canceling ability 3 (approximately 2.7 times as capable as noise canceling ability 1)<br>11: Noise canceling ability 4 (approximately 5 times as capable as noise canceling ability 1) | R/W |

Note: Noise canceling ability varies according to the manufacturing condition, temperature, and $V_{CC}$. When $\overline{\text{NMI}}$ or $\overline{\text{IRQ}}$ pin is in use, input pulses over longer intervals than the specified minimum interval for input.

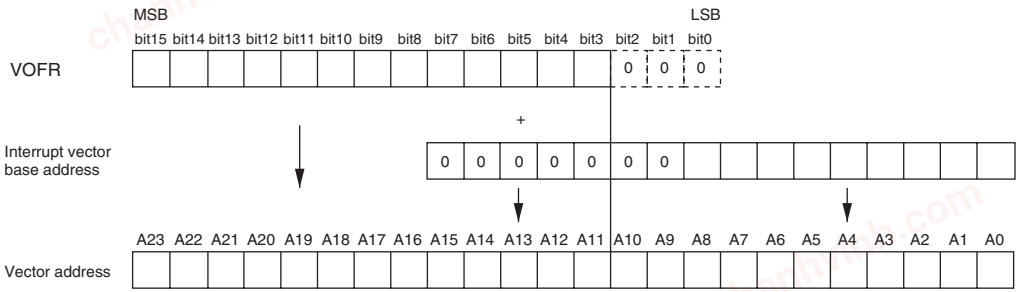
chanhvinh.com



### 4.2.7 Interrupt Vector Offset Register (VOFR)

Address: H'FF0526

| Bit: | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | bit15 | bit14 | bit13 | bit12 | bit11 | bit10 | bit9 | bit8 | bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |




VOFR is a 16-bit readable/writable register that sets an offset for an interrupt vector address. Interrupt vector areas other than the trace interrupt area and trap instruction interrupt area can be varied with the offset. The upper 13 bits are used to set the offset for the interrupt vector address (A23 to A11). Bits 2 to 0 are reserved. The write value should always be 0. This register also can be accessed in 8-bit units.

The vector address can be obtained by adding the VOFR value to the interrupt vector base address as shown above, except for the trace interrupt and trap instruction interrupt.

This register is initialized to H'0000 by a reset.

Note: When the DTC is used while the interrupt vector offset register (VOFR) holds an offset-address setting, the VOFR setting has no effect. The DTC refers to the vector definition in the vector address area defined by the initial value (H'0000) of VOFR, which leads to the execution of exception processing.
Whenever the DTC is used, set the VOFR to H'0000 (its default value).

chanhvinh.com



### 4.2.8 Event Link Interrupt Control Status Register (ELCSR)

Address: H'FF0528

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | ELIE2 | ELIE1 | ELF2 | ELF1 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 to 4 | — | Reserved | These bits are read as 0. The write value should be 0. | — |
| 3 | ELIE2 | ELC interrupt 2 enable | 0: ELF2 interrupts are disabled.<br>1: ELF2 interrupts are enabled. | R/W |
| 2 | ELIE1 | ELC interrupt 1 enable | 0: ELF1 interrupts are disabled.<br>1: ELF1 interrupts are enabled. | R/W |
| 1 | ELF2 | ELC interrupt flag 2 | [Setting condition]<br>• When the event selected by ELSR30 occurs*[1]<br>[Clearing conditions]<br>• When 1 is read from this bit and then 0 is written to the same bit.<br>• When the DTC is activated by an ELF2 interrupt, and the DISEL bit in MRB of the DTC is 0.*[2] | R/W |
| 0 | ELF1 | ELC interrupt flag 1 | [Setting condition]<br>• When the event selected by ELSR12 occurs*[1]<br>[Clearing conditions]<br>• When 1 is read from this bit and then 0 is written to the same bit.<br>• When the DTC is activated by an ELF1 interrupt, and the DISEL bit in MRB of the DTC is 0.*[2] | R/W |

Notes: 1. For details, see section 12, Event Link Controller.
2. When the DTC is activated by an ELF1 or ELF2 interrupt, the event link source module is not affected.

chanhvinh.com



## 4.3 Interrupt Sources

### 4.3.1 External Interrupt sources

There are nine external interrupts: NMI and IRQ7 to IRQ0. These external interrupts can be used to cause the device to exit from standby mode.

#### (1) NMI Interrupt

The nonmaskable interrupt request (NMI) is the highest-priority interrupt, and always accepted by the CPU regardless of the interrupt control mode or the status of the CPU interrupt mask bits. The NMIEG bit in INTCR can be used to select whether an interrupt is requested at a rising edge or a falling edge on the $\overline{\text{NMI}}$ pin.

#### (2) IRQ7 to IRQ0 Interrupts

Interrupts IRQ7 to IRQ0 are generated by an input signal at pins $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$. IRQ7 to IRQ0 interrupts have the following features:

- Using ISCR, it is possible to select whether an interrupt on the $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ input pins is generated by a falling edge, rising edge, or both edges.
- Enabling or disabling of interrupt requests IRQ7 to IRQ0 can be selected with IER.
- The interrupt priority level can be set with IPR.
- The status of interrupt requests IRQ7 to IRQ0 is indicated in ISR. ISR flags can be cleared to 0 by software.

A block diagram of interrupts IRQ7 to IRQ0 is shown in figure 4.2.

chanhvinh.com



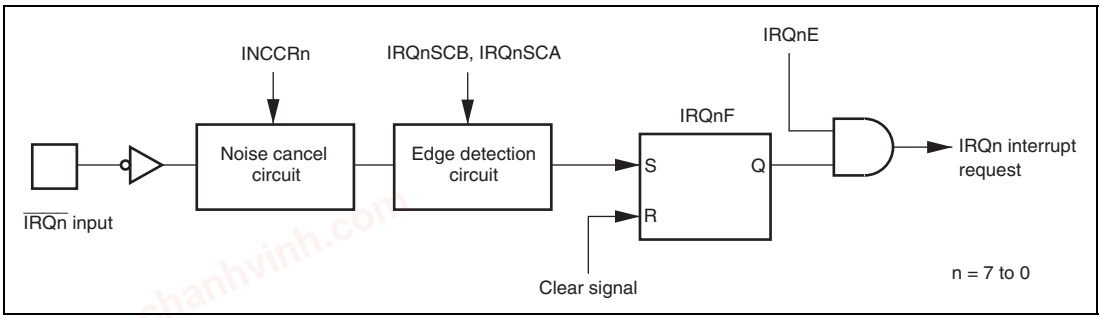



**Figure 4.2 Block Diagram of IRQ7 to IRQ0 Interrupt**

### 4.3.2 Internal Interrupts

The sources for internal interrupts from on-chip peripheral modules have the following features:

- For each on-chip peripheral module there are flags that indicate the interrupt request status, and enable bits that select enabling or disabling of these interrupts. They can be controlled independently. When the enable bit is set to 1, an interrupt request is issued to the interrupt controller.
- The interrupt priority level can be set by means of IPR.
- The DTC can be activated by a peripheral module interrupt request.
- When the DTC is activated by an interrupt request, it is not affected by the interrupt control mode or CPU interrupt mask bit.

chanhvinh.com



## 4.4 Interrupt Exception Handling Vector Table

Table 4.3 shows interrupt exception handling sources, vector addresses, and interrupt priorities.

For default priorities, the lower the vector number, the higher the priority. When interrupt control mode 2 is set, priorities among modules can be changed by the IPR. Modules set at the same priority will conform to their default priorities. Priorities within a module are fixed.

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

chanhvinh.com



**Table 4.3 Interrupt Sources, Vector Addresses, and Interrupt Priorities**

| Origin of Interrupt Source | Interrupt Source | Vector Number | Vector Address*[1] | DTCER | IPR | Priority |
|---|---|---|---|---|---|---|
| $\overline{\text{RES}}$ Pin | Reset | 0 | H'0000 to H'0003 | — | — | High |
| WDT | 1. $\overline{\text{RES}}$ pin reset | | | | | |
| LVD | 2. WDT overflow | | | | | |
| | 3. LVD reset | | | | | |
| | 4. Software reset | | | | | |
| — | Reserved | 1 to 4 | H'0004 to H'0013 | — | — | |
| CPU | Trace | 5 | H'0014 to H'0017 | — | — | |
| — | Reserved | 6 | H'0018 to H'001B | — | — | |
| External pin | NMI | 7 | H'001C to H'001F | — | — | |
| CPU | TRAPA0 (TRAPA #0 instruction) | 8 | H'0020 to H'0023 | — | — | |
| | TRAPA0 (TRAPA #1 instruction) | 9 | H'0024 to H'0027 | — | — | |
| | TRAPA0 (TRAPA #2 instruction) | 10 | H'0028 to H'002B | — | — | |
| | TRAPA0 (TRAPA #3 instruction) | 11 | H'002C to H'002F | — | — | |
| — | Reserved | 12 to 15 | H'0030 to H'003F | — | — | |
| FLASH | IFMBSYA (access when flash memory busy) | 16 | H'0040 to H'0043 | — | — | |
| | IFLRDY (flash memory ready) | 17 | H'0044 to H'0047 | — | IPRA7 and IPRA6 | |
| WDT | IWDT (WDT periodic interrupt) | 18 | H'0048 to H'004B | — | IPRA5 and IPRA4 | |
| LVD | ILVINT1 (low-voltage detected interrupt 1) | 19 | H'004C to H'004F | — | IPRA3 and IPRA2 | |
| | ILVINT2 (low-voltage detected interrupt 2) | 20 | H'0050 to H'0053 | — | | |
| CPG | ICKSW (clock switching interrupt) | 21 | H'0054 to H'0057 | — | IPRA1 and IPRA0 | Low |

chanhvinh.com



| Origin of Interrupt Source | Interrupt Source | Vector Number | Vector Address*1 | DTCER | IPR | Priority |
|---|---|---|---|---|---|---|
| External pin | IRQ0 | 22 | H'0058 to H'005B | DTCEA7 | IPRB7 and IPRB6 | High |
| | IRQ1 | 23 | H'005C to H'005F | DTCEA6 | IPRB5 and IPRB4 | |
| | IRQ2 | 24 | H'0060 to H'0063 | DTCEA5 | IPRB3 and IPRB2 | |
| | IRQ3 | 25 | H'0064 to H'0067 | DTCEA4 | IPRB1 and IPRB0 | |
| | IRQ4 | 26 | H'0068 to H'006B | DTCEA3 | IPRC7 and IPRC6 | |
| | IRQ5 | 27 | H'006C to H'006F | DTCEA2 | IPRC5 and IPRC4 | |
| | IRQ6 | 28 | H'0070 to H'0073 | DTCEA1 | IPRC3 and IPRC2 | |
| | IRQ7 | 29 | H'0074 to H'0077 | DTCEA0 | IPRC1 and IPRC0 | |
| A/D converter unit 1 | IADEND_1 (conversion end) | 30 | H'0078 to H'007B | DTCEB7 | IPRD7 and IPRD6 | |
| | IADCMP_1 (compare condition match) | 31 | H'007C to H'007F | DTCEB6 | | |
| A/D converter unit 2*2 | IADEND_2 (conversion end) | 32 | H'0080 to H'0083 | DTCEB5 | IPRD5 and IPRD4 | |
| | IADCMP_2 (compare condition match) | 33 | H'0084 to H'0087 | DTCEB4 | | |
| DTC | ISWDTEND (data transfer end) | 34 | H'0088 to H'008B | — | IPRD3 and IPRD2 | |
| ELC | ELC1FP (ELSR12 event generation) | 35 | H'008C to H'008F | DTCEB3 | IPRD1 and IPRD0 | |
| | ELC2FP (ELSR30 event generation) | 36 | H'0090 to H'0093 | DTCEB2 | | Low |

chanhvinh.com



| Origin of Interrupt Source | Interrupt Source | Vector Number | Vector Address*[1] | DTCER | IPR | Priority |
|---|---|---|---|---|---|---|
| SCI3 channel 1 | SCI3_1 ERI<br>1. Overrun error<br>2. Parity error<br>3. Framing error | 37 | H'0094 to H'0097 | — | IPRE7 and IPRE6 | High |
| | SCI3_1 RXI | 38 | H'0098 to H'009B | DTCEB1 | | |
| | SCI3_1 TXI | 39 | H'009C to H'009F | DTCEB0 | | |
| | SCI3_1 TEI | 40 | H'00A0 to H'00A3 | — | | |
| SCI3 channel 2 | SCI3_2 ERI<br>1. Overrun error<br>2. Parity error<br>3. Framing error | 41 | H'00A4 to H'00A7 | — | IPRE5 and IPRE4 | |
| | SCI3_2 RXI | 42 | H'00A8 to H'00AB | DTCEC7 | | |
| | SCI3_2 TXI | 43 | H'00AC to H'00AF | DTCEC6 | | |
| | SCI3_2 TEI | 44 | H'00B0 to H'00B3 | — | | |
| SCI3 channel 3 | SCI3_3 ERI<br>1. Overrun error<br>2. Parity error<br>3. Framing error | 45 | H'00B4 to H'00B7 | — | IPRE3 and IPRE2 | |
| | SCI3_3 RXI | 46 | H'00B8 to H'00BB | DTCEC5 | | |
| | SCI3_3 TXI | 47 | H'00BC to H'00BF | DTCEC4 | | |
| | SCI3_3 TEI | 48 | H'00C0 to H'00C3 | — | | |
| — | Reserved | 49 to 58 | H'00C4 to H'00EB | — | — | |
| IIC2/SSU | 1. IIC-BUS mode<br>— NAKI<br>— STPI<br>2. Clock synchronous mode<br>— Overrun<br>3. SSU mode<br>— Overrun (OEI)<br>— Conflict (CEI) | 59 | H'00EC to H'00EF | — | IPRF3 and IPRF2 | |
| | RXI | 60 | H'00F0 to H'00F3 | DTCED7 | | |
| | TXI | 61 | H'00F4 to H'00F7 | DTCED6 | | |
| | TEI | 62 | H'00F8 to H'00FB | — | | Low |

chanhvinh.com



| Origin of Interrupt Source | Interrupt Source | Vector Number | Vector Address*[1] | DTCER | IPR | Priority |
|---|---|---|---|---|---|---|
| — | Reserved | 63 to 68 | H'00FC to H'0113 | — | — | High |
| Timer RA/ HW-LIN | 1. Timer RA<br>— ITAUD<br>2. HW-LIN<br>— Bus conflict detection (BCDCT)<br>— Sync Break detection (SBDCT)<br>— Sync Field measurement end (SFDCT) | 69 | H'0114 to H'0117 | — | IPRG5 and IPRG4 | |
| Timer RB | ITBUD | 70 | H'0118 to H'011B | — | IPRG3 and IPRG2 | |
| Timer RC*[3] | ITCMA (input capture A/compare match A) | 71 | H'011C to H'011F | DTCED3 | IPRG1 and IPRG0 | |
| | ITCMB (input capture B/compare match B) | 72 | H'0120 to H'0123 | DTCED2 | | |
| | ITCMC (input capture C/compare match C) | 73 | H'0124 to H'0127 | DTCED1 | | |
| | ITCMD (input capture D/compare match D) | 74 | H'0128 to H'012B | DTCED0 | | |
| | ITCOV counter overflow | 75 | H'012C to H'012F | — | | |
| Timer RD unit 0 channel 0 | ITDMA0_0 (input capture A/compare match A) | 76 | H'0130 to H'0133 | DTCEE7 | IPRH7 and IPRH6 | |
| | ITDMB0_0 (input capture B/compare match B) | 77 | H'0134 to H'0137 | DTCEE6 | | Low |

chanhvinh.com



| Origin of Interrupt Source | Interrupt Source | Vector Number | Vector Address*[1] | DTCER | IPR | Priority |
|---|---|---|---|---|---|---|
| Timer RD unit 0 channel 0 | ITDMC0_0 (input capture C/compare match C) | 78 | H'0138 to H'013B | DTCEE5 | IPRH7 and IPRH6 | High |
| | ITCMD0_0 (input capture D/compare match D) | 79 | H'013C to H'013F | DTCEE4 | | |
| | ITDOV0_0 overflow | 80 | H'0140 to H'0143 | — | | |
| | ITDUD0_0 underflow | 81 | H'0144 to H'0147 | — | | |
| Timer RD unit 0 channel 1 | ITDMA0_1 (input capture A/compare match A) | 82 | H'0148 to H'014B | DTCEE3 | IPRH5 and IPRH4 | |
| | ITDMB0_1 (input capture B/compare match B) | 83 | H'014C to H'014F | DTCEE2 | | |
| | ITDMC0_1 (input capture C/compare match C) | 84 | H'0150 to H'0153 | DTCEE1 | | |
| | ITCMD0_1 (input capture D/compare match D) | 85 | H'0154 to H'0157 | DTCEE0 | | |
| | ITDOV0_1 overflow | 86 | H'0158 to H'015B | — | | |
| Timer RD unit 1 channel 2*[4] | ITDMA1_2 (input capture A/compare match A) | 87 | H'015C to H'015F | DTCEF7 | IPRH3 and IPRH2 | |
| | ITDMB1_2 (input capture B/compare match B) | 88 | H'0160 to H'0163 | DTCEF6 | | |
| | ITDMC1_2 (input capture C/compare match C) | 89 | H'0164 to H'0167 | DTCEF5 | | |
| | ITCMD1_2 (input capture D/compare match D) | 90 | H'0168 to H'016B | DTCEF4 | | |
| | ITDOV1_2 overflow | 91 | H'016C to H'016F | — | | |
| | ITDUD1_2 underflow | 92 | H'0170 to H'0173 | — | | Low |

chanhvinh.com



| Origin of Interrupt Source | Interrupt Source | Vector Number | Vector Address*[1] | DTCER | IPR | Priority |
|---|---|---|---|---|---|---|
| Timer RD unit 1 channel 3*[4] | ITDMA1_3 (input capture A/compare match A) | 93 | H'0174 to H'0177 | DTCEE3 | IPRH1 and IPRH0 | High |
| | ITDMB1_3 (input capture B/compare match B) | 94 | H'0178 to H'017B | DTCEE2 | | |
| | ITDMC1_3 (input capture C/compare match C) | 95 | H'017C to H'017F | DTCEE1 | | |
| | ITCMD1_3 (input capture D/compare match D) | 96 | H'0180 to H'0183 | DTCEE0 | | |
| | ITDOV1_3 overflow | 97 | H'0184 to H'0187 | — | | |
| — | Reserved | 98, 99 | H'0188 to H'018F | — | — | |
| Timer RE | Second interrupt | 100 | H'0190 to H'0193 | DTCEG4 | IPRI7 and IPRI6 | |
| | Minute interrupt | 101 | H'0194 to H'0197 | DTCEG3 | | |
| | Hour interrupt | 102 | H'0198 to H'019B | DTCEG2 | | |
| | Day interrupt | 103 | H'019C to H'019F | DTCEG1 | | |
| | Week interrupt | 104 | H'01A0 to H'01A3 | DTCEG0 | | |
| | Compare match | 105 | H'01A4 to H'01A7 | — | | |
| — | Reserved | 106 to 108 | H'01A8 to H'01B3 | — | — | |
| Timer RG | ITGMA (input capture A/compare match A) | 109 | H'01B4 to H'01B7 | DTCEH3 | IPRI3 and IPRI2 | |
| | ITGMB (input capture B/compare match B) | 110 | H'01B8 to H'01BB | DTCEH2 | | |
| | ITGOV | 111 | H'01BC to H'01BF | — | | |
| | ITGUD | 112 | H'01C0 to H'01C3 | — | | Low |

Notes: 1. Lower 16 bits of the vector address when VOFR = H'0000
2. Provided for the H8S/20223 and H8S/20235 Groups only. This area is reserved for the other groups.
3. Provided for the H8S/20103 and H8S/20115 Groups only. This area is reserved for the other groups.
4. Not provided for the H8S/20103 and H8S/20115 Groups.

chanhvinh.com



## 4.5 Interrupt Control Modes and Interrupt Operation

The interrupt controller has two modes: interrupt control mode 0 and interrupt control mode 2. Interrupt operations differ depending on the interrupt control mode. The interrupt control mode is selected by INTCR. Table 4.4 shows the differences between interrupt control mode 0 and interrupt control mode 2.

**Table 4.4 Interrupt Control Modes**

| Interrupt Control Mode | Priority Setting Registers | Interrupt Mask Bits | Description |
|---|---|---|---|
| 0 | Default | I | The priorities of interrupt sources are fixed at the default settings.<br>Interrupt sources except for NMI is masked by the I bit. |
| 2 | IPR | I1 and I0 | Four priority levels except for NMI can be set with IPR.<br>Four-level interrupt mask control is performed by bits I1 and I0. |

### 4.5.1 Interrupt Control Mode 0

In interrupt control mode 0, interrupt requests except for NMI is masked by the I bit in CCR of the CPU. Figure 4.3 shows a flowchart of the interrupt acceptance operation in this case.

1. If an interrupt source occurs when the corresponding interrupt enable bit is set to 1, an interrupt request is sent to the interrupt controller.
2. If the I bit is set to 1, only an NMI interrupt is accepted, and other interrupt requests are held pending. If the I bit is cleared, an interrupt request is accepted.
3. When interrupt requests are sent to the interrupt controller, the highest-ranked interrupt request according to the priority system is accepted, and other interrupt requests are retained.
4. When the CPU accepts an interrupt request, it starts interrupt exception handling after execution of the current instruction has been completed.
5. The PC and CCR are saved to the stack area by interrupt exception handling. The PC saved on the stack shows the address of the first instruction to be executed after returning from the interrupt handling routine.
6. The I bit in CCR is set to 1. This masks all interrupts except NMI.
7. The CPU generates a vector address for the accepted interrupt request and starts execution of the interrupt handling routine at the address indicated by the contents of the start address in the vector table.

chanhvinh.com



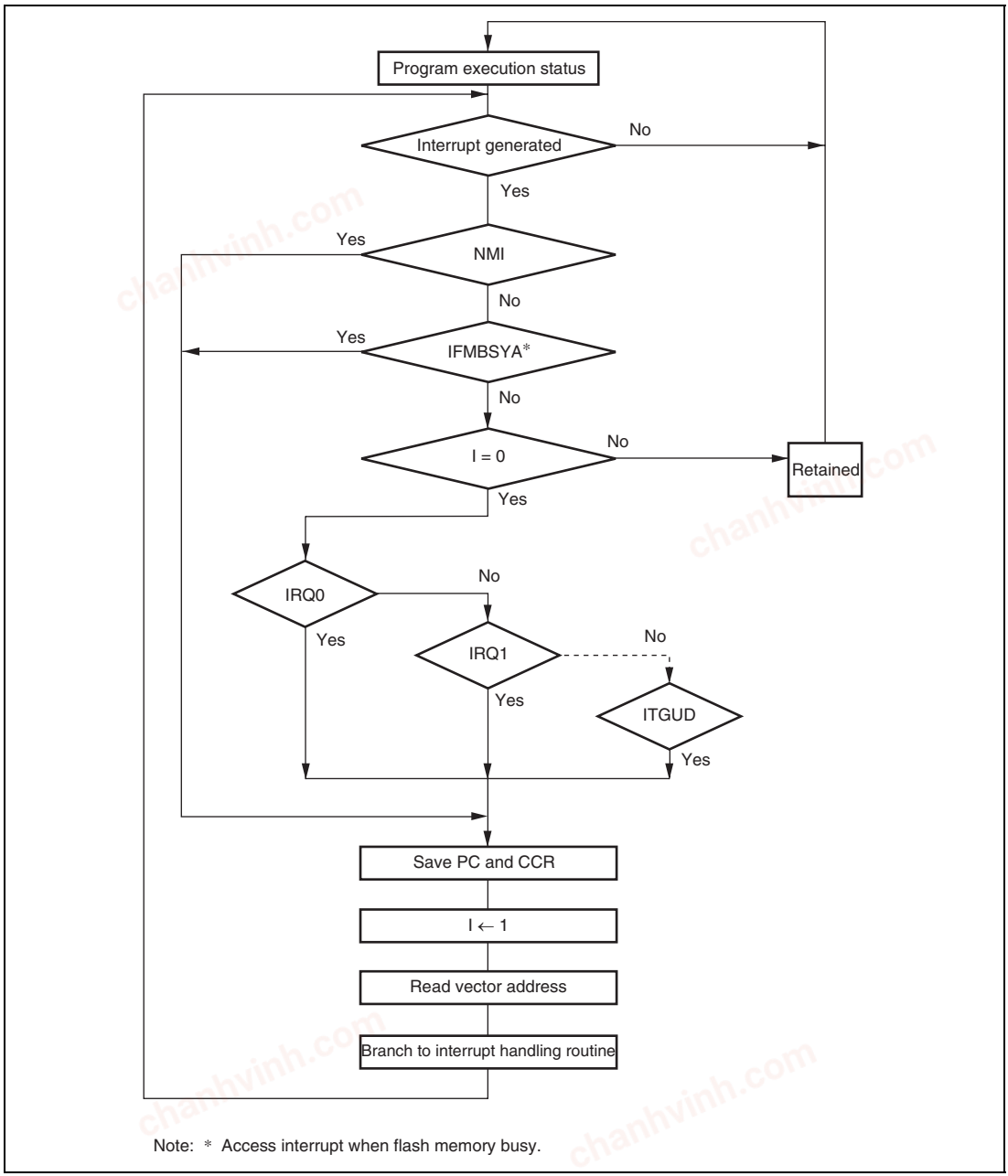



**Figure 4.3 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 0**

chanhvinh.com



### 4.5.2 Interrupt Control Mode 2

In interrupt control mode 2, mask control is executed in four levels for interrupt requests except NMI by comparing the EXR interrupt mask level (I1 and I0 bits*) in the CPU and the IPR setting. Figure 4.4 shows a flowchart of the interrupt acceptance operation.

1. If an interrupt source occurs when the corresponding interrupt enable bit is set to 1, an interrupt request is sent to the interrupt controller.
2. When interrupt requests are sent to the interrupt controller, the interrupt with the highest priority according to the interrupt priority levels set in IPR is selected, and lower-priority interrupt requests are held pending. If the same priority are generated at the same time, the interrupt request is selected according to the default priority system shown in table 4.3.
3. Next, the priority of the selected interrupt request is compared with the interrupt mask level set in EXR. An interrupt request with a priority no higher than the mask level set at that time is held pending, and only an interrupt request with a priority higher than the interrupt mask level is accepted.
4. When the CPU accepts an interrupt request, it starts interrupt exception handling after execution of the current instruction has been completed.
5. The PC, CCR, and EXR are saved to the stack area by interrupt exception handling. The PC saved on the stack shows the address of the first instruction to be executed after returning from the interrupt handling routine.
6. The T bit in EXR is cleared to 0. The interrupt mask level is rewritten with the priority level of the accepted interrupt. If the accepted interrupt is NMI, the interrupt mask level is set to H'3.
7. The CPU generates a vector address for the accepted interrupt request and starts execution of the interrupt handling routine at the address indicated by the contents of the start address in the vector table.

Note: * The I2 bit does not affect the mask control.

chanhvinh.com



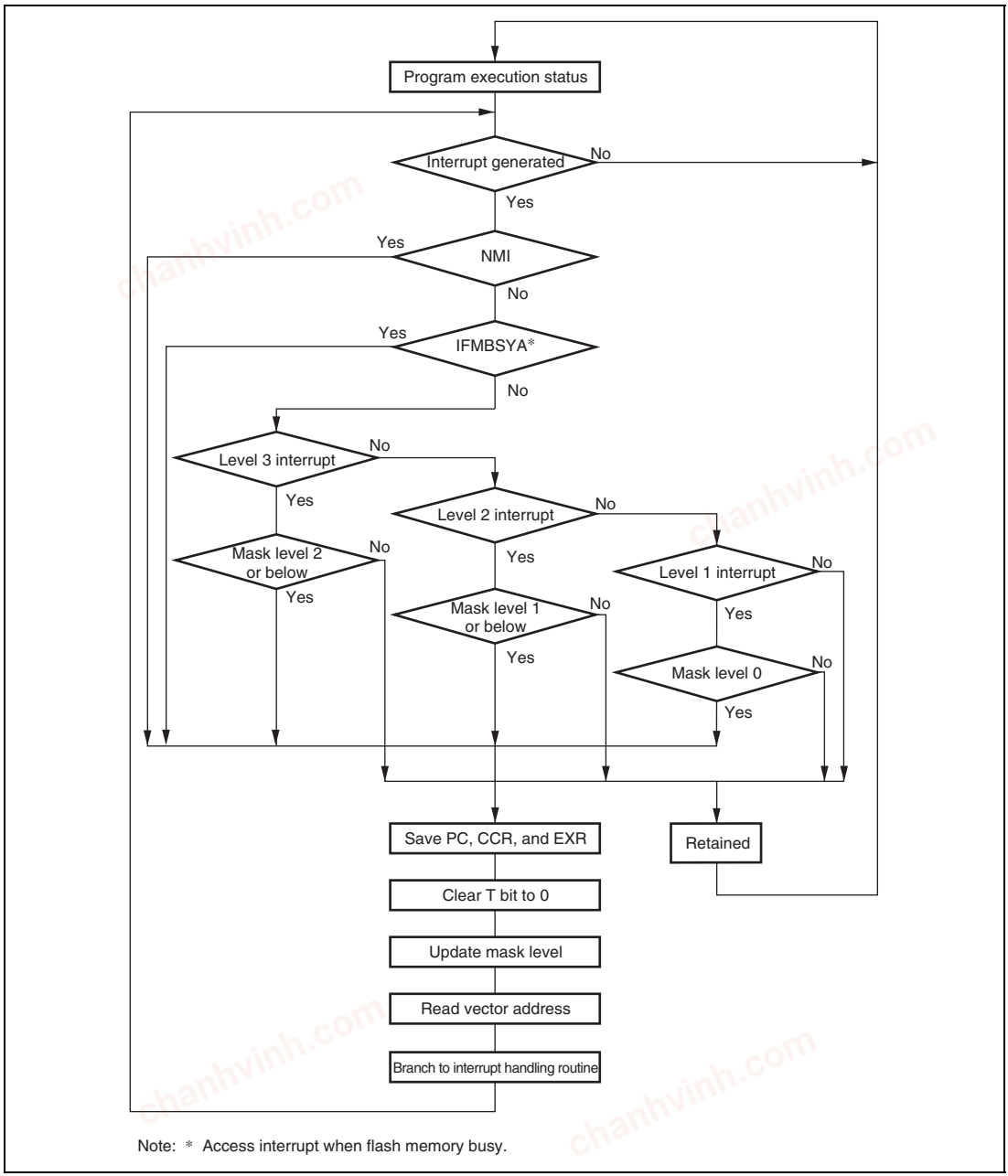



**Figure 4.4 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 2**

chanhvinh.com



### 4.5.3 Interrupt Exception Handling Sequence

Figure 4.5 shows the interrupt exception handling sequence. The example shown is for the case where interrupt control mode 0, the program area, and stack area are in on-chip memory.

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

chanhvinh.com



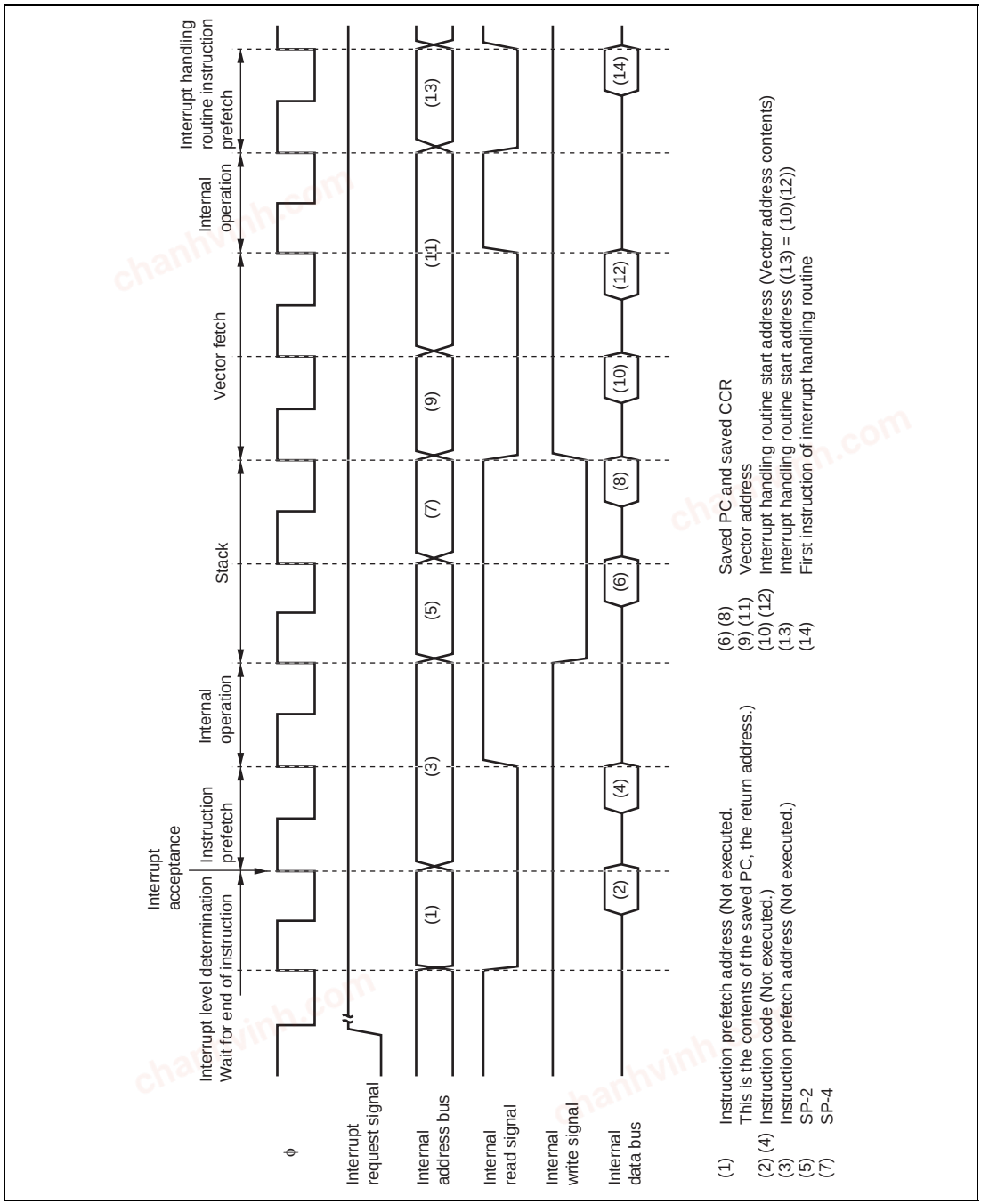



**Figure 4.5 Interrupt Exception Handling**

chanhvinh.com



### 4.5.4 Interrupt Response Time

Table 4.5 shows interrupt response time, the interval between generation of an interrupt request and execution of the first instruction in the interrupt handling routine.

**Table 4.5 Interrupt Response Times**

| No. | Execution Status | Interrupt Control Mode 0 | Interrupt Control Mode 2 |
|---|---|---|---|
| 1 | Interrupt priority determination*1 | 3 | |
| 2 | Number of wait states until executing instruction ends*2 | 1 to 21 | |
| 3 | PC, CCR, EXR stack | 2 | 3 |
| 4 | Vector fetch | 2 | |
| 5 | Instruction fetch*3 | 2 | |
| 6 | Internal processing*4 | 2 | |
| Total (using on-chip memory) | | 12 to 32 | 13 to 33 |

Notes: 1. Two states in case of internal interrupt
2. Refers to MULXS and DIVXS instructions.
3. Prefetch after interrupt acceptance and interrupt handling routine prefetch
4. Internal processing after interrupt acceptance and internal processing after vector fetch

### 4.5.5 DTC Activation by Interrupt

The DTC can be activated by an interrupt request. In this case, the following options are available:

1. Interrupt request to CPU
2. Activation request to DTC
3. Both of the above

For details of interrupt requests that can be used to activate the DTC, see table 4.3 and section 11, Data Transfer Controller (DTC).

chanhvinh.com



## 4.6 Usage Notes

### 4.6.1 Conflict between Interrupt Generation and Disabling

When an interrupt enable bit is cleared to 0 to mask interrupt requests, the masking becomes effective after execution of the instruction. When an interrupt enable bit is cleared to 0 by an instruction such as BCLR or MOV, if an interrupt is generated during execution of the instruction, the interrupt concerned is still enabled on completion of the instruction, and so interrupt exception handling for that interrupt will be executed after completion of the instruction. However, if there is an interrupt request of higher priority than that interrupt, interrupt exception handling with the higher-priority interrupt is executed, and that lower-priority interrupt will be ignored. The same also applies when an interrupt source flag is cleared to 0. Figure 4.6 shows an example in which the IRQ0E bit in IER is cleared to 0. The above conflict does not occur if an enable bit or interrupt source flag is cleared to 0 while the interrupt is masked.

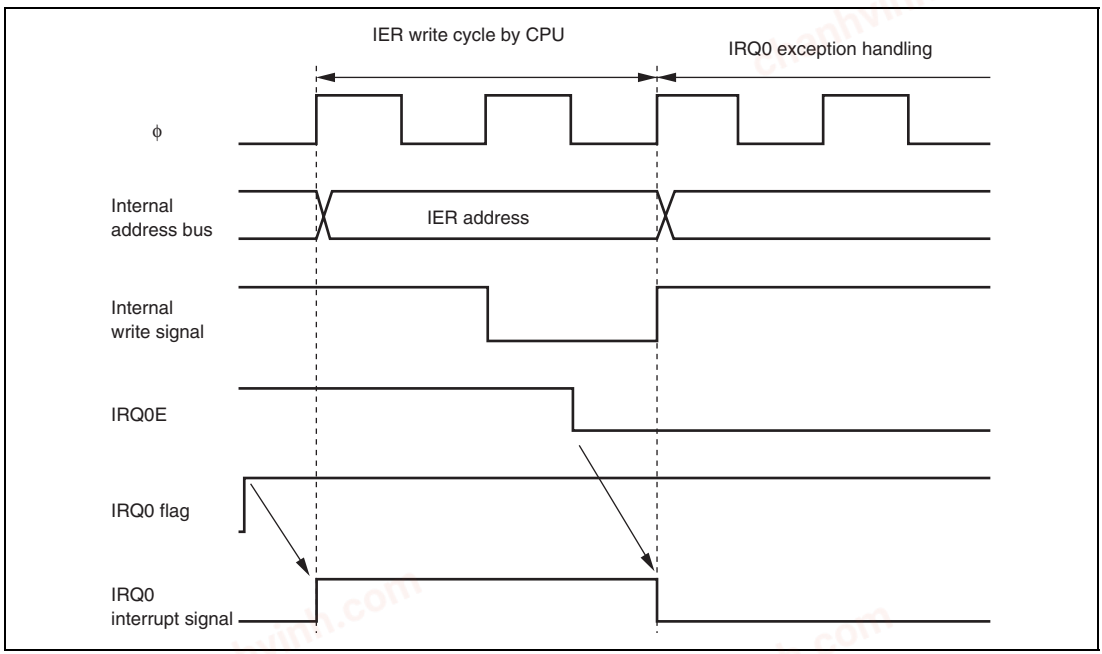



**Figure 4.6 Conflict between Interrupt Generation and Disabling**

chanhvinh.com



### 4.6.2 Instructions that Disable Interrupts

Instructions that disable interrupts are LDC, ANDC, ORC, and XORC. After any of these instructions is executed, all interrupts including NMI are disabled and the next instruction is always executed. When the I bit is set by one of these instructions, the new value becomes valid after two states that execution of the instruction ends.

### 4.6.3 Time when Interrupts are Disabled

There are time when interrupt acceptance is disabled by the interrupt controller. The interrupt controller disables interrupt acceptance for a 3-state period after the CPU has updated the mask level with an LDC, ANDC, ORC, or XORC instruction.

### 4.6.4 Interrupts during Execution of EEPMOV Instruction

Interrupt operation differs between the EEPMOV.B instruction and the EEPMOV.W instruction.

With the EEPMOV.B instruction, an interrupt request (including NMI) issued during the transfer is not accepted until the transfer is completed.

With the EEPMOV.W instruction, if an interrupt request is issued during the transfer, interrupt exception handling starts at a break in the transfer cycle. The PC value saved on the stack in this case is the address of the next instruction. Therefore, if an interrupt is generated during execution of an EEPMOV.W instruction, the following coding should be used.

```
L1:   EEPMOV.W
      MOV.W     R4,R4
      BNE       L1
```

chanhvinh.com



### 4.6.5 Changing PMR, ISCRH, ISCRL and INCCR

When the PMR, ISCRH, ISCRL, and INCCR are modified to change an IRQ7 to IRQ0 interrupt function, the interrupt request flag bit may be set to 1 at an unintended time. To prevent this, the pin function should be changed when the interrupt request is disabled, then the interrupt request flag should be cleared to 0 after a specific interval time*.

Figure 4.7 shows the procedure to modify PMR (port mode register), ISCRH, ISCRL, and INCCR and clear the interrupt request flags.

Note: Two states + a minimum interval for input ($t_{IH}$/$t_{IL}$)

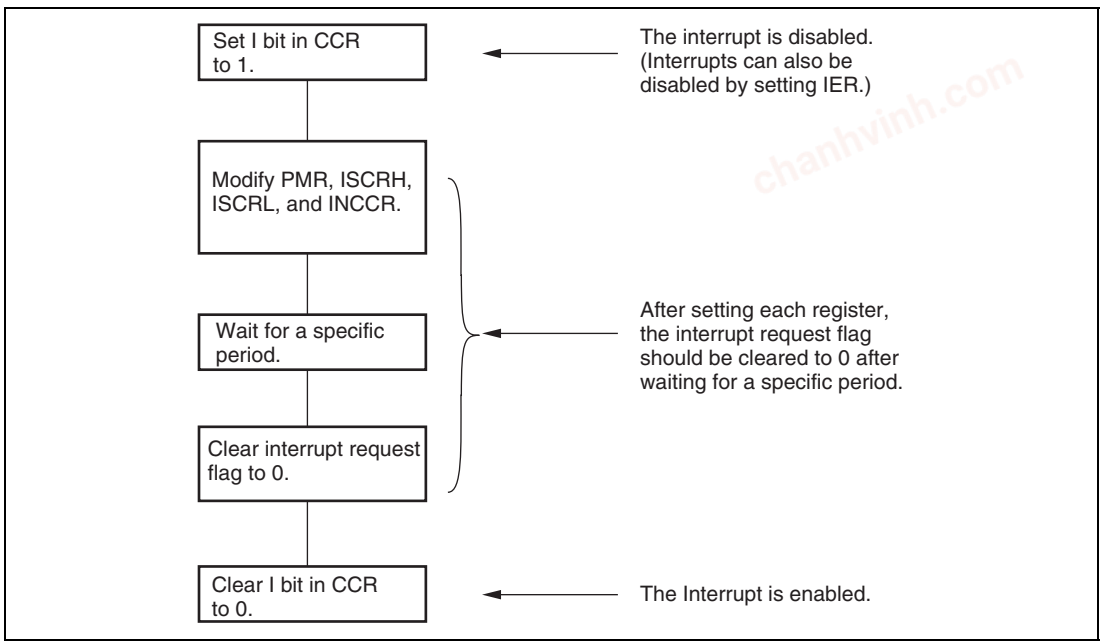



**Figure 4.7 Procedure to Modify PMR, ISCRH, ISCRL, and INCCR and Clear Interrupt Request Flag**

### 4.6.6 IRQ Status Register (ISR)

Depending on the pin state after a reset, IRQnF may be set to 1. Therefore, always read ISR and clear it to 0 after resets.

chanhvinh.com



### 4.6.7 $\overline{\text{NMI}}$ Pin

The $\overline{\text{NMI}}$ pin is also used to set up entry to boot mode on exit from the reset state. In using the $\overline{\text{NMI}}$ pin, note that the low-level should not be being applied to the $\overline{\text{NMI}}$ pins on exit from the reset state (including power-on reset). In general, it is recommended that the connection of a pull-up resistor to the $\overline{\text{NMI}}$ pin.

chanhvinh.com



# Section 5 Clock Pulse Generator

The clock pulse generator is comprised of the main oscillator, a duty correction circuit, a low-speed OCO (OCO: on-chip oscillator), a sub-oscillator, a clock selection circuit, a system clock divider, a PSC divider for peripheral modules, and a ϕs divider for the bus master and memory.

Table 5.1 lists clock source symbols and their meanings used in this manual.

**Table 5.1 Clock Source Symbols**

| Symbol | Description |
|---|---|
| ϕloco | Low-speed OCO output |
| ϕosc | Main oscillator output clock |
| ϕsub | Sub-oscillator output clock |
| ϕhigh | High-speed clock (ϕosc) |
| ϕlow | Low-speed clock (ϕloco or ϕsub) |
| ϕbase | System reference clock |
| ϕ | System operation clock |
| ϕs | Bus master operation clock |

chanhvinh.com



## 5.1 Overview

- Choice of three clock sources:
  ϕloco, ϕsub, and ϕosc
- Main oscillation backup function
  By detecting a ϕosc stop, it is possible to automatically switch the system clock to ϕlow.
- Clock switching interrupt function
  When the system clock is switched from ϕosc to ϕloco, a CPU interrupt can be generated if enabled.

Figure 5.1 shows a block diagram of the clock pulse generation circuit.

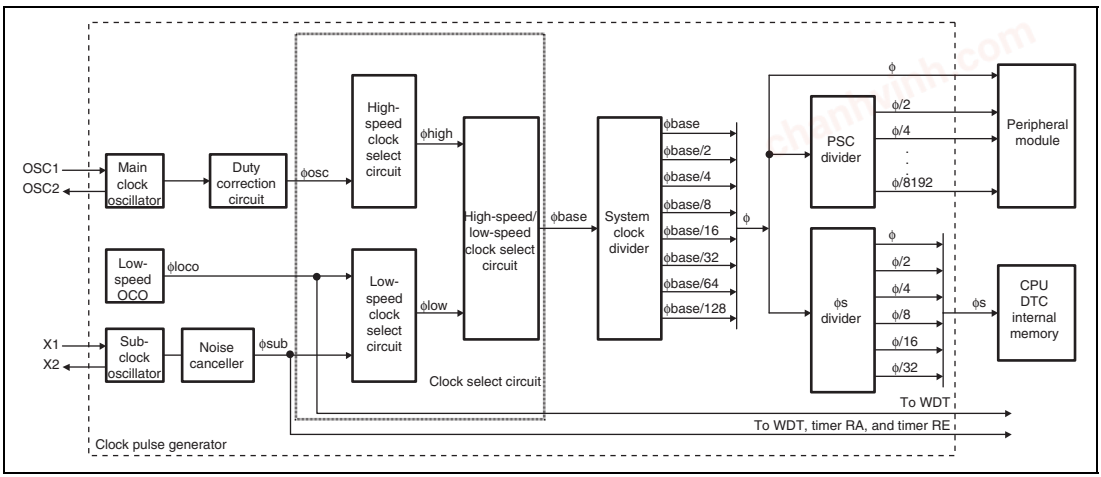



**Figure 5.1 Block Diagram of Clock Pulse Generation Circuit**

chanhvinh.com



The system reference clock (ϕbase) is the basic clock on which the CPU and on-chip peripheral modules operate. ϕbase can be divided by a value from 1 to 128 in the system clock divider, and the divided clock is supplied as the system clock ϕ. The system clock ϕ is divided by a value from 2 to 8192 in the PSC divider, and the divided clock can be supplied to on-chip peripheral modules. The system clock ϕ is also divided by a value from 1 to 32 in the ϕs divider, and the divided clock can be supplied to the bus master and memory.

After release from a reset, ϕbase is switched to the low-speed OCO.

## 5.2 Register Descriptions

- Backup control register (BAKCR)
- System clock control register (SYSCCR)
- Power-down control register 1 (LPCR1)
- Power-down control register 2 (LPCR2)
- Power-down control register 3 (LPCR3)
- OSC oscillation settling control status register (OSCCSR)

chanhvinh.com



### 5.2.1 Backup Control Register (BAKCR)

Address: H'FF06D4

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | WI | WE | OSCBAKE | BAKCKSEL | CKSWIE | CKSWIF | OSCHLT | — |
| Value after reset: | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | WI | Write inhibit | 0: Writing is permitted.<br>1: Writing is inhibited. | W |
| 6 | WE | Write enable | 0: Writing is disabled.<br>1: Writing is enabled.<br>[Setting condition]<br>When 0 is written to WI and 1 is written to WE at the same time.<br>[Clearing condition]<br>When 0 is written to WI and WE at the same time. | R/W |
| 5 | OSCBAKE | External clock backup enable | 0: External clock backup is disabled.<br>1: External clock backup is enabled.* | R/W |
| 4 | BAKCKSEL | Backup destination clock source select | 0: φlow<br>1: Setting prohibited | R/W |
| 3 | CKSWIE | Clock switching interrupt enable | 0: Interrupt requests are disabled.<br>1: Interrupt requests are enabled. | R/W |
| 2 | CKSWIF | Clock switching interrupt flag | 0: A clock switching interrupt request has not been generated.<br>1: A clock switching interrupt request has been generated.<br>[Setting condition]<br>When the system clock for the LSI is switched from φosc to φlow while OSCBAKE is 1.<br>[Clearing condition]<br>When 1 is read from the bit and then 0 is written to the same bit. | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 1 | OSCHLT | Main oscillator stop detect flag | 0: The external main oscillator is oscillating.<br>1: The external main oscillator is stopped.<br>[Setting condition]<br>When the external main oscillator is stopped while OSCBAKE is 1. | R |
| 0 | — | Reserved | This bit is read as 0. The write value should be 0. | — |

Notes: Be sure to use MOV instructions to write values to this register.

* Enable the external clock backup after completion of switching of the system clock from ϕloco to ϕosc. If enabling of the backup function is followed by switching of the system clock from ϕloco to ϕosc, switching of the clock signal may fail due to erroneous detection by the clock-stop detection circuit. Furthermore, if an application is also using the backup function in the period of switching of the system clock signal between ϕlow and ϕosc, disable the backup function while flow is in use as the system clock signal.

- WI bit (write inhibit)

  This register can be written to only when this bit is 0. This bit is always read as 1.

- WE bit (write enable)

  Bits 5 to 2 in this register can be written to when this bit is 1.

- OSCBAKE bit (external clock backup enable)

  The main oscillator stop detect circuit is enabled when this bit is 1. When this LSI operates at the external main oscillator clock, the backup function is enabled.

  By detecting a ϕosc stop, the system clock is automatically switched to ϕlow.

- CKSWIE bit (clock switching interrupt enable)

  The main clock switching interrupt requests are enabled when this bit is 1.

- CKSWIF bit (clock switching interrupt enable)

  This is a clock switching interrupt request flag.

- OSCHLT bit (main oscillator stop detect flag)

  When the OSCBAKE bit is 1, this bit indicates the results of external oscillator stop detection. This bit, however, simply indicates whether the oscillator is active or not; it does not indicate a stable oscillation. When OSCBAKE is 0, this bit is always read as 0. An oscillator stop is detected when the external oscillator is between 0 to 2 MHz.

chanhvinh.com



### 5.2.2 System Clock Control Register (SYSCCR)

Address: H'FF06D0

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | WI | WE | PHIHSEL | PHILSEL | — | SUBNC[1:0] | | — |
| Value after reset: | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | WI | Write inhibit | 0: Writing is permitted.<br>1: Writing is inhibited. | W |
| 6 | WE | Write enable | 0: Writing is disabled.<br>1: Writing is enabled.<br>[Setting condition]<br>When 0 is written to WI and 1 is written to WE at the same time.<br>[Clearing condition]<br>When 0 is written to WI and WE at the same time. | R/W |
| 5 | PHIHSEL | ϕhigh clock source select | 0: Setting prohibited<br>1: ϕosc<br>[Setting condition]<br>When 1 is written to this bit while CKSWIF in BAKCR is 0. | R/W |
| 4 | PHILSEL | ϕlow clock source select | 0: ϕloco<br>1: ϕsub | R/W |
| 3 | — | Reserved | This bit is read as 0. The write value should be 0. | — |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 2, 1 | SUBNC [1:0]* | ϕsub noise canceler sampling function setting | 00: The sampling circuit is disabled.<br>01: Sampling is performed at ϕbase/4.<br>10: Sampling is performed at ϕbase/16.<br>11: Setting prohibited | R/W |
| 0 | — | Reserved | This bit is read as 0. The write value should be 0. | — |

Notes: A MOV instruction should be used to write to this register.

* Only enable the sampling circuit after completion of switching of the system clock from ϕloco to ϕosc. If the sampling circuit is enabled while the system clock is ϕloco, the supply of ϕ subclock will be stopped during the period of waiting for stable ϕosc oscillation when the system clock is switched from ϕloco to ϕosc.

Furthermore, if an application is also using the sampling circuit of the noise canceler for the ϕ subclock signal in the period of switching of the system clock signal between ϕlow and ϕosc, disable the sampling circuit of the noise canceler for the ϕ subclock signal while ϕlow remains in use as the system clock signal.

- WI bit (write inhibit)

  This register can be written to only when this bit is 0. This bit is always read as 1.

- WE bit (write enable)

  Bits 5, 4, 2, and 1 in this register can be written to when this bit is 1.

- PHIHSEL bit (ϕhigh clock source select)

  This bit is 1 when 0 is written to the WI bit in BAKCR at CKSWIF = 0 and WE = 1 and then 1 is written to this bit.

- PHILSEL bit (ϕlow clock source select)

  When 0 is written to WI and 1 is written to this bit at WE = 1, this bit is 1.

  When 0 is written to WI and this bit at WE = 1, this bit is 0.

- SUBNC[1:0] bits (ϕsub noise canceler sampling function setting)

  Selects a sampling clock for the subclock oscillator noise canceler. Enable the sampling circuit when ϕsub is selected as a clock source for the timer RE, timer RA, and watchdog timer.

Note: The frequency of the low-speed on-chip oscillator varies greatly according to the power supply voltage and operating temperature. In designing application systems, allow sufficient margins for frequency variation.

chanhvinh.com



### 5.2.3 Power-Down Control Register 1 (LPCR1)

Address: H'FF06D1

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | WI | WE | SSBY | PSCSTP | SLEEPRS | STBYRS | — | PHIBSEL |
| Value after reset: | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | WI | Write inhibit | 0: Writing is permitted.<br>1: Writing is inhibited. | W |
| 6 | WE | Write enable | 0: Writing is disabled.<br>1: Writing is enabled.<br>[Setting condition]<br>When 0 is written to WI and 1 is written to WE at the same time.<br>[Clearing condition]<br>When 0 is written to WI and WE at the same time. | R/W |
| 5 | SSBY | Software standby | 0: A transition is made to sleep mode.<br>1: A transition is made to standby mode. | R/W |
| 4 | PSCSTP | PSC divider stop | 0: PSC divider is operating.<br>1: PSC divider is stopped*. | R/W |
| 3 | SLEEPRS | $\phi$ source select for recovery from sleep mode | 0: $\phi$low<br>1: $\phi$high | R/W |
| 2 | STBYRS | $\phi$ source select for recovery from standby mode | 0: $\phi$low<br>1: $\phi$high | R/W |
| 1 | — | Reserved | This bit is read as 0. The write value should be 0. | — |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 0 | PHIBSEL | φbase clock source select | 0: φlow<br>1: φhigh<br>[Setting conditions]<br>• When 1 is written to this bit.<br>• When the system returns from sleep mode while SLEEPRS is 1.<br>• When the system returns from standby mode while STBYRS is 1.<br>[Clearing conditions]<br>• When 0 is written to this bit.<br>• When the main oscillator backup is generated while BAKCKSEL in BAKCR is 0.<br>• When the system returns from sleep mode while SLEEPRS is 0.<br>• When the system returns from standby mode while STBYRS is 0. | R/W |

Notes: A MOV instruction should be used to write to this register.

* Operations of the peripheral modules using the φ clock are not affected by this bit setting.

- WI bit (write inhibit)

  This register can be written to only when this bit is 0. This bit is always read as 1.

- WE bit (write enable)

  Bits 5 to 2 in this register can be written to when this bit is 1.

- SSBY bit (software standby)

  Selects a mode to be entered after the SLEEP instruction is executed.

- PSCSTP bit (PSC divider stop)

  Stops the PSC divider circuit when this bit is 1. Peripheral modules using φ/2 to φ/8192 clocks stop operation. (The register values are retained.)

- SLEEPRS bit (φ source select for recovery from sleep mode)

  Selects a clock source to be used when a transition is made from sleep mode to active mode.

chanhvinh.com



- STBYRS bit (ϕ source select for recovery from standby mode)
  Selects a clock source to be used when a transition is made from standby mode to active mode.

- PHIBSEL bit (ϕbase clock source select)
  Selects a clock source for the ϕbase to be used in active mode or sleep mode.

### 5.2.4 Power-Down Control Register 2 (LPCR2)

Address: H'FF06D2

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | WI | WE | — | — | — | PHI[2:0] | | |
| Value after reset: | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | WI | Write inhibit | 0: Writing is permitted.<br>1: Writing is inhibited. | W |
| 6 | WE | Write enable | 0: Writing is disabled.<br>1: Writing is enabled.<br>[Setting condition]<br>When 0 is written to WI and 1 is written to WE at the same time.<br>[Clearing condition]<br>When 0 is written to WI and WE at the same time. | R/W |
| 5 to 3 | — | Reserved | These bits are read as 0. The write value should be 0. | — |
| 2 to 0 | PHI[2:0] | System clock ϕ select | 000: ϕbase<br>001: ϕbase/2<br>010: ϕbase/4<br>011: ϕbase/8<br>100: ϕbase/16<br>101: ϕbase/32<br>110: ϕbase/64<br>111: ϕbase/128 | R/W |

Note: A MOV instruction should be used to write to this register.

chanhvinh.com



- WI (write inhibit)
  This register can be written to only when this bit is 0. This bit is always read as 1.
- WE bit (write enable)
  Bits 2 to 0 in this register can be written to when this bit is 1.
- PHI2 bit to PHI0 bit (system clock ϕ select)
  Selects a clock source for the system clock ϕ to be used in active mode or sleep mode. The clock is changed immediately after this bit is set.

### 5.2.5 Power-Down Control Register 3 (LPCR3)

Address: H'FF06D3

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | WI | WE | STBYINT | SLEEPINT | — | PHIS[2:0] | | |
| Value after reset: | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | WI | Write inhibit | 0: Writing is permitted.<br>1: Writing is inhibited. | W |
| 6 | WE | Write enable | 0: Writing is disabled.<br>1: Writing is enabled.<br>[Setting condition]<br>When 0 is written to WI and 1 is written to WE at the same time.<br>[Clearing condition]<br>When 0 is written to WI and WE at the same time. | R/W |
| 5 | STBYINT | Standby mode interrupt generation flag | 0: No interrupt has occurred in standby mode.<br>1: An interrupt has occurred in standby mode.<br>[Setting condition]<br>When an interrupt is generated in standby mode.<br>[Clearing condition]<br>When an interrupt is generated in states other than standby mode. | R |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 4 | SLEEPINT | Sleep mode interrupt generation flag | 0: No interrupt has occurred in sleep mode.<br>1: An interrupt has occurred in sleep mode.<br>[Setting condition]<br>When an interrupt is generated in sleep mode.<br>[Clearing condition]<br>When an interrupt is generated in states other than sleep mode. | R |
| 3 | — | Reserved | This bit is read as 0. The write value should be 0. | — |
| 2 to 0 | PHIS[2:0] | Bus master operation clock φs select | 000: φ<br>001: φ/2<br>010: φ/4<br>011: φ/8<br>100: φ/16<br>101: φ/32<br>110: Setting prohibited<br>111: Setting prohibited | R/W |

Note: A MOV instruction should be used to write to this register.

- WI bit (write inhibit)

  This register can be written to only when this bit is 0. This bit is always read as 1.

- WE bit (write enable)

  Bits 2 to 0 in this register can be written to when this bit is 1.

- STBYINT bit (standby mode interrupt generation flag)

  This bit is set to 1 when an interrupt is generated in standby mode. This bit is cleared to 0 when an interrupt is generated in the other state.

- SLEEPINT bit (sleep mode interrupt generation flag)

  This bit is set to 1 when an interrupt is generated in sleep mode. This bit is cleared to 0 when an interrupt is generated in the other state.

- PHIS[2:0] bits (bus master operation clock φs select)

  Selects a clock source for the bus master operation clock φs to be used in active mode or sleep mode. The clock is changed immediately after these bits are set.

chanhvinh.com



### 5.2.6 OSC Oscillation Settling Control Status Register (OSCCSR)

Address: H'FF06D5

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | OSCWEF | — | — | — | STS[3:0] | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | OSCWEF | ϕosc oscillation settling wait state completed flag | 0: Number of wait states for a stable ϕosc oscillation has not elapsed.<br>1: Number of wait states for a stable ϕosc oscillation has elapsed.<br>[Setting condition]<br>• The number of states specified by the STS[3:0] bits having elapsed since starting of the main clock oscillator<br>[Clearing conditions]<br>• Switching of the functions of the PJ0 and PJ1 pins from general I/O to the oscillator functions<br>• A transition to standby mode while the PJ0 and PJ1 pins are functioning as oscillator pins<br>• Detection of stoppage of ϕosc oscillation while the backup function is enabled. | R |
| 6 to 4 | — | Reserved | These bits are read as 0. The write value should be 0. | — |
| 3 to 0 | STS[3:0] | ϕosc oscillation settling time select 3 to 0 | Specifies the number of wait states for a stable ϕosc oscillation. For the relationship between assigned values and the numbers of wait states, see table 5.2. | R/W |

- STS3 bit to STS0 bit (ϕosc oscillation settling time select 3 to 0)

  Specifies the number of wait states for a stable ϕosc oscillation. The count clock is ϕosc. Table 5.2 shows the relationship between assigned values and the numbers of wait states. If the system reference clock is ϕosc when the system returns from the standby mode, or when the system reference clock is switched to ϕosc, set these bits so that wait time will be 6.5 ms or greater depending on the frequency of the oscillator.

  The watchdog timer is enabled in the initial state. When switching the clock while the watchdog timer is enabled, take account of the oscillation settling time in adjusting the overflow cycle of the watchdog timer.

chanhvinh.com



If the ϕosc is already oscillating stably or the ϕosc is an external clock input, wait time can be selected from 16 states (STS[3:0]=B'0000).

**Table 5.2 Relationship between Operation Frequency and Number of Wait States**

| Bit | | | | Number of | Operation Frequency | | | | |
|---|---|---|---|---|---|---|---|---|---|
| STS3 | STS2 | STS1 | STS0 | Wait States | 20 MHz | 16 MHz | 10 MHz | 8 MHz | 4 MHz |
| 0 | 0 | 0 | 0 | 16 states | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0 | 0 | 0 | 1 | 32 states | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 |
| 0 | 0 | 1 | 0 | 64 states | 0.00 | 0.00 | 0.01 | 0.01 | 0.02 |
| 0 | 0 | 1 | 1 | 128 states | 0.01 | 0.01 | 0.01 | 0.02 | 0.03 |
| 0 | 1 | 0 | 0 | 256 states | 0.01 | 0.02 | 0.03 | 0.03 | 0.06 |
| 0 | 1 | 0 | 1 | 512 states | 0.03 | 0.03 | 0.05 | 0.06 | 0.13 |
| 0 | 1 | 1 | 0 | 1024 states | 0.05 | 0.06 | 0.10 | 0.13 | 0.26 |
| 0 | 1 | 1 | 1 | 2048 states | 0.10 | 0.13 | 0.20 | 0.26 | 0.51 |
| 1 | 0 | 0 | 0 | 4096 states | 0.20 | 0.26 | 0.41 | 0.51 | 1.02 |
| 1 | 0 | 0 | 1 | 8192 states | 0.41 | 0.51 | 0.82 | 1.02 | 2.05 |
| 1 | 0 | 1 | 0 | 16384 states | 0.82 | 1.02 | 1.64 | 2.05 | 4.10 |
| 1 | 0 | 1 | 1 | 32768 states | 1.64 | 2.05 | 3.28 | 4.10 | 8.19 |
| 1 | 1 | 0 | 0 | 65536 states | 3.28 | 4.10 | 6.55 | 8.19 | 16.38 |
| 1 | 1 | 0 | 1 | 131072 states | 6.55 | 8.19 | 13.11 | 16.38 | 32.77 |
| 1 | 1 | 1 | 0 | 262144 states | 13.11 | 16.38 | 26.21 | 32.77 | 65.54 |
| 1 | 1 | 1 | 1 | 262144 states | 13.11 | 16.38 | 26.21 | 32.77 | 65.54 |

chanhvinh.com



## 5.3 Operation of Selection of System Reference Clock

After a reset, this LSI enters active mode operating in low-speed clocks. The user, by means of software, can change the system reference clock from a low-speed OCO clock to the main oscillator clock or a sub-oscillator clock.

Figure 5.2 shows a transition diagram between system reference clock states. Table 5.3 shows conditions under which clock sources can be switched.

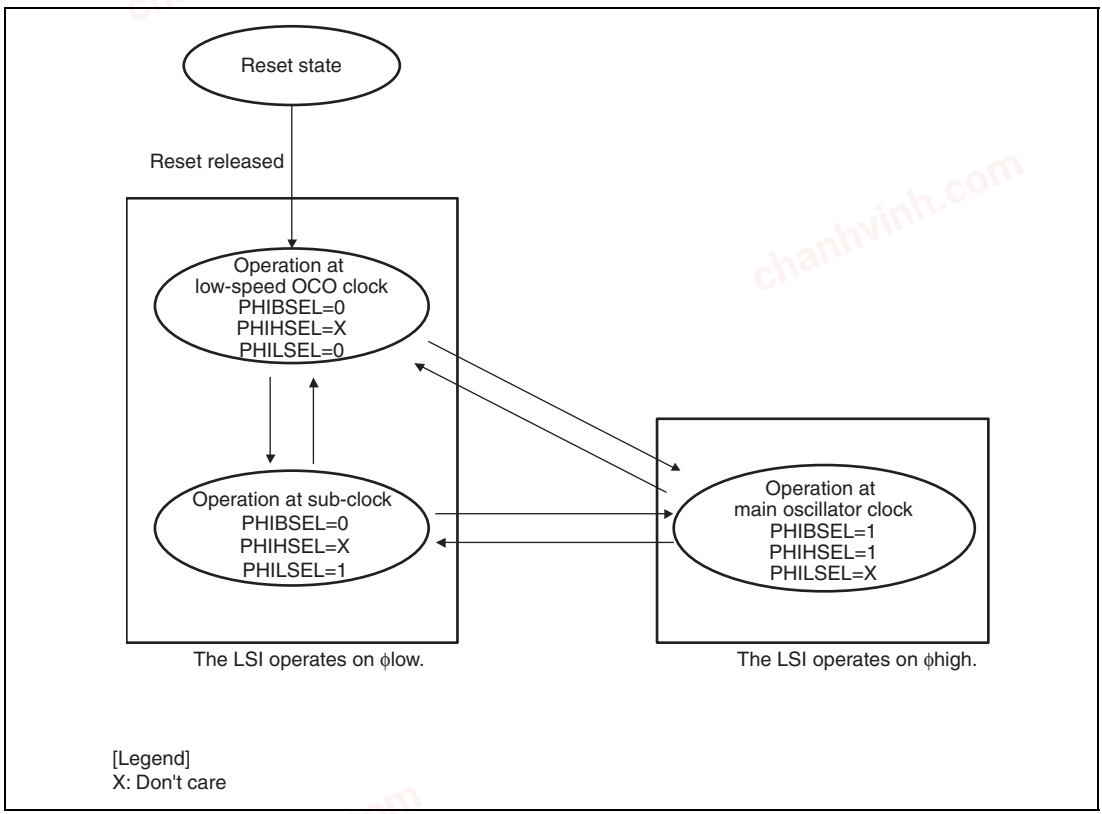



**Figure 5.2 Transition Diagram between LSI System Reference Clock States**

chanhvinh.com



**Table 5.3 Clock Source Switching**

| Bit | | | |
|---|---|---|---|
| **PHIBSEL** | **PHIHSEL** | **PHILSEL** | **Switching Operation** |
| 0 | Don't care | 0 → 1 | φloco → φsub |
| 0 | Don't care | 1 → 0 | φsub → φloco |
| 0 → 1 | 1 | 0 | φloco → φosc |
| 1 → 0 | 1 | 0 | φosc → φloco |
| 0 → 1 | 1 | 1 | φsub → φosc |
| 1 → 0 | 1 | 1 | φosc → φsub |

chanhvinh.com



Table 5.4 shows the low-speed OCO, main oscillator, and sub-oscillator operation states in each operating mode (system state).

**Table 5.4 Clock Operation States in Each Operating Mode**

| System State | System Clock | Main Oscillator State | Low-Speed OCO State | Sub-Oscillator State |
|---|---|---|---|---|
| Reset released | φloco | Stopped | Oscillating | Oscillating*3 |
| Active mode, sleep mode | φosc | Depending on user setting*2 | Oscillating | |
| | φloco | Depending on user setting*1 | Oscillating | |
| | φsub | Depending on user setting*1 | Oscillating | |
| Standby mode | None | Stopped | Oscillating | |

Notes: 1. Can be set with the PMRJ[1:0] bits in PMRJ.
2. Backup operation is performed by selecting the oscillation stop with the PMRJ[1:0] bits in PMRJ when the backup function is enabled.
3. A crystal resonator should be connected when a sub-oscillator clock is used. To switch the system reference clock to φsub immediately after the power-on, oscillation settling time for the sub-oscillator should be ensured.

chanhvinh.com



### 5.3.1 Switching System Reference Clock to ϕosc

Figure 5.3 shows a flowchart of the process in which the system reference clock is switched from ϕloco to ϕosc.

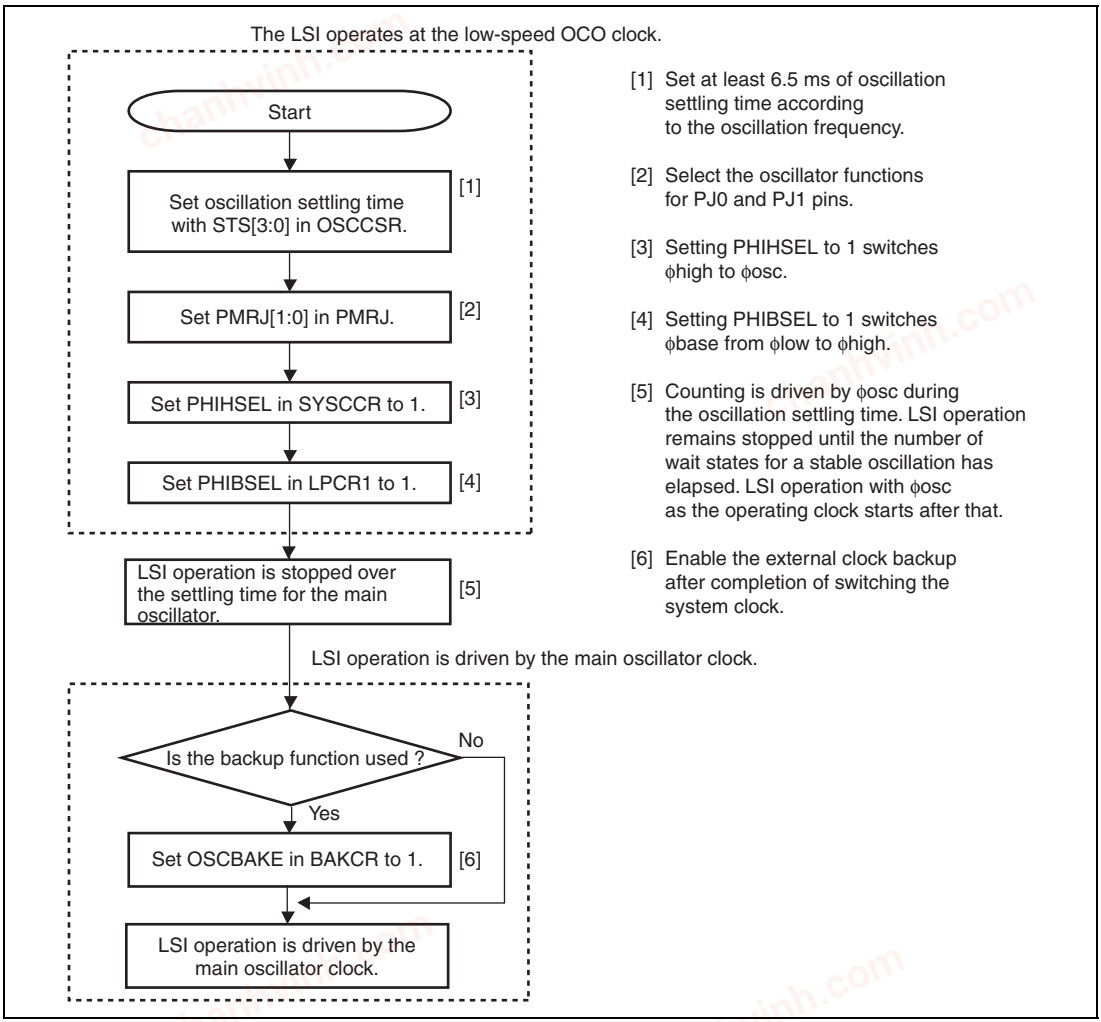



**Figure 5.3 Flowchart of Clock Switching from ϕloco to ϕosc (1)**

chanhvinh.com



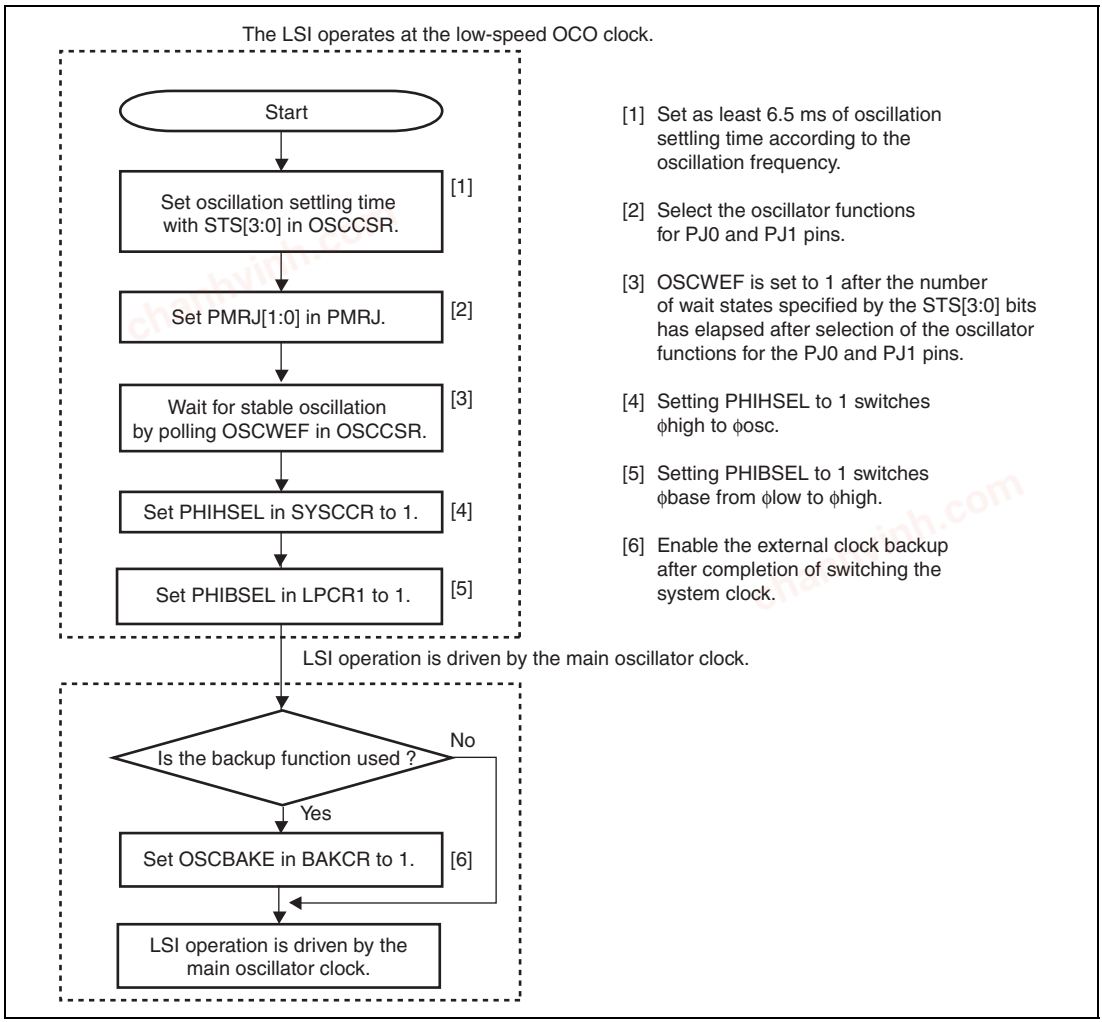



**Figure 5.4 Flowchart of Clock Switching from ϕloco to ϕosc (2)**

chanhvinh.com



### 5.3.2 Clock Change Timing

#### (1) Switching Division Ratio for the Same Clock Source

Figure 5.5 shows a division ratio switching timing chart for the same clock source.

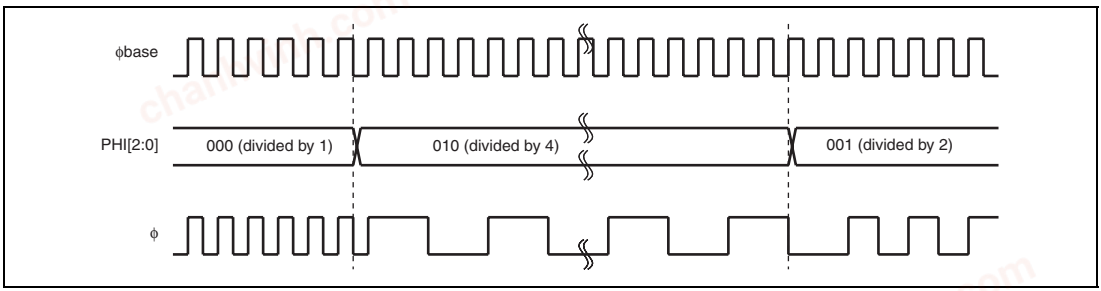


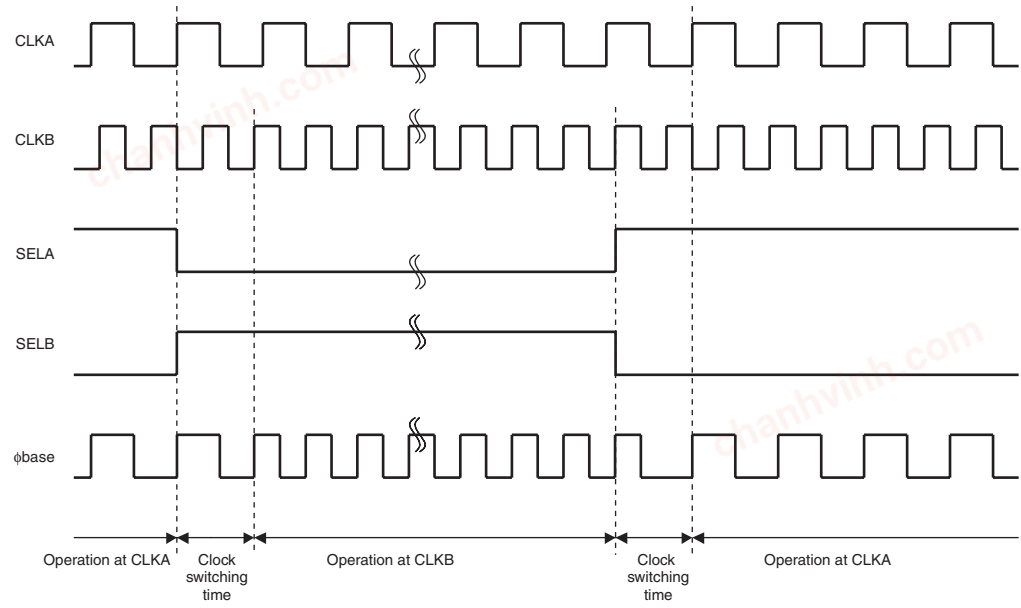

**Figure 5.5 Timing of Division Ratio Switching for the Same Clock Source**

chanhvinh.com



### (2) Switching System Reference Clock Source

Figures 5.6 and 5.7 show clock source switching timing charts for the system reference clock.




[Legend]
CLKA: Clock A
CLKB: Clock B
SELA: CLKA select signal
SELB: CLKB select signal
ϕbase: System base clock

Note: Clock switching time is a period from the clock select signal change to the second high level of the destination clock.

**Figure 5.6 Timing of Clock Source Switching (When the switching destination clock source is active)**

chanhvinh.com



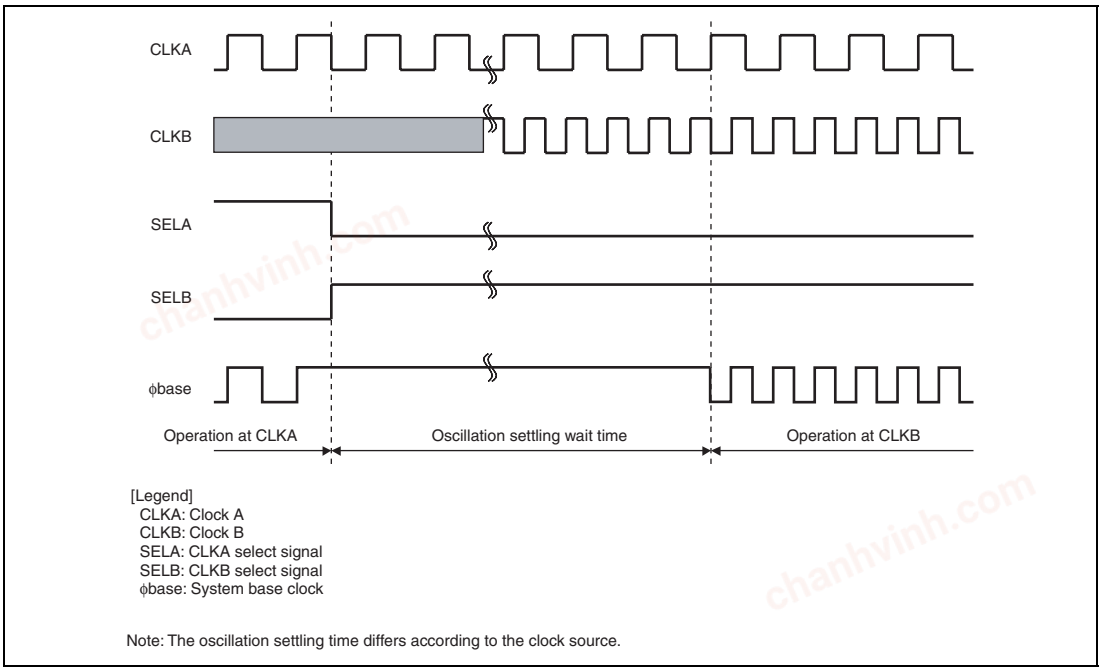



**Figure 5.7 Timing of Clock Source Switching (When the switching destination clock source is stopped)**

Oscillation stabilization wait time varies with switching destination clock sources. If the destination clock is ϕosc, wait time is specified by the STS[3:0] bit of the OSCCSR. For oscillation stabilization wait time values, see table 5.2.

During oscillation stabilization wait time, the ϕbase stops; therefore, any module that operates with the ϕbase as a base, including the bus master, stops. The register retains the pre-switching value.

chanhvinh.com



### 5.3.3 Backup Operation

If the operating clock for the system is ϕosc and the backup function is enabled, when the main oscillator detects an oscillation halt condition, the system clock automatically switches to ϕlow, according to BAKCKSEL in BAKCR. The period from the stopping of the main oscillator to the time the system clock is operating from ϕlow will be clock halt detection time + oscillation stabilization wait time for the backup clock. Time to wait for oscillation stabilization is 0 ms if the target backup clock is already oscillating when stopping of the main oscillator is detected.

To reduce power consumption, it is also possible to set the target backup clock to a stopped state when the backup function is enabled. In that case, the target backup clock is automatically activated when stopping of the main clock oscillator is detected. The system clock is then changed after a certain amount of oscillation settling time. This LSI circuit may malfunction during operation with the back-up function. Accordingly, usage with the watchdog timer is recommended.

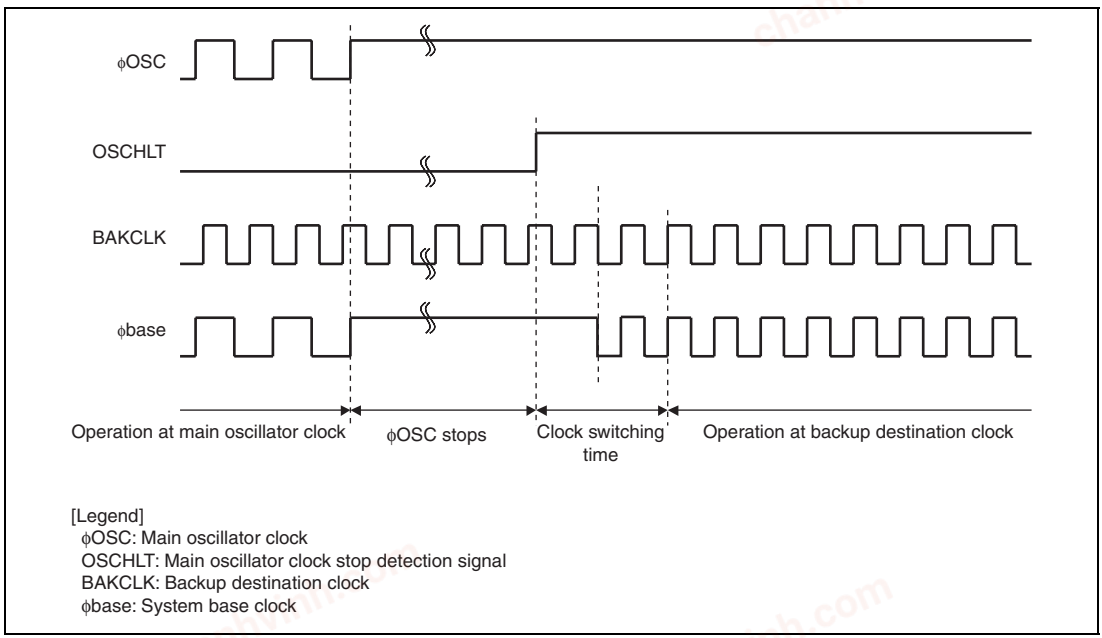



**Figure 5.8 Timing of Backup Operation When Main Oscillator Stops at High Level (When the backup destination clock is active)**

chanhvinh.com



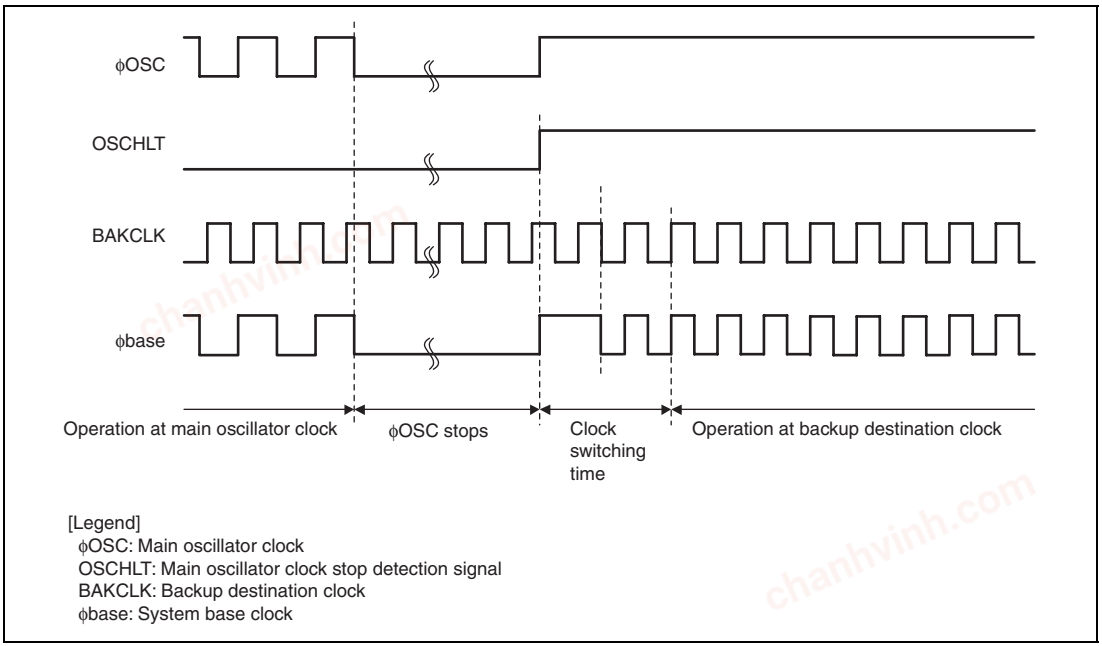



**Figure 5.9 Timing of Backup Operation When Main Oscillator Stops at Low Level (When the backup destination clock is active)**

chanhvinh.com



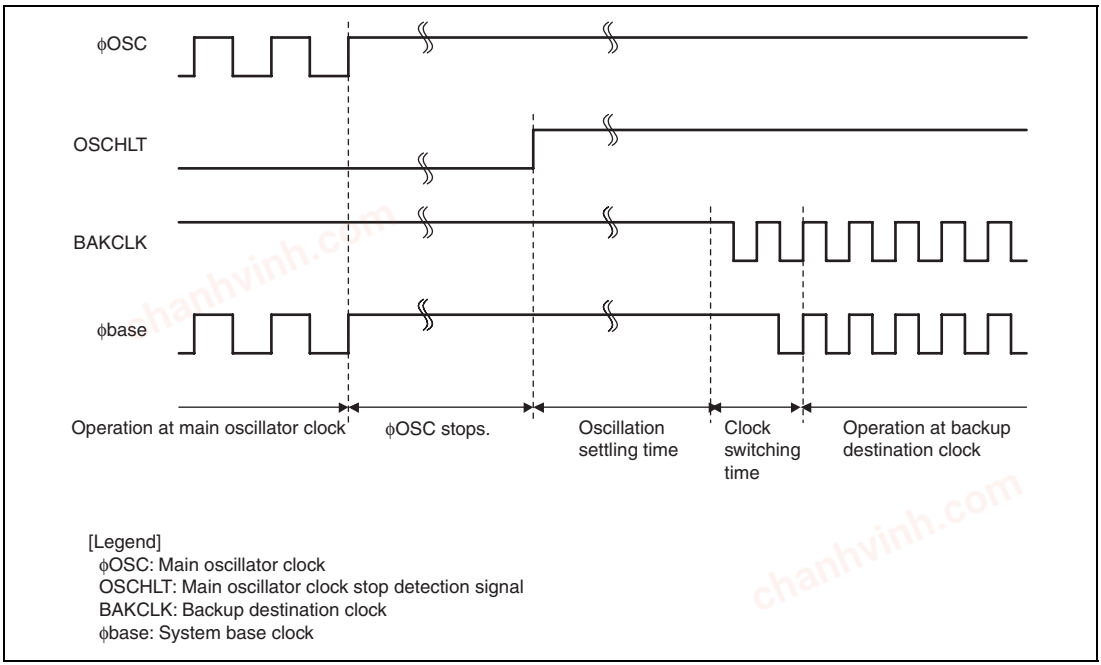



**Figure 5.10 Timing of Backup Operation When Main Oscillator Stops at High Level (When the backup destination clock is stopped)**

chanhvinh.com



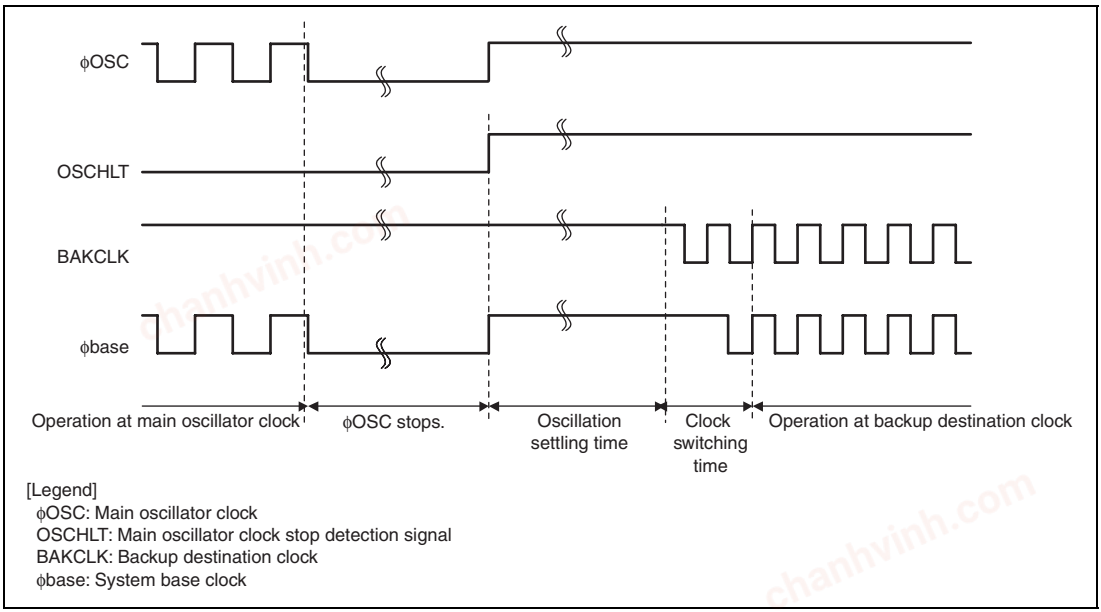



**Figure 5.11 Timing of Backup Operation When Main Oscillator Stops at Low Level (When the backup destination clock is stopped)**

chanhvinh.com



## 5.4 Main Clock Oscillator

This LSI has two methods to supply external clock pulses into it: connecting a crystal or ceramic resonator, and an external clock. For setting PJ0/OSC1 and PJ1/OSC2/CLKOUT for a crystal resonator or an external clock, see section 10.10.1, Port Mode Register J (PMRJ). Figure 5.12 shows a block diagram of the Main clock oscillator.

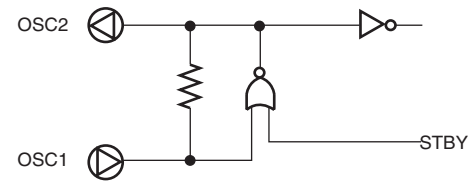



**Figure 5.12 Block Diagram of Main Clock Oscillator**

### 5.4.1 Connecting Crystal Resonator

Figure 5.13 shows an example of connecting a crystal resonator. An AT-cut parallel-resonance crystal resonator should be used. A damping resistor Rd should be added, if necessary. Since the resistor values vary depending on the resonator, use values recommended by the resonator manufacturer.

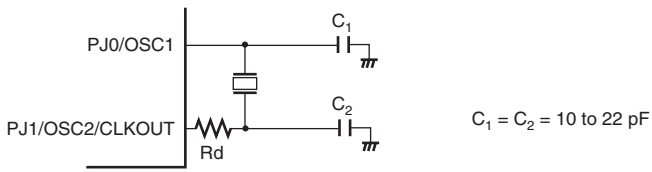



**Figure 5.13 Example of Connection to Crystal Resonator**

chanhvinh.com



### 5.4.2 Connecting Ceramic Resonator

Figure 5.14 shows an example of connecting a ceramic resonator. A damping resistor Rd should be added, if necessary. Since the resistor values vary depending on the resonator, use values recommended by the resonator manufacturer.

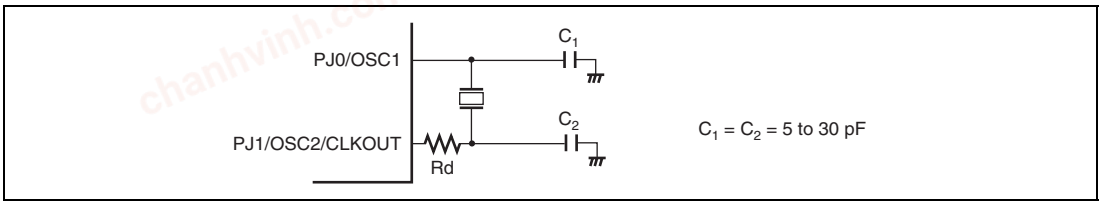



**Figure 5.14 Example of Connection to Ceramic Resonator**

### 5.4.3 External Clock Input Method

To use the external clock, input the external clock on pin OSC1. Figure 5.15 shows an example of connection. The duty cycle of the external clock signal must be 45 to 55%.

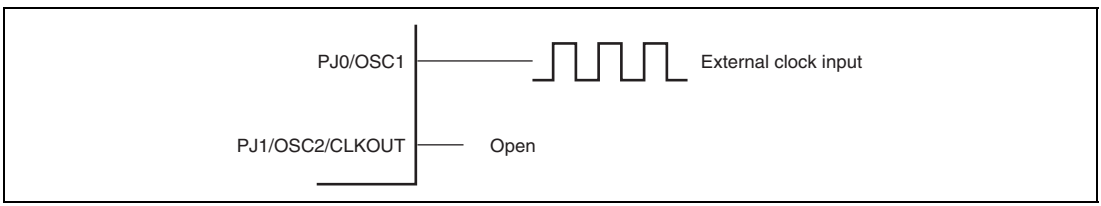



**Figure 5.15 Example of External Clock Input**

Note: To input the external clock, set the PMRJ[1:0] bits to 01. Do not input the external clock while PMRJ[1:0] bits are set to 11.

chanhvinh.com



## 5.5 Subclock Oscillator

Figure 5.16 shows a block diagram of the subclock oscillator.

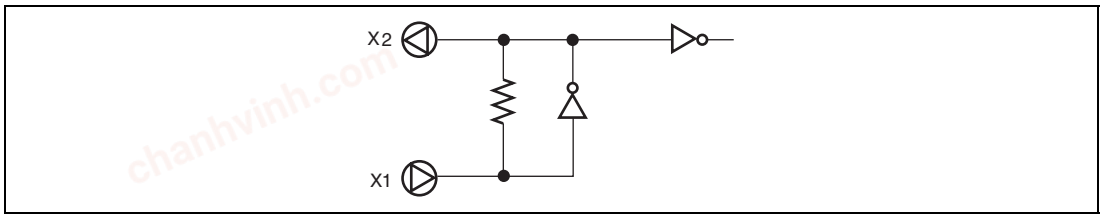



**Figure 5.16 Block Diagram of Subclock Oscillator**

### 5.5.1 Connecting 32.768-kHz Crystal Resonator

Clock pulses can be supplied to the subclock divider by connecting a 32.768-kHz crystal resonator, as shown in figure 5.17. A damping resistor Rd should be added, if necessary. Since the resistor values vary depending on the resonator, use values recommended by the resonator manufacturer.

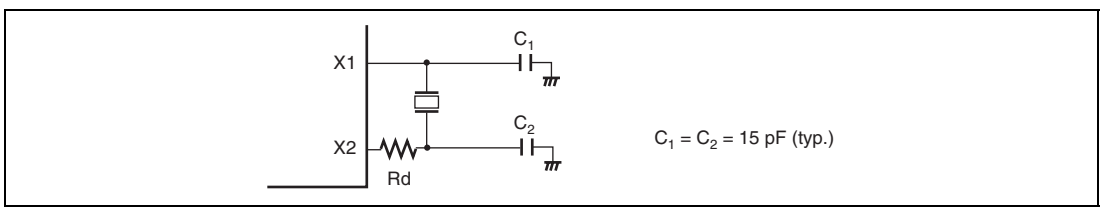



**Figure 5.17 Typical Connection to 32.768-kHz Crystal Resonator**

### 5.5.2 Pin Connection when not Using Subclock

When the subclock is not used, connect pin X1 to VSS and leave pin X2 open, as shown in figure 5.18.

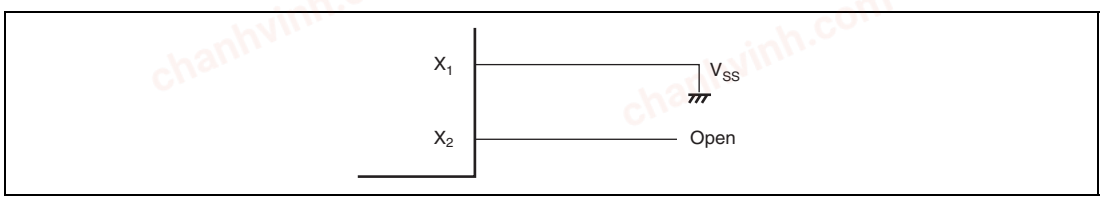



**Figure 5.18 Pin Connection when not Using Subclock**

chanhvinh.com



## 5.6 Prescaler

The prescaler is a 13-bit counter using the system clock (ϕ) as its input clock. The outputs, which are divided clocks, are used as internal clocks by the on-chip peripheral modules. The prescaler is initialized to H'0000 and stops counting after a reset. It starts counting when the PSCSTP bit in LPCR1 is cleared. The prescaler counter cannot be accessed by the CPU.

The outputs from the prescaler is shared by the on-chip peripheral modules. The division ratio can be set separately for each on-chip peripheral module. The clock input to the prescaler is a system clock with the division ratio specified by bits PHI2 to PHI0 in LPCR2.

chanhvinh.com



## 5.7 Usage Notes

### 5.7.1 Note on Resonators

Resonator characteristics are closely related to board design and should be carefully evaluated by the user, referring to the examples shown in this section. Resonator circuit parameters will differ depending on the resonator element, stray capacitance of the PCB, and other factors. Suitable values should be determined in consultation with the resonator element manufacturer. Design the circuit so that the resonator element never receives voltages exceeding its maximum rating.

### 5.7.2 Notes on Board Design

When using a crystal resonator (ceramic resonator), place the resonator and its load capacitors as close as possible to pins OSC1 and OSC2. Other signal lines should be routed away from the oscillator circuit to prevent induction from interfering with correct oscillation (see figure 5.19).

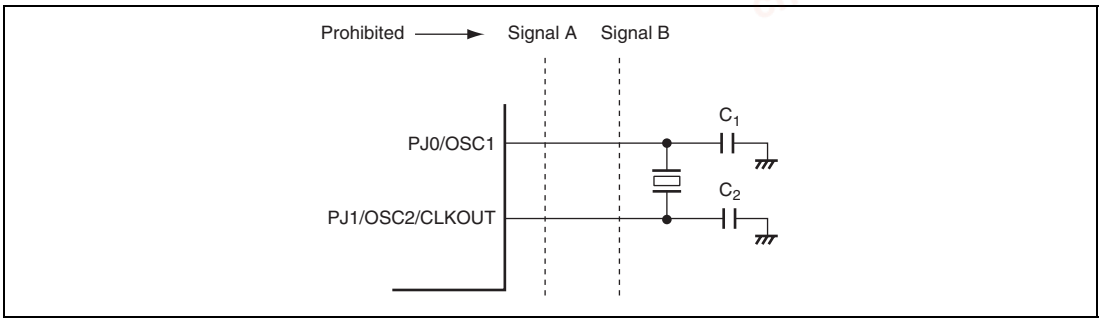



**Figure 5.19 Example of Incorrect Board Design**

chanhvinh.com



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

chanhvinh.com



# Section 6 Power-Down Modes

In addition to normal active mode, this LSI can enter either of the two power-down modes after release from a reset, in which power consumption is reduced. As other measures for reduced power consumption, this LSI also has a bus-master-clock division function for the low-speed-operation of bus masters, module standby function which allows the selective stopping of on-chip peripheral modules, and a PSC-divider stopping function. Further power consumption is possible by selecting the low-speed on-chip oscillator clock ϕloco, or sub-oscillator clock ϕsub as the source of the system clock ϕ to operate the LSI at a low speed. After release from a reset, all of the peripheral functions except timer RE are in the module standby state. Make the settings for the operation of module in the corresponding registers after the module standby state is released.

- Active Mode

  The CPU and on-chip peripheral modules operate on the system clock ϕ. The system clock frequency can be selected from among ϕbase to ϕbase/128, where ϕbase is the system reference clock.

- Sleep Mode

  The CPU is stopped. On-chip peripheral modules operate on the system clock ϕ.

- Standby Mode

  The CPU and all the on-chip peripheral modules are stopped. However, timer RE can operate when the realtime clock mode is selected. The watchdog timer (WDT) also operates when the low-speed OCO or subclock ϕsub is selected as the WDT clock source.

- Bus Master Clock Division Function

  For the bus masters CPU and DTC, ROM, and RAM, the operating clock ϕs can be divided independently of the clock supplied to the peripheral modules. The bus master clock ϕs can be selected from among ϕ to ϕ/32.

- PSC Divider Stop Function

  The PSC divider can be stopped through software setting. Specifically, the peripheral modules using ϕ/2 to ϕ/8192 are stopped (register values are retained), whereas the ones using ϕ remain operating.

- Module Standby Function

  Power consumption can be reduced by halting individual on-chip peripheral modules that are not in use.

chanhvinh.com



## 6.1 Register Descriptions

The registers related to power-down modes are listed below.

- Power-down control register 1 (LPCR1)
- Power-down control register 2 (LPCR2)
- Power-down control register 3 (LPCR3)
- Module standby control register 1 (MSTCR1)
- Module standby control register 2 (MSTCR2)
- Module standby control register 3 (MSTCR3)

### 6.1.1 Power-Down Control Registers 1, 2, and 3 (LPCR1, LPCR2, LPCR3)

LPCR1, LPCR2, and LPCR3 control power-down modes. For details, see section 5, Clock Pulse Generator.

### 6.1.2 Module Standby Control Register 1 (MSTCR1)

Address: H'FFFFDC

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | MSTWDT | — | MSTAD1 | MSTAD2 | MSTDA | MSTDTC | — | — |
| Value after reset: | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | MSTWDT | Watchdog timer module standby | 0: Operating state<br>1: Standby state | R/W |
| 6 | — | Reserved | This bit is read as 0. The write value should be 0. | — |
| 5 | MSTAD1 | A/D converter unit 1 module standby | 0: Operating state<br>1: Standby state | R/W |
| 4 | MSTAD2 | A/D converter unit 2 module standby* | 0: Operating state<br>1: Standby state | R/W |
| 3 | MSTDA | D/A converter module standby | 0: Operating state<br>1: Standby state | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 2 | MSTDTC | DTC module standby | 0: Operating state<br>1: Standby state | R/W |
| 1, 0 | — | Reserved | These bits are read as 0. The write value should be 0. | — |

Note: * When a peripheral module is in the module standby state, the registers of the module cannot be accessed.

- MSTWDT bit (watchdog timer module standby)
  When this bit is set to 1, the WDT enters the standby state. Note that if the low-speed OCO is selected as the WDT count clock, the WDT operates regardless of the setting of this bit but the WDT registers cannot be accessed.

- MSTAD1 bit (A/D converter unit 1 module standby)
  When this bit is set to 1, A/D converter unit 1 enters the standby state.

- MSTAD2 bit (A/D converter unit 2 module standby)
  When this bit is set to 1, A/D converter unit 2 enters the standby state.
  A/D converter unit 2 is not available on the H8S/20103, H8S/20203, H8S/20115, and H8S/20215 Groups; this bit is reserved on these devices. For a write-access, write 1 to this bit.

- MSTDA bit (D/A converter module standby)
  When this bit is set to 1, the D/A converter enters the standby state.

- MSTDTC bit (DTC module standby)
  When this bit is set to 1, the DTC enters the standby state.

chanhvinh.com



### 6.1.3 Module Standby Control Register 2 (MSTCR2)

Address: H'FFFFDD

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | MSTSCI3_1 | MSTSCI3_2 | MSTSCI3_3 | — | — | MSTICSU | — | — |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | MSTSCI3_1 | SCI3 channel 1 module standby | 0: Operating state<br>1: Standby state | R/W |
| 6 | MSTSCI3_2 | SCI3 channel 2 module standby | 0: Operating state<br>1: Standby state | R/W |
| 5 | MSTSCI3_3 | SCI3 channel 3 module standby | 0: Operating state<br>1: Standby state | R/W |
| 4, 3 | — | Reserved | These bits are read as 1. The write value should be 1. | — |
| 2 | MSTICSU | IIC2/SSU module standby | 0: Operating state<br>1: Standby state | R/W |
| 1, 0 | — | Reserved | These bits are read as 1. The write value should be 1. | — |

Notes: 1. When a peripheral module is in the module standby state, the registers of the module cannot be accessed.
2. When writing to this register, write 1s to the reserved bits.

- MSTSCI3_1 (SCI3 channel 1 module standby)
  When this bit is set to 1, SCI3 channel 1 enters the standby state.
- MSTSCI3_2 (SCI3 channel 2 module standby)
  When this bit is set to 1, SCI3 channel 2 enters the standby state.
- MSTSCI3_3 (SCI3 channel 3 module standby)
  When this bit is set to 1, SCI3 channel 3 enters the standby state.
- MSTICSU (IIC2/SSU module standby)
  When this bit is set to 1, the IIC2 or SSU enters the standby state.

chanhvinh.com



### 6.1.4 Module Standby Control Register 3 (MSTCR3)

Address: H'FFFFDE

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | MSTTMRA | MSTTMRB | MSTTMRC | MSTTMRD1 | MSTTMRD2 | MSTTMRG | — | MSTTMRE |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | MSTTMRA | Timer RA module standby | 0: Operating state<br>1: Standby state | R/W |
| 6 | MSTTMRB | Timer RB module standby | 0: Operating state<br>1: Standby state | R/W |
| 5 | MSTTMRC | Timer RC module standby | 0: Operating state<br>1: Standby state | R/W |
| 4 | MSTTMRD1 | Timer RD unit 0 module standby | 0: Operating state<br>1: Standby state | R/W |
| 3 | MSTTMRD2 | Timer RD unit 1 module standby | 0: Operating state<br>1: Standby state | R/W |
| 2 | MSTTMRG | Timer RG module standby | 0: Operating state<br>1: Standby state | R/W |
| 1 | — | Reserved | This bit is read as 1. The write value should be 1. | — |
| 0 | MSTTMRE | Timer RE module standby | 0: Operating state<br>1: Standby state | R/W |

Notes: 1. When a peripheral module is in the module standby state, the registers of the module cannot be accessed.
2. When writing to this register, write 1s to the reserved bits.

chanhvinh.com



- MSTTMRA bit (timer RA module standby)
  When this bit is set to 1, timer RA enters the standby state.

- MSTTMRB bit (timer RB module standby)
  When this bit is set to 1, timer RB enters the standby state.

- MSTTMRC bit (timer RC module standby)
  When this bit is set to 1, timer RC enters the standby state.
  Timer RC is not available on the H8S/20203, H8S/20223, H8S/20215, and H8S/20235 Groups; this bit is reserved on these devices. For a write-access, write 1 to this bit.

- MSTTMRD1 bit (timer RD unit 0 module standby)
  When this bit is set to 1, timer RD unit 0 enters the standby state.

- MSTTMRD2 bit (timer RD unit 1 module standby)
  When this bit is set to 1, timer RD unit 1 enters the standby state.
  Timer RD unit 1 is not available on the H8S/20103 and H8S/20115 Groups; this bit is reserved on the device. For a write-access, write 1 to this bit.

- MSTTMRG bit (timer RG module standby)
  When this bit is set to 1, timer RG enters the standby state.

- MSTTMRE bit (timer RE module standby)
  When this bit is set to 1, timer RE enters the standby state. Note that if the ϕsub is selected as the count clock in realtime clock mode or output-compare mode, timer RE operates regardless of the setting of this bit but the timer RE registers cannot be accessed.

chanhvinh.com



## 6.2 Mode Transitions and States of LSI

Figure 6.1 shows the possible transitions among the operating modes. SLEEP instructions are used to cause a transition from the program execution state to the program halt state. Interrupts are used to return from the program halt state to the program execution state. When the $\overline{\text{RES}}$ pin is driven low or any other internal reset occurs, this LSI is placed in the reset state from any mode. After release from a reset, this LSI is placed in active mode.

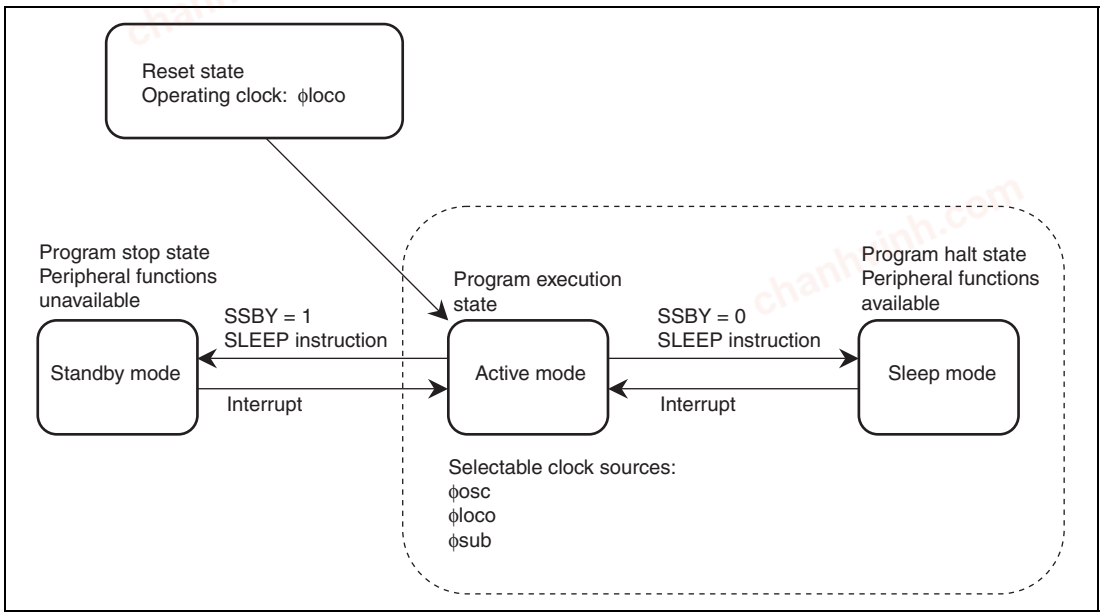



**Figure 6.1 Mode Transition Diagram**

chanhvinh.com



Table 6.1 shows the internal states of the LSI in each mode.

**Table 6.1 Internal State in Each Operating Mode**

| Function | | LPCR1 PSCSTP = 0 | | LPCR1 PSCSTP = 1 | | |
|---|---|---|---|---|---|---|
| | | Active Mode | Sleep Mode | Active Mode | Sleep Mode | Standby Mode |
| System clock | | Functioning | Functioning | Functioning | Functioning | Stopped |
| CPU | Instruction execution | Functioning | Stopped | Functioning | Stopped | Stopped |
| | Registers | Functioning | Retained | Functioning | Retained | Retained |
| DTC | | Functioning | Functioning | Functioning | Functioning | Stopped |
| ELC | | Functioning | Functioning | Functioning*1 | Functioning*1 | Retained |
| RAM | | Functioning | Functioning | Functioning | Functioning | Retained |
| I/O ports | | Functioning | Functioning | Functioning | Functioning | Register contents are retained, but output goes to the high-impedance state. |
| External interrupts | IRQ7 to IRQ0, $\overline{\text{NMI}}$ | Functioning | Functioning | Functioning | Functioning | Functioning |
| Peripheral modules | Timer RA, timer RB, timer RC, timer RD_0, timer RD_1 | Functioning | Functioning | Retained*2 | Retained*2 | Retained |
| | Timer RE | Functioning | Functioning | Functioning in realtime clock mode and retained in output-compare mode. | | Functioning in realtime clock mode and retained in output-compare mode. |
| | Timer RG | Functioning | Functioning | Retained*2 | Retained*2 | Retained |
| | Watchdog timer | Functioning | Functioning | Retained*3 | Retained*3 | Retained*3 |
| | SCI3_1, SCI3_2, SCI3_3 | Functioning | Functioning | Retained*2 | Retained*2 | Reset |

chanhvinh.com



| Function | | LPCR1 PSCSTP = 0 Active Mode | LPCR1 PSCSTP = 0 Sleep Mode | LPCR1 PSCSTP = 1 Active Mode | LPCR1 PSCSTP = 1 Sleep Mode | Standby Mode |
|---|---|---|---|---|---|---|
| Peripheral modules | IIC2/SSU | Functioning | Functioning | Retained | Retained | Reset |
| | A/D converter_1, A/D converter_2 | Functioning | Functioning | Retained*4 | Retained*4 | Reset |
| | D/A converter | Functioning | Functioning | Functioning | Functioning | Reset |

Notes: 1. The timers are stopped if $\phi/2$ to $\phi/8192$ is selected as the clock source of the event-generation timer.
2. The timers operate if $\phi$ is selected as the count clock. The timers are stopped if $\phi/2$ to $\phi/8192$ is selected as the count clock.
3. The WDT operates if the low-speed OCO or subclock $\phi$sub is selected as its clock source.
4. The A/D converters operate when A/D conversion time = 43 states (max) is selected. The A/D converters are retained when the other conversion time is set.

### 6.2.1 Active Mode

In active mode, the CPU, DTC, and all the on-chip peripheral modules operate on the system clock $\phi$. The system clock frequency can be selected from among $\phi$base, $\phi$base/2, $\phi$base/4, $\phi$base/8, $\phi$base/16, $\phi$base/32, $\phi$base/64, and $\phi$base/128 according to the PHI[2:0] setting in LPCR2.

### 6.2.2 Sleep Mode

When a SLEEP instruction is executed in active mode with the SSBY bit = 0 in LPCR1, a transition to sleep mode is made. In sleep mode, the CPU is stopped but the DTC and all the on-chip peripheral modules operate on the system clock. CPU register contents are retained.

When an interrupt is requested, sleep mode is canceled causing a transition to active mode and interrupt exception handling starts. Sleep mode cannot be canceled if the I bit in CCR is 1 or the requested interrupt is masked by the interrupt enable bit. After sleep mode is canceled, the high-speed or low-speed clock is selected as the system clock source depending on the SLEEPRS bit setting in LPCR1.

When the $\overline{\text{RES}}$ pin is driven low or any other internal reset occurs, sleep mode is canceled causing a transition to the reset state.

chanhvinh.com



### 6.2.3 Standby Mode

When a SLEEP instruction is executed in active mode with the SSBY bit = 1 in LPCR1, a transition to standby mode is made. In standby mode, clock oscillation is stopped and thus the CPU, DTC, and all the on-chip peripheral modules (except timer RE and WDT) are stopped. However, as long as the rated voltage is supplied, the following contents are retained: the CPU registers, the registers of some on-chip peripheral modules, and on-chip RAM. Additionally, on-chip RAM contents will be retained as long as the voltage rated as the RAM data retention voltage is provided. The I/O ports go to the high-impedance state.

When an interrupt is requested, standby mode is canceled causing a transition to active mode and interrupt exception handling starts. Standby mode cannot be canceled if the I bit in CCR is 1 or the requested interrupt is masked by the interrupt enable bit. After standby mode is canceled, the high-speed or low-speed clock is selected as the system clock source depending on the STBYRS bit setting in LPCR1.

When the $\overline{\text{RES}}$ pin is driven low or any other internal reset occurs, standby mode is canceled causing a transition to the reset state.

## 6.3 Bus Master Clock Division Function

In active or sleep mode, the operating clock for the CPU, DTC, on-chip ROM, and on-chip RAM can be divided independently of the clock supplied to the peripheral modules. Using a divided clock can reduce power consumption.

The operating clock $\phi$s for the bus masters and the on-chip ROM and on-chip RAM can be selected from among $\phi$, $\phi/2$, $\phi/4$, $\phi/8$, $\phi/16$, and $\phi/32$ according to the PHIS[2:0] setting in LPCR3.

### 6.3.1 Reset States

For reset states, see section 3.3, Reset.

chanhvinh.com



## 6.4 Module Standby Function

The module standby function is available for any peripheral module. When a module is set to the module standby state, the clock supply to the module stops placing the module in the power-down state. Setting the corresponding bit to the module in MSTCR to 1 places the module in the module standby state and clearing the bit cancels the module standby state. After release from a reset, all the modules except timer RE are in the module standby state; to use a module, cancel the module standby state of it.

Note that the registers of the module in the module standby state cannot be accessed.

## 6.5 PSC Divider Stop Function

When the peripheral modules do not use the PSC divider output, the PSC divider can be stopped by setting the PSCSTP bit in LPCR1 to 1.

When the PSC divider is stopped, the peripheral modules using $\phi/2$ to $\phi/8192$ can be stopped as shown in table 6.1 (register values are retained). Before setting the PSCSTP bit to 1, set the peripheral modules using the PSC divider output to the module standby state.

After release from a reset, the PSC divider is stopped since the PSCSTP bit is set to 1. For the PSCSTP bit, see section 5.2.3, Power-Down Control Register 1 (LPCR1).

chanhvinh.com



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

chanhvinh.com



# Section 7 ROM

The features of the on-chip flash memory are described below.

## 7.1 Overview

- Programming/erasing method

  Four bytes are programmed simultaneously. A single block is erased at a time; only one block should be erased at a time even when the entire ROM area is to be erased.
- Programming/erasing time

  Programmable ROM programming time: 150 µs (typ.) for 4-byte simultaneous programming, i.e., 38 µs (typ.) per byte

  Data flash programming time: 300 µs (typ.) for 4-byte simultaneous programming, i.e., 75 µs (typ.) per byte

  Erasing time: 200 ms (typ.) per block for the programmable ROM and data flash areas.
- Reprogramming capability: The programmable ROM area can be reprogrammed up to 1000 times and the data flash area can be reprogrammed up to 10000 times.
- Two on-board programming modes

  Boot mode: The on-chip SCI can be used for programming/erasing the user ROM area. In this mode, the communication bit rate between the host and this LSI can be automatically adjusted.

  User mode: Any interface can be used for programming/erasing the user ROM area.
- Programmer mode

  A PROM programmer is used for programming/erasing.
- Protection function

  Flash memory can be protected against erroneous programming and erasure.

  Lock-bit protection function can be set through software.
- PROM-programmer protection/Boot-mode protection

  By writing specified data to a specified address range in user ROM, protection of the user-ROM area in boot mode and PROM-programmer mode can be established.
- Access cycle

  Programmable ROM: One state

  Data flash: Two states

chanhvinh.com



## 7.2 Block Configuration

Figure 7.1 shows the blocks of the flash memory. The user ROM area contains the programmable ROM area for storing the microcomputer's operating program and the data flash area for storing data. In the figure, the thick-line frames each indicate an erasure block (erasing unit); the thin-line frames each indicate a programming unit. The values in the frames are addresses. Erasure can be done in erasure-block units shown in the figure 7.1. Programming can be done in 2-word or 4-byte units, each of which begins at the address whose lower four-bit value is H'0, H'4, H'8, or H'C.

chanhvinh.com



H8S/20103, H8S/20203, and H8S/20223
(programmable ROM: 128 Kbytes, data flash: 8 Kbytes)

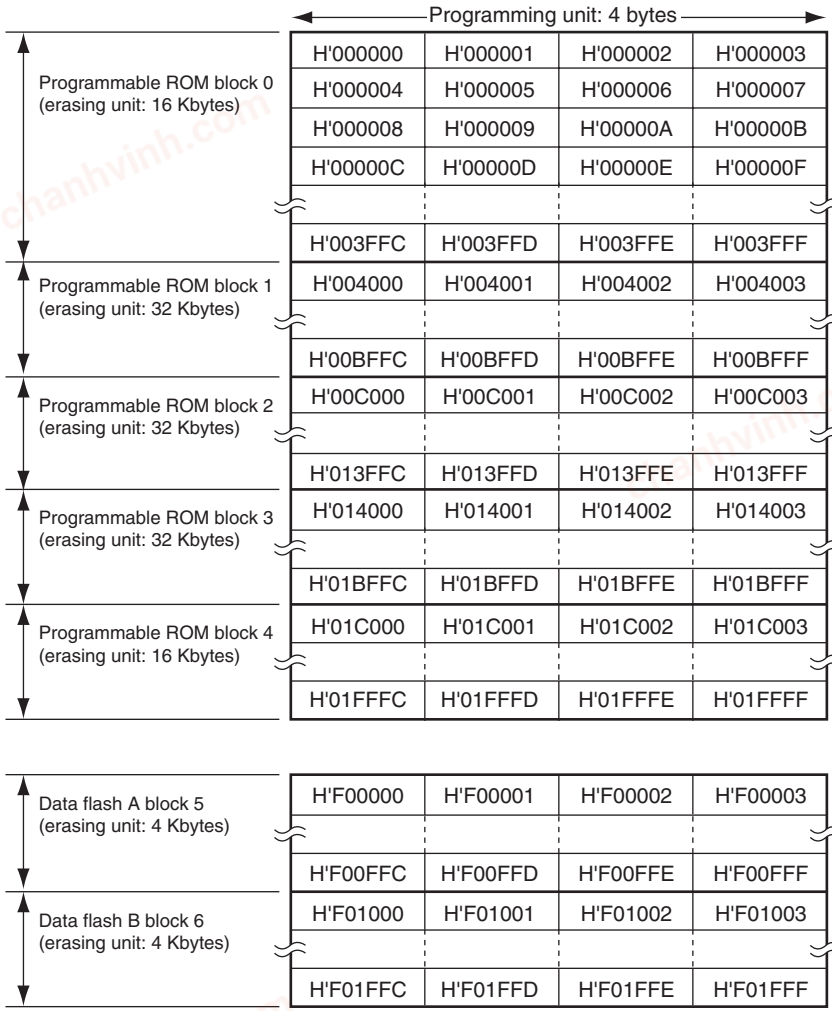



**Figure 7.1 Block Configuration of Flash Memory (1)**

chanhvinh.com



H8S/20102, H8S/20202, and H8S/20222
(programmable ROM: 96 Kbytes, data flash: 8 Kbytes)

Programming unit: 4 bytes

| | | | | |
|---|---|---|---|---|
| Programmable ROM block 0 (erasing unit: 16 Kbytes) | H'000000 | H'000001 | H'000002 | H'000003 |
| | H'000004 | H'000005 | H'000006 | H'000007 |
| | H'000008 | H'000009 | H'00000A | H'00000B |
| | H'00000C | H'00000D | H'00000E | H'00000F |
| | ≈ | | | ≈ |
| | H'003FFC | H'003FFD | H'003FFE | H'003FFF |
| Programmable ROM block 1 (erasing unit: 32 Kbytes) | H'004000 | H'004001 | H'004002 | H'004003 |
| | ≈ | | | ≈ |
| | H'00BFFC | H'00BFFD | H'00BFFE | H'00BFFF |
| Programmable ROM block 2 (erasing unit: 32 Kbytes) | H'00C000 | H'00C001 | H'00C002 | H'00C003 |
| | ≈ | | | ≈ |
| | H'013FFC | H'013FFD | H'013FFE | H'013FFF |
| Programmable ROM block 3 (erasing unit: 16 Kbytes) | H'014000 | H'014001 | H'014002 | H'014003 |
| | ≈ | | | ≈ |
| | H'017FFC | H'017FFD | H'017FFE | H'017FFF |

| | | | | |
|---|---|---|---|---|
| Data flash A block 5 (erasing unit: 4 Kbytes) | H'F00000 | H'F00001 | H'F00002 | H'F00003 |
| | ≈ | | | ≈ |
| | H'F00FFC | H'F00FFD | H'F00FFE | H'F00FFF |
| Data flash B block 6 (erasing unit: 4 Kbytes) | H'F01000 | H'F01001 | H'F01002 | H'F01003 |
| | ≈ | | | ≈ |
| | H'F01FFC | H'F01FFD | H'F01FFE | H'F01FFF |

**Figure 7.1 Block Configuration of Flash Memory (2)**

chanhvinh.com



H8S/20115, H8S/20215, and H8S/20235
(programmable ROM: 256 Kbytes, data flash: 8 Kbytes)

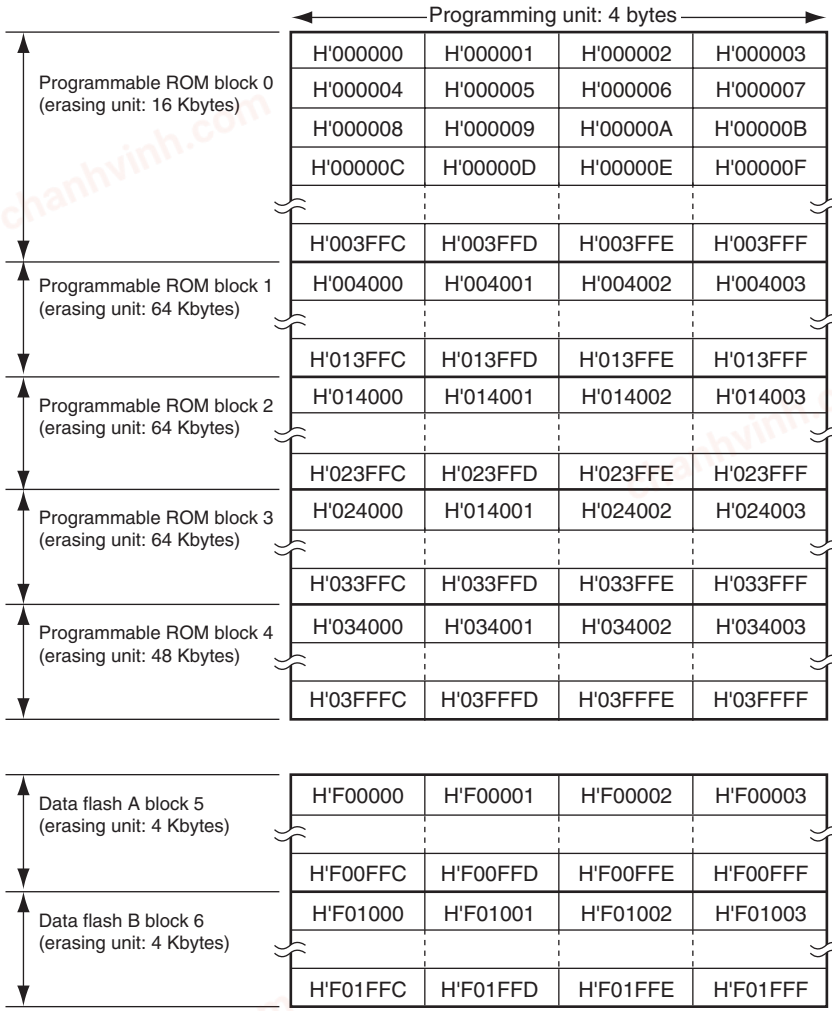



**Figure 7.1 Block Configuration of Flash Memory (3)**

chanhvinh.com



H8S/20114, H8S/20214, and H8S/20234
(programmable ROM: 192 Kbytes, data flash: 8 Kbytes)

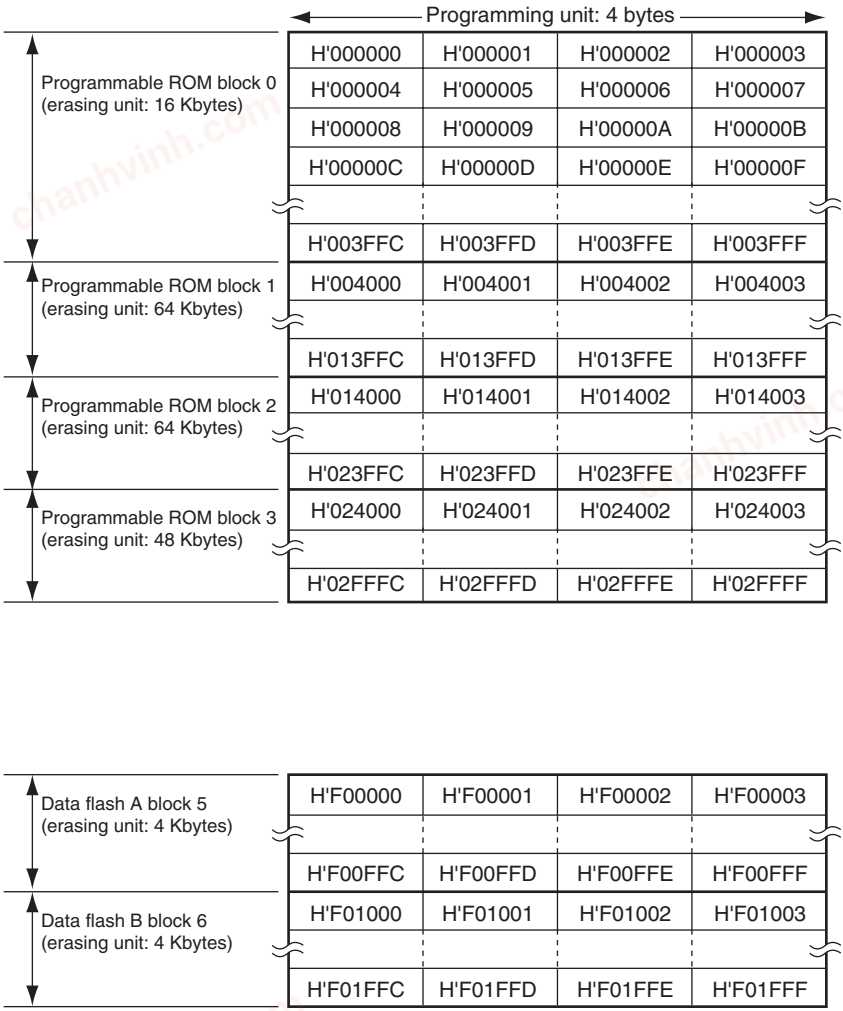



**Figure 7.1 Block Configuration of Flash Memory (4)**

chanhvinh.com



## 7.3 CPU Reprogramming Mode

In CPU reprogramming mode, the user ROM area can be reprogrammed by executing the software commands by the CPU. The software commands should be issued to the specific area to be reprogrammed in the user ROM area.

If an interrupt is requested during erasure operation in CPU reprogramming mode, erasure can be suspended to process the interrupt. This is referred to as erase-suspend function. In erase-suspend mode, the user ROM area can be read through programming.

The CPU has two reprogramming modes, EW0 mode and EW1 mode. Table 7.1 shows differences between the two modes.

**Table 7.1 Differences between EW0 Mode and EW1 Mode**

| Item | EW0 Mode | EW1 Mode |
|---|---|---|
| Area in which a reprogramming-control program can be located | User ROM area | User ROM area |
| Area in which a reprogramming-control program can be executed | A reprogramming-control program must be transferred to RAM before execution. | A reprogramming-control program can be executed in the user ROM area. |
| Area which can be reprogrammed | User ROM area | User ROM area excluding the blocks in which a reprogramming-control program is located. |
| Limitations on software commands | None | The program and erasure commands must not be executed on any block in which a reprogramming-control program is located. |
| Mode after software command execution | Read-array mode | Read-array mode |

chanhvinh.com



| Item | EW0 Mode | EW1 Mode |
|---|---|---|
| CPU state during auto-programming and auto-erasure | Operating state | Hold state (I/O ports retain the states in which they have been before the command is executed.) |
| Flash memory state detection | Read the FMPRSF, FMERSF, and FMEBSF bits in FLMSTR in a program. | |
| Conditions of transition to erase-suspend state | Both the FMSPEN and FMSPREQ bits in FLMCR2 are set to 1.<br>Or, both the FMSPEN and FMISPE bits in FLMCR2 are set to 1 and an interrupt is requested. | The FMSPEN bit in FLMCR2 is set to 1 and an interrupt is requested. |
| Conditions of Interrupt generation | • The flash memory returns from the busy state to the ready state*[1].<br>• The user ROM area is read in the busy state*[1]. | Use of interrupts prohibited. |
| Usage of DTC | Usable*[2] | Usable*[2]*[3] |

Notes: 1. To avoid the generation of access to the user ROM area, set VOFR so that the variable vectors and interrupt processing routines are allocated to RAM.
2. Allocate DTC vectors and processing routines to RAM. Do not use the DTC for access to the user ROM area during E/W processing. If this is ignored, values read will be invalid.
3. Do not use the DTC if the reprogramming-control program is allocated to RAM.

chanhvinh.com



### 7.3.1 EW0 Mode

EW0 mode can be selected by transferring the reprogramming-control program to the RAM, branching to the program in the RAM, setting the FMEWMOD bit in FLMCR1 to 0, and setting the FMCMDEN bit in FLMCR1 to 1 (to enable software commands), in this order.

Programming and erasure operations can be controlled through software commands. Completion of the software command and related information can be read out from the FLMSTR register.

To cause a transition to erase-suspend mode during erasure, set both the FMSPEN and FMSPREQ bits in FLMCR2 to 1 (to enable a transition to erase-suspend mode and to request a transition to erase-suspend mode, respectively). Then wait for the transition time to erase-suspend mode (approximately 50 µs), check that the FMRDY bit in FLMSTR is 1 (ready state), and access the user ROM area. Setting the FMSPREQ bit to 0 resumes erasure.

When the interrupt is used, set the interrupt vector offset register (VOFR) such that access to the user ROM area is not generated. That is, the vectors should have addresses within the RAM and point to interrupt processing routines that are also in the RAM. If the user ROM area is to be read while software commands are enabled (the FMCMDEN bit in FLMCR1 is 1), set bus master operation clock ϕs to a frequency below 5 MHz.

### 7.3.2 EW1 Mode

EW1 mode can be selected by setting the FMEWMOD bit in FLMCR1 to 1, and then setting the FMCMDEN bit in FLMCR1 to 1 (to enable software commands).

Programming and erasure operations can be controlled through software commands. Completion of the software command and related information can be read out from the FLMSTR register.

To cause a transition to erase-suspend mode during erasure, set the FMSPEN bit in FLMCR2 to 1 (to enable a transition to erase-suspend mode), and then execute the erasure command. Note that the interrupt for causing a transition to erase-suspend mode must be enabled beforehand. This allows the interrupt request to be accepted when the transition time to erase-suspend mode has elapsed after the erasure command is executed.

When an interrupt is requested, the FMSPREQ bit is automatically set to 1 (to request a transition to erase-suspend mode), thus suspending erasure. If erasure has not been completed at the end of interrupt processing (FMERSF = 1 in FLMSTR), resume erasure by setting the FMSPREQ bit to 0.

chanhvinh.com



## 7.4 Register Descriptions

- Flash memory control register 1 (FLMCR1)
- Flash memory control register 2 (FLMCR2)
- Flash memory data flash protect register (DFPR)
- Flash memory status register (FLMSTR)

### 7.4.1 Flash Memory Control Register 1 (FLMCR1)

Address: H'FF0660

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | FMLBD | FMWUS | FMEWMOD | FMCMDEN |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7, 6 | — | Reserved | These bits are read as 0. The write value should be 0. | — |
| 5, 4 | — | Reserved | | — |
| 3 | FMLBD*1*2 | Lock bit disable | 0: The lock bits are enabled.<br>1: The lock bits are disabled. | R/W |
| 2 | FMWUS | CPU reprogramming-instruction unit select | 0: Reprogramming through byte instructions<br>1: Reprogramming through word instructions | R/W |
| 1 | FMEWMOD | EW mode select | 0: EW0 mode<br>1: EW1 mode | R/W |
| 0 | FMCMDEN *1*2*3*4 | Flash memory software command enable | 0: Flash memory software commands are disabled.<br>1: Flash memory software commands are enabled. | R/W |

Notes: 1. When setting the bit to 1, first clear the bit to 0 and then immediately set the bit to 1; do not allow any interrupt to be generated between these operations.
2. The bit is cleared to 0 when the FMRDY bit changes from 0 to 1.
3. Set the FMEWMOD bit and then set the FMCMDEN bit to 1.
4. When setting the FMCMDEN bit to 1 while the FMEWMOD bit is 0, be sure to execute the program in the RAM.

FLMCR1 enables/disables reprogramming/erasure, selects the reprogramming/erasure mode, enables/disables lock bits, and selects the reprogramming unit of the flash memory. For specific use, see section 7.6, Programming/Erasing.

chanhvinh.com



- FMLBD bit (lock bit disable)

  This bit disables the lock-bit function. Setting FMLBD to 1 enables erasing/programming the block to which the lock-bit protection is applied. For the relationship between the FMLBD bit and the lock bit for the block, see table 7.2 below. Command sequence error occurs when the erasing/programming command is executed while disabling the erase program.

**Table 7.2 Relationship between FMLBD, Lock Bit, and Corresponding Erasure/Programming Operation**

| FMLBD | Lock Bit | Erasure/Programming Operation |
|---|---|---|
| 1 | — | Erasure/programming possible |
| 0 | 1 (erased state) | |
| | 0 (programmed state) | Erasure/programming impossible |

- FMWUS bit (CPU reprogramming-instruction unit select)

  Setting the FMWUS bit to 0 enables software commands to be issued through byte instructions. Setting the FMWUS bit to 1 enables software commands to be issued through word instructions. For software commands, see section 7.6.1, Software Commands.

- FMEWMOD bit (EW mode select)

  Setting the FMEWMOD bit to 0 and the FMCMDEN bit to 1 selects EW0 mode. Setting the FMEWMOD and FMCMDEN bits to 1 selects EW1.

- FMCMDEN bit (flash memory software command enable)

  Setting the FMCMDEN bit to 1 enables software commands to be accepted. For issuing software commands to the data flash areas, appropriately set the flash memory data flash protect register (DFPR), which is described in section 7.4.3.

chanhvinh.com



### 7.4.2 Flash Memory Control Register 2 (FLMCR2)

Address: H'FF0661

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | FMRDYIE | FMBSYRDIE | FMISPE | FMSPREQ | FMSPEN |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7, 6 | — | Reserved | These bits are read as 0. The write value should be 0. | — |
| 5 | — | Reserved | | — |
| 4 | FMRDYIE *1*2 | Flash read-ready interrupt enable | 0: The ready interrupt is disabled.<br>1: The ready interrupt is enabled. | R/W |
| 3 | FMBSYRDIE *1*3 | Flash busy-read interrupt enable | 0: The busy-read interrupt is disabled.<br>1: The busy-read interrupt is enabled. | R/W |
| 2 | FMISPE*4 | Suspend-request enable by interrupt request | 0: Transition to erase-suspend mode by an interrupt request is disabled.<br>1: Transition to erase-suspend mode by an interrupt request is enabled. | R/W |
| 1 | FMSPREQ *1*5*6*7 | Erase suspend | 0: Erasure is resumed.<br>1: Transition to erase-suspend mode is made. | R/W |
| 0 | FMSPEN *4*8 | Erase-suspend enable | 0: Erase suspend is disabled.<br>1: Erase suspend is enabled. | R/W |

Notes:
1. For programming the flash memory, set the FMSPEN bit to 1.
2. The FMRDYIE bit is cleared to 0 when the FMCMDEN bit changes from 0 to 1.
3. The FMBSYRDIE bit is cleared to 0 when the FMCMDEN bit changes from 0 to 1.
4. When setting the bit to 1, first clear the bit to 0 and then immediately set the bit to 1; do not allow any interrupt to be generated between these operations.
5. The FMSPREQ bit is set to 1 when an interrupt is generated if the FMSPEN bit is 1 in EW1 mode.
6. The FMSPREQ bit is set to 1 when an interrupt is generated if the FMSPEN and FMISPE bits are 1 in EW0 mode.
7. The FMSPREQ bit is cleared to 0 when the FMRDY bit changes from 0 to 1 upon completion of E/W.
8. The FMSPEN bit is cleared to 0 when the FMRDY bit changes from 0 to 1 if the FMSPREQ bit is 0.

chanhvinh.com



FLMCR2 enables/disables flash memory interrupts, enables/controls a transition to erase-suspend mode.

- FMRDYIE bit (flash read-ready interrupt enable)
  Setting the FMRDYIE bit to 1 enables an interrupt to be generated when the flash memory changes from the busy state to the ready state.

- FMBSYRDIE bit (flash busy-read interrupt enable)
  Setting the FMBSYRDIE bit to 1 enables an interrupt to be generated when the user ROM area is accessed while the flash memory is in the busy state.

- FMISPE bit (suspend-request enable by interrupt request)
  Setting the FMISPE bit to 1 in EW0 mode allows the FMSPREQ bit to be automatically set to 1 (to request a transition to erase-suspend mode) thus causing a transition to erase-suspend mode when an interrupt is requested.

- FMSPREQ bit (erase suspend)
  Setting the FMSPREQ bit to 1 causes a transition to erase-suspend mode. To resume erasure, set the FMSPREQ bit to 0.

- FMSPEN bit (erase-suspend enable)
  Setting the FMSPEN bit to 1 enables a transition to erase-suspend mode.

chanhvinh.com



### 7.4.3 Flash Memory Data Flash Protect Register (DFPR)

Address: H'FF0662

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | DFPR1 | DFPR0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 to 2 | — | Reserved | These bits are read as 0. The write value should be 0. | — |
| 1 | DFPR1 | Data flash B E/W disable*1*2 | 0: E/W of data flash B is enabled.<br>1: E/W of data flash B is disabled. | R/W |
| 0 | DFPR0 | Data flash A E/W disable*1*2 | 0: E/W of data flash A is enabled.<br>1: E/W of data flash A is disabled. | R/W |

Notes: 1. When setting the bit to 0, first set the bit to 1 and then immediately set the bit to 0; do not allow any interrupt to be generated between these operations.
2. The DFPR bits are set to 1 when the FMCMDEN bit changes from 0 to 1.

DFPR enables/disables reprogramming of data flash areas in block units. Before reprogramming the data flash areas, cancel the protection against reprogramming.

- DFPR1 bit (data flash B E/W disable)

  Setting the DFPR1 bit to 1 disables software commands to be issued to data flash B. Setting the DFPR1 bit to 0 enables software commands to be issued to data flash B.

- DFPR0 bit (data flash A E/W disable)

  Setting the DFPR0 bit to 1 disables software commands to be issued to data flash A. Setting the DFPR0 bit to 0 enables software commands to be issued to data flash A.

chanhvinh.com



### 7.4.4 Flash Memory Status Register (FLMSTR)

Address: H'FF0663

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | FMRDYIF | FMBSYRDIF | FMEBSF | FMERSF | FMPRSF | — | — | FMRDY |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | FMRDYIF *1*2*3 | Flash read-ready interrupt request flag | 0: The flash read-ready interrupt is not being requested.<br>1: The flash read-ready interrupt is being requested.<br>[Setting condition]<br>• FMRDY changes from 0 to 1.<br>[Clearing condition]<br>• "1" is read from FMRDYIF and then the bit is cleared to 0. | R/W |
| 6 | FMBSYRDIF *2*3*4 | Flash busy-read interrupt request flag | 0: The flash busy-read interrupt is not being requested.<br>1: The flash busy-read interrupt is being requested.<br>[Setting condition]<br>• The user ROM area is accessed while the FMRDY is 0.<br>[Clearing condition]<br>• "1" is read from FMBSYRDIF and then the bit is cleared to 0. | R/W |
| 5 | FMEBSF *3*5 | Erasure/blank-checking status flag | 0: Successful end<br>1: End with an error<br>[Setting conditions]<br>• The erasure command is executed and results in unsuccessful erasure.<br>• The blank-checking command is executed and it is detected that the specified block is not blank.<br>[Clearing condition]<br>• The clear-status command is issued. | R |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 4 | FMERSF | Erase-suspend flag | 0: Erase-suspend function is not being used.<br>1: Erase-suspend function is being used.<br>[Setting condition]<br>• Erase-suspend mode is being used.<br>[Clearing condition]<br>• Erase-suspend mode is not being used. | R |
| 3 | FMPRSF *3*5 | Programming status flag | 0: Successful end<br>1: End with an error<br>[Setting conditions]<br>• The programming command is executed and results in unsuccessful programming.<br>• The lock-bit program command is executed and results in unsuccessful programming.<br>[Clearing condition]<br>• The clear-status command is issued. | R |
| 2 | — | Reserved | This bit is read as 0. The write value should be 0. | — |
| 1 | — | Reserved | Reading this bit returns the value same as the FMRDY value. The write value should be 1. | — |
| 0 | FMRDY | Flash memory ready/busy status flag | 0: Busy (programming or erasure in progress)<br>1: Ready<br>[Setting condition]<br>• The flash memory is not being programmed or erased.<br>[Clearing condition]<br>• The flash memory is being programmed or erased. | R |

Notes: 1. The FMRDYIF bit is set to 1 when the FMRDY bit changes from 0 to 1.
2. When setting the bit to 0, first read 1 from the bit and then write 0 to the bit.
3. The bit cannot be set to 1 through software.
4. The FMBSYRDIF bit is set to 1 when the ROM area is accessed while the FMRDY bit is 0.
5. The bit is cleared to 0 when the clear-status command is executed.

chanhvinh.com



- FMRDYIF (flash read-ready interrupt request flag)

  The FMRDYIF bit indicates that the flash memory has changed from the busy state to the ready state. When the FMRDYIF bit is set to 1 while the FMRDYIE bit is 1, an interrupt request is generated.

- FMBSYRDIF (flash busy-read interrupt request flag)

  The FMBSYRDIF bit indicates that the user ROM area is accessed while the flash memory is in the busy state. When the FMBSYRDIF bit is set to 1 while the FMBSYRDIE bit is 1, an interrupt request is generated.

- FMEBSF (erasure/blank-checking status flag)

  The FMEBSF bit is a read-only bit indicating the state when erasure/blank-checking command is executed.

- FMERSF (erase-suspend flag)

  The FMERSF bit is a read-only bit indicating the state of erase-suspend mode.

- FMPRSF (programming status flag)

  The FMPRSF bit is a read-only bit indicating the state when programming command is executed.

- FMRDY (flash memory ready/busy status flag)

  The FMRDY bit indicates the state of flash memory operation.

chanhvinh.com



## 7.5 On-Board Programming

The flash memory can be programmed/erased on board (boot mode and user mode), or by using a PROM programmer (programmer mode). When the reset is released, this LSI enters one of these modes depending on the levels of the signals input on the TEST, $\overline{\text{NMI}}$, and ports, as shown in table 7.3. The levels of these signals must be fixed at least 80 µs before the reset is released.

When this LSI enters boot mode, the built-in boot program is initiated. The boot program transfers the programming-control program to the on-chip RAM, erases the flash memory areas entirely, and then executes the programming-control program. Boot mode is useful for on-board initial programming as well as forced recovery when programming/erasure in user mode is disabled. User mode is useful for erasing and reprogramming the specified blocks, which function is achieved by branching to the programming/erasure processing programs prepared by the user.

**Table 7.3 Pin Levels and Programming Mode Selection**

| TEST | NMI | P85 | PB3 | PB2 | PB1 | PB0 | LSI Modes after Release from a Reset |
|---|---|---|---|---|---|---|---|
| 0 | 1 | × | × | × | × | × | User mode |
| 0 | 0 | 1 | × | × | × | × | Boot mode |
| 1 | × | × | 0 | 0 | 0 | 0 | Programmer mode |

Note: ×: Do not care.

### 7.5.1 Boot Mode

In boot mode, control commands and data for programming are transmitted from the externally connected host via SCI3_1 to program/erase the user ROM area.

In boot mode, it is necessary to prepare the tool for transmitting control commands and data for programming, and the data for programming in the host. Asynchronous mode is used for serial communication. Figure 7.2 shows the system configuration in boot mode. Although interrupt requests are ignored in boot mode, interrupt requests should be disabled by the system.

chanhvinh.com



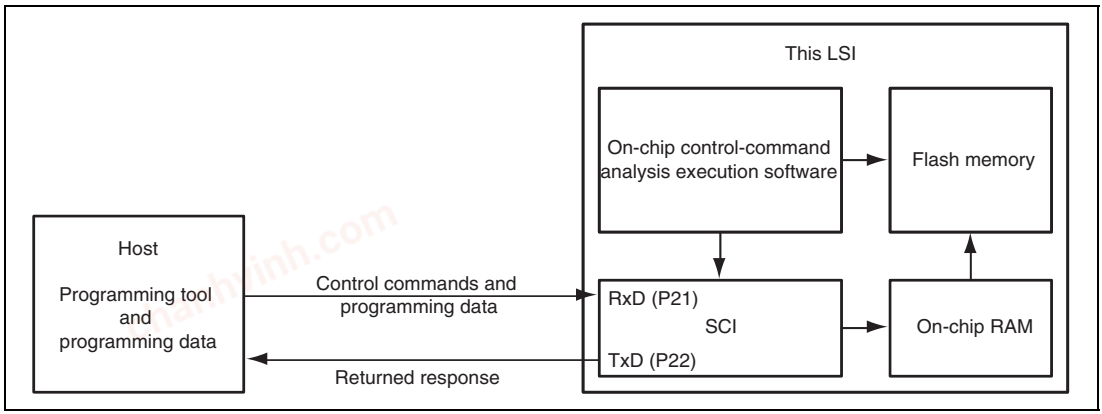



**Figure 7.2 System Configuration in Boot Mode**

### (1) Serial Interface Settings by Host

SCI3_1 is set to asynchronous mode, in which the serial transmission/reception format is set to 8-bit data, one stop-bit, and no parity.

When this LSI enters boot mode, the built-in boot program is initiated. When the boot program is initiated, this LSI measures the low-level period of asynchronous serial communication data (H'00) transmitted continuously from the host. This LSI then calculates the bit rate of transmission from the host, and adjusts the SCI bit rate so that it should match that of the host.

After completing the bit rate adjustment, this LSI transmits one H'00 byte to the host to signal completion of bit rate adjustment. When successfully having received this completion signal, the host should transmit one H'55 byte to this LSI. When not, boot mode should be initiated again.

Table 7.4 shows the automatically adjustable bit rates for the host.

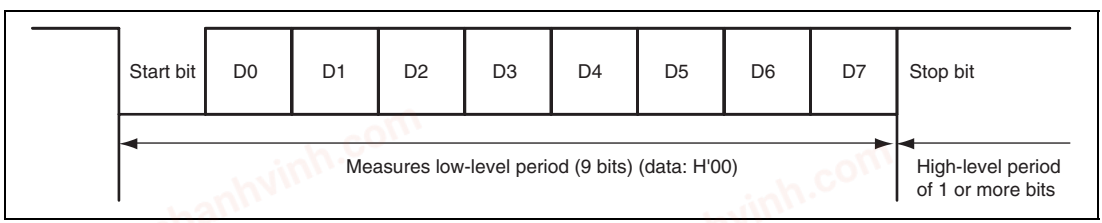



**Figure 7.3 Automatic Adjustment of Bit Rates**

chanhvinh.com



**Table 7.4 Automatically Adjustable Bit Rates for the Host**

| Host Bit Rate |
|---|
| 9600 bps |
| 4800 bps |
| 2400 bps |

Note: Automatic adjustment of the SCI bit rate to 9600 bps will not be possible in some cases. If the signal indicating completion of bit-rate adjustment is not transmitted, restart the LSI in boot mode after reducing the bit rate.

chanhvinh.com



### (2) State Transition

Figure 7.4 shows the state transitions in boot mode.

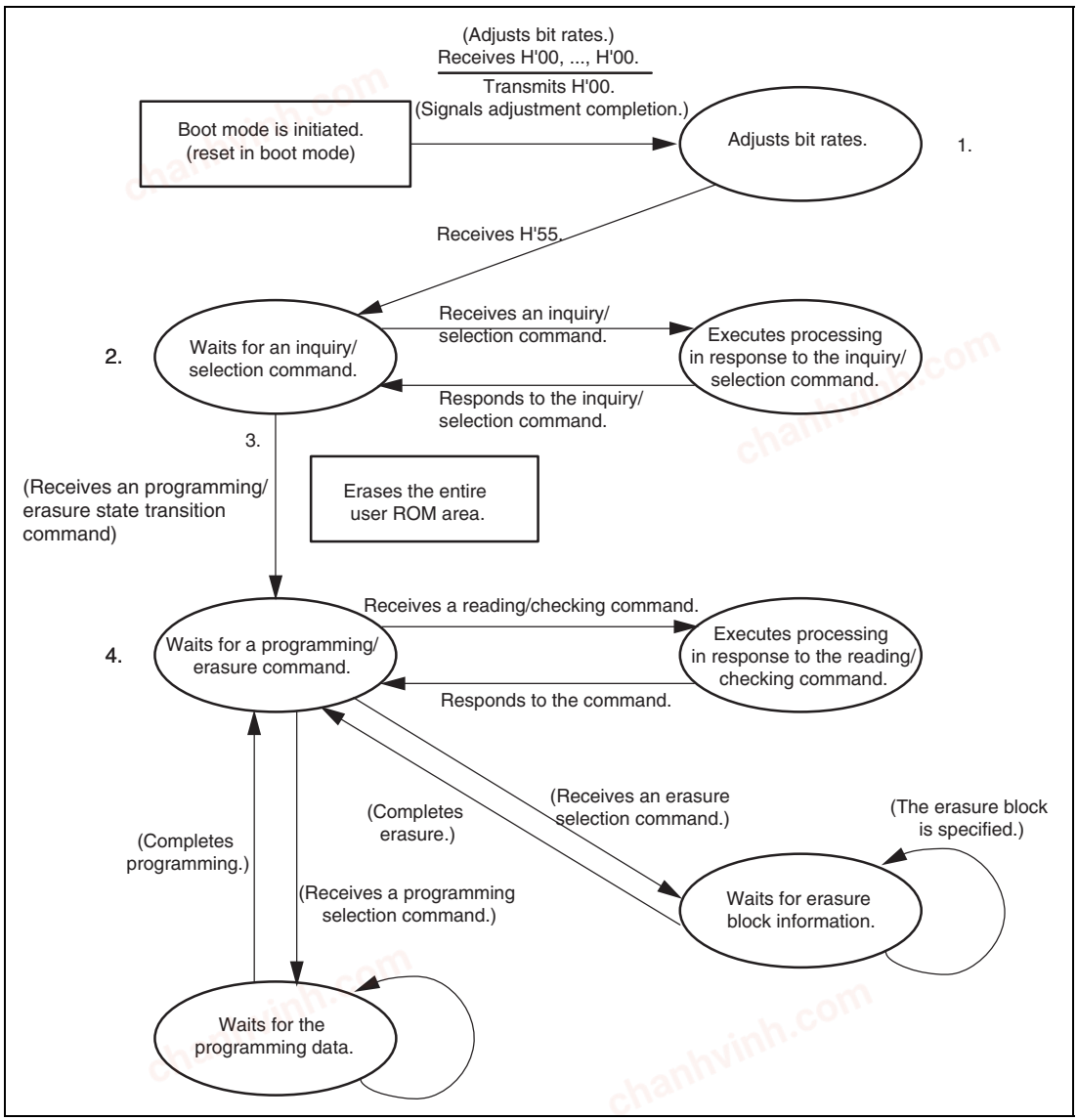



**Figure 7.4 State Transitions in Boot Mode**

chanhvinh.com



1. After boot mode is initiated, this LSI adjusts the SCI3_1 bit rate so that it should match the host's bit rate.
2. This LSI sends the requested information to the host in response to inquiries regarding the size, configuration, and start addresses of the user ROM areas, information on the supported devices, etc.
3. On receiving a programming/erasure state transition command, this LSI erases the entire user ROM area automatically.
4. When completing erasure of the user ROM area, this LSI enters the programming/erasure-command wait state. After transmission of the programming selection command, the host should transmit the address at which the programming should start and the programming data. When programming is completed, the host should transmit H'FFFFFFFF as the programming start address to terminate programming. This allows this LSI to return to the programming/erasure-command wait state from the programming-data wait state. If the above programming-termination command is once issued to an area in an erasure block and when that block is to be programmed again, erase the block before programming. Figure 7.5 shows an example of an erasure block containing the area that has been already programmed.

   On receiving an erasure selection command, this LSI enters the erasure-block-information wait state. After transmission of the erasure selection command, the host should transmit the erasure block number. When erasure is completed, the host should transmit H'FF as the erasure block number. This allows this LSI to return to the programming/erasure-command wait state from the erasure-block-information wait state. Note that erasure is necessary only when programming is once done in boot mode and then only a specific block is to be reprogrammed without applying a reset-start. If the necessary programming can be done in a single operation, such erasure processing is unnecessary because all the blocks are erased before this LSI enters the wait state for programming, erasure, or other commands. In addition to the programming/erasure commands, there are commands for sum checking and blank checking (erasure checking) of the user ROM areas, memory reading, and acquiring the current state information.

Note that data can be read from the user ROM area only after the user ROM area has been automatically erased and then programmed.

chanhvinh.com



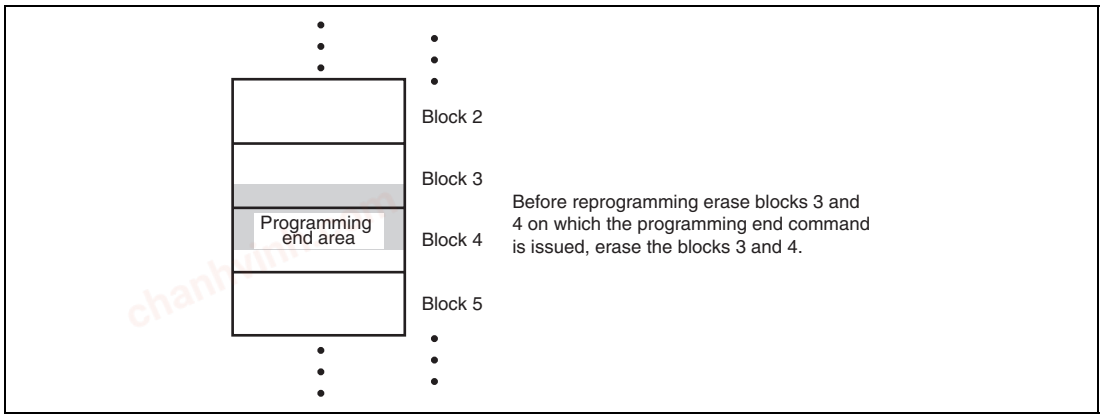



**Figure 7.5 Example of Erase Block Including Programmed Area**

### 7.5.2 Specifications of Standard Serial Communication Interface in Boot Mode

The boot program activated in boot mode communicates with the host via the on-chip SCI3_1. The following describes the specifications of the serial communication interface between the host and the boot program.

The boot program has three states.

1. Bit-rate adjustment state

   In this state, the boot program adjusts the SCI3_1 bit rate to match that of the host to perform serial communication with the host. When this LSI is started up in boot mode, the boot program is activated and enters the bit-rate adjustment state, in which it receives command from the host and adjusts the bit rate accordingly. After bit rate adjustment is completed, the boot program enters the inquiry/selection state.
2. Inquiry/selection state

   In this state, the boot program responds to inquiry commands from the host. The device, clock mode, and bit rate are selected. Upon completion of selection, the boot program enters the programming/erasure state in response to the programming/erasure state transition command. Before entering the programming/erasure state, the boot program transfers the erasure-related libraries to the on-chip RAM and erases the user ROM areas.
3. Programming/erasure state

   In this state, the boot program executes programming/erasure. The boot program transfers the program for programming/erasure to the on-chip RAM according to the command transmitted from the host and executes programming/erasure. The boot program also executes sum checking and blank checking as directed using the corresponding commands.

chanhvinh.com



Figure 7.6 shows the boot program states and processing flow.

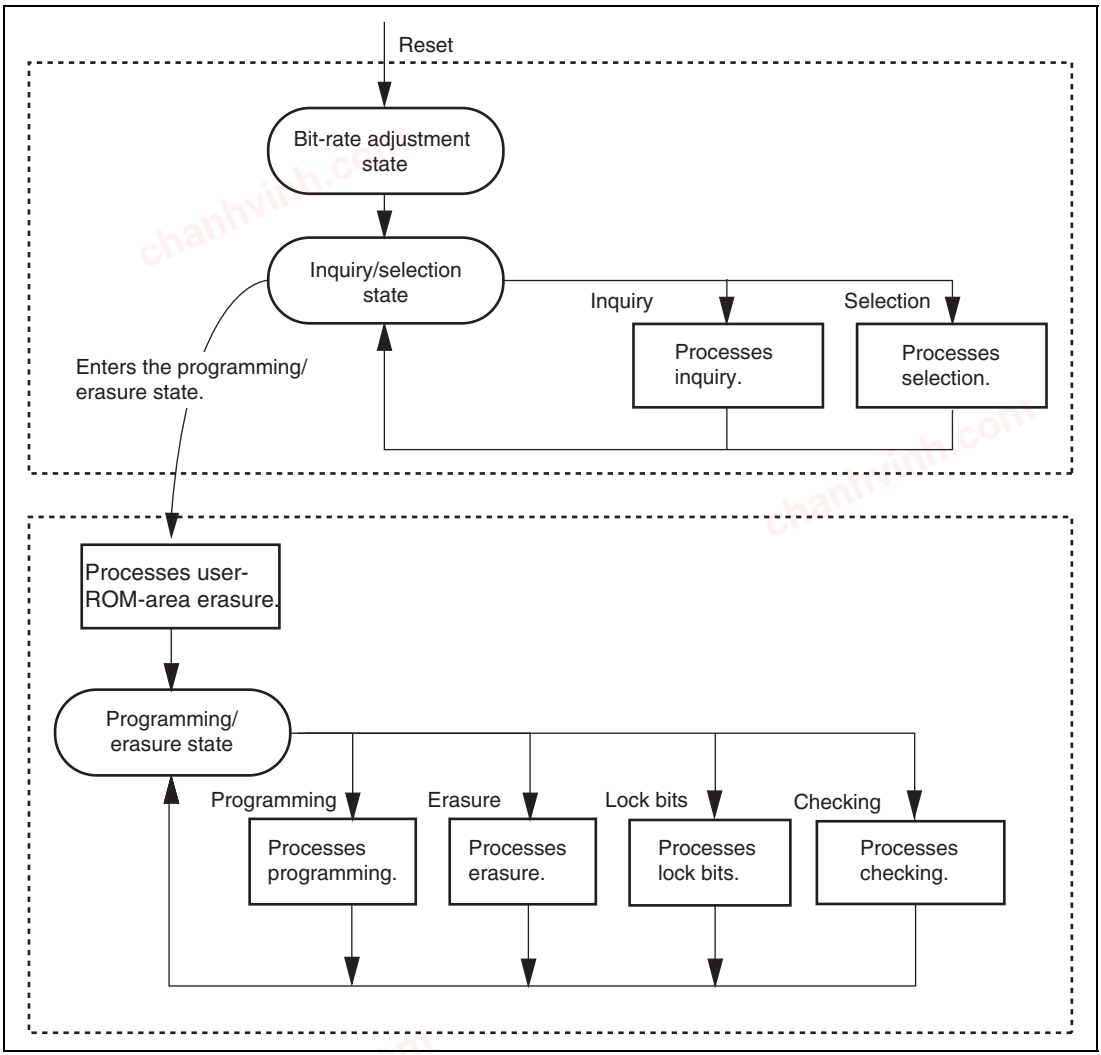



**Figure 7.6 Boot Program States and Processing Flow**

chanhvinh.com



### (1) Bit-Rate Adjustment State

In the bit-rate adjustment state, the boot program measures the low-level period of H'00 transmitted from the host and calculates the bit rate according to the measurement. The bit rate can be changed using the new-bit-rate selection command. On completion of bit rate adjustment, the boot program enters the inquiry/selection state. Figure 7.7 shows the sequence of bit rate adjustment.

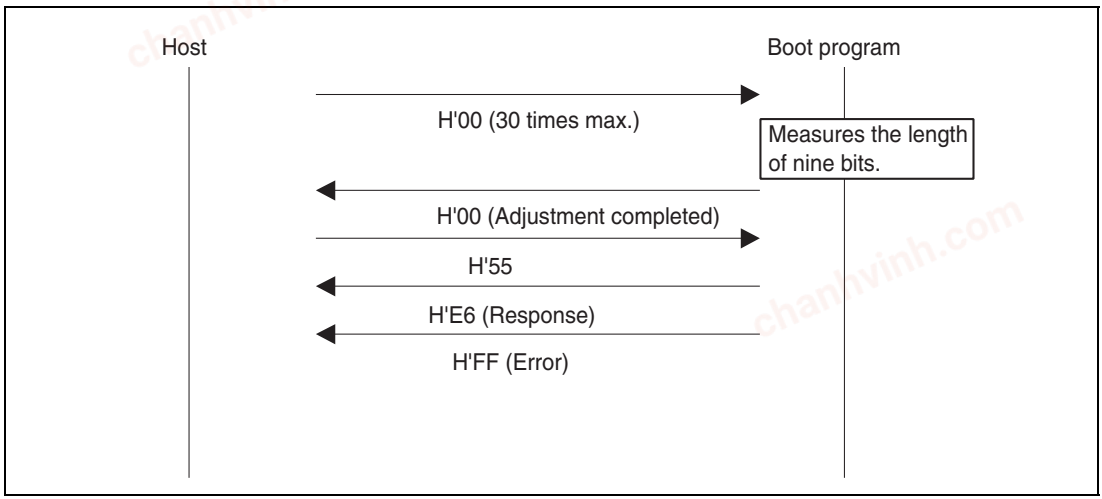



**Figure 7.7 Sequence of Bit Rate Adjustment**

### (2) Communication Protocol

1. One-character command or one-character response

   A command or response consisting of one character used for inquiry or ACK code indicating a successful end.

2. n-character command or n-character response

   A command or response requiring 128 bytes of data used as a selection command or a response to an inquiry. The length of programming data is defined separately and so data size (length) is omitted here.

3. Error response

   Response to a command which has caused an error; two bytes, consisting of the error response and error code.

4. 128-byte programming command

   This command does not include its data size information. The data size can be acquired from the response to the programming-size inquiry command.

chanhvinh.com



5. Response to a memory read command
   This response includes 4-byte size information.

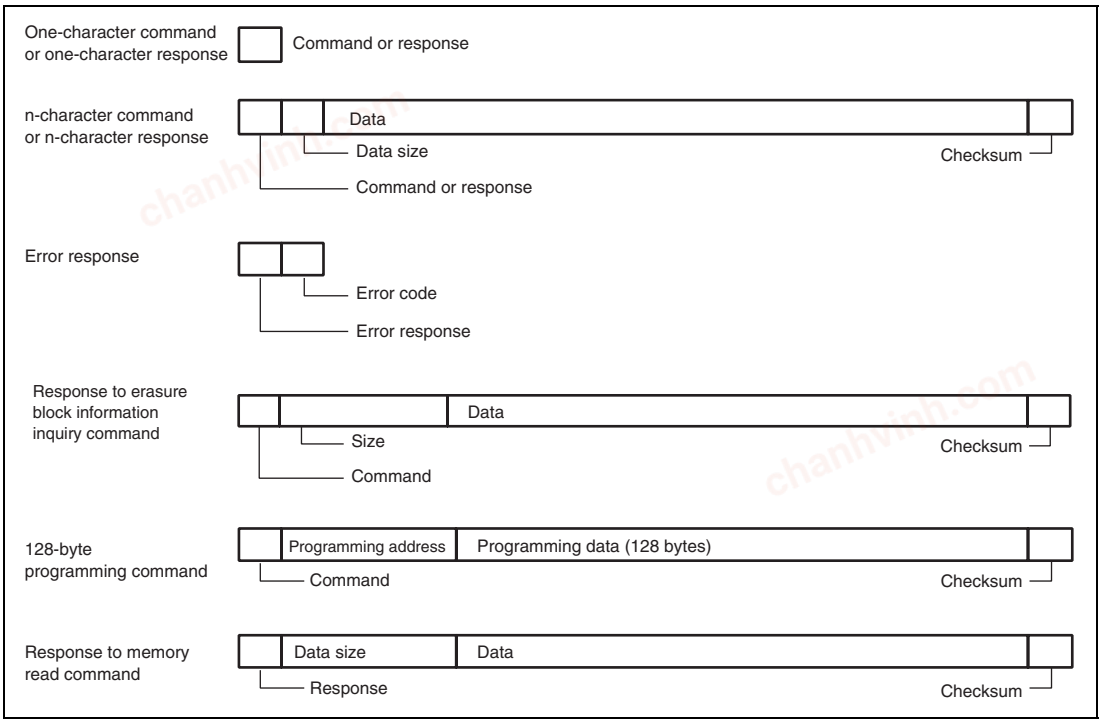



**Figure 7.8 Formats in Communication Protocol**

- Command (1 byte): Inquiry, selection, programming, erasure, checking, etc.
- Response (1 byte): Response to an inquiry
- Size (1 or 2 bytes): The size of transmit/receive data excluding the command code, response, size, and checksum, or the size of data in a response to erasure block information inquiry command
- Data (n bytes): Particular data for a command or response
- Checksum (1 byte): This is set so that the total sum of the values from the command- code field or response through the SUM field should be H'00.
- Error response (1 byte): Error response to a command
- Error code (1 byte): Type of an error that has occurred
- Programming address (4 bytes): Address for programming
- Programming data (n bytes): Data for programming. "n" is known from the response to the programming-size inquiry command. (Data size for this LSI is fixed 128 bytes.)

chanhvinh.com



- Data size (4 bytes): Size (four-byte length) of data in the response to the memory read command.

### (3) Inquiry/Selection State

In this state, the boot program returns the information on the flash ROM in response to inquiry commands from the host, and selects the device, clock mode, and bit rate in response to the relevant selection commands.

Table 7.5 lists inquiry/selection commands.

chanhvinh.com



**Table 7.5 Inquiry/Selection Commands**

| Command | Command Name | Function |
|---|---|---|
| H'20 | Supported-device inquiry | Obtains the device code and product name. |
| H'10 | Device selection | Selects a device (code) |
| H'21 | Clock mode inquiry | Obtains the number of selectable clock modes and the values corresponding to each of the modes. |
| H'11 | Clock mode selection | Selects the clock mode |
| H'22 | Frequency-division-ratio inquiry | Obtains the number of frequency types, the number of frequency division ratios, and the specific frequency division ratio values |
| H'23 | Operating-frequency inquiry | Obtains the maximum and minimum operating values for operating frequency of the main and peripheral clocks |
| H'25 | Programmable ROM information inquiry | Obtains the number of programmable ROM area and start and end addresses of each area |
| H'26 | Erasure-block information inquiry | Obtains the number of blocks and the start and end addresses of each block |
| H'27 | Programming-size inquiry | Obtains the size of the unit for programming |
| H'2A | Data flash inquiry | Checks whether or not data flash is present |
| H'2B | Data flash information inquiry | Obtains the number of data flash areas and the start and end addresses of each area |
| H'2C | Clock switching information inquiry | Checks whether or not switching between an internal and external clock source is possible |
| H'3F | New bit-rate selection | Selects a new bit rate |
| H'40 | Programming/erasure state transition | Erases the user ROM area and entering the programming/erasure state |
| H'4F | Boot-program state inquiry | Checks the state of processing by the boot program. |
| H'60 | ID authentication | Obtains the ID code of the programmable ROM. |

The selection commands should be transmitted from the host in the following order: device selection (H'10), clock mode selection (H'11), and new bit-rate selection (H'3F). If the same selection command is transmitted more than one time, the last one is valid.

chanhvinh.com



All the commands in table 7.5 except for the boot-program-state inquiry command (H'4F) are valid until the boot program accepts the programming/erasure state transition command (H'40). That is, until the transition command is accepted, the host can repeatedly send inquiry and selection commands in table 7.5. The boot-program-state inquiry command (H'4F) is valid even after the boot program accepts the transition-to-programming/erasure-state command (H'40).

### (a) Supported-Device Inquiry

In response to a supported-device inquiry command, the boot program returns the device codes of the supported devices and the device name of the boot program.

| Command | H'20 |
|---|---|

- Command H'20 (1 byte): Supported-device inquiry

| Response | H'30 | Size | Number of devices | |
|---|---|---|---|---|
| | Number of characters | Device code | | Device name |
| | SUM | | | |

- Response H'30 (1 byte): Response to a supported-device inquiry command
- Size (1 byte): The size of transmit/receive data excluding the response-command, size, and checksum fields. Here, it refers to the total size used by the number-of-devices, number-of-characters, device-code, and device-name fields.
- Number of devices (1 byte): The number of device types supported by the boot program in the microcomputer.
- Number of characters (1 byte): The number of characters in the device-code and device-name fields.
- Device code (4 bytes): The supported device code (ASCII code)
- Device name (128 bytes): The supported device name (ASCII code)
- SUM (1 byte): Checksum

chanhvinh.com



### (b) Device Selection

In response to a device selection command, the boot program sets the specified supported device as the selected device. The boot program will return the information on the selected device in response to the subsequent inquiries.

| | | | | |
|---|---|---|---|---|
| Command | H'10 | Size | Device code | SUM |

- Command H'10 (1 byte): Device selection
- Size (1 byte): The number of characters in the device-code field (fixed to four).
- Device code (4 bytes): The device code that has been returned in response to a supported-device inquiry command (ASCII code)
- SUM (1 byte): Checksum

| | |
|---|---|
| Response | H'06 |

- Response H'06 (1 byte): Response to a device selection command
  The ACK code is returned when the specified device code corresponds to one of the supported devices.

Error response

| | |
|---|---|
| H'90 | ERROR |

- Error response H'90 (1 byte): Error response to a device selection command
- ERROR (1 byte): Error code
  H'11: Checksum error
  H'21: Device code error indicating device code disagreement

### (c) Clock Mode Inquiry

In response to a clock mode inquiry, the boot program returns the information on the selectable clock modes.

| | |
|---|---|
| Command | H'21 |

- Command H'21 (1 byte): Clock mode inquiry

| | | | | | |
|---|---|---|---|---|---|
| Response | H'31 | Size | Mode | … | SUM |

chanhvinh.com



- Response H'31 (1 byte): Response to a clock mode inquiry command
- Size (1 byte): The total size of the mode fields
- Mode (1 byte): Selectable clock modes (example: H'00 denotes clock mode)
- SUM (1 byte): Checksum

### (d) Clock Mode Selection

In response to a clock mode selection command, the boot program sets the specified clock mode as the selected clock mode. The boot program will return the information on the selected clock mode in response to the subsequent inquiries.

The clock-mode selection command should be transmitted after the device selection command (H'10).

| | | | | |
|---|---|---|---|---|
| Command | H'11 | Size | Mode | SUM |

- Command H'11 (1 byte): Clock mode selection
- Size (1 byte): The number of characters in the mode field (fixed to one).
- Mode (1 byte): The clock mode that has been returned in response to a clock mode inquiry command.
- SUM (1 byte): Checksum

| | |
|---|---|
| Response | H'06 |

- Response H'06 (1 byte): Response to a clock mode selection command
  The ACK code is returned when the specified clock mode corresponds to one of the selectable clock modes.

Error response

| | |
|---|---|
| H'91 | ERROR |

- Error response H'91 (1 byte): Error response to a clock mode selection command
- ERROR (1 byte): Error code
  H'11: Checksum error
  H'22: Clock mode error indicating clock mode disagreement

Even if value H'00 or H'01 has been returned in response to the clock-mode inquiry command as the number of modes, it is required to select the clock mode for each value.

chanhvinh.com



### (e) Frequency-Division-Ratio Inquiry

In response to a frequency-division-ratio inquiry command, the boot program returns the information on the selectable frequency division ratios.

| Command | H'22 |
|---|---|

- Command H'22 (1 byte): Frequency-division-ratio inquiry

| Response | H'32 | Size | Number of types | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Number of Frequency division ratios | Frequency division ratio | … | | | | | |
| | … | | | | | | | |
| | SUM | | | | | | | |

- Response H'32 (1 byte): Response to a frequency-division-ratio inquiry command
- Size (1 byte): The total size of the number-of-types, number-of-frequency-division-ratios, and frequency-division-ratio fields.
- Number of types (1 byte): The number of operating clock signals for which frequency division ratios can be selected for the device.

  (For example, the value is H'02 if frequency division ratio settings can be made for the frequencies of the main and peripheral module operating clock signals.)
- Number of frequency division ratios (1 byte): The number of selectable frequency division ratios for each operating clock signal.

  (For example, the number of selectable frequency division ratios for the main or peripheral module operating clock signal.)
- Frequency division ratio (1 byte):

  The negative numerical value by which the frequency is divided. (Example: H'FE (-2) when the frequency is divided by two.)

  As many frequency-division-ratio fields are repeated as the number of corresponding frequency division ratios; and the combinations of the former and latter fields are repeated as many times as the number of types (= number of operating clock signals).
- SUM (1 byte): Checksum

chanhvinh.com



### (f) Operating-Frequency Inquiry

In response to an operating-frequency inquiry command, the boot program returns the number of operating frequency types and the respective maximum and minimum frequencies.

| Command | H'23 |
|---|---|

- Command H'23 (1 byte): Operating-frequency inquiry

| Response | H'33 | Size | Number of operating frequencies |
|---|---|---|---|
| | Minimum operating frequency | | Maximum operating frequency |
| | … | | |
| | SUM | | |

- Response H'33 (1 byte): Response to an operating-frequency inquiry command
- Size (1 byte): The total size of the number-of-operating-frequencies, maximum-frequency, and minimum-frequency fields.
- Number of operating frequencies (1 byte): The number of operating frequency types required for the device

  (For example, the value is H'02 if the main and peripheral module operating frequencies are required.)
- Minimum operating frequency (2 bytes): The minimum frequency of a frequency-multiplied or divided clock signal

  The values in the minimum- and maximum-operating-frequency fields are obtained by multiplying the operating frequency (MHz; to the second decimal place) by 100. (For example, when the frequency is 20.00 MHz, 20.00 is multiplied by 100 to be 2000, and so H'07D0 is returned here.)
- Maximum operating frequency (2 bytes): The maximum frequency of a frequency-multiplied or divided clock signal.

  As many pairs of the minimum- and maximum-operating-frequency fields are continued as the number of operating frequencies (= number of operating frequency types).
- SUM (1 byte): Checksum

chanhvinh.com



### (g) Programmable ROM Information Inquiry

In response to a programmable ROM information inquiry command, the boot program returns the number of programmable ROM areas and their addresses.

| Command | H'25 |
|---|---|

- Command H'25 (1 byte): Programmable ROM information inquiry

| Response | H'35 | Size | Number of areas |
|---|---|---|---|
| | Start address of an area | | End address of an area |
| | … | | |
| | SUM | | |

- Response H'35 (1 byte): Response to a programmable ROM information inquiry command
- Size (1 byte): The total size of the fields for the number of areas, start addresses of areas, and end addresses of areas.
- Number of areas (1 byte): The number of consecutive programmable ROM areas
  (H'01 is returned when the programmable ROM areas are continuous.)
- Start address of an area (4 bytes): Address where the area starts
- End address of an area (4 bytes): Address where the area ends
  The number of address pairs is the same as the number of areas.
- SUM (1 byte): Checksum

### (h) Erasure-Block Information Inquiry

In response to an erasure-block information inquiry command, the boot program returns the number of erasure blocks and their addresses.

| Command | H'26 |
|---|---|

- Command H'26 (1 byte): Erasure-block information inquiry

| Response | H'36 | Size | Number of blocks |
|---|---|---|---|
| | Start address of the block | | End address of the block |
| | … | | |
| | SUM | | |

chanhvinh.com



- Response H'36 (1 byte): Response to an erasure-block information inquiry command
- Size (2 bytes): The total size of the number-of-blocks, start-address-of-the-block, and end-address-of-the-block fields.
- Number of blocks (1 byte): The number of flash memory blocks to be erased
- Start address of the block (4 bytes): The start address of the block
- End address of the block (4 bytes): The end address of the block

  As many pairs of the start-address-of-the-block and end-address-of-the-block fields are continued as the number of blocks.
- SUM (1 byte): Checksum

### (i) Programming-Size Inquiry

In response to a programming-size inquiry command, the boot program returns the size of a unit for programming.

| Command | H'27 |
|---|---|

- Command H'27 (1 byte): Programming-size inquiry

| Response | H'37 | Size | Programming size | SUM |
|---|---|---|---|---|

- Response H'37 (1 byte): Response to a programming-size inquiry command
- Size (1 byte): The number of characters in the programming-size field (fixed to 2)
- Programming size (2 bytes): The size of a unit for programming

  Programming data is received in the unit specified here.
- SUM (1 byte): Checksum

chanhvinh.com



### (j) Data Flash Inquiry

In response to a data flash inquiry command, the boot program returns an indicator of whether or not data-flash memory is present.

| Command | H'2A |
|---|---|

- Command H'2A (1 byte): Data flash inquiry

| Response | H'3A | Size | Presence | SUM |
|---|---|---|---|---|

- Response H'3A (1 byte): Response to a data flash inquiry command
- Size (1 byte): The number of characters in the presence field (fixed to 1)
- Presence (1 bytes): Presence of data flash
  H'00: Data flash is not present.
  H'01: Data flash is present.
- SUM (1 byte): Checksum value that makes the sum of the bytes from the command to the SUM byte become H'00.

### (k) Data Flash Information Inquiry

In response to a data flash information inquiry command, the boot program returns the number of data flash areas and their addresses.

| Command | H'2B |
|---|---|

- Command H'2B (1 byte): Data flash information inquiry

| Response | H'3B | Size | Number of areas |
|---|---|---|---|
| | Start address of an area | | End address of an area |
| | … | | |
| | SUM | | |

- Response H'3B (1 byte): Response to a data flash information inquiry command
- Size (1 byte): The total size of the fields for the number of areas, start addresses of areas, and end addresses of areas.

chanhvinh.com



- Number of areas (1 byte): The number of contiguous data-flash areas

  H'01: Consecutive data flash areas

  H'00: No data flash area
- Start address of an area (4 bytes): Address where the area starts. This value is omitted if there is no data-flash area.
- End address of an area (4 bytes): Address where the area ends. This value is omitted if there is no data-flash area.

  The number of address pairs is the same as the number of areas.
- SUM (1 byte): Checksum value that makes the sum of the bytes from the command to the SUM byte become H'00.

### (l) Clock Switching Information Inquiry

In response to a clock switching information inquiry command, the boot program returns an indicator of whether or not switching between an internal and external clock source is possible.

| | | |
|---|---|---|
| Command | H'2C | |

- Command H'2C (1 byte): Clock switching information inquiry

| | | | | |
|---|---|---|---|---|
| Response | H'3C | Size | Switching possibility | SUM |

- Response H'3C (1 byte): Response to a clock switching information inquiry command
- Size (1 byte): The number of characters in the switching possibility field (fixed to 1)
- Switching possibility (1 bytes): Possibility of switching between internal and external clock sources

  H'00: Switching is impossible.

  H'01: Switching is possible.
- SUM (1 byte): Checksum value that makes the sum of the bytes from the command to the SUM byte become H'00.

chanhvinh.com



### (m) New Bit-Rate Selection

In response to a new bit-rate selection command, the boot program changes the bit rate settings to those of the specified one, and responds to the acknowledgement from the host at the new bit rate.

The new bit-rate selection command should be transmitted after the clock-mode selection command.

| Command | H'3F | Size | Bit rate | Input frequency |
|---|---|---|---|---|
| | Number of frequency division ratio types | Frequency division ratio 1 | Frequency division ratio 2 | |
| | SUM | | | |

- Command H'3F (1 byte): New bit-rate selection
- Size (1 byte): The total size of the bit-rate, input-frequency, number-of-frequency-division-ratio-types, and frequency-division-ratio fields.
- Bit rate (2 bytes): New bit rate

  The value to be set here is obtained by dividing the desired bit rate by 100. (For example, to select the bit rate of 19200 bps, 19200 is divided by 100 to be 192, and so H'00C0 should be set.)
- Input frequency (2 bytes): The frequency of the clock input to the boot program.

  Bit 15: Clock source (0: external clock, 1: on-chip oscillator)

  Bits 14 to 0: A frequency of the clock to be input to the device

  The value to be set here is obtained by multiplying the input frequency (MHz; to the second decimal place) by 100. (For example, to select the input frequency of 20.00 MHz, 20.00 is multiplied by 100 to be 2000, and so H'07D0 should be set.)
- Number of frequency division ratio types (1 byte): The number of selectable frequency division ratios for the device.

  (The value is usually H'02 because the main and peripheral module operating frequencies can be usually selected.)
- Frequency division ratio 1 (1 byte): Frequency division ratio for the main operating frequency.

  The negative numerical value by which the frequency is divided. (Example: H'FE (-2) when the frequency is divided by two.)

chanhvinh.com



- Frequency division ratio 2 (1 byte): Frequency division ratio for the peripheral operating frequency.

  The negative numerical value by which the frequency is divided. (Example: H'FE (-2) when the frequency is divided by two.)
- SUM (1 byte): Checksum

| Response | H'06 |
|---|---|

- Response H'06 (1 byte): Response to the new bit-rate selection command

  The ACK code is returned when selection is possible.

Error response

| H'BF | ERROR |
|---|---|

- Error response H'BF (1 byte): Error response to a new bit-rate selection command
- ERROR (1 byte): Error code

  H'11: Checksum error

  H'24: Bit-rate selection error indicating that the specified bit rate is not selectable.

  H'25: Input frequency error indicating that the specified input frequency is not within the range from the minimum to maximum values.

  H'26: Frequency division ratio error indicating disagreement of frequency division ratios

  H'27: Operating frequency error indicating that the specified operating frequency is not within the range from the minimum to maximum values.

### (4) Checking Received Data

The following describes how the received data is checked.

1. Input frequency

   Clock source checking: The value of bit 15 indicates a clock source (the external clock or on-chip oscillator).

   Frequency checking: The value of the received input frequency (bits 14 to 0) is checked to see if it is within the range from minimum to maximum values of the input frequency of the selected clock mode of the selected device. If not, an input frequency error is generated.
2. Frequency division ratio

   The value of the received frequency division ratio is checked to see if it corresponds to the frequency division ratio value of the selected clock mode of the selected device. If not, a frequency division ratio error is generated.

chanhvinh.com



3. Operating frequency

   The operating frequency is calculated from the received input frequency and frequency division ratio. The input frequency is the frequency of the clock signal supplied to the LSI, whereas the operating frequency is the frequency at which the LSI actually operates. The following formula is used for the calculation.

   Operating frequency = input frequency/frequency division ratio

   The obtained operating frequency is checked to see if it is within the range from the minimum to maximum values of the operating frequency of the selected clock mode of the selected device. If not, an operating frequency error is generated.

4. Bit rate

   From the peripheral operating frequency (ϕ) and bit rate (B), the value (n) of the clock select bits (CKS) in the serial mode register (SMR) and the value (N) in the bit rate register (BRR) are calculated to determine the error. The error determined is checked to see if it is smaller than 4%. If not, a bit-rate selection error is generated. The following formula is used for the calculation.

$$\text{Error (\%)} = \left\{ \frac{\phi \times 10^6}{(N+1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

When selection of the new bit rate is possible, the boot program returns an ACK code to the host and then makes the necessary register settings to select the new bit rate. The host then transmits an ACK code at the new bit rate and the boot program responds to it at the new bit rate.

| Acknowledge | H'06 |
|---|---|

- Acknowledge H'06 (1 byte): Acknowledgement of the new bit rate

| Response | H'06 |
|---|---|

- Response H'06 (1 byte): Response to acknowledgement of the new bit rate

Figure 7.9 shows the sequence of new bit-rate selection.

chanhvinh.com



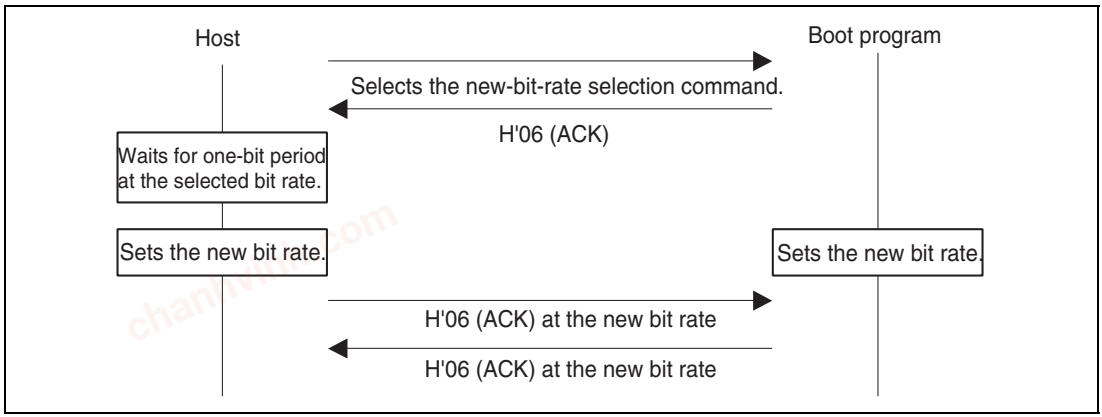



**Figure 7.9 Sequence of New Bit-Rate Selection**

### (5) Programming/Erasure State Transition (Without ID Checking)

In response to a programming/erasure state transition command, the boot program transfers the erasure program to erase the data in the user ROM area. On completion of this erasure, the boot program returns the ACK code and enters the programming/erasure state.

Before transmitting the programming selection command and data for programming, the host should select the device, clock mode, and new bit rate for this LSI using the device selection, clock-mode selection, and new-bit-rate selection commands; and then transmit a programming/erasure state transition command to the boot program.

| Command | H'40 |
|---|---|

- Command H'40 (1 byte): Programming/erasure state transition

| Response | H'06 |
|---|---|

- Response H'06 (1 byte): Response to the programming/erasure state transition command (without ID checking)

  The ACK code is returned when the user ROM area have been successfully erased after transfer of the erasure program.

Error response

| H'C0 | H'51 |
|---|---|

chanhvinh.com



- Error response H'C0 (1 byte): Error response to the programming/erasure state transition command.
- Error code H'51 (1 byte): Erasure error indicating that erasure was unsuccessful because of an error.

### (6) Programming/Erasure State Transition (With ID Checking)

In response to a programming/erasure state transition command, the boot program checks the ID code if the location of user ROM indicated in table 7.6 contains a control code (H'52 or H'45).

| Command | H'40 |
|---|---|

- Command H'40 (1 byte): Programming/erasure state transition

| Response | H'16 |
|---|---|

- Response H'16 (1 byte): Response to a programming/erasure state transition command (with ID checking).

**Table 7.6 Address of the Protection Code for the User ROM Area**

| | H'000004 | H'000005 | H'000006 | H'000007 | H'000010 | H'000011 | H'000012 | H'000013 |
|---|---|---|---|---|---|---|---|---|
| Boot mode | Control code | Authentication ID (56 bits) | | | | | | |

### (7) ID Checking

After the response to the command for the transition with ID checking, the command for ID checking and ID code against which the actual ID code is to be checked are sent from the host.

| Command | H'60 | Size | ID (16 bytes) | SUM |
|---|---|---|---|---|

- Command H'60 (1 byte): ID Checking
- Size (1 byte): The number of bytes of transmitted data, excluding the command, size, and checksum fields (fixed to H'10).

chanhvinh.com



- ID (16 bytes): The ID value is in the eight lower-order bytes. The value of the eight higher-order bytes is H'FF.

  For example, when the ID is the eight-byte value H'55112233, H'44556677, the value H'FFFFFFFF, H'FFFFFFFF, H'55112233, H'44556677 will match the ID.

  When the ID is the six-byte value H'55112233, H'4455, the value H'FFFFFFFF, H'FFFFFFFF, H'55112233, H'4455FFFF will match the ID.
- SUM (1 byte): Checksum

| Response | H'06 |
|---|---|

- Response H'06 (1 byte): Response to an ID checking command (with ID checking)

  The ACK code is returned if ID checking is successful.

Error response

| H'E0 | H'xx |
|---|---|

- Error response H'E0 (1 byte): Error response to an ID checking command
- ERROR (1 byte): Error code

  H'11: Checksum error

  H'61: ID code mismatch error

  H'63: Erasure error when ID code mismatch occurs.

### (8) Checksum Errors

When the host detects an abnormal checksum value, the host is capable of resending the same command if this is judged to be appropriate. The boot program does not have a function for resending.

When the boot program determines that the value of a sum check was incorrect, it returns a checksum error.

Error response

| H'xx | H'11 |
|---|---|

- Response H'xx (1 byte): Error response is the same as the command byte except that bit 7 is set to 1.

  For example, the error response corresponding to the device selection command H'10 is H'90.
- Error code H'11 (1 byte): Checksum error

chanhvinh.com



### (9) Command Errors

Command errors are caused by undefined commands, incorrect command sequence, and unacceptable commands. For example, sending a clock-mode selection command before a device selection command and sending an inquiry command after a programming/erasure state transition command both cause command errors.

Error response

| | |
|---|---|
| H'80 | H'xx |

- Error response H'80 (1 byte): Command error
- Command H'xx (1 byte): Received command

### (10) Order of Commands

In the inquiry/selection state, commands should be sent in the following order.

1. Send the supported-device inquiry command (H'20) to get the list of supported devices.
2. Select a device according to the returned device information, and send the device selection command (H'10).
3. Send the clock-mode inquiry command (H'21) to inquire about clock modes.
4. Select a clock mode from among the returned clock modes, and send the clock-mode selection command (H'11).
5. After selection of the device and clock mode, send the frequency-division-ratio inquiry command (H'22) and operating-frequency inquiry command (H'23) to get the information necessary for selecting a new bit rate.
6. Select a new bit rate according to the returned information on the frequency division ratios and operating frequencies, send the new bit-rate selection command (H'3F).
7. After selection of the new bit rate, send the programmable ROM information inquiry command (H'25), erasure-block-information inquiry command (H'26), and programming-size inquiry command (H'27) to get the information necessary for programming/erasing the user ROM area.
8. After each inquiry of step [7], send the programming/erasure state transition command (H'40) to cause a transition to the programming/erasure state.

chanhvinh.com



### (11) Programming/Erasure State

In the programming/erasure state, the boot program selects the form of programming in response to the programming selection command and then writes the data in response to the 128-byte programming command; or the boot program erases the desired blocks in response to the erasure selection and block erasure commands. Table 7.7 lists the programming/erasure commands.

**Table 7.7 Programming/Erasure Commands**

| Command | Command Name | Function |
|---|---|---|
| H'43 | User-ROM-area programming selection | Transfers the control program for user-ROM area programming. |
| H'50 | 128-byte programming | Executes 128-byte programming. |
| H'48 | Erasure selection | Transfers the erasure-control program. |
| H'58 | Block erasure | Erases the block data. |
| H'52 | Memory read | Reads data from memory. |
| H'4B | Programmable ROM sum check | Executes sum checking of the programmable ROM area. |
| H'4D | Programmable ROM blank check | Executes blank checking of the programmable ROM area. |
| H'4F | Boot-program state inquiry | Obtains about the processing state of the boot program. |
| H'61 | Data flash sum check | Executes sum checking for the data flash area |
| H'62 | Data flash blank check | Executes blank checking for the data flash area |
| H'71 | Lock-bit state read | Reads the state of a lock bit. |
| H'77 | Lock-bit program | Executes lock-bit programming. |
| H'75 | Disabling lock-bit | Disables the function of lock bits. |
| H'7A | Enabling lock-bit | Enables the function of lock bits. |

1. Programming

   Programming is performed by using the programming selection command and the 128-byte programming command.

   First, the host sends the programming selection command, and selects the form and area of programming.

chanhvinh.com



Next, the host sends the 128-byte programming command. The boot program assumes that the 128 bytes of data included in the 128-byte programming command should be programmed according to the form of programming selected by the preceding programming selection command. To program more than 128 bytes, repeatedly send 128-byte programming commands. To terminate programming, the host should send the 128-byte programming command with address H'FFFFFFFF. On completion of programming, the boot program waits for the next programming/erasure selection command.

The sequence of programming by the programming selection command and 128-byte programming command is shown in figure 7.10.

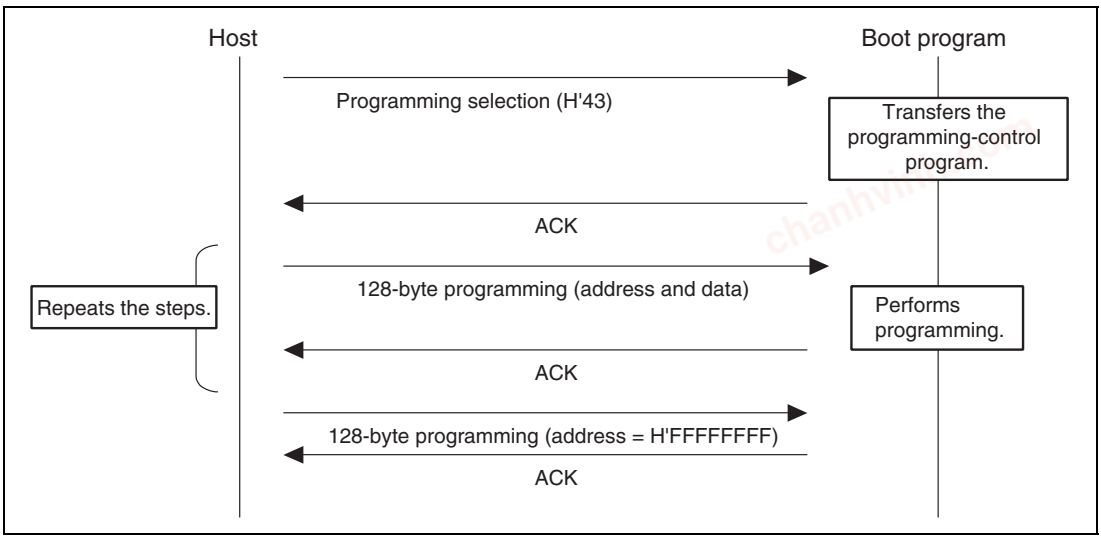



**Figure 7.10 Programming Sequence**

2. Erasure

Erasure is performed by using the erasure selection command and the block erasure command.

First, select erasure by the erasure selection command and then actually erase a specific block using the block erasure command. To erase multiple blocks, repeatedly send block erasure commands. To terminate erasure, the host should send the block erasure command with block number H'FF. On completion of erasure, the boot program waits for the next programming/erasure selection command.

The sequence of erasure by the erasure selection command and block erasure command is shown in figure 7.11.

chanhvinh.com



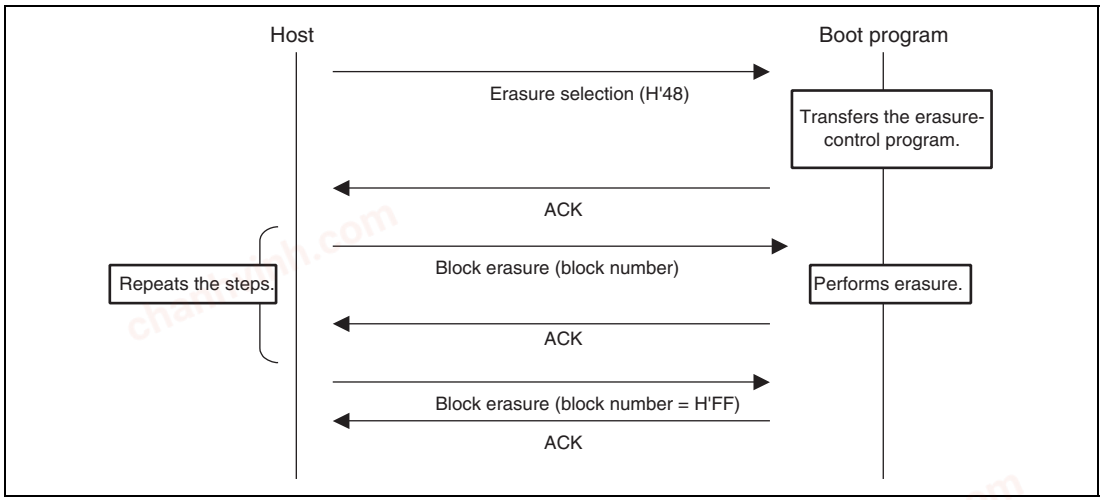



**Figure 7.11 Erasure Sequence**

### (a) User-ROM-Area Programming Selection

In response to a user-ROM-area programming selection command, the boot program transfers the relevant programming-control program according to which the user ROM area is programmed.

| Command | H'43 |
|---|---|

- Command H'43 (1 byte): User-ROM-area programming selection

| Response | H'06 |
|---|---|

- Response H'06 (1 byte): Response to a user-ROM-area programming selection command.
  The ACK code is returned upon completion of transferring the programming-control program.

Error response

| H'C3 | ERROR |
|---|---|

- Error response H'C3 (1 byte): Error response to a user-ROM-area programming selection command
- ERROR (1 byte): Error code
  H'54: Selection processing error (processing was not completed because of a transfer error)

chanhvinh.com



### (b) 128-Byte Programming

In response to a 128-byte programming command, the boot program programs the user ROM area according to the programming-control program transferred in response to the user-ROM-area programming selection command.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Command | H'50 | Address | | | | | | |
| | Data | … | | | | | | |
| | … | | | | | | | |
| | SUM | | | | | | | |

- Command H'50 (1 byte): 128-byte programming
- Address for programming (4 bytes): Address at which programming starts
  The address should be the multiple of the size returned in response to the programming-size inquiry command.
  [Example] H'00, H'01, H'00, H'00: H'00010000
- Programming data (128 bytes): Data for programming
  The size of the programming data is the size returned in response to the programming-size inquiry command.
- SUM (1 byte): Checksum

| | |
|---|---|
| Response | H'06 |

- Response H'06 (1 byte): Response to a 128-byte programming command.
  The ACK code is returned upon completion of the requested programming.

Error response

| | |
|---|---|
| H'D0 | ERROR |

- Error response H'D0 (1 byte): Error response to a 128-byte programming command
- ERROR (1 byte): Error code
  H'11: Checksum error
  H'2A: Address error. This error indicates that the address is not within the specified range of areas.
  H'53: Programming error (programming failed because of an error in programming)

chanhvinh.com



The specified address should be on a boundary corresponding to the unit of programming (programming size). For example, when the programming size is 128 bytes, the lower 8 bits of the address should be either H'00 or H'80. When less than 128 bytes of data are to be programmed, the host should transmit the data after padding the vacant bytes with H'FF.

To terminate programming, send the 128-byte programming command with H'FFFFFFFF in the address-for-programming field. This informs the boot program that the data has been completely sent; the boot program then waits for the next programming/erasure selection command.

| Command | H'50 | Address | SUM |
|---|---|---|---|

- Command H'50 (1 byte): 128-byte programming
- Address for programming (4 bytes): Terminating code (H'FF, H'FF, H'FF, H'FF)
- SUM (1 byte): Checksum

| Response | H'06 |
|---|---|

- Response H'06 (1 byte): Response to a 128-byte programming command.
  The ACK code is returned upon termination of the programming process.

Error response

| H'D0 | ERROR |
|---|---|

- Error response H'D0 (1 byte): Error response to a 128-byte programming command
- ERROR (1 byte): Error code
  H'11: Checksum error
  H'53: Programming error (programming failed because of an error in programming)

### (c) Erasure Selection

In response to an erasure selection command, the boot program transfers the relevant erasure-control program. The data in the user ROM area is erased using the transferred erasure-control program.

| Command | H'48 |
|---|---|

- Command H'48 (1 byte): Erasure selection

| Response | H'06 |
|---|---|

chanhvinh.com



- Response H'06 (1 byte): Response to an erasure selection command.
  The ACK code is returned upon completion of transferring the erasure-control program.

Error response

| | |
|---|---|
| H'C8 | ERROR |

- Error response H'C8 (1 byte): Error response to an erasure selection command
- ERROR (1 byte): Error code
  H'54: Selection processing error (processing was not completed because of a transfer error)

### (d) Block Erasure

In response to a block erasure command, the boot program erases the data in the specified block.

| Command | H'58 | Size | Block number | SUM |
|---|---|---|---|---|

- Command H'58 (1 byte): Block erasure
- Size (1 byte): The number of characters in the block-number field (fixed to 1)
- Block number (1 byte): The number specific to the block to be erased
- SUM (1 byte): Checksum

| Response | H'06 |
|---|---|

- Response H'06 (1 byte): Response to a block erasure command.
  The ACK code is returned when the specified block has been erased.

Error response

| | |
|---|---|
| H'D8 | ERROR |

- Error response H'D8 (1 byte): Error response to a block erasure command
- ERROR (1 byte): Error code
  H'11: Checksum error
  H'29: Block number error (the specified block number is incorrect)
  H'51: Erasure error (an error occurred during erasure)

chanhvinh.com



On receiving the command with H'FF as the block number, the boot program terminates erasure processing and waits for the next programming/erasure selection command.

| | | | | |
|---|---|---|---|---|
| Command | H'58 | Size | Block number | SUM |

- Command H'58 (1 byte): Block erasure
- Size (1 byte): The number of characters in the block number field (fixed to 1)
- Block number (1 byte): H'FF (erasure terminating code)
- SUM (1 byte): Checksum

| | |
|---|---|
| Response | H'06 |

- Response H'06 (1 byte): Response to the block erasure command for terminating erasure processing; ACK code is returned upon termination of the erasure process.

To perform erasure again after issuing the command with H'FF as the block number, start the process by sending an erasure selection command.

**(e) Memory Read**

In response to a memory read command, the boot program returns the data stored in the specified address.

| | | | | | |
|---|---|---|---|---|---|
| Command | H'52 | Size | Area | Address for reading | |
| | Reading size | | | | SUM |

- Command H'52 (1 byte): Memory read
- Size (1 byte): The total size of the area, address-for-reading, and reading-size fields (fixed to 9)
- Area (1 byte):
  H'01: User ROM area
  Specifying an incorrect area causes an address error.
- Address for reading (4 bytes): Address where reading starts
- Reading size (4 bytes): The amount of data to be read
- SUM (1 byte): Checksum

chanhvinh.com



| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Response | H'52 | Reading size | | | | | | |
| | Data | … | | | | | | |
| | SUM | | | | | | | |

- Response H'52 (1 byte): Response to a memory read command
- Reading size (4 bytes): The amount of data to be read
- Data (128 bytes): The specified amount of data to be read out starting at the specified address
- SUM (1 byte): Checksum

Error response

| | |
|---|---|
| H'D2 | ERROR |

- Error response H'D2 (1 byte): Error response to a memory read command
- ERROR (1 byte): Error code
  H'11: Checksum error
  H'2A: Address error (the specified address for reading is not in the MAT)
  H'2B: Size error (the specified size (amount) is greater than the size of the MAT)

### (f) Programmable ROM Sum Check

In response to a programmable ROM sum check command, the boot program adds all the data bytes in the user ROM area and returns the result.

| | |
|---|---|
| Command | H'4B |

- Command H'4B (1 byte): Programmable ROM sum check

| | | | | |
|---|---|---|---|---|
| Response | H'5B | Size | Checksum for the MAT | SUM |

- Response H'5B (1 byte): Response to a programmable ROM sum check command
- Size (1 byte): The number of characters in the checksum-for-the-MAT field (fixed to 4)
- Checksum for the MAT (4 bytes): Result of checksum calculation for the programmable ROM area; the total of all the data in the MAT, in byte units.
- SUM (1 byte): Checksum (for this response)

chanhvinh.com



### (g) Programmable ROM Blank Check

In response to a programmable ROM blank check command, the boot program checks to see if the whole the programmable ROM area is blank and returns the result.

| Command | H'4D |
|---|---|

- Command H'4D (1 byte): Programmable ROM blank check

| Response | H'06 |
|---|---|

- Response H'06 (1 byte): Response to a programmable ROM blank check command
  The ACK code is returned when the whole area is blank (all bytes are H'FF).

Error response

| H'CD | H'52 |
|---|---|

- Error response H'CD (1 byte): Error response to a programmable ROM blank check command
- Error code H'52 (1 byte): Non-erased error

### (h) Data Flash Sum Check

In response to a data flash sum check command, the boot program obtains the sum of all bytes in the data flash areas and returns the result.

| Command | H'61 |
|---|---|

- Command H'61 (1 byte): Data flash sum check

| Response | H'71 | Size | Checksum | SUM |
|---|---|---|---|---|

- Response H'71 (1 byte): Response to a data flash sum check command
- Size (1 byte): The number of characters in the checksum field (fixed to 4)
- Checksum (4 bytes): Result of checksum calculation for the data flash areas; the byte-wise sum of all data in the areas.
- SUM (1 byte): Checksum value that makes the sum of the bytes from the command to the SUM byte become H'00.

chanhvinh.com



### (i) Data Flash Blank Check

In response to a data flash blank check command, the boot program checks to see if the whole data flash areas are blank and returns an indicator of the result.

| Command | H'62 |
|---|---|

- Command H'62 (1 byte): Data flash blank check

| Response | H'06 |
|---|---|

- Response H'06 (1 byte): Response to a data flash blank check command.
  The ACK code is returned when the whole areas are blank (all bytes are H'FF).

Error response

| H'E2 | H'52 |
|---|---|

- Error response H'E2 (1 byte): Error response to a data flash blank check command
- Error code H'52 (1 byte): Non-erased error

### (j) Lock-Bit State Read

The lock bit for an area of user ROM (other than the data flash area) is read, and the result is returned.

| Command | H'71 | Size | Area | Medium address | Upper address | SUM |
|---|---|---|---|---|---|---|

- Command H'71 (1 byte): Lock-bit state read
- Size (1 byte): The total size of the area, medium address, and upper address fields (fixed to 3).
- Area (1 byte): H'01: User ROM area
- Medium address (1 byte): Middle-order bits (bits 8 to 15) of the address where the block ends
- Upper address (1 byte): Higher-order bits (bits 16 to 23) of the address where the block ends
- SUM (1 byte): Checksum value such that the sum of all bytes from the command to the SUM field becomes H'00.

| Response | STATUS |
|---|---|

chanhvinh.com



- STATUS (1 byte): The value 0 for bit 6 indicates the locked state.
- STATUS (1 byte): The value 1 for bit 6 indicates the unlocked state.

Error response

| | |
|---|---|
| H'F1 | ERROR |

- Error response H'F1 (1 byte): Error response to a lock-bit state read command
- ERROR (1 byte): Error code
  H'11: Checksum error
  H'2A: Address error. This error indicates that the specified block address is incorrect.

### (k) Lock-Bit Program

This command locks a specified block (other than the data flash area).

| Command | H'77 | Size | Area | Medium address | Upper address | SUM |
|---|---|---|---|---|---|---|

- Command H'77 (1 byte): Lock-bit program
- Size (1 byte): The total size of the area, medium address, and upper address fields (fixed to 3).
- Area (1 byte): H'01: User ROM area
- Medium address (1 byte): Middle-order bits (bits 8 to 15) of the address where the block ends
- Upper address (1 byte): Higher-order bits (bits 16 to 23) of the address where the block ends
- SUM (1 byte): Checksum

| Response | H'06 |
|---|---|

- Response H'06 (1 byte): Response to a lock-bit program command (ACK code)

Error response

| | |
|---|---|
| H'F7 | ERROR |

- Error response H'F7 (1 byte): Error response to an lock-bit program command
- ERROR (1 byte): Error code
  H'11: Checksum error
  H'2A: Address error
  H'53: Programming error. This error indicates that an error occurred in programming of the lock bit.

chanhvinh.com



### (l) Enabling Lock Bit

This command enables the function of lock bits.

| Command | H'7A |
|---|---|

- Command H'7A (1 byte): Enabling lock bit

| Response | H'06 |
|---|---|

- Response H'06 (1 byte): Response to an enabling-lock-bit command (ACK code)

### (m) Disabling Lock Bit

This command disables the function of lock bits.

| Command | H'75 |
|---|---|

- Command H'75 (1 byte): Disabling lock bit

| Response | H'06 |
|---|---|

- Response H'06 (1 byte): Response to a disabling-lock-bit command (ACK code)

### (n) Boot-Program State Inquiry

In response to a boot-program state inquiry command, the boot program returns its current state and error information. This inquiry can be made either in the inquiry/selection state or the programming/erasure state.

| Command | H'4F |
|---|---|

- Command H'4F (1 byte): Boot-program state inquiry

| Response | H'5F | Size | STATUS | ERROR | SUM |
|---|---|---|---|---|---|

- Response H'5F (1 byte): Response to a boot-program state inquiry command
- Size (1 byte): The number of characters in the STATUS and ERROR fields (fixed to 2)
- STATUS (1 byte): State of the boot program

chanhvinh.com



- ERROR (1 byte): Error information
  ERROR = 0: Success
  ERROR ≠ 0: Error
- SUM (1 byte): Checksum

**Table 7.8 State Codes**

| Code | Description |
|---|---|
| H'11 | Waiting for device selection |
| H'12 | Waiting for clock mode selection |
| H'13 | Waiting for bit rate selection |
| H'1F | Waiting for transition to programming/erasure state (bit rate selection completed) |
| H'31 | Programming or erasure state (programming/erasure in progress) |
| H'3F | Waiting for programming/erasure selection (erasure completed) |
| H'4F | Waiting to receive data for programming (programming completed) |
| H'5F | Waiting for erasure block specification (erasure completed) |

chanhvinh.com



**Table 7.9 Error Codes**

| Code | Description |
|---|---|
| H'00 | No error |
| H'11 | Checksum error |
| H'12 | Programming size error |
| H'21 | Device-code disagreement error |
| H'22 | Clock-mode disagreement error |
| H'24 | Bit-rate selection disable error |
| H'25 | Input frequency error |
| H'26 | Frequency division ratio error |
| H'27 | Operating frequency error |
| H'29 | Block number error |
| H'2A | Address error |
| H'2B | Data size error |
| H'51 | Erasure error |
| H'52 | Non-erased error |
| H'53 | Programming error |
| H'54 | Selection processing error |
| H'61 | ID code mismatch error |
| H'63 | Erasure error when ID code mismatch occurs |
| H'80 | Command error |
| H'FF | Bit-rate-adjustment acknowledge error |

### 7.5.3 Programming/Erasing in User Mode

On-board programming/erasing of individual flash memory blocks is also possible in user mode by branching to the user programming/erasure-control program. The user must set the branching conditions and provide the on-board means of supplying the programming data. The flash memory must contain the user programming/ erasure-control program or a program that allows the user programming/erasure-control program to be supplied externally. As the flash memory itself cannot be read during programming/erasing, transfer the user programming/erasure-control program to the on-chip RAM to execute, as in boot mode. Figure 7.12 shows a sample procedure for programming/erasing in user mode. Prepare user programming/erasure-control program in accordance with the description in section 7.6, Programming/Erasing.

chanhvinh.com



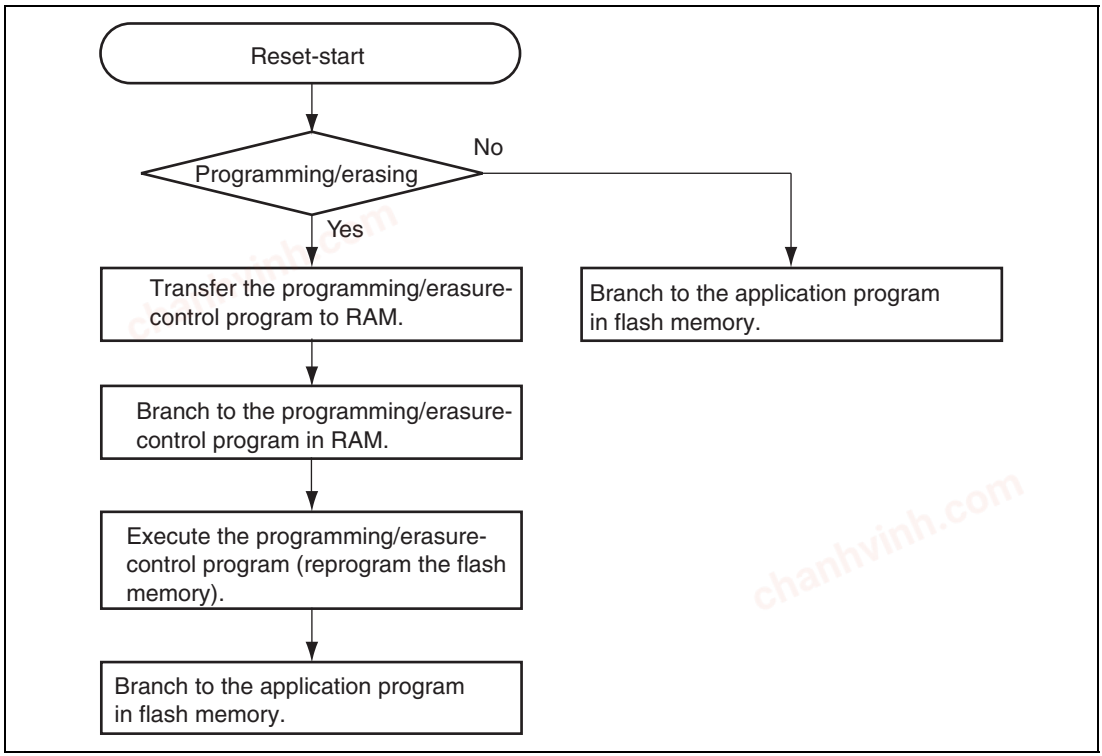



**Figure 7.12 Sample Programming/Erasing Procedure in User Mode (EW0 Mode)**

chanhvinh.com



## 7.6 Programming/Erasing

The CPU reprogramming method is employed to program and erase flash memory on board, in which the CPU executes software commands.

### 7.6.1 Software Commands

Table 7.10 shows a list of software commands through word instructions and table 7.11 shows a list of software commands through byte instructions. Whether an instruction is to be byte-length or word-length is specified by the FMWUS bit in FLMCR1.

**Table 7.10 Software Commands (in Word Instructions: FMWUS = 1)**

| Software Command | First Command Cycle | | | Second Command Cycle | | | Third Command Cycle | | | Command Use in Modes | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Mode | Addr. | Data | Mode | Addr. | Data | Mode | Addr. | Data | EW0 | EW1 |
| Erasure | Write | × | H'2020 | Write | BA | H'D0D0 | | | | Possible | Possible |
| Programming | Write | WA | H'4141 | Write | WA | WD1 | Write | WA | WD2 | Possible | Possible |
| Blank checking | Write | × | H'2525 | Write | BA | H'D0D0 | | | | Possible | Possible |
| Lock-bit program | Write | × | H'7777 | Write | BA | H'D0D0 | | | | Possible | Possible |
| Read-array | Write | × | H'FFFF | | | | | | | Possible | — |
| Clear-status | Write | × | H'5050 | | | | | | | Possible | Possible |
| Lock-bit reading | Write | × | H'7171 | Read | BA | H'xxxx | | | | Possible | Impossible |

[Legend]

×: Arbitrary address in the user ROM area

xx: Eight-bit arbitrary data

BA: Arbitrary address in a block

WA: Programming address. (The lower two bits of an address are ignored. WA should be the same in each command cycle.)

WDn: Programming data (16 bits)

chanhvinh.com



**Table 7.11 Software Commands (in Byte Instructions: FMWUS = 0)**

| Software Command | First Command Cycle | | | Second Command Cycle | | | Third Command to Fifth Command Cycle | | | Command Use in Modes | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Mode | Addr. | Data | Mode | Addr. | Data | Mode | Addr. | Data | EW0 | EW1 |
| Erasure | Write | × | H'20 | Write | BA | H'D0 | | | | Possible | Possible |
| Programming | Write | WA | H'41 | Write | WA | WD1 | Write | WA | WD2 to WD4 | Possible | Possible |
| Blank checking | Write | × | H'25 | Write | BA | H'D0 | | | | Possible | Possible |
| Lock-bit program | Write | × | H'77 | Write | BA | H'D0 | | | | Possible | Possible |
| Read-array | Write | × | H'FF | | | | | | | Possible | — |
| Clear-status | Write | × | H'50 | | | | | | | Possible | Possible |
| Lock-bit reading | Write | × | H'71 | Read | BA | H'xx | | | | Possible | Impossible |

[Legend]

×: Arbitrary address in the user ROM area

xx: Eight-bit arbitrary data

BA: Arbitrary address in a block

WA: Programming address. (The lower two bits of an address are ignored. WA should be the same in each command cycle.)

WDn: Programming data (8 bits)

### (1) Initialization for CPU Reprogramming Mode

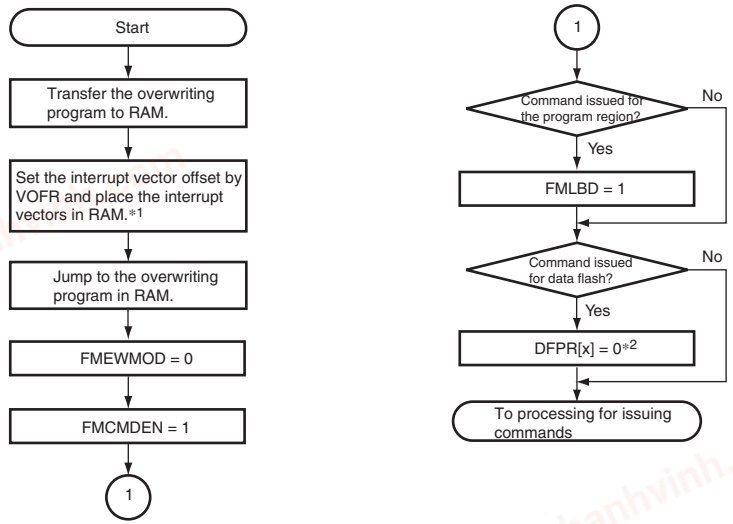
Before software commands are issued, settings for CPU reprogramming mode must be made and issuing of software commands must be permitted.

Figure 7.13 shows initialization for CPU reprogramming mode.

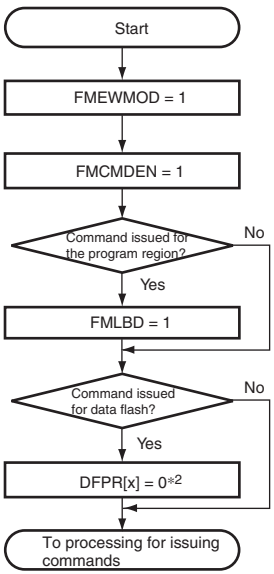
chanhvinh.com



Flow of initialization for EW0 mode*1




Flow of initialization for EW1 mode*1




Notes: For any interrupts that are in use, allocate the interrupt vector entries and interrupt routines to RAM.
If interrupts are not to be used, allocation to RAM is not necessary.
1. Within the flow, set the CPU overwriting unit selection bit (FMWUS) to select the unit of overwriting.
2. Set the DFPR according to the area of data-flash memory for which commands are to be issued.

**Figure 7.13 Initialization for E/W Mode**

chanhvinh.com



### (2) Erasure

When H'20 is written in the first command cycle and H'D0 is written to any address in the block in the second command cycle, erase/erase-verify of the specified block is automatically started.

Completion of erasure is indicated by the FMRDY bit in FLMSTR. The FMRDY bit is read as 0 during erasure, and read as 1 after erasure completion.

After erasure completion, the erasure result can be checked by reading the FMEBSF bit in FLMSTR. (See the description in (9) below, Full Status Checking.)

Note that if the lock bit is 0 (locked) in the specified block and the FMLBD bit is 0 (lock bit enabled), an erasure command is not accepted for the specified block.

Figures 7.14 and 7.15 show the flowcharts when the erase-suspend function is not used and when used, respectively.

When the erase-suspend function is being employed and erasure is resumed immediately after a period in erase-suspend mode, instruction fetching with normal incrimination of the program counter will not be possible. To avoid this problem, add three NOP instructions immediately after the instruction that writes FMSPREQ = 0. Furthermore, do not use the DTC when erasure has been suspended in EW1 mode and the reprogramming control program has been allocated to RAM.

In EW1 mode, do not execute this command for the block in which the reprogramming-control program is located.

The FMRDY bit in FLMSTR changes to 0 when erasure is started, and changes to 1 when completed.

chanhvinh.com



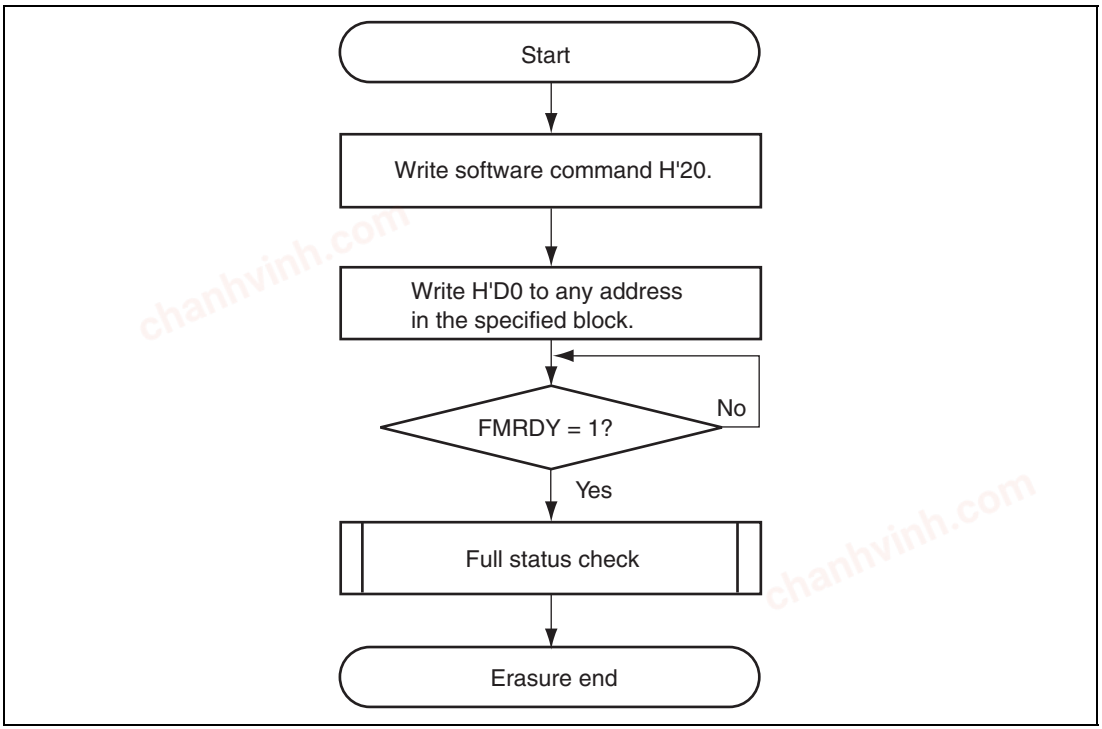
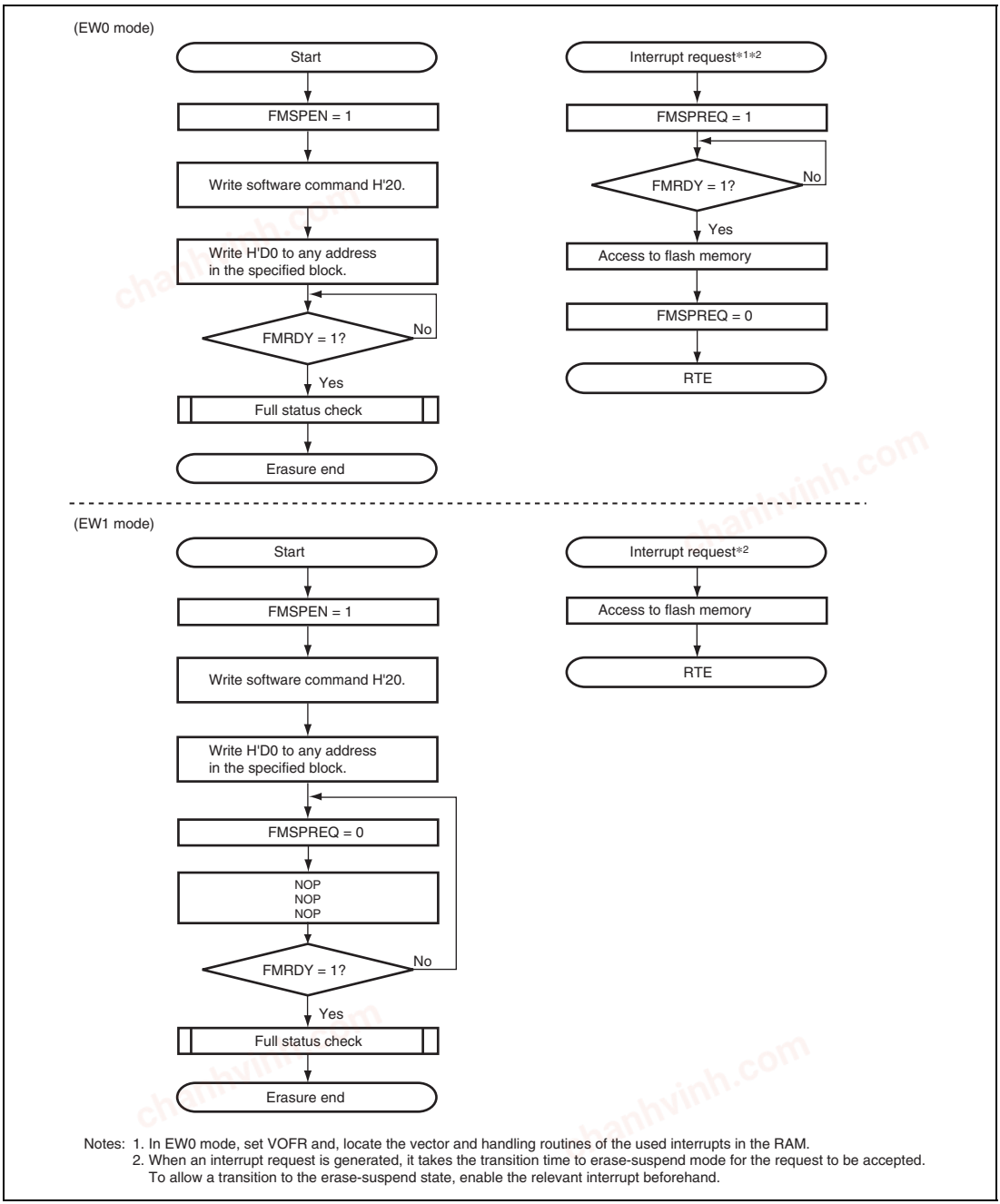



**Figure 7.14 Flowchart When Erase-Suspend Function is Not Used**

chanhvinh.com






Notes: 1. In EW0 mode, set VOFR and, locate the vector and handling routines of the used interrupts in the RAM.
2. When an interrupt request is generated, it takes the transition time to erase-suspend mode for the request to be accepted. To allow a transition to the erase-suspend state, enable the relevant interrupt beforehand.

**Figure 7.15 Flowchart When Erase-Suspend Function is Used**

chanhvinh.com



### (3) Programming

A programming command is used to program data in the flash memory in 4-byte units.

Command or data size can be set depending on the FMWUS bit in FLMCR1. Setting the FMWUS bit to 0 enables using byte instructions. When H'41 is written in the first command cycle and data is written to the programming address in the second through fifth command cycles, programming and verifying are automatically started*.

Setting the FMWUS bit to 1 enables using word instructions. When H'4141 is written in the first command cycle and data is written to the programming address in the second and third command cycles, programming and verifying are started*.

Completion of programming is indicated by the FMRDY bit in FLMSTR. The FMRDY bit is read as 0 during programming, and read as 1 after programming completion.

After programming completion, the programming result can be checked by reading the FMPRSF bit in FLMSTR. (See the description in (9) below, Full Status Checking.)

Figure 7.16 shows the programming flowchart.

Do not additionally program the already-programmed addresses.

Note that if the lock bit is 0 (locked) in the specified block and the FMLBD bit is 0 (lock bit enabled), a programming command is not accepted for the specified block.

In EW1 mode, do not execute this command for the block in which the reprogramming-control program is located.

The FMRDY bit in FLMSTR changes to 0 when programming is started, and changes to 1 when completed.

Note: * The lower two bits of the programming addresses are ignored.

chanhvinh.com



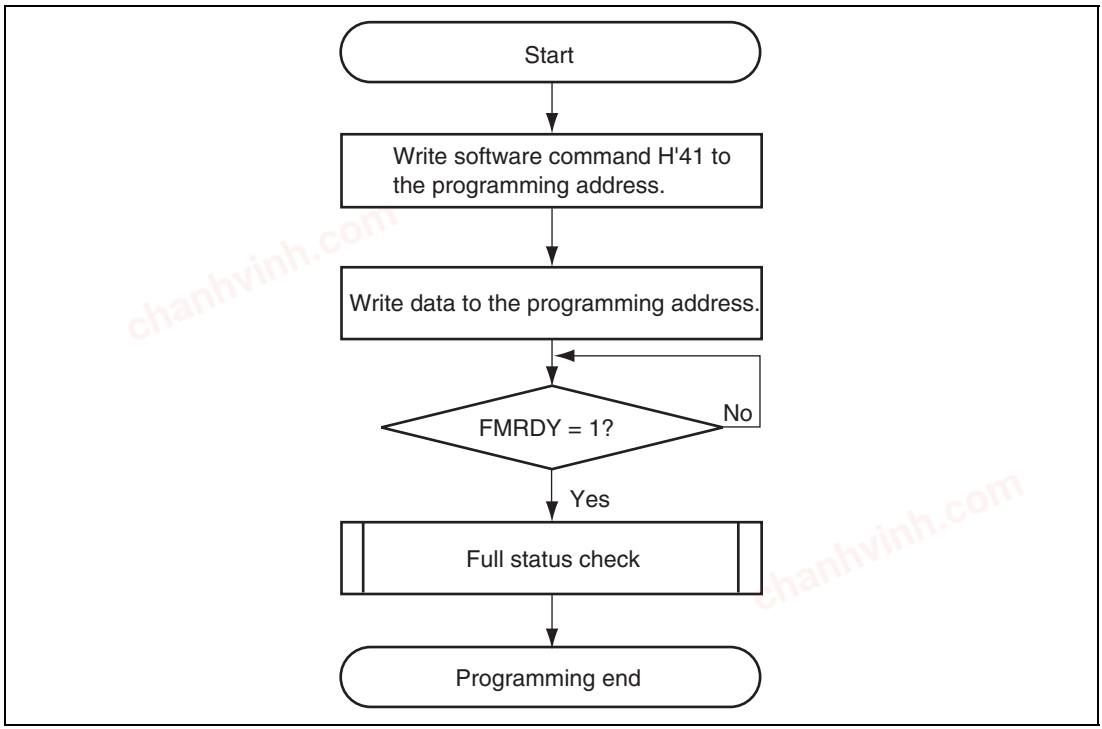



**Figure 7.16 Programming Flowchart**

chanhvinh.com



### (4) Blank Checking

When H'25 is written in the first command cycle and H'D0 is written to any address in the block in the second command cycle, blank checking of the specified block is started.

Completion of blank checking is indicated by the FMRDY bit in FLMSTR. The FMRDY bit is read as 0 during blank checking, and read as 1 after blank checking completion.

After blank checking completion, the blank checking result can be checked by reading the FMEBSF bit in FLMSTR. (See the description in (9) below, Full Status Checking.)

Figure 7.17 shows the blank checking flowchart.

The FMRDY bit in FLMSTR changes to 0 when blank checking is started, and changes to 1 when completed.

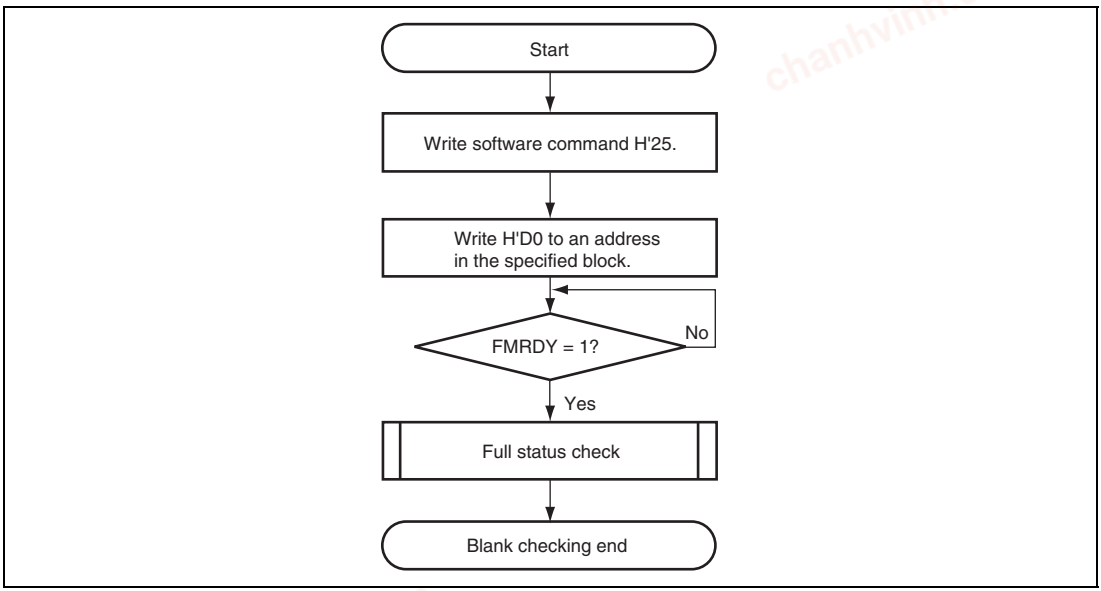



**Figure 7.17 Blank Checking Flowchart**

chanhvinh.com



### (5) Lock-Bit Program

When H'77 is written in the first command cycle and H'D0 is written to any address in the block in the second command cycle, lock-bit programming of the specified block is started.

Completion of lock-bit programming is indicated by the FMRDY bit in FLMSTR. The FMRDY bit is read as 0 during lock-bit programming, and read as 1 after lock-bit programming completion.

After lock-bit programming completion, the lock-bit programming result can be checked by reading the FMPRSF bit in FLMSTR. (See the description in (9) below, Full Status Checking.)

Figure 7.18 shows the lock-bit programming flowchart.

The FMRDY bit in FLMSTR changes to 0 when lock-bit programming is started, and changes to 1 when completed.

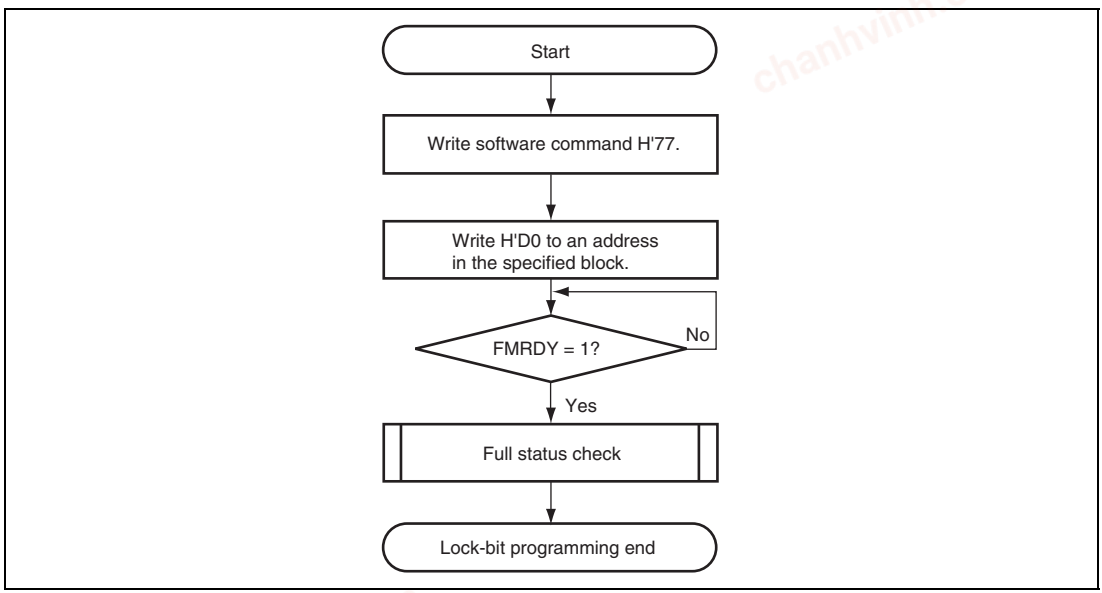



**Figure 7.18 Lock-Bit Programming Flowchart**

chanhvinh.com



### (6) Read-Array Command

A read-array command is to cause a transition to a mode in which data can be read from flash memory.

When H'FF is written in the first command cycle, a transition to read array mode is caused. When the specified addresses are read out in the subsequent command cycles, data is read from the specified addresses.

Since read-array mode is retained until any other command is written, multiple addresses can be read successively.

### (7) Lock-Bit Reading Command

This command is used to read the value of the lock bit in flash memory.

Writing H'71 in the first command cycle and reading from the specifying block address (BA) in the second command cycle returns the value of the lock bit. If a word instruction is used for reading, the value of the lock bit will be reflected in bits 6 and 14 of the read-out word. If a byte instruction is used, the value of the lock bit will be reflected in bit 6. Execute the lock-bit reading command in EW0 mode.

### (8) Status Clearing Command

A clear-status command is used to clear the status flag to 0.

When H'50 is written in the first command cycle, the FMPRSF and FMEBSF bits in FLMSTR are cleared to 0.

### (9) Full Status Checking

When any command (other than the read-array command, lock bit reading command and clear-status command) is issued, full-status checking is performed to confirm whether or not there was an error.

When an error occurs, the FMPRSF and FMEBSF bits in FLMSTR are set to 1, indicating the occurrence of the relevant errors.

Table 7.12 shows the bit values in FLMSTR and the corresponding errors. Figure 7.19 shows the full status checking flowchart and procedures of handling each error.

chanhvinh.com



**Table 7.12 Bit Values in FLMSTR and Corresponding Errors**

| Bit Values in FLMSTR | | | |
|---|---|---|---|
| **FMEBSF** | **FMPRSF** | **Error** | **Error Generation Conditions** |
| 0 | 0 | Successful end | |
| 0 | 1 | Programming error | The programming command is executed and results in unsuccessful programming. |
| | | Lock-bit programming error | The lock-bit program command is executed and results in unsuccessful programming. |
| 1 | 0 | Erasure error | The erasure command is executed and results in unsuccessful erasure. |
| | | Blank checking error | The blank checking command is executed and it is detected that the specified block is not blank. |
| 1 | 1 | Command sequence error | • A command is not written correctly.<br>• A data value other than H'D0 and H'FF is written in the last cycle of the command that consists of two command cycles.<br>• The erasure command is input in erase-suspend mode.<br>• The programming command is input for the suspended block in erase-suspend mode. |

chanhvinh.com



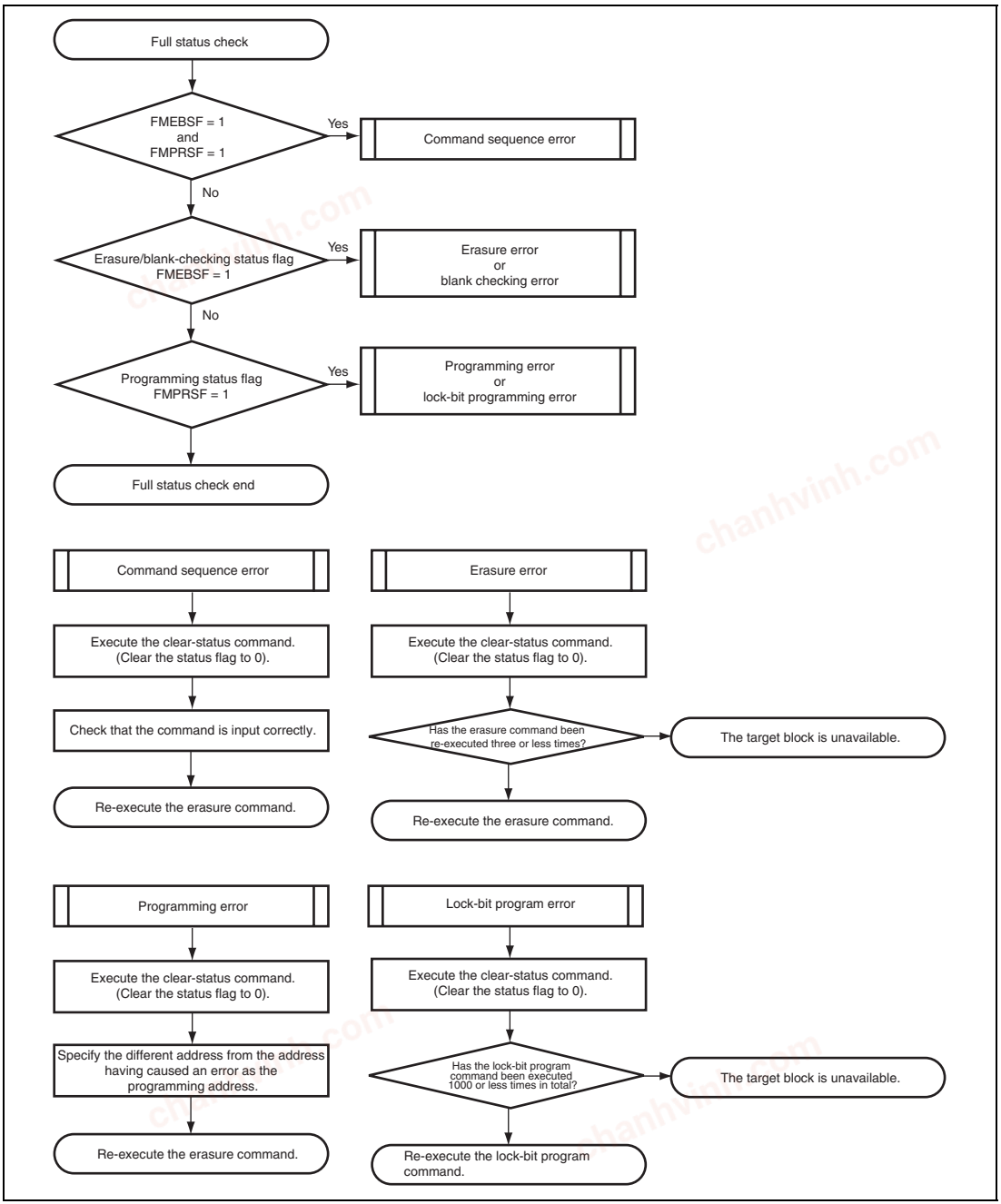



**Figure 7.19 Full Status Checking Flowchart and Procedures of Handling Errors**

chanhvinh.com



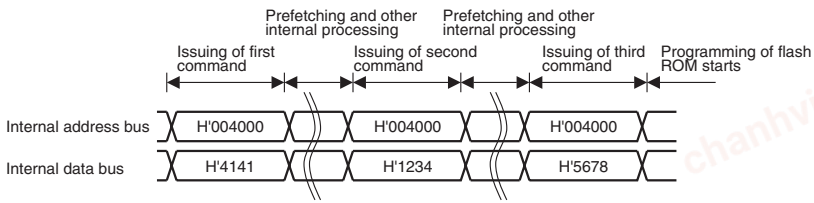
### (10) Example of Issuing Commands

Figures 7.20 and 7.21 show examples of issuing programming commands and erasure commands, respectively. Figure 7.22 shows examples of issuing read-array commands.

● Using word-length instructions to issue programming commands (FMWUS = 1)

| Target address for writing | Data |
|---|---|
| H'004000 | H'12 |
| H'004001 | H'34 |
| H'004002 | H'56 |
| H'004003 | H'78 |

```
[Programming Example]
@MOV.W  #H'4141,R0          ; Programming command
@MOV.W  #H'1234,R1          ; Writing of data
@MOV.W  #H'5678,R2          ; Writing of data
@MOV.W  R0, @H'00004000     ; First command
@MOV.W  R1, @H'00004000     ; Second command
@MOV.W  R2, @H'00004000     ; Third command
```




● Using byte-length instructions to issue programming commands (FMWUS = 0)

| Target address for writing | Data |
|---|---|
| H'004000 | H'12 |
| H'004001 | H'34 |
| H'004002 | H'56 |
| H'004003 | H'78 |

```
[Programming Example]
@MOV.B  #H'41, R0L          ; Programming command
@MOV.W  #H'1234,R1          ; Writing of data
@MOV.W  #H'5678,R2          ; Writing of data
@MOV.B  R0L, @H'00004000    ; First command
@MOV.B  R1H, @H'00004000    ; Second command
@MOV.B  R1L, @H'00004000    ; Third command
@MOV.B  R2H, @H'00004000    ; Fourth command
@MOV.B  R2L, @H'00004000    ; Fifth command
```

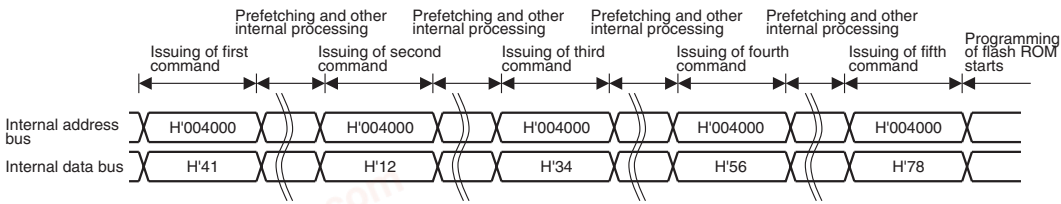



**Figure 7.20 Examples of Issuing Programming Commands**

chanhvinh.com

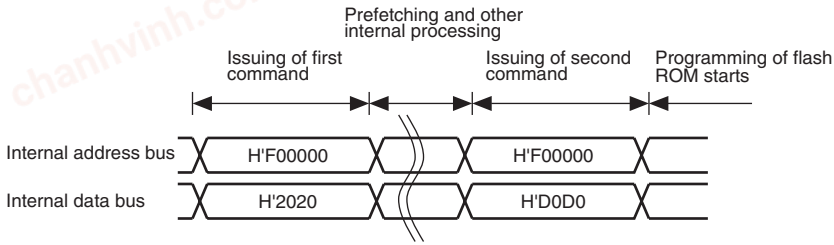


● Using word-length instructions to issue erasure commands (FMWUS = 1)

[Erasure Setting]
Erasure block = Data flash A

[Programming Example]

```
@MOV.W  #H'2020,R0          ; Erasure command
@MOV.W  #H'D0D0,R1          ; Erasure command
@MOV.W  R0,@H'00F00000      ; First command
@MOV.W  R1,@H'00F00000      ; Second command
```




● Using byte-length instructions to issue erasure commands (FMWUS = 0)

[Erasure Setting]
Erasure block = Data flash A

[Programming Example]

```
@MOV.B  #H'20, R0L          ; Erasure command
@MOV.B  #H'D0, R0H          ; Erasure command
@MOV.B  R0L,@H'00F00000     ; First command
@MOV.B  R0H,@H'00F00000     ; Second command
```

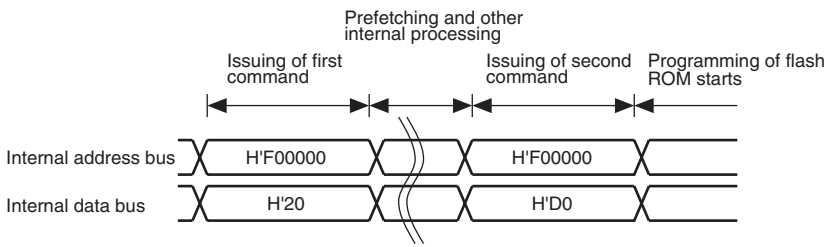



**Figure 7.21 Examples of Issuing Erasure Commands**

chanhvinh.com

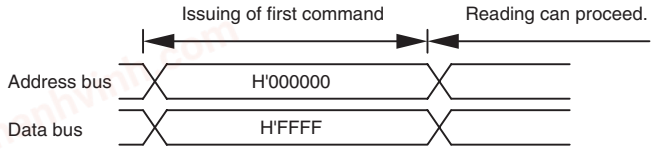


● Using word-length instructions to issue read-array commands (FMWUS = 1)

[Read-Array Setting]
Address = Program-ROM area

[Programming Example]

```
@MOV.W   #H'FFFF, R0        ; Read-array command
@MOV.W   R0, @H'00000000    ; First command
```




● Using byte-length instructions to issue read-array commands (FMWUS = 0)

[Read-Array Setting]
Address = Program-ROM area

[Programming Example]

```
@MOV.B   #H'FF, R0L         ; Read-array command
@MOV.B   R0L,@H'00000000    ; First command
```

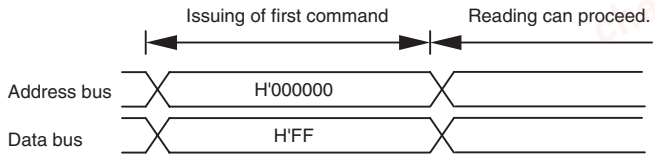



**Figure 7.22 Examples of Issuing Read-Array Commands**

chanhvinh.com



## 7.7 Protection

Three modes are available to protect the flash memory against reading, programming, and erasing: software protection, lock-bit protection, and protection to restrict access in programmer mode and boot mode.

### 7.7.1 Software Protection

Software commands can be disabled by clearing the FMCMDEN bit in the flash memory control register (FLMCR1) through software. In this state, no software commands are executed even if input.

Data flash areas can be protected in block units by using the flash memory data flash protect register (DFPR). Setting bits DFPR1 and DFPR0 in DFPR to 1 places all the data flash areas in protect mode.

### 7.7.2 Lock-Bit Protection

The programming/erasure commands can be disabled by programming the lock bits using the lock-bit program command. In this state, the erasure/programming commands are not executed even if input. This prevents flash memory from being erroneously erased or programmed due to CPU runaway.

The protection function can be temporarily disabled by setting the FMLBD bit in FLMCR1 to 1. To clear the lock bit, erase the specified block. Note that lock bits are unavailable in data flash areas.

chanhvinh.com



### 7.7.3 PROM Programmer Protection/Boot Mode Protection

PROM programmer protection/boot mode protection is enabled by writing the specified data to the user ROM area shown in the table 7.13.

The protection function can be disabled by using a PROM programmer or on-board programmer to delete the entire user ROM area.

Table 7.14 shows the specifications for PROM programmer protection and table 7.15 shows the specifications of protection in boot mode.

**Table 7.13 Address Range of the Protection Code in User ROM**

| | H'000004 | H'000005 | H'000006 | H'000007 | H'000010 | H'000011 | H'000012 | H'000013 |
|---|---|---|---|---|---|---|---|---|
| PROM programmer | Control code | Not used | | | | | | |
| Boot mode | | Authentication ID code (56 bits) | | | | | | |

**Table 7.14 Specifications for PROM Programmer Protection**

| Control code* | Protection State | Operation to be Carried Out |
|---|---|---|
| H'FF | PROM programmer protection is disabled. | Possible operations; reading/programming/ erasing by PROM programmer. |
| Other than above | PROM programmer protection is enabled. | Possible operations; programming/erasing by PROM programmer.<br>However, reading is not possible. |

Note: * Used together with control code for boot mode protection.

chanhvinh.com



**Table 7.15 Specifications for Boot Mode Protection**

| Control code* | Protection State | Operation in Serial Connection |
|---|---|---|
| Other than above | Protection is disabled. | Entire blocks are deleted. |
| H'45 | ID authentication protection 1*[2] | Possible for reading/programming/erasing if the ID was authenticated.<br>If the ID was not authenticated, entire blocks are deleted. |
| H'52 | ID authentication protection 2 | Possible for reading/programming/erasing if the ID was authenticated.<br>If the ID was not authenticated, authentication is performed again. |
| | ID authentication protection 2+*[3] | If control code is H'52 and the special code (H'50, H'72, H'6F, H'74, H'65, H'63 and H'74) is written to the authentication ID bytes, processing for serial connections will not be accepted. |

Notes: 1. Used together with the control code for the PROM programmer.
2. Re-authentication can be performed up to 3 times in case of error in the ID code.
3. Once this setting has been made, serial connections are not accepted unless a PROM programmer is used to delete the setting.

## 7.8 Programmer Mode

In programmer mode, flash memory areas can be programmed/erased using a PROM programmer via a socket adapter, just as a discrete flash memory can be. Use the PROM programmer that supports this product.

chanhvinh.com



## 7.9 Usage Notes

### (1) Prohibited Instruction

In EW0 mode, the following instruction cannot be used because it refers to the data in the flash memory area.

- TRAPA

### (2) Interrupts

Table 7.16 shows interrupt handling in CPU reprogramming mode.

**Table 7.16 Interrupt Handling in CPU Reprogramming Mode**

| Mode | State | When Interrupt Request is Received | When Watchdog Timer Reset, LVD Reset, Software Reset, or Pin Reset, Interrupt Request is Generated |
|---|---|---|---|
| EW0 | During erasure command | Interrupts can be handled if interrupt vectors are located in the RAM. For details, see section 4.2.7, Interrupt vector offset register (VOFR). | Immediately after a reset is generated, a software command is forcibly terminated, and flash memory and LSI are reset.<br>Because of the forced termination, it might be impossible to read correct values from the block or address for which the software command has been executed; execute erasure again after restarting and confirm that erasure is completed successfully.<br>The watchdog timer does not stop even during command execution; initialize the timer periodically. |
| | During programming command | | |
| | During lock-bit program command | | |
| | During blank checking command | | |

chanhvinh.com



| Mode | State | When Interrupt Request is Received | When Watchdog Timer Reset, LVD Reset, Software Reset, or Pin Reset, Interrupt Request is Generated |
|---|---|---|---|
| EW1 | During erasure command (erase-suspend function not used) | Erasure is given priority, keeping the interrupt request waiting. When erasure is completed, execution of the interrupt processing is started. | Immediately after a reset is generated, a software command is forcibly terminated, and flash memory and LSI are reset.<br>Because of the forced termination, it might be impossible to read correct values from the block or address for which the software command has been executed; execute erasure again after restarting and confirm that erasure is completed successfully.<br>Since the watchdog timer does not stop even during command execution, set the overflow time of the watchdog timer longer than the erasure/programming execution time. |
| | During erasure command (erase-suspend function used) | After the transition time to erase-suspend mode, erasure is suspended starting execution of the interrupt processing. When the interrupt processing is completed, setting the FMSPREQ bit in FLMCR2 to 0 resumes erasure. | |
| | During programming command | A software command is given priority, keeping the interrupt request waiting. When the software command is completed, execution of the interrupt processing is started. | |
| | During lock-bit program command | | |
| | During blank checking command | | |

### (3) Method of Access

When writing values to the protected bits indicated below, start by writing 0 to the bit and then write 1 to it or by writing 1 to the bit and then write 0 to it. Do not allow the generation of any interrupt or any access to other I/O registers between the two operations. For writing, always use the MOV instruction.

#### (a) Bits that are set to 1 by writing 0 and then 1 consecutively

- FLMCR1: FMLBD and FMCMDEN bits
- FLMCR2: FMISPE and FMSPEN bits

#### (b) Bits that are cleared to 0 by writing 1 and then 0 consecutively

- DFPR: DFPR1 and DFPR0 bits

chanhvinh.com



The example below is of code for use when the FMCMDEN and FMLBD bits in FLMCR1 are to be changed from 0 to 1.

```
MOV.B  @FLMCR1,R0L   :FLMCR1=H'04      R0L=H'04      R0H=H'xx
MOV.B  @FLMCR1,R0H   :FLMCR1=H'04      R0L=H'04      R0H=H'04
BSET   #0,R0H        :FLMCR1=H'04      R0L=H'04      R0H=H'05
BSET   #3,R0H        :FLMCR1=H'04      R0L=H'04      R0H=H'0D
MOV.B  R0L,@FLMCR1   :FLMCR1=H'04      R0L=H'04      R0H=H'0D
MOV.B  R0H,@FLMCR1   :FLMCR1=H'0D      R0L=H'04      R0H=H'0D
```

### (4) Reprogramming User ROM Area

When it is necessary to reprogram the block containing the reprogramming-control program, use boot mode. This is because if the power supply voltage drops in EW0 mode while the block containing the reprogramming-control program is being reprogrammed, the reprogramming-control program cannot be correctly reprogrammed, and this might disable further reprogramming of the flash memory. Only proceed with overwriting of the programming-control program after securing ample stabilization time for the power supply.

### (5) Program

Do not program the already-programmed addresses.

### (6) LSI Mode Transition

During software command execution, do not cause a transition to standby mode or sleep mode.

### (7) Reset during Execution of Software Command in Flash Memory

Do not apply a pin reset, LVD reset, or watchdog timer reset during execution of the programming, lock-bit programming, blank-checking, and erasure commands. If applied, the currently executed command is forcibly terminated. In this case, execute the erasure command of the specified block again and confirm that erasure is completed successfully.

### (8) Reading the User ROM Area while the Mode is EW0 and Software Commands are Enabled

If the user ROM area is to be read while software commands are enabled and the reprogramming mode is EW0, set the bus master operation clock ϕs to a frequency below 5 MHz.

chanhvinh.com



### (9) Frequent Reprogramming

For systems that will be frequently reprogrammed, follow the below procedure to reduce the effective number of reprogramming operations. As far as is possible, write data at appropriately increasing addresses until no blank areas remain, and then erase the whole block. For example, in a case where the data are written in 16-byte sets, write the maximum of 256 16-byte sets, and then erase the whole block when further reprogramming is required. This procedure can reduce the effective number of reprogramming operations.

For control of the number of reprogramming operations, we recommend keeping a record of the number of times each block has been reprogrammed.

### (10) Occurrence of Erase Errors during Erasure Operation

When an erase error has occurred during an erasure operation, issue a clear-status command and then an erasure command. Repeat this procedure at least three times so that the error does not occur.

### (11) Points for Caution when Using the Erase-Suspend Function

In CPU reprogramming mode, if the erase-suspend function is used during erasure of the user ROM area, operations to erase the user ROM area will not be completed in some cases.

1. Conditions for the problem

   In CPU reprogramming mode, repeated use of the erase-suspend function at a certain interval within the period of erasure when the user ROM area is being erased in EW0 mode or EW1 mode

   (1) EW0 mode

   — Repeated generation of interrupts with a certain interval

   — Repeated setting of the FMSPREQ bit in FLMCR2 to 1 with a certain interval

   (2) EW1 mode

   — Repeated generation of interrupts with a certain interval
2. To avoid the problem

   Do not use repeatedly the erase-suspend function at the certain interval. If the erase-suspend function is to be repeatedly used within the period of erasure, employ software control to ensure that the interval between requests satisfies the conditions given in formula 2-1 below to avoid the same interval.

   $T2 > T1 + TD$ or $T2 < T1 - TD$ (2-1)

   — TD = 1.0 µs (certain period that does not depend on the operating frequency of the product)

   — T1: Interval between previous and current requests for suspension of erasure

   — T2: Interval between current and next requests for suspension of erasure

chanhvinh.com



# Section 8 RAM

The H8S/20103, H8S/20203, H8S/20223, H8S/20115, H8S/20215, and H8S/20235 Group LSIs have an on-chip high-speed static RAM. The RAM is connected to the CPU via a 16-bit data bus, enabling the CPU to access both byte data and word data in one state.

| Product Classification | | RAM Size | RAM Address |
|---|---|---|---|
| 64 pins | H8S/20103 | 8 Kbytes | H'FFDF80 to H'FFFF7F |
| | H8S/20102 | 8 Kbytes | H'FFDF80 to H'FFFF7F |
| | H8S/20115 | 12 Kbytes | H'FFCF80 to H'FFFF7F |
| | H8S/20114 | 12 Kbytes | H'FFCF80 to H'FFFF7F |
| 80 pins | H8S/20223 | 8 Kbytes | H'FFDF80 to H'FFFF7F |
| | H8S/20222 | 8 Kbytes | H'FFDF80 to H'FFFF7F |
| | H8S/20203 | 8 Kbytes | H'FFDF80 to H'FFFF7F |
| | H8S/20202 | 8 Kbytes | H'FFDF80 to H'FFFF7F |
| | H8S/20215 | 12 Kbytes | H'FFCF80 to H'FFFF7F |
| | H8S/20214 | 12 Kbytes | H'FFCF80 to H'FFFF7F |
| | H8S/20235 | 12 Kbytes | H'FFCF80 to H'FFFF7F |
| | H8S/20234 | 12 Kbytes | H'FFCF80 to H'FFFF7F |

chanhvinh.com



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

chanhvinh.com



# Section 9 Peripheral I/O Mapping Controller

The peripheral function mapping controller (PMC) is composed of registers that are used to select the functions of multiplexed pins. The multiplexed pins are divided into two groups: group 1 and group 2. Group 1 consists of ports 1 to 3, 5, and 6, and group 2 consists of ports 8, 9*, and A. Tables 9.1 and 9.2 list the functions of the multiplexed pins in each group.

Note: Port 9 is not available on the H8S/20103 and H8S/20115 Groups.

**Table 9.1 Multiplexed Pin Functions (Ports 1, 2, 3, 5, and 6)**

| Group 1 | Pin Name | Function 1 | Function 2 | Function 3 | Function 4 | Function 5 | Function 6 |
|---|---|---|---|---|---|---|---|
| Port 1<br>Port 2<br>Port 3<br>Port 5<br>Port 6 | Pm7 | $\overline{\text{IRQ7}}$ input | TXD_2 output | TXD_3 output | SSI input/output | FTIOD1 input/output | $\overline{\text{ADTRG2}}$ input |
| | Pm6 | $\overline{\text{IRQ6}}$ input | RXD_2 input | RXD_3 input | $\overline{\text{SCS}}$ input/output | FTIOC1 input/output | $\overline{\text{ADTRG1}}$ input |
| | Pm5 | $\overline{\text{IRQ5}}$ input | SCK3_2 input/output | SCK3_3 input/output | SSCK input/output | FTIOB1 input/output | $\overline{\text{TRDOI_1}}$ input |
| | Pm4 | $\overline{\text{IRQ4}}$ input | $\overline{\text{TRDOI_0}}$ input | FTCI input* | SSO input/output | FTIOA1 input/output | TRAIO input/output |
| | Pm3 | $\overline{\text{IRQ3}}$ input | $\overline{\text{TRCOI}}$ input* | FTIOD input/output* | TGIOB input/output | FTIOD0 input/output | TRAO output |
| | Pm2 | $\overline{\text{IRQ2}}$ input | TXD output | FTIOC input/output* | TGIOA input/output | FTIOC0 input/output | TRBO output |
| | Pm1 | $\overline{\text{IRQ1}}$ input | RXD input | FTIOB input/output* | TCLKB input | FTIOB0 input/output | TRGB input |
| | Pm0 | $\overline{\text{IRQ0}}$ input | SCK3 input/output | FTIOA input/output* | TCLKA input | FTIOA0 input/output | TREO output |
| Initially mapped port | | Port 1 | Port 2 | Port 3 | Port 5 | Port 6 | None |

[Legend] m = 1, 2, 3, 5 and 6

Notes: 1. The timer RC is not available on the H8S/20203, H8S/20223, H8S/20215, and H8S/20235 Groups; therefore, the function cannot be selected for these groups.

2. The SCL and SDA functions for the IIC2 module are not allocatable to pins other than P56 and P57.

chanhvinh.com



**Table 9.2 Multiplexed Pin Functions (Ports 8, 9, and A)**

| Group 2 | Pin Name | Function 1 | Function 2 | Function 3 | Function 4 | Function 5*2 | Function 6 |
|---|---|---|---|---|---|---|---|
| Port 8<br>Port 9<br>Port A*1 | Pm7 | $\overline{\text{IRQ7}}$ input | — | TXD output | TREO output | FTIOD3 input/output | TXD_3 output |
| | Pm6 | $\overline{\text{IRQ6}}$ input | — | RXD input | TRBO output | FTIOC3 input/output | RXD_3 input |
| | Pm5 | $\overline{\text{IRQ5}}$ input | — | SCK3 input/output | TRAIO input/output | FTIOB3 input/output | SCK3_3 input/output |
| | Pm4 | $\overline{\text{IRQ4}}$ input | — | — | TRGB input | FTIOA3 input/output | — |
| | Pm3 | $\overline{\text{IRQ3}}$ input | — | — | TRAO output | FTIOD2 input/output | — |
| | Pm2 | $\overline{\text{IRQ2}}$ input | — | — | — | FTIOC2 input/output | — |
| | Pm1 | $\overline{\text{IRQ1}}$ input | — | — | — | FTIOB2 input/output | — |
| | Pm0 | $\overline{\text{IRQ0}}$ input | — | — | — | FTIOA2 input/output | — |
| Initially mapped port | | None | None | None | Port 8 | Port 9 | None |

[Legend] n = 8, 9, and A
—: Reserved

Notes: 1. Port A is multiplexed with A/D converter analog input in the H8S/20223 and H8S/20235 Groups. Therefore, the multiplexed functions cannot be selected for the port.
The PA3 to PA0 pins are multiplexed with A/D converter analog input in the H8S/20203 and H8S/20215 Groups. Therefore, the multiplexed functions cannot be selected for the port pins.
2. Function 5 cannot be selected for the H8S/20103 and H8S/20115 Groups.

chanhvinh.com



## 9.1 Register Descriptions

- Peripheral function mapping register write-protect register (PMCWPR)
- Port 1 peripheral function mapping register 1 (PMCR11)
- Port 1 peripheral function mapping register 2 (PMCR12)
- Port 1 peripheral function mapping register 3 (PMCR13)
- Port 1 peripheral function mapping register 4 (PMCR14)
- Port 2 peripheral function mapping register 1 (PMCR21)
- Port 2 peripheral function mapping register 2 (PMCR22)
- Port 2 peripheral function mapping register 3 (PMCR23)
- Port 2 peripheral function mapping register 4 (PMCR24)
- Port 3 peripheral function mapping register 1 (PMCR31)
- Port 3 peripheral function mapping register 2 (PMCR32)
- Port 3 peripheral function mapping register 3 (PMCR33)
- Port 3 peripheral function mapping register 4 (PMCR34)
- Port 5 peripheral function mapping register 1 (PMCR51)
- Port 5 peripheral function mapping register 2 (PMCR52)
- Port 5 peripheral function mapping register 3 (PMCR53)
- Port 5 peripheral function mapping register 4 (PMCR54)
- Port 6 peripheral function mapping register 1 (PMCR61)
- Port 6 peripheral function mapping register 2 (PMCR62)
- Port 6 peripheral function mapping register 3 (PMCR63)
- Port 6 peripheral function mapping register 4 (PMCR64)
- Port 8 peripheral function mapping register 3 (PMCR83)
- Port 8 peripheral function mapping register 4 (PMCR84)
- Port 9 peripheral function mapping register 1 (PMCR91)*[1]
- Port 9 peripheral function mapping register 2 (PMCR92)*[1]
- Port 9 peripheral function mapping register 3 (PMCR93)*[1]
- Port 9 peripheral function mapping register 4 (PMCR94)*[1]
- Port A peripheral function mapping register 3 (PMCRA3)*[2]
- Port A peripheral function mapping register 4 (PMCRA4)*[2]

Notes: 1. PMCR91 to PMCR94 are not available on the H8S/20103 and H8S/20115 Groups.
2. PMCRA3 and PMCRA4 are not available on the H8S/20223 and H8S/20235 Groups.

chanhvinh.com



### 9.1.1 Peripheral Function Mapping Register Write-Protect Register (PMCWPR)

Address: H'FF0065

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | B0WI | PMCRWE | — | — | — | — | — | — |
| Value after reset: | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | B0WI | Bit 6 write protect | 0: Writing to PMCRWE (bit 6 in this register) is enabled.<br>1: Writing to the PMCRWE bit is disabled. | W |
| 6 | PMCRWE | PMCR write enable | 0: Writing to PMCR is disabled.<br>1: Writing to PMCR is enabled. | R/W |
| 5 to 0 | — | Reserved | These bits are always read as 0. The write value should always be 0. | — |

Note: A MOV instruction should be used to rewrite this register.

- B0WI bit (Bit 6 write protect)

  Only when the write value to this bit is 0, PMCRWE (bit 6 in this register) can be modified. This bit is always read as 1.

chanhvinh.com



### 9.1.2 Port Group 1

**Peripheral Function Mapping Registers 1 to 4 (PMCRn1 to PMCRn4 (n = 1, 2, 3, 5, and 6))**

**(1) Port 1**

**(a) Port 1 Peripheral Function Mapping Register 1 (PMCR11)**

Address: H'FF0040

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | P11MD[2:0] | | | — | P10MD[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 6 to 4 | P11MD[2:0] | P11 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ1}}$ input (initial value)<br>010: RXD input (SCI3_1)<br>011: FTIOB input/output (timer RC)*2<br>100: TCLKB input (timer RG)<br>101: FTIOB0 input/output (timer RD_0)<br>110: TRGB input (timer RB)<br>111: Setting prohibited | R/W |
| 3 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 2 to 0 | P10MD[2:0] | P10 function select*1 | 000: Setting prohibited<br>001: $\overline{\text{IRQ0}}$ input (initial value)<br>010: SCK3 input/output (SCI3_1)<br>011: FTIOA input/output (timer RC)*2<br>100: TCLKA input (timer RG)<br>101: FTIOA0 input/output (timer RD_0)<br>110: TREO output (timer RE)<br>111: Setting prohibited | R/W |

Notes: 1. For the H8S/20103 and H8S/20115 Groups, P10 is not provided and P10MD[2:0] are reserved. The initial value is B'001. The write value should be B'001.
2. This function cannot be selected for the H8S/20203, H8S/20223, H8S/20215, and H8S/20235 Groups.

chanhvinh.com



### (b) Port 1 Peripheral Function Mapping Register 2 (PMCR12)

Address: H'FF0041

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | P13MD[2:0] | | | — | P12MD[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 6 to 4 | P13MD[2:0] | P13 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ3}}$ input (initial value)<br>010: $\overline{\text{TRCOI}}$ input (timer RC)*<br>011: FTIOD input/output (timer RC)*<br>100: TGIOB input/output (timer RG)<br>101: FTIOD0 input/output (timer RD_0)<br>110: TRAO output (timer RA)<br>111: Setting prohibited | R/W |
| 3 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 2 to 0 | P12MD[2:0] | P12 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ2}}$ input (initial value)<br>010: TXD output (SCI3_1)<br>011: FTIOC input/output (timer RC)*<br>100: TGIOA input/output (timer RG)<br>101: FTIOC0 input/output (timer RD_0)<br>110: TRBO output (timer RB)<br>111: Setting prohibited | R/W |

Note: * This function cannot be selected for the H8S/20203, H8S/20223, H8S/20215, and H8S/20235 Groups.

chanhvinh.com



### (c) Port 1 Peripheral Function Mapping Register 3 (PMCR13)

Address: H'FF0042

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | P15MD[2:0] | | | — | P14MD[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 6 to 4 | P15MD[2:0] | P15 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ5}}$ input (initial value)<br>010: SCK3_2 input/output (SCI3_2)<br>011: SCK3_3 input/output (SCI3_3)<br>100: SSCK input/output*4 (SSU)<br>101: FTIOB1 input/output (timer RD_0)<br>110: $\overline{\text{TRDOI_1}}$ input (timer RD_1)*2<br>111: Setting prohibited | R/W |
| 3 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 2 to 0 | P14MD[2:0] | P14 function select*1 | 000: Setting prohibited<br>001: $\overline{\text{IRQ4}}$ input (initial value)<br>010: $\overline{\text{TRDOI_0}}$ input (timer RD_0)<br>011: FTCI input (timer RC)*3<br>100: SSO input/output*4 (SSU)<br>101: FTIOA1 input/output (timer RD_0)<br>110: TRAIO input/output (timer RA)<br>111: Setting prohibited | R/W |

Notes: 1. For the H8S/20103 and H8S/20115 Groups, P14 is not provided and P14MD[2:0] are reserved. The initial value is B'001. The write value should be B'001.
2. This function cannot be selected for the H8S/20103 and H8S/20115 Groups.
3. This function cannot be selected for the H8S/20203, H8S/20223, H8S/20215, and H8S/20235 Groups.
4. If the SSCK output pin or the SSO output pin is set, the NMOS open-drain output cannot be selected.

chanhvinh.com



### (d) Port 1 Peripheral Function Mapping Register 4 (PMCR14)

Address: H'FF0043

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | P17MD[2:0] | | | — | P16MD[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 6 to 4 | P17MD[2:0] | P17 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ7}}$ input (initial value)<br>010: TXD_2 output (SCI3_2)<br>011: TXD_3 output (SCI3_3)<br>100: SSI input/output (SSU)<br>101: FTIOD1 input/output (timer RD_0)<br>110: $\overline{\text{ADTRG2}}$ input (AD_2)<br>111: Setting prohibited | R/W |
| 3 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 2 to 0 | P16MD[2:0] | P16 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ6}}$ input (initial value)<br>010: RXD_2 input (SCI3_2)<br>011: RXD_3 input (SCI3_3)<br>100: $\overline{\text{SCS}}$ input/output* (SSU)<br>101: FTIOC1 input/output (timer RD_0)<br>110: $\overline{\text{ADTRG1}}$ input (AD_1)<br>111: Setting prohibited | R/W |

Note: * If the $\overline{\text{SCS}}$ output pin of the SSU is set, the NMOS open-drain output cannot be selected.

chanhvinh.com



## (2) Port 2

### (a) Port 2 Peripheral Function Mapping Register 1 (PMCR21)

Address: H'FF0044

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | P21MD[2:0] | | | — | P20MD[2:0] | | |
| Value after reset: | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 6 to 4 | P21MD[2:0] | P21 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ1}}$ input<br>010: RXD input (SCI3_1) (initial value)<br>011: FTIOB input/output (timer RC)*<br>100: TCLKB input (timer RG)<br>101: FTIOB0 input/output (timer RD_0)<br>110: TRGB input (timer RB)<br>111: Setting prohibited | R/W |
| 3 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 2 to 0 | P20MD[2:0] | P20 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ0}}$ input<br>010: SCK3 input/output (SCI3_1) (initial value)<br>011: FTIOA input/output (timer RC)*<br>100: TCLKA input (timer RG)<br>101: FTIOA0 input/output (timer RD_0)<br>110: TREO output (timer RE)<br>111: Setting prohibited | R/W |

Note: * The timer RC is not available on the H8S/20203, H8S/20223, H8S/20215, and H8S/20235 Groups. These bits are reserved and the function cannot be selected for these groups.

chanhvinh.com



### (b) Port 2 Peripheral Function Mapping Register 2 (PMCR22)

Address: H'FF0045

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | P23MD[2:0] | | | — | P22MD[2:0] | | |
| Value after reset: | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |

| Bit | Bit Name | Initial Value | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 6 to 4 | P23MD[2:0] | P23 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ3}}$ input<br>010: $\overline{\text{TRCOI}}$ input (timer RC) (initial value)<br>011: FTIOD input/output (timer RC)*<br>100: TGIOB input/output (timer RG)<br>101: FTIOD0 input/output (timer RD_0)<br>110: TRAO output (timer RA)<br>111: Setting prohibited | R/W |
| 3 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 2 to 0 | P22MD[2:0] | P22 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ2}}$ input<br>010: TXD output (SCI3_1) (initial value)<br>011: FTIOC input/output (timer RC)*<br>100: TGIOA input/output (timer RG)<br>101: FTIOC0 input/output (timer RD_0)<br>110: TRBO output (timer RB)<br>111: Setting prohibited | R/W |

Note: * The timer RC is not available on the H8S/20203, H8S/20223, H8S/20215, and H8S/20235 Groups. These bits are reserved and the function cannot be selected for these groups.

chanhvinh.com



### (c) Port 2 Peripheral Function Mapping Register 3 (PMCR23)

Address: H'FF0046

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | P25MD[2:0] | | | — | P24MD[2:0] | | |
| Value after reset: | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 6 to 4 | P25MD[2:0] | P25 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ5}}$ input<br>010: SCK3_2 input/output (SCI3_2) (initial value)<br>011: SCK3_3 input/output (SCI3_3)<br>100: SSCK input/output*3 (SSU)<br>101: FTIOB1 input/output (timer RD_0)<br>110: $\overline{\text{TRDOI_1}}$ input (timer RD_1)*1<br>111: Setting prohibited | R/W |
| 3 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 2 to 0 | P24MD[2:0] | P24 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ4}}$ input<br>010: $\overline{\text{TRDOI_0}}$ input (timer RD_0) (initial value)<br>011: FTCI input (timer RC)*2<br>100: SSO input/output*3 (SSU)<br>101: FTIOA1 input/output (timer RD_0)<br>110: TRAIO input/output (timer RA)<br>111: Setting prohibited | R/W |

Notes: 1. This function cannot be selected for the H8S/20103 and H8S/20115 Groups.
2. This function cannot be selected for the H8S/20203, H8S/20223, H8S/20215, and H8S/20235 Groups.
3. If the SSCK output pin or the SSO output pin is set, the NMOS open-drain output cannot be selected.

chanhvinh.com



### (d) Port 2 Peripheral Function Mapping Register 4 (PMCR24)

Address: H'FF0047

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | P27MD[2:0] | | | — | P26MD[2:0] | | |
| Value after reset: | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 6 to 4 | P27MD[2:0] | P27 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ7}}$ input<br>010: TXD_2 output (SCI3_2) (initial value)<br>011: TXD_3 output (SCI3_3)<br>100: SSI input/output (SSU)<br>101: FTIOD1 input/output (timer RD_0)<br>110: $\overline{\text{ADTRG2}}$ input (AD_2)<br>111: Setting prohibited | R/W |
| 3 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 2 to 0 | P26MD[2:0] | P26 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ6}}$ input<br>010: RXD_2 input (SCI3_2) (initial value)<br>011: RXD_3 input (SCI3_3)<br>100: $\overline{\text{SCS}}$ input/output* (SSU)<br>101: FTIOC1 input/output (timer RD_0)<br>110: $\overline{\text{ADTRG1}}$ input (AD_1)<br>111: Setting prohibited | R/W |

Note: * If the $\overline{\text{SCS}}$ output pin of the SSU is set, the NMOS open-drain output cannot be selected.

chanhvinh.com



## (3) Port 3

### (a) Port 3 Peripheral Function Mapping Register 1 (PMCR31)

Address: H'FF0048

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | P31MD[2:0] | | | — | P30MD[2:0] | | |
| Value after reset: | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 6 to 4 | P31MD[2:0] | P31 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ1}}$ input<br>010: RXD input (SCI3_1)<br>011: FTIOB input/output (timer RC)* (initial value)<br>100: TCLKB input (timer RG)<br>101: FTIOB0 input/output (timer RD_0)<br>110: TRGB input (timer RB)<br>111: Setting prohibited | R/W |
| 3 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 2 to 0 | P30MD[2:0] | P30 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ0}}$ input<br>010: SCK3 input/output (SCI3_1)<br>011: FTIOA input/output (timer RC)* (initial value)<br>100: TCLKA input (timer RG)<br>101: FTIOA0 input/output (timer RD_0)<br>110: TREO output (timer RE)<br>111: Setting prohibited | R/W |

Note: * The timer RC is not available on the H8S/20203, H8S/20223, H8S/20215, and H8S/20235 Groups. No function is selected in the initial state for these groups.

chanhvinh.com



### (b) Port 3 Peripheral Function Mapping Register 2 (PMCR32)

Address: H'FF0049

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | P33MD[2:0] | | | — | P32MD[2:0] | | |
| Value after reset: | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 6 to 4 | P33MD[2:0] | P33 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ3}}$ input<br>010: $\overline{\text{TRCOI}}$ input (timer RC)*<br>011: FTIOD input/output (timer RC)* (initial value)<br>100: TGIOB input/output (timer RG)<br>101: FTIOD0 input/output (timer RD_0)<br>110: TRAO output (timer RA)<br>111: Setting prohibited | R/W |
| 3 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 2 to 0 | P32MD[2:0] | P32 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ2}}$ input<br>010: TXD output (SCI3_1)<br>011: FTIOC input/output (timer RC)* (initial value)<br>100: TGIOA input/output (timer RG)<br>101: FTIOC0 input/output (timer RD_0)<br>110: TRBO output (timer RB)<br>111: Setting prohibited | R/W |

Note: * The timer RC is not available on the H8S/20203, H8S/20223, H8S/20215, and H8S/20235 Groups. No function is selected in the initial state for these groups.

chanhvinh.com



### (c) Port 3 Peripheral Function Mapping Register 3 (PMCR33)

Address: H'FF004A

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | P35MD[2:0] | | | — | P34MD[2:0] | | |
| Value after reset: | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 6 to 4 | P35MD[2:0] | P35 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ5}}$ input<br>010: SCK3_2 input/output (SCI3_2)<br>011: SCK3_3 input/output (SCI3_3) (initial value)<br>100: SSCK input/output*3 (SSU)<br>101: FTIOB1 input/output (timer RD_0)<br>110: $\overline{\text{TRDOI_1}}$ input (timer RD_1)*2<br>111: Setting prohibited | R/W |
| 3 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 2 to 0 | P34MD[2:0] | P34 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ4}}$ input<br>010: $\overline{\text{TRDOI_0}}$ input (timer RD_0)<br>011: FTCI input (timer RC)*1 (initial value)<br>100: SSO input/output*3 (SSU)<br>101: FTIOA1 input/output (timer RD_0)<br>110: TRAIO input/output (timer RA)<br>111: Setting prohibited | R/W |

Notes: 1. The timer RC is not available on the H8S/20203, H8S/20223, H8S/20215, and H8S/20235 Groups. No function is selected in the initial state for these groups.
2. This function cannot be selected for the H8S/20103 and H8S/20115 Groups.
3. If the SSCK output pin or the SSO output pin is set, the NMOS open-drain output cannot be selected.

chanhvinh.com



### (d) Port 3 Peripheral Function Mapping Register 4 (PMCR34)

Address: H'FF004B

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | P37MD[2:0] | | | — | P36MD[2:0] | | |
| Value after reset: | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 6 to 4 | P37MD[2:0] | P37 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ7}}$ input<br>010: TXD_2 output (SCI3_2)<br>011: TXD_3 output (SCI3_3) (initial value)<br>100: SSI input/output (SSU)<br>101: FTIOD1 input/output (timer RD_0)<br>110: $\overline{\text{ADTRG2}}$ input (AD_2)*1<br>111: Setting prohibited | R/W |
| 3 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 2 to 0 | P36MD[2:0] | P36 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ6}}$ input<br>010: RXD_2 input (SCI3_2)<br>011: RXD_3 input (SCI3_3) (initial value)<br>100: $\overline{\text{SCS}}$ input/output*2 (SSU)<br>101: FTIOC1 input/output (timer RD_0)<br>110: $\overline{\text{ADTRG1}}$ input (AD_1)<br>111: Setting prohibited | R/W |

Notes: 1. This function cannot be selected for the H8S/20103, H8S/20203, H8S/20115, and H8S/20215 Groups.

2. If the $\overline{\text{SCS}}$ output pin of the SSU is set, the NMOS open-drain output cannot be selected.

chanhvinh.com



## (4) Port 5

### (a) Port 5 Peripheral Function Mapping Register 1 (PMCR51)

Address: H'FF0050

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | P51MD[2:0] | | | — | P50MD[2:0] | | |
| Value after reset: | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 6 to 4 | P51MD[2:0] | P51 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ1}}$ input<br>010: RXD input (SCI3_1)<br>011: FTIOB input/output (timer RC)*<br>100: TCLKB input (timer RG) (initial value)<br>101: FTIOB0 input/output (timer RD_0)<br>110: TRGB input (timer RB)<br>111: Setting prohibited | R/W |
| 3 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 2 to 0 | P50MD[2:0] | P50 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ0}}$ input<br>010: SCK3 input/output (SCI3_1)<br>011: FTIOA input/output (timer RC)*<br>100: TCLKA input (timer RG) (initial value)<br>101: FTIOA0 input/output (timer RD_0)<br>110: TREO output (timer RE)<br>111: Setting prohibited | R/W |

Note: * The timer RC is not available on the H8S/20203, H8S/20223, H8S/20215, and H8S/20235 Groups. These bits are reserved and the function cannot be selected for these groups.

chanhvinh.com



### (b) Port 5 Peripheral Function Mapping Register 2 (PMCR52)

Address: H'FF0051

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | P53MD[2:0] | | | — | P52MD[2:0] | | |
| Value after reset: | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 6 to 4 | P53MD[2:0] | P53 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ3}}$ input<br>010: $\overline{\text{TRCOI}}$ input (timer RC)*<br>011: FTIOD input/output (timer RC)*<br>100: TGIOB input/output (timer RG) (initial value)<br>101: FTIOD0 input/output (timer RD_0)<br>110: TRAO output (timer RA)<br>111: Setting prohibited | R/W |
| 3 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 2 to 0 | P52MD[2:0] | P52 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ2}}$ input<br>010: TXD output (SCI3_1)<br>011: FTIOC input/output (timer RC)*<br>100: TGIOA input/output (timer RG) (initial value)<br>101: FTIOC0 input/output (timer RD_0)<br>110: TRBO output (timer RB)<br>111: Setting prohibited | R/W |

Note: * The timer RC is not available on the H8S/20203, H8S/20223, H8S/20215, and H8S/20235 Groups. These bits are reserved and the function cannot be selected for these groups.

chanhvinh.com



### (c) Port 5 Peripheral Function Mapping Register 3 (PMCR53)

Address: H'FF0052

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | P55MD[2:0] | | | — | P54MD[2:0] | | |
| Value after reset: | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 6 to 4 | P55MD[2:0] | P55 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ5}}$ input<br>010: SCK3_2 input/output (SCI3_2)<br>011: SCK3_3 input/output (SCI3_3)<br>100: SSCK input/output*3 (SSU) (initial value)<br>101: FTIOB1 input/output (timer RD_0)<br>110: $\overline{\text{TRDOI_1}}$ input/output (timer RD_1)*2<br>111: Setting prohibited | R/W |
| 3 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 2 to 0 | P54MD[2:0] | P54 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ4}}$ input<br>010: $\overline{\text{TRDOI_0}}$ input (timer RD_0)<br>011: FTCI input (timer RC)*1<br>100: SSO input/output*3 (SSU) (initial value)<br>101: FTIOA1 input/output (timer RD_0)<br>110: TRAIO input/output (timer RA)<br>111: Setting prohibited | R/W |

Notes: 1. The timer RC is not available on the H8S/20203, H8S/20223, H8S/20215, and H8S/20235 Groups. These bits are reserved and the function cannot be selected for these groups.
2. This function cannot be selected for the H8S/20103 and H8S/20115 Groups.
3. If the NMOS open-drain output is selected for the SSCK output pin or the SSO output pin, use the PMC to allocate that pin from port 5.

chanhvinh.com



### (d) Port 5 Peripheral Function Mapping Register 4 (PMCR54)

Address: H'FF0053

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | P57MD[2:0] | | | — | P56MD[2:0] | | |
| Value after reset: | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 6 to 4 | P57MD[2:0] | P57 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ7}}$ input<br>010: TXD_2 output (SCI3_2)<br>011: TXD_3 output (SCI3_3)<br>100: SSI/SCL input/output*1 (SSU/IIC2) (initial value)<br>101: FTIOD1 input/output (timer RD_0)<br>110: $\overline{\text{ADTRG2}}$ input (AD_2)<br>111: Setting prohibited | R/W |
| 3 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 2 to 0 | P56MD[2:0] | P56 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ6}}$ input<br>010: RXD_2 input (SCI3_2)<br>011: RXD_3 input (SCI3_3)<br>100: $\overline{\text{SCS}}$/SDA input/output*1*2 (SSU/IIC2) (initial value)<br>101: FTIOC1 input/output (timer RD_0)<br>110: $\overline{\text{ADTRG1}}$ input (AD_1)<br>111: Setting prohibited | R/W |

chanhvinh.com



Notes: 1. When the IIC2/SSU is used as the IIC2 function, the SCL and SDA functions can be allocated to the P56 and P57 pins only because SCL and SDA require buffers dedicated for IIC input/output. The PMC can not be used to allocate the SCL and SDA functions to other pins. When the IIC2/SSU is used for the SSU function except *2, there is no restriction.
The P56 and P57 pins have different characteristics from other pins. When these pins are used as the SCL and SDA pins for the IIC2, they provide NMOS open drain output. When the P56 and P57 pins are used for other output functions, they provide NMOS push-pull output and characteristics of the high level output is different from that of the CMOS output.

2. If the NMOS open-drain output is selected for the $\overline{SCS}$ output pin, use the PMC to allocate that pin from port 5

chanhvinh.com



## (5) Port 6

### (a) Port 6 Peripheral Function Mapping Register 1 (PMCR61)

Address: H'FF0054

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | P61MD[2:0] | | | — | P60MD[2:0] | | |
| Value after reset: | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 6 to 4 | P61MD[2:0] | P61 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ1}}$ input<br>010: RXD input (SCI3_1)<br>011: FTIOB input/output (timer RC)*<br>100: TCLKB input (timer RG)<br>101: FTIOB0 input/output (timer RD_0) (initial value)<br>110: TRGB input (timer RB)<br>111: Setting prohibited | R/W |
| 3 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 2 to 0 | P60MD[2:0] | P60 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ0}}$ input<br>010: SCK3 input/output (SCI3_1)<br>011: FTIOA input/output (timer RC)*<br>100: TCLKA input (timer RG)<br>101: FTIOA0 input/output (timer RD_0) (initial value)<br>110: TREO output (timer RE)<br>111: Setting prohibited | R/W |

Note: * The timer RC is not available on the H8S/20203, H8S/20223, H8S/20215, and H8S/20235 Groups. These bits are reserved and the function cannot be selected for these groups.

chanhvinh.com



### (b) Port 6 Peripheral Function Mapping Register 2 (PMCR62)

Address: H'FF0055

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | P63MD[2:0] | | | — | P62MD[2:0] | | |
| Value after reset: | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 6 to 4 | P63MD[2:0] | P63 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ3}}$ input<br>010: $\overline{\text{TRCOI}}$ input (timer RC)*<br>011: FTIOD input/output (timer RC)*<br>100: TGIOB input/output (timer RG)<br>101: FTIOD0 input/output (timer RD_0) (initial value)<br>110: TRAO output (timer RA)<br>111: Setting prohibited | R/W |
| 3 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 2 to 0 | P62MD[2:0] | P62 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ2}}$ input<br>010: TXD output (SCI3_1)<br>011: FTIOC input/output (timer RC)*<br>100: TGIOA input/output (timer RG)<br>101: FTIOC0 input/output (timer RD_0) (initial value)<br>110: TRBO output (timer RB)<br>111: Setting prohibited | R/W |

Note: * The timer RC is not available on the H8S/20203, H8S/20223, H8S/20215, and H8S/20235 Groups. These bits are reserved and the function cannot be selected for these groups.

chanhvinh.com



### (c) Port 6 Peripheral Function Mapping Register 3 (PMCR63)

Address: H'FF0056

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | P65MD[2:0] | | | — | P64MD[2:0] | | |
| Value after reset: | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 6 to 4 | P65MD[2:0] | P65 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ5}}$ input<br>010: SCK3_2 input/output (SCI3_2)<br>011: SCK3_3 input/output (SCI3_3)<br>100: SSCK input/output*3 (SSU)<br>101: FTIOB1 input/output (timer RD_0) (initial value)<br>110: $\overline{\text{TRDOI_1}}$ input (timer RD_1)*2<br>111: Setting prohibited | R/W |
| 3 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 2 to 0 | P64MD[2:0] | P64 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ4}}$ input<br>010: $\overline{\text{TRDOI_0}}$ input (timer RD_0)<br>011: FTCI input (timer RC)*1<br>100: SSO input/output*3 (SSU)<br>101: FTIOA1 input/output (timer RD_0) (initial value)<br>110: TRAIO input/output (timer RA)<br>111: Setting prohibited | R/W |

Notes: 1. The timer RC is not available on the H8S/20203, H8S/20223, H8S/20215, and H8S/20235 Groups. These bits are reserved and the function cannot be selected for these groups.
2. This function cannot be selected for the H8S/20103 and H8S/20115 Groups.
3. If the SSCK output pin or the SSO output pin is set, the NMOS open-drain output cannot be selected.

chanhvinh.com



### (d) Port 6 Peripheral Function Mapping Register 4 (PMCR64)

Address: H'FF0057

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | P67MD[2:0] | | | — | P66MD[2:0] | | |
| Value after reset: | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 6 to 4 | P67MD[2:0] | P67 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ7}}$ input<br>010: TXD_2 output (SCI3_2)<br>011: TXD_3 output (SCI3_3)<br>100: SSI input/output (SSU)<br>101: FTIOD1 input/output (timer RD_0) (initial value)<br>110: $\overline{\text{ADTRG2}}$ input (AD_2) *1<br>111: Setting prohibited | R/W |
| 3 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 2 to 0 | P66MD[2:0] | P66 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ6}}$ input<br>010: RXD_2 input (SCI3_2)<br>011: RXD_3 input (SCI3_3)<br>100: $\overline{\text{SCS}}$ input/output*2 (SSU)<br>101: FTIOC1 input/output (timer RD_0) (initial value)<br>110: $\overline{\text{ADTRG1}}$ input (AD_1)<br>111: Setting prohibited | R/W |

Notes: 1. This function cannot be selected for the H8S/20103, H8S/20203, H8S/20115, and H8S/20215 Groups.
2. If the $\overline{\text{SCS}}$ output pin of the SSU is set, the NMOS open-drain output cannot be selected.

chanhvinh.com



### 9.1.3 Port Group 2

### Peripheral Function Mapping Registers 1 to 4 (PMCRn1 to PMCRn4 (n = 8, 9, and A)

#### (1) Port 8

#### (a) Port 8 Peripheral Function Mapping Register 3 (PMCR83)

Address: H'FF005E

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | P85MD[2:0] | | | — | — | — | — |
| Value after reset: | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 6 to 4 | P85MD[2:0] | P85 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ5}}$ input<br>010: Setting prohibited<br>011: SCK3 input/output (SCI3_1)<br>100: TRAIO input/output (timer RA) (initial value)<br>101: FTIOB3 input/output (timer RD_1)*<br>110: SCK3_3 input/output (SCI3_3)<br>111: Setting prohibited | R/W |
| 3 to 0 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |

Note: * This function cannot be selected for the H8S/20103 and H8S/20115 Groups.

chanhvinh.com



### (b) Port 8 Peripheral Function Mapping Register 4 (PMCR84)

Address: H'FF005F

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | P87MD[2:0] | | | — | P86MD[2:0] | | |
| Value after reset: | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 6 to 4 | P87MD[2:0] | P87 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ7}}$ input<br>010: Setting prohibited<br>011: TXD output (SCI3_1)<br>100: TREO output (timer RE) (initial value)<br>101: FTIOD3 input/output (timer RD_1)*<br>110: TXD_3 output (SCI3_3)<br>111: Setting prohibited | R/W |
| 3 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 2 to 0 | P86MD[2:0] | P86 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ6}}$ input<br>010: Setting prohibited<br>011: RXD input (SCI3_1)<br>100: TRBO output (timer RB) (initial value)<br>101: FTIOC3 input/output (timer RD_1)*<br>110: RXD_3 input (SCI3_3)<br>111: Setting prohibited | R/W |

Note: * This function cannot be selected for the H8S/20103 and H8S/20115 Groups.

chanhvinh.com



## (2) Port 9

### (a) Port 9 Peripheral Function Mapping Register 1 (PMCR91)

Address: H'FF0060

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | P91MD[2:0] | | | — | P90MD[2:0] | | |
| Value after reset: | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 6 to 4 | P91MD[2:0] | P91 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ1}}$ input<br>010: Setting prohibited<br>011: Setting prohibited<br>100: Setting prohibited<br>101: FTIOB2 input/output (timer RD_1) (initial value)<br>110: Setting prohibited<br>111: Setting prohibited | R/W |
| 3 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 2 to 0 | P90MD[2:0] | P90 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ0}}$ input<br>010: Setting prohibited<br>011: Setting prohibited<br>100: Setting prohibited<br>101: FTIOA2 input/output (timer RD_1) (initial value)<br>110: Setting prohibited<br>111: Setting prohibited | R/W |

Note: PMCR91 is not available on the H8S/20103 and H8S/20115 Groups.

chanhvinh.com



### (b) Port 9 Peripheral Function Mapping Register 2 (PMCR92)

Address: H'FF0061

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | P93MD[2:0] | | | — | P92MD[2:0] | | |
| Value after reset: | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 6 to 4 | P93MD[2:0] | P93 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ3}}$ input<br>010: Setting prohibited<br>011: Setting prohibited<br>100: TRAO output (timer RA)<br>101: FTIOD2 input/output (timer RD_1) (initial value)<br>110: Setting prohibited<br>111: Setting prohibited | R/W |
| 3 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 2 to 0 | P92MD[2:0] | P92 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ2}}$ input<br>010: Setting prohibited<br>011: Setting prohibited<br>100: Setting prohibited<br>101: FTIOC2 input/output (timer RD_1) (initial value)<br>110: Setting prohibited<br>111: Setting prohibited | R/W |

Note: PMCR92 is not available on the H8S/20103 and H8S/20115 Groups.

chanhvinh.com



### (c) Port 9 Peripheral Function Mapping Register 3 (PMCR93)

Address: H'FF0062

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | P95MD[2:0] | | | — | P94MD[2:0] | | |
| Value after reset: | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 6 to 4 | P95MD[2:0] | P95 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ5}}$ input<br>010: Setting prohibited<br>011: SCK3 input/output (SCI3_1)<br>100: TRAIO input/output (timer RA)<br>101: FTIOB3 input/output (timer RD_1) (initial value)<br>110: SCK3_3 input/output (SCI3_3)<br>111: Setting prohibited | R/W |
| 3 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 2 to 0 | P94MD[2:0] | P94 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ4}}$ input<br>010: Setting prohibited<br>011: Setting prohibited<br>100: TRGB input (timer RB)<br>101: FTIOA3 input/output (timer RD_1) (initial value)<br>110: Setting prohibited<br>111: Setting prohibited | R/W |

Note: PMCR93 is not available on the H8S/20103 and H8S/20115 Groups.

chanhvinh.com



### (d) Port 9 Peripheral Function Mapping Register 4 (PMCR94)

Address: H'FF0063

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | P97MD[2:0] | | | — | P96MD[2:0] | | |
| Value after reset: | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 6 to 4 | P97MD[2:0] | P97 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ7}}$ input<br>010: Setting prohibited<br>011: TXD output (SCI3_1)<br>100: TREO output (timer RE)<br>101: FTIOD3 input/output (timer RD_1) (initial value)<br>110: TXD_3 output (SCI3_3)<br>111: Setting prohibited | R/W |
| 3 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 2 to 0 | P96MD[2:0] | P96 function select | 000: Setting prohibited<br>001: $\overline{\text{IRQ6}}$ input<br>010: Setting prohibited<br>011: RXD input (SCI3_1)<br>100: TRBO output (timer RB)<br>101: FTIOC3 input/output (timer RD_1) (initial value)<br>110: RXD_3 input (SCI3_3)<br>111: Setting prohibited | R/W |

Note: PMCR94 is not available on the H8S/20103 and H8S/20115 Groups.

chanhvinh.com



## (3) Port A

### (a) Port A Peripheral Function Mapping Register 3 (PMCRA3)

Address: H'FF0066

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | PA5MD[2:0] | | | — | PA4MD[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 6 to 4 | PA5MD[2:0] | PA5 function select | 000: No function selected (initial value)<br>001: $\overline{\text{IRQ5}}$ input<br>010: Setting prohibited<br>011: SCK3 input/output (SCI3_1)<br>100: TRAIO input/output (timer RA)<br>101: FTIOB3 input/output (timer RD_1)*<br>110: SCK3_3 input/output (SCI3_3)<br>111: Setting prohibited | R/W |
| 3 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 2 to 0 | PA4MD[2:0] | PA4 function select | 000: No function selected (initial value)<br>001: $\overline{\text{IRQ4}}$ input<br>010: Setting prohibited<br>011: Setting prohibited<br>100: TRGB input (timer RB)<br>101: FTIOA3 input/output (timer RD_1)*<br>110: Setting prohibited<br>111: Setting prohibited | R/W |

Notes: PMCRA3 is not available on the H8S/20223 and H8S/20235 Groups.

* This function cannot be selected for the H8S/20103 and H8S/20115 Groups.

chanhvinh.com



### (b) Port A Peripheral Function Mapping Register 4 (PMCRA4)

Address: H'FF0067

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | PA7MD[2:0] | | | — | PA6MD[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 6 to 4 | PA7MD[2:0] | PA7 function select | 000: No function selected (initial value)<br>001: $\overline{\text{IRQ7}}$ input<br>010: Setting prohibited<br>011: TXD output (SCI3_1)<br>100: TREO output (timer RE)<br>101: FTIOD3 input/output (timer RD_1)*<br>110: TXD_3 output (SCI3_3)<br>111: Setting prohibited | R/W |
| 3 | — | Reserved | This bit is always read as 0. The write value should always be 0. | — |
| 2 to 0 | PA6MD[2:0] | PA6 function select | 000: No function selected (initial value)<br>001: $\overline{\text{IRQ6}}$ input<br>010: Setting prohibited<br>011: RXD input (SCI3_1)<br>100: TRBO output (timer RB)<br>101: FTIOC3 input/output (timer RD_1)*<br>110: RXD_3 input (SCI3_3)<br>111: Setting prohibited | R/W |

Notes: PMCRA4 is not available on the H8S/20223 and H8S/20235 Groups.

* This function cannot be selected for the H8S/20103 and H8S/20115 Groups.

chanhvinh.com



## 9.2 Usage Notes

### 9.2.1 Procedures for Setting Multiplexed Port Functions

Use the following procedures to set a function for a multiplexed port.

1. Clear the relevant port mode register (PMR) bit to 0 to select the general input function.
2. Set PMCWPR to enable writing to the relevant peripheral function mapping register (PMCR).
3. Select a function using the peripheral function mapping register (PMCR).
4. Set the PMCRWE bit in PMCWPR to 0 to disable writing to PMCR.
5. Set the PMR bit to 1 as necessary to activate the selected multiplexed function.

### 9.2.2 Notes on Setting PMC Registers

1. A function of a multiplexed port should be set when the relevant PMR bit is 0. If a function is set when PMR is 1, an unintended edge may be input for the input function or unintended pulses may be output for the output function.
2. Only the functions that can be selected by PMCR should be set. If the other functions are set, operation cannot be guaranteed.
3. The same function must not be assigned to multiple pins by the PMC.
4. Port A also has an analog input function for the A/D converter. When port A is used as analog input, the relevant bit in PMRA should be set to 0 to select general I/O.

chanhvinh.com



# Section 10 I/O Ports

The H8S/20103 and H8S/20115 Groups have fifty-five general I/O ports, and the H8S/20223, H8S/20203, H8S/20215, and H8S/20235 Groups each have sixty-nine general I/O ports. The general I/O ports are divided into three groups: the digital I/O ports that can also be used as I/O pins of the on-chip peripheral modules or external interrupt input pins, the ports that can also be used as analog input ports, and the ports that can also be used as external oscillation pins. Although all the ports are set as general input ports immediately after a reset, the pin functions can be selected by setting the appropriate register.

Pin functions of the digital I/O ports are selected by the peripheral function mapping controller (PMC). For details, see section 9, Peripheral I/O Mapping Controller. All pins of general I/O ports can be set as high-power ports. For the permissible total output current, see section 28, Electrical Characteristics.

## 10.1 Port 1

Figure 10.1 shows the pin configuration of port 1.

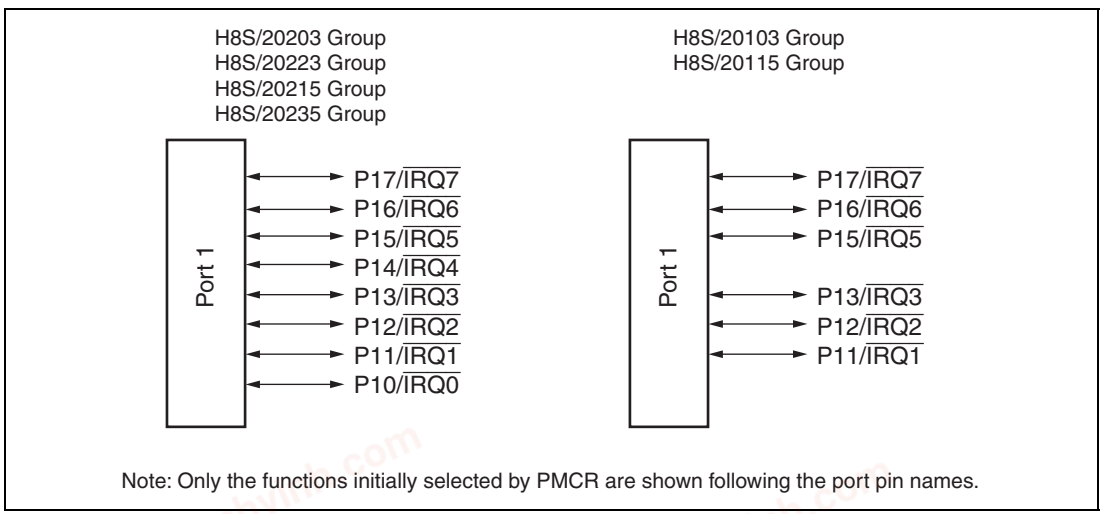



**Figure 10.1 Port 1 Pin Configuration**

chanhvinh.com



Port 1 has the following registers.

- Port mode register 1 (PMR1)
- Port control register 1 (PCR1)
- Port data register 1 (PDR1)
- Port pull-up control register 1 (PUCR1)
- Port drive control register 1 (PDVR1)

### 10.1.1 Port Mode Register 1 (PMR1)

Address: H'FF0000

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PMR17 | PMR16 | PMR15 | PMR14 | PMR13 | PMR12 | PMR11 | PMR10 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PMR17 | Port 17 mode | 0: General I/O port | R/W |
| 6 | PMR16 | Port 16 mode | 1: The function selected by the peripheral I/O mapping controller (PMC). | R/W |
| 5 | PMR15 | Port 15 mode | | R/W |
| 4 | PMR14 | Port 14 mode | PMR1 is a register that selects the function of the multiplexed pins: general I/O function or the function selected by the PMC. | R/W |
| 3 | PMR13 | Port 13 mode | | R/W |
| 2 | PMR12 | Port 12 mode | In the H8S/20103 and H8S/20115 Groups, bits PMR14 and PMR10 are reserved. Only 0 should be written to these bits. | R/W |
| 1 | PMR11 | Port 11 mode | | R/W |
| 0 | PMR10 | Port 10 mode | | R/W |

chanhvinh.com



### 10.1.2 Port Control Register 1 (PCR1)

Address: H'FFFFF0

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PCR17 | PCR16 | PCR15 | PCR14 | PCR13 | PCR12 | PCR11 | PCR10 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PCR17 | Port 17 control | 0: When the corresponding pin is designated as a general I/O port, the pin functions as an input port.<br>1: When the corresponding pin is designated as a general I/O port, the pin functions as an output port.<br>When the corresponding pin is designated in PMR1 as a general I/O pin, setting a PCR bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port.<br>In the H8S/20103 and H8S/20115 Groups, bits PCR14 and PCR10 are reserved. Only 0 should be written to these bits. | R/W |
| 6 | PCR16 | Port 16 control | | R/W |
| 5 | PCR15 | Port 15 control | | R/W |
| 4 | PCR14 | Port 14 control | | R/W |
| 3 | PCR13 | Port 13 control | | R/W |
| 2 | PCR12 | Port 12 control | | R/W |
| 1 | PCR11 | Port 11 control | | R/W |
| 0 | PCR10 | Port 10 control | | R/W |

chanhvinh.com



### 10.1.3 Port Data Register 1 (PDR1)

Address: H'FFFFE0

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PDR17 | PDR16 | PDR15 | PDR14 | PDR13 | PDR12 | PDR11 | PDR10 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PDR17 | Port 17 data | 0: Low level | R/W |
| 6 | PDR16 | Port 16 data | 1: High level | R/W |
| 5 | PDR15 | Port 15 data | PDR1 is a register that stores output data for port 1 pins. When PCR1 bits are set to 1, the values stored in PDR1 are output. | R/W |
| 4 | PDR14 | Port 14 data | | R/W |
| 3 | PDR13 | Port 13 data | When PDR1 is read while PCR1 bits are set to 1, the values stored in PDR1 are read. If PDR1 is read while PCR1 bits are cleared to 0, the pin states are read regardless of the value stored in PDR1. | R/W |
| 2 | PDR12 | Port 12 data | | R/W |
| 1 | PDR11 | Port 11 data | | R/W |
| 0 | PDR10 | Port 10 data | In the H8S/20103 and H8S/20115 Groups, bits PDR14 and PDR10 are reserved. Only 0 should be written to these bits. | R/W |

chanhvinh.com



### 10.1.4 Port Pull-Up Control Register 1 (PUCR1)

Address: H'FF0010

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PUCR17 | PUCR16 | PUCR15 | PUCR14 | PUCR13 | PUCR12 | PUCR11 | PUCR10 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PUCR17 | Port 17 pull-up control | 0: The pull-up MOS of corresponding pin is disabled.<br>1: The pull-up MOS of corresponding pin is enabled.<br>This function is valid only for the pin set as general input, and for the input pin with a multiplexed function.<br>In the H8S/20103 and H8S/20115 Groups, bits PUCR14 and PUCR10 are reserved. Only 0 should be written to these bits. | R/W |
| 6 | PUCR16 | Port 16 pull-up control | | R/W |
| 5 | PUCR15 | Port 15 pull-up control | | R/W |
| 4 | PUCR14 | Port 14 pull-up control | | R/W |
| 3 | PUCR13 | Port 13 pull-up control | | R/W |
| 2 | PUCR12 | Port 12 pull-up control | | R/W |
| 1 | PUCR11 | Port 11 pull-up control | | R/W |
| 0 | PUCR10 | Port 10 pull-up control | | R/W |

chanhvinh.com



### 10.1.5 Port Drive Control Register 1 (PDVR1)

Address: H'FF0030

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PDVR17 | PDVR16 | PDVR15 | PDVR14 | PDVR13 | PDVR12 | PDVR11 | PDVR10 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PDVR17 | Port 17 drive control | 0: Normal output<br>1: High-current drive output | R/W |
| 6 | PDVR16 | Port 16 drive control | PDVR1 is a register that controls drive capability of the output pins in a bit unit. | R/W |
| 5 | PDVR15 | Port 15 drive control | In the H8S/20103 and H8S/20115 Groups, bits PDVR14 and PDVR10 are reserved. Only 0 should be written to these bits. | R/W |
| 4 | PDVR14 | Port 14 drive control | | R/W |
| 3 | PDVR13 | Port 13 drive control | | R/W |
| 2 | PDVR12 | Port 12 drive control | | R/W |
| 1 | PDVR11 | Port 11 drive control | | R/W |
| 0 | PDVR10 | Port 10 drive control | | R/W |

chanhvinh.com



## 10.2 Port 2

Figure 10.2 shows the pin configuration of port 2.

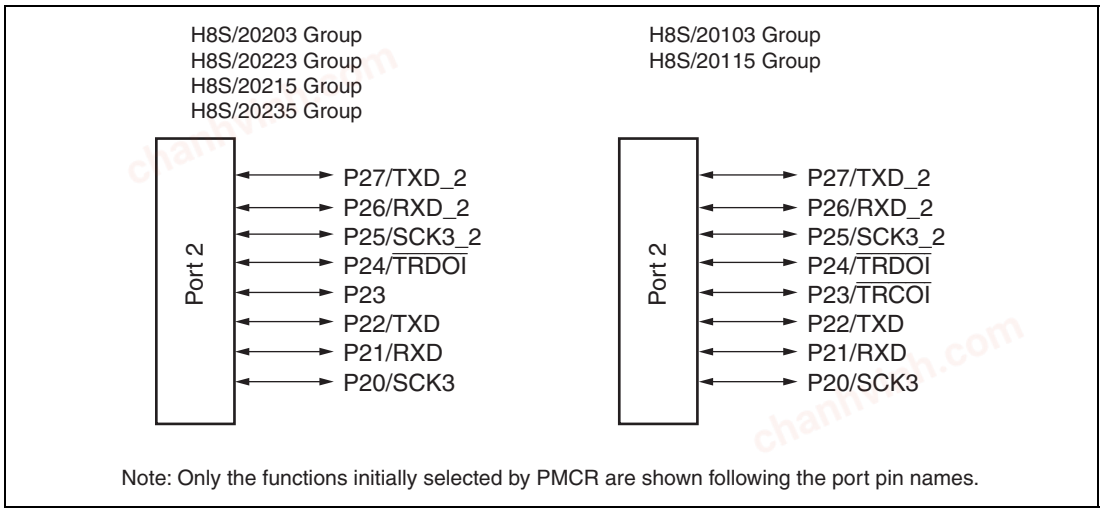



**Figure 10.2 Port 2 Pin Configuration**

Port 2 has the following registers.

- Port mode register 2 (PMR2)
- Port control register 2 (PCR2)
- Port data register 2 (PDR2)
- Port pull-up control register 2 (PUCR2)
- Port drive control register 2 (PDVR2)

chanhvinh.com



### 10.2.1 Port Mode Register 2 (PMR2)

Address: H'FF0001

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PMR27 | PMR26 | PMR25 | PMR24 | PMR23 | PMR22 | PMR21 | PMR20 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PMR27 | Port 27 mode | 0: General I/O port<br>1: The function selected by the peripheral function mapping controller (PMC).<br>PMR2 is a register that selects the function of the multiplexed pins: general I/O function or the function selected by the PMC. | R/W |
| 6 | PMR26 | Port 26 mode | | R/W |
| 5 | PMR25 | Port 25 mode | | R/W |
| 4 | PMR24 | Port 24 mode | | R/W |
| 3 | PMR23 | Port 23 mode | | R/W |
| 2 | PMR22 | Port 22 mode | | R/W |
| 1 | PMR21 | Port 21 mode | | R/W |
| 0 | PMR20 | Port 20 mode | | R/W |

chanhvinh.com



### 10.2.2 Port Control Register 2 (PCR2)

Address: H'FFFFF1

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PCR27 | PCR26 | PCR25 | PCR24 | PCR23 | PCR22 | PCR21 | PCR20 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PCR27 | Port 27 control | 0: When the corresponding pin is designated as a general I/O port, the pin functions as an input port.<br>1: When the corresponding pin is designated as a general I/O port, the pin functions as an output port.<br>PCR2 is a register that selects inputs/outputs in bit units for pins to be used as general I/O ports of port 2. | R/W |
| 6 | PCR26 | Port 26 control | | R/W |
| 5 | PCR25 | Port 25 control | | R/W |
| 4 | PCR24 | Port 24 control | | R/W |
| 3 | PCR23 | Port 23 control | | R/W |
| 2 | PCR22 | Port 22 control | | R/W |
| 1 | PCR21 | Port 21 control | | R/W |
| 0 | PCR20 | Port 20 control | | R/W |

- PCR27 bit to PCR20 bit (port 27 to 20 control)
  When the corresponding pin is designated in PMR2 as a general I/O pin, setting a PCR2 bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port.

chanhvinh.com



### 10.2.3 Port Data Register 2 (PDR2)

Address: H'FFFFE1

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PDR27 | PDR26 | PDR25 | PDR24 | PDR23 | PDR22 | PDR21 | PDR20 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PDR27 | Port 27 data | 0: Low level | R/W |
| 6 | PDR26 | Port 26 data | 1: High level | R/W |
| 5 | PDR25 | Port 25 data | PDR2 is a register that stores output data for port 2 pins. When PCR2 bits are set to 1, the values stored in PDR2 are output. | R/W |
| 4 | PDR24 | Port 24 data | | R/W |
| 3 | PDR23 | Port 23 data | When PDR2 is read while PCR2 bits are set to 1, the values stored in PDR2 are read. If PDR2 is read while PCR2 bits are cleared to 0, the pin states are read regardless of the value stored in PDR2. | R/W |
| 2 | PDR22 | Port 22 data | | R/W |
| 1 | PDR21 | Port 21 data | | R/W |
| 0 | PDR20 | Port 20 data | | R/W |

chanhvinh.com



### 10.2.4 Port Pull-Up Control Register 2 (PUCR2)

Address: H'FF0011

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PUCR27 | PUCR26 | PUCR25 | PUCR24 | PUCR23 | PUCR22 | PUCR21 | PUCR20 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PUCR27 | Port 27 pull-up control | 0: The pull-up MOS of corresponding pin is disabled.<br>1: The pull-up MOS of corresponding pin is enabled.<br>PUCR2 is a register that controls the pull-up MOS in bit units of the pins set as the input ports. | R/W |
| 6 | PUCR26 | Port 26 pull-up control | | R/W |
| 5 | PUCR25 | Port 25 pull-up control | | R/W |
| 4 | PUCR24 | Port 24 pull-up control | | R/W |
| 3 | PUCR23 | Port 23 pull-up control | | R/W |
| 2 | PUCR22 | Port 22 pull-up control | | R/W |
| 1 | PUCR21 | Port 21 pull-up control | | R/W |
| 0 | PUCR20 | Port 20 pull-up control | | R/W |

- PUCR27 bit to PUCR20 bit (port 27 to 20 pull-up control)

  This function is valid only for the pin set as general input, and for the input pin with a function selected by the PMC.

chanhvinh.com



### 10.2.5 Port Drive Control Register 2 (PDVR2)

Address: H'FF0031

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PDVR27 | PDVR26 | PDVR25 | PDVR24 | PDVR23 | PDVR22 | PDVR21 | PDVR20 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PDVR27 | Port 27 drive control | 0: Normal output<br>1: High-current drive output<br>PDVR2 is a register that controls drive capability of the output pins in a bit unit. | R/W |
| 6 | PDVR26 | Port 26 drive control | | R/W |
| 5 | PDVR25 | Port 25 drive control | | R/W |
| 4 | PDVR24 | Port 24 drive control | | R/W |
| 3 | PDVR23 | Port 23 drive control | | R/W |
| 2 | PDVR22 | Port 22 drive control | | R/W |
| 1 | PDVR21 | Port 21 drive control | | R/W |
| 0 | PDVR20 | Port 20 drive control | | R/W |

chanhvinh.com



## 10.3 Port 3

Figure 10.3 shows the pin configuration of port 3.

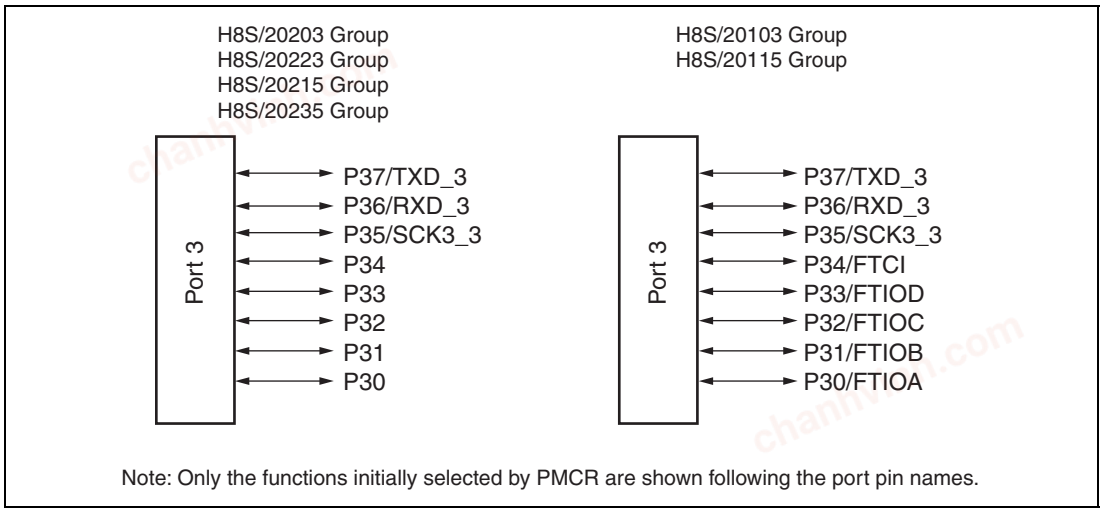



**Figure 10.3 Port 3 Pin Configuration**

Port 3 has the following registers.

- Port mode register 3 (PMR3)
- Port control register 3 (PCR3)
- Port data register 3 (PDR3)
- Port pull-up control register 3 (PUCR3)
- Port drive control register 3 (PDVR3)

chanhvinh.com



### 10.3.1 Port Mode Register 3 (PMR3)

Address: H'FF0002

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PMR37 | PMR36 | PMR35 | PMR34 | PMR33 | PMR32 | PMR31 | PMR30 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PMR37 | Port 37 mode | 0: General I/O port<br>1: The function selected by the peripheral function mapping controller (PMC).<br>PMR3 is a register that selects the function of the multiplexed pins: general I/O function or the function selected by the PMC. | R/W |
| 6 | PMR36 | Port 36 mode | | R/W |
| 5 | PMR35 | Port 35 mode | | R/W |
| 4 | PMR34 | Port 34 mode | | R/W |
| 3 | PMR33 | Port 33 mode | | R/W |
| 2 | PMR32 | Port 32 mode | | R/W |
| 1 | PMR31 | Port 31 mode | | R/W |
| 0 | PMR30 | Port 30 mode | | R/W |

chanhvinh.com



### 10.3.2 Port Control Register 3 (PCR3)

Address: H'FFFFF2

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PCR37 | PCR36 | PCR35 | PCR34 | PCR33 | PCR32 | PCR31 | PCR30 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PCR37 | Port 37 control | 0: When the corresponding pin is designated as a general I/O port, the pin functions as an input port.<br>1: When the corresponding pin is designated as a general I/O port, the pin functions as an output port.<br>PCR3 is a register that selects inputs/outputs in bit units for pins to be used as general I/O ports of port 3. | R/W |
| 6 | PCR36 | Port 36 control | | R/W |
| 5 | PCR35 | Port 35 control | | R/W |
| 4 | PCR34 | Port 34 control | | R/W |
| 3 | PCR33 | Port 33 control | | R/W |
| 2 | PCR32 | Port 32 control | | R/W |
| 1 | PCR31 | Port 31 control | | R/W |
| 0 | PCR30 | Port 30 control | | R/W |

- PCR37 bit to PCR30 bit (port 37 to 30 control)
  When the corresponding pin is designated in PMR3 as a general I/O pin, setting a PCR3 bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port.

chanhvinh.com



### 10.3.3 Port Data Register 3 (PDR3)

Address: H'FFFFE2

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PDR37 | PDR36 | PDR35 | PDR34 | PDR33 | PDR32 | PDR31 | PDR30 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PDR37 | Port 37 data | 0: Low level<br>1: High level<br>PDR3 is a register that stores output data for port 3 pins. When PCR3 bits are set to 1, the values stored in PDR3 are output.<br>When PDR3 is read while PCR3 bits are set to 1, the values stored in PDR3 are read. If PDR3 is read while PCR3 bits are cleared to 0, the pin states are read regardless of the value stored in PDR3. | R/W |
| 6 | PDR36 | Port 36 data | | R/W |
| 5 | PDR35 | Port 35 data | | R/W |
| 4 | PDR34 | Port 34 data | | R/W |
| 3 | PDR33 | Port 33 data | | R/W |
| 2 | PDR32 | Port 32 data | | R/W |
| 1 | PDR31 | Port 31 data | | R/W |
| 0 | PDR30 | Port 30 data | | R/W |

chanhvinh.com



### 10.3.4 Port Pull-Up Control Register 3 (PUCR3)

Address: H'FF0012

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PUCR37 | PUCR36 | PUCR35 | PUCR34 | PUCR33 | PUCR32 | PUCR31 | PUCR30 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PUCR37 | Port 37 pull-up control | 0: The pull-up MOS of corresponding pin is disabled.<br>1: The pull-up MOS of corresponding pin is enabled.<br>PUCR3 is a register that controls the pull-up MOS in bit units of the pins set as the input ports. | R/W |
| 6 | PUCR36 | Port 36 pull-up control | | R/W |
| 5 | PUCR35 | Port 35 pull-up control | | R/W |
| 4 | PUCR34 | Port 34 pull-up control | | R/W |
| 3 | PUCR33 | Port 33 pull-up control | | R/W |
| 2 | PUCR32 | Port 32 pull-up control | | R/W |
| 1 | PUCR31 | Port 31 pull-up control | | R/W |
| 0 | PUCR30 | Port 30 pull-up control | | R/W |

- PUCR37 bit to PUCR30 bit (port 37 to 30 pull-up control)

  This function is valid only for the pin set as general input, and for the input pin with a function selected by the PMC.

chanhvinh.com



### 10.3.5 Port Drive Control Register 3 (PDVR3)

Address: H'FF0032

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PDVR37 | PDVR36 | PDVR35 | PDVR34 | PDVR33 | PDVR32 | PDVR31 | PDVR30 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PDVR37 | Port 37 drive control | 0: Normal output<br>1: High-current drive output<br>PDVR3 is a register that controls drive capability of the output pins in a bit unit. | R/W |
| 6 | PDVR36 | Port 36 drive control | | R/W |
| 5 | PDVR35 | Port 35 drive control | | R/W |
| 4 | PDVR34 | Port 34 drive control | | R/W |
| 3 | PDVR33 | Port 33 drive control | | R/W |
| 2 | PDVR32 | Port 32 drive control | | R/W |
| 1 | PDVR31 | Port 31 drive control | | R/W |
| 0 | PDVR30 | Port 30 drive control | | R/W |

chanhvinh.com



## 10.4 Port 5

Figure 10.4 shows the pin configuration of port 5.

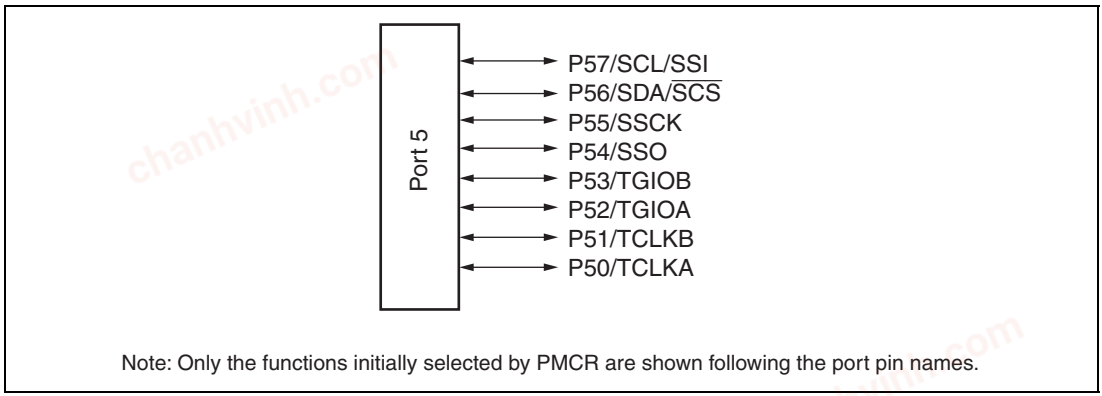



**Figure 10.4 Port 5 Pin Configuration**

Port 5 has the following registers.

- Port mode register 5 (PMR5)
- Port control register 5 (PCR5)
- Port data register 5 (PDR5)
- Port pull-up control register 5 (PUCR5)
- Port drive control register 5 (PDVR5)

chanhvinh.com



### 10.4.1 Port Mode Register 5 (PMR5)

Address: H'FF0004

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PMR57 | PMR56 | PMR55 | PMR54 | PMR53 | PMR52 | PMR51 | PMR50 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PMR57 | Port 57 mode | 0: General I/O port<br>1: The function selected by the peripheral function mapping controller (PMC).<br>PMR5 is a register that selects the function of the multiplexed pins: general I/O function or the function selected by the PMC. | R/W |
| 6 | PMR56 | Port 56 mode | | R/W |
| 5 | PMR55 | Port 55 mode | | R/W |
| 4 | PMR54 | Port 54 mode | | R/W |
| 3 | PMR53 | Port 53 mode | | R/W |
| 2 | PMR52 | Port 52 mode | | R/W |
| 1 | PMR51 | Port 51 mode | | R/W |
| 0 | PMR50 | Port 50 mode | | R/W |

chanhvinh.com



### 10.4.2 Port Control Register 5 (PCR5)

Address: H'FFFFF4

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PCR57 | PCR56 | PCR55 | PCR54 | PCR53 | PCR52 | PCR51 | PCR50 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PCR57 | Port 57 control | 0: When the corresponding pin is designated as a general I/O port, the pin functions as an input port.<br>1: When the corresponding pin is designated as a general I/O port, the pin functions as an output port.<br>PCR5 is a register that selects inputs/outputs in bit units for pins to be used as general I/O ports of port 5. | R/W |
| 6 | PCR56 | Port 56 control | | R/W |
| 5 | PCR55 | Port 55 control | | R/W |
| 4 | PCR54 | Port 54 control | | R/W |
| 3 | PCR53 | Port 53 control | | R/W |
| 2 | PCR52 | Port 52 control | | R/W |
| 1 | PCR51 | Port 51 control | | R/W |
| 0 | PCR50 | Port 50 control | | R/W |

- PCR57 bit to PCR50 bit (port 57 to 50 control)
  When the corresponding pin is designated in PMR5 as a general I/O pin, setting a PCR5 bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port.

chanhvinh.com



### 10.4.3 Port Data Register 5 (PDR5)

Address: H'FFFFE4

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PDR57 | PDR56 | PDR55 | PDR54 | PDR53 | PDR52 | PDR51 | PDR50 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PDR57 | Port 57 data | 0: Low level | R/W |
| 6 | PDR56 | Port 56 data | 1: High level | R/W |
| 5 | PDR55 | Port 55 data | PDR5 is a register that stores output data for port 5 pins. When PCR5 bits are set to 1, the values stored in PDR5 are output. | R/W |
| 4 | PDR54 | Port 54 data | | R/W |
| 3 | PDR53 | Port 53 data | When PDR5 is read while PCR5 bits are set to 1, the values stored in PDR5 are read. If PDR5 is read while PCR5 bits are cleared to 0, the pin states are read regardless of the value stored in PDR5. | R/W |
| 2 | PDR52 | Port 52 data | | R/W |
| 1 | PDR51 | Port 51 data | | R/W |
| 0 | PDR50 | Port 50 data | | R/W |

chanhvinh.com



### 10.4.4 Port Pull-Up Control Register 5 (PUCR5)

Address: H'FF0014

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | PUCR55 | PUCR54 | PUCR53 | PUCR52 | PUCR51 | PUCR50 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | These bits are read as 0. The write value should be 0. | — |
| 6 | — | Reserved | | — |
| 5 | PUCR55 | Port 55 pull-up control | 0: The pull-up MOS of corresponding pin is disabled.<br>1: The pull-up MOS of corresponding pin is enabled.<br>PUCR5 is a register that controls the pull-up MOS in bit units of the pins set as the input ports. | R/W |
| 4 | PUCR54 | Port 54 pull-up control | | R/W |
| 3 | PUCR53 | Port 53 pull-up control | | R/W |
| 2 | PUCR52 | Port 52 pull-up control | | R/W |
| 1 | PUCR51 | Port 51 pull-up control | | R/W |
| 0 | PUCR50 | Port 50 pull-up control | | R/W |

- PUCR55 bit to PUCR50 bit (port 55 to 50 pull-up control)

  This function is valid only for the pin set as general input, and for the input pin with a function selected by the PMC.

chanhvinh.com



### 10.4.5 Port Drive Control Register 5 (PDVR5)

Address: H'FF0034

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | PDVR55 | PDVR54 | PDVR53 | PDVR52 | PDVR51 | PDVR50 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is read as 0. The write value should be 0. | — |
| 6 | — | Reserved | | — |
| 5 | PDVR55 | Port 55 drive control | 0: Normal output<br>1: High-current drive output<br>PDVR5 is a register that controls drive capability of the output pins in a bit unit.<br>When pins P56 and P57 are set as general output, they function as NMOS push-pull output and thus drive capability cannot be selected for them. | R/W |
| 4 | PDVR54 | Port 54 drive control | | R/W |
| 3 | PDVR53 | Port 53 drive control | | R/W |
| 2 | PDVR52 | Port 52 drive control | | R/W |
| 1 | PDVR51 | Port 51 drive control | | R/W |
| 0 | PDVR50 | Port 50 drive control | | R/W |

Note: When pins P56 and P57 are set as general output, they function as NMOS push-pull output, and have characteristics different from those of other CMOS outputs. When set as SDA and SCL of IIC2, they function as NMOS open-drain output. For details, see section 28, Electrical Characteristics.

chanhvinh.com



## 10.5 Port 6

Figure 10.5 shows the pin configuration of port 6.

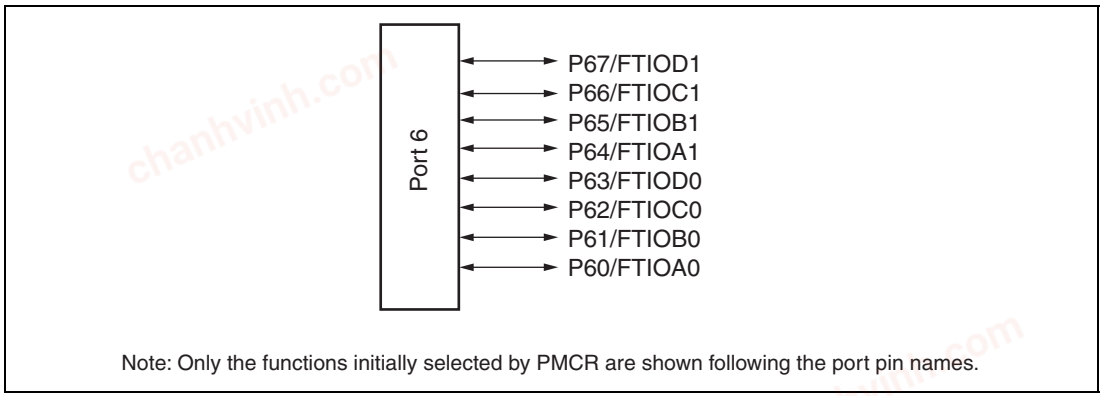



**Figure 10.5 Port 6 Pin Configuration**

Port 6 has the following registers.

- Port mode register 6 (PMR6)
- Port control register 6 (PCR6)
- Port data register 6 (PDR6)
- Port pull-up control register 6 (PUCR6)
- Port drive control register 6 (PDVR6)

chanhvinh.com



### 10.5.1 Port Mode Register 6 (PMR6)

Address: H'FF0005

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PMR67 | PMR66 | PMR65 | PMR64 | PMR63 | PMR62 | PMR61 | PMR60 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PMR67 | Port 67 mode | 0: General I/O port<br>1: The function selected by the peripheral function mapping controller (PMC).<br>PMR6 is a register that selects the function of the multiplexed pins: general I/O function or the function selected by the PMC. | R/W |
| 6 | PMR66 | Port 66 mode | | R/W |
| 5 | PMR65 | Port 65 mode | | R/W |
| 4 | PMR64 | Port 64 mode | | R/W |
| 3 | PMR63 | Port 63 mode | | R/W |
| 2 | PMR62 | Port 62 mode | | R/W |
| 1 | PMR61 | Port 61 mode | | R/W |
| 0 | PMR60 | Port 60 mode | | R/W |

chanhvinh.com



### 10.5.2 Port Control Register 6 (PCR6)

Address: H'FFFFF5

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PCR67 | PCR66 | PCR65 | PCR64 | PCR63 | PCR62 | PCR61 | PCR60 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PCR67 | Port 67 control | 0: When the corresponding pin is designated as a general I/O port, the pin functions as an input port.<br>1: When the corresponding pin is designated as a general I/O port, the pin functions as an output port.<br>PCR6 is a register that selects inputs/outputs in bit units for pins to be used as general I/O ports of port 6. | R/W |
| 6 | PCR66 | Port 66 control | | R/W |
| 5 | PCR65 | Port 65 control | | R/W |
| 4 | PCR64 | Port 64 control | | R/W |
| 3 | PCR63 | Port 63 control | | R/W |
| 2 | PCR62 | Port 62 control | | R/W |
| 1 | PCR61 | Port 61 control | | R/W |
| 0 | PCR60 | Port 60 control | | R/W |

- PCR67 bit to PCR60 bit (port 67 to 60 control)
  When the corresponding pin is designated in PMR6 as a general I/O pin, setting a PCR6 bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port.

chanhvinh.com



### 10.5.3 Port Data Register 6 (PDR6)

Address: H'FFFFE5

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PDR67 | PDR66 | PDR65 | PDR64 | PDR63 | PDR62 | PDR61 | PDR60 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PDR67 | Port 67 data | 0: Low level<br>1: High level<br>PDR6 is a register that stores output data for port 6 pins. When PCR6 bits are set to 1, the values stored in PDR6 are output.<br>When PDR6 is read while PCR6 bits are set to 1, the values stored in PDR6 are read. If PDR6 is read while PCR6 bits are cleared to 0, the pin states are read regardless of the value stored in PDR6. | R/W |
| 6 | PDR66 | Port 66 data | | R/W |
| 5 | PDR65 | Port 65 data | | R/W |
| 4 | PDR64 | Port 64 data | | R/W |
| 3 | PDR63 | Port 63 data | | R/W |
| 2 | PDR62 | Port 62 data | | R/W |
| 1 | PDR61 | Port 61 data | | R/W |
| 0 | PDR60 | Port 60 data | | R/W |

chanhvinh.com



### 10.5.4 Port Pull-Up Control Register 6 (PUCR6)

Address: H'FF0015

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PUCR67 | PUCR66 | PUCR65 | PUCR64 | PUCR63 | PUCR62 | PUCR61 | PUCR60 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PUCR67 | Port 67 pull-up control | 0: The pull-up MOS of corresponding pin is disabled.<br>1: The pull-up MOS of corresponding pin is enabled.<br>PUCR6 is a register that controls the pull-up MOS in bit units of the pins set as the input ports. | R/W |
| 6 | PUCR66 | Port 66 pull-up control | | R/W |
| 5 | PUCR65 | Port 65 pull-up control | | R/W |
| 4 | PUCR64 | Port 64 pull-up control | | R/W |
| 3 | PUCR63 | Port 63 pull-up control | | R/W |
| 2 | PUCR62 | Port 62 pull-up control | | R/W |
| 1 | PUCR61 | Port 61 pull-up control | | R/W |
| 0 | PUCR60 | Port 60 pull-up control | | R/W |

- PUCR67 bit to PUCR60 bit (port 67 to 60 pull-up control)

  This function is valid only for the pin set as general input, and for the input pin with a function selected by the PMC.

chanhvinh.com



### 10.5.5 Port Drive Control Register 6 (PDVR6)

Address: H'FF0035

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PDVR67 | PDVR66 | PDVR65 | PDVR64 | PDVR63 | PDVR62 | PDVR61 | PDVR60 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

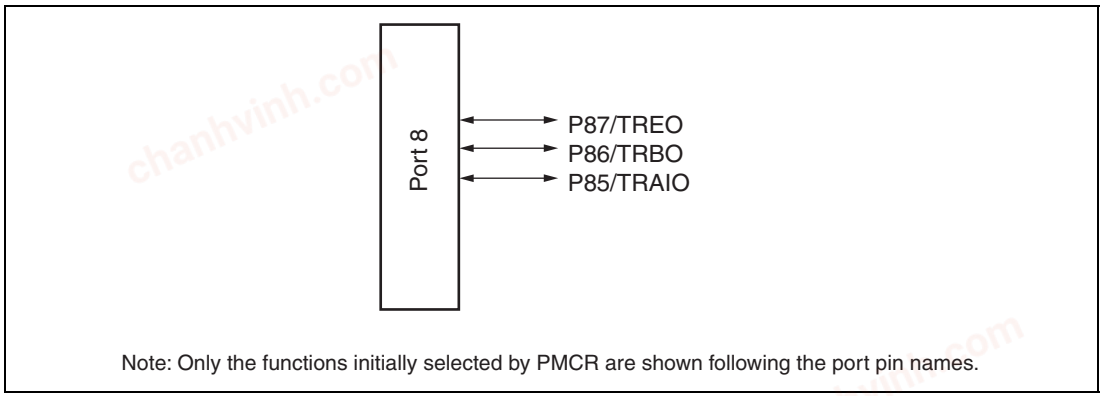
| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PDVR67 | Port 67 drive control | 0: Normal output<br>1: High-current drive output<br>PDVR6 is a register that controls drive capability of the output pins in a bit unit. | R/W |
| 6 | PDVR66 | Port 66 drive control | | R/W |
| 5 | PDVR65 | Port 65 drive control | | R/W |
| 4 | PDVR64 | Port 64 drive control | | R/W |
| 3 | PDVR63 | Port 63 drive control | | R/W |
| 2 | PDVR62 | Port 62 drive control | | R/W |
| 1 | PDVR61 | Port 61 drive control | | R/W |
| 0 | PDVR60 | Port 60 drive control | | R/W |

chanhvinh.com



## 10.6 Port 8

Figure 10.6 shows the pin configuration of port 8.




Note: Only the functions initially selected by PMCR are shown following the port pin names.

**Figure 10.6 Port 8 Pin Configuration**

Port 8 has the following registers.

- Port mode register 8 (PMR8)
- Port control register 8 (PCR8)
- Port data register 8 (PDR8)
- Port pull-up control register 8 (PUCR8)
- Port drive control register 8 (PDVR8)

chanhvinh.com



### 10.6.1 Port Mode Register 8 (PMR8)

Address: H'FF0005

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PMR67 | PMR66 | PMR65 | PMR64 | PMR63 | PMR62 | PMR61 | PMR60 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PMR87 | Port 87 mode | 0: General I/O port<br>1: The function selected by the peripheral function mapping controller (PMC).<br>PMR8 is a register that selects the function of the multiplexed pins: general I/O function or the function selected by the PMC. | R/W |
| 6 | PMR86 | Port 86 mode | | R/W |
| 5 | PMR85 | Port 85 mode | | R/W |
| 4 to 0 | — | Reserved | These bits are read as 0. The write value should be 0. | — |

chanhvinh.com



### 10.6.2 Port Control Register 8 (PCR8)

Address: H'FFFFF7

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PCR87 | PCR86 | PCR85 | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PCR87 | Port 87 control | 0: When the corresponding pin is designated as a general I/O port, the pin functions as an input port.<br>1: When the corresponding pin is designated as a general I/O port, the pin functions as an output port.<br>PCR8 is a register that selects inputs/outputs in bit units for pins to be used as general I/O ports of port 8. | R/W |
| 6 | PCR86 | Port 86 control | | R/W |
| 5 | PCR85 | Port 85 control | | R/W |
| 4 to 0 | — | Reserved | These bits are read as 0. The write value should be 0. | — |

- PCR87 bit to PCR85 bit (port 87 to 85 control)

  When the corresponding pin is designated in PMR8 as a general I/O pin, setting a PCR8 bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port.

chanhvinh.com



### 10.6.3 Port Data Register 8 (PDR8)

Address: H'FFFFE7

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PDR87 | PDR86 | PDR85 | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PDR87 | Port 87 data | 0: Low level | R/W |
| 6 | PDR86 | Port 86 data | 1: High level | R/W |
| 5 | PDR85 | Port 85 data | PDR8 is a register that stores output data for port 8 pins. When PCR8 bits are set to 1, the values stored in PDR8 are output.<br>When PDR8 is read while PCR8 bits are set to 1, the values stored in PDR8 are read. If PDR8 is read while PCR8 bits are cleared to 0, the pin states are read regardless of the value stored in PDR8. | R/W |
| 4 to 0 | — | Reserved | These bits are read as 0. The write value should be 0. | — |

chanhvinh.com



### 10.6.4 Port Pull-Up Control Register 8 (PUCR8)

Address: H'FF0017

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PUCR87 | PUCR86 | PUCR85 | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PUCR87 | Port 87 pull-up control | 0: The pull-up MOS of corresponding pin is disabled. | R/W |
| 6 | PUCR86 | Port 86 pull-up control | 1: The pull-up MOS of corresponding pin is enabled. | R/W |
| 5 | PUCR85 | Port 85 pull-up control | PUCR8 is a register that controls the pull-up MOS in bit units of the pins set as the input ports. | R/W |
| 4 to 0 | — | Reserved | These bits are read as 0. The write value should be 0. | — |

- PUCR87 bit to PUCR85 bit (port 87 to 85 pull-up control)

  This function is valid only for the pin set as general input, and for the input pin with a function selected by the PMC.

chanhvinh.com



### 10.6.5 Port Drive Control Register 8 (PDVR8)

Address: H'FF0037

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PDVR87 | PDVR86 | PDVR85 | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PDVR87 | Port 87 drive control | 0: Normal output<br>1: High-current drive output | R/W |
| 6 | PDVR86 | Port 86 drive control | PDVR8 is a register that controls drive capability of the output pins in a bit unit. | R/W |
| 5 | PDVR85 | Port 85 drive control | | R/W |
| 4 to 0 | — | Reserved | These bits are read as 0. The write value should be 0. | — |

### 10.6.6 Notes on Using Port 8

When using on-chip debugger function, set port 8 as general I/O port using PMR8.

chanhvinh.com



## 10.7 Port 9

Figure 10.7 shows the pin configuration of port 9. Port 9 is not available on the H8S/20103 and H8S/20115 Groups.

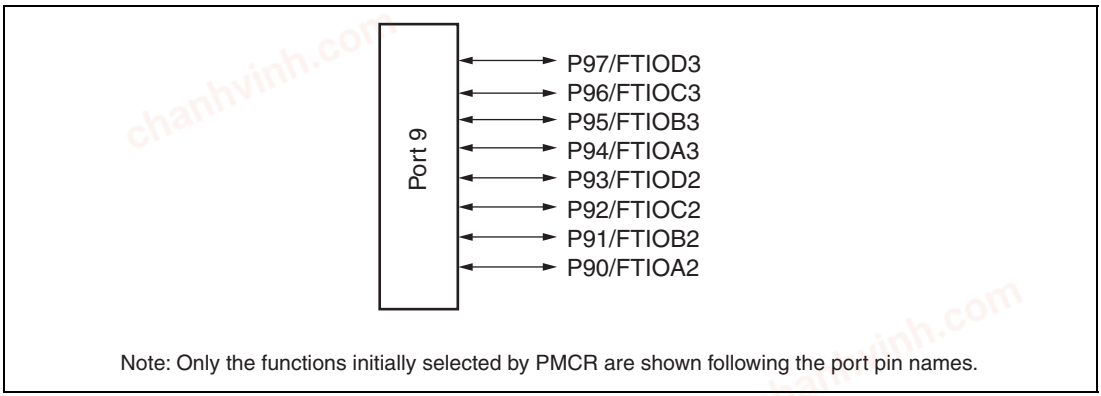



**Figure 10.7 Port 9 Pin Configuration**

Port 9 has the following registers.

- Port mode register 9 (PMR9)
- Port control register 9 (PCR9)
- Port data register 9 (PDR9)
- Port pull-up control register 9 (PUCR9)
- Port drive control register 9 (PDVR9)

chanhvinh.com



### 10.7.1 Port Mode Register 9 (PMR9)

Address: H'FF0008

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PMR97 | PMR96 | PMR95 | PMR94 | PMR93 | PMR92 | PMR91 | PMR90 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PMR97 | Port 97 mode | 0: General I/O port<br>1: The function selected by the peripheral function mapping controller (PMC).<br>PMR9 is a register that selects the function of the multiplexed pins: general I/O function or the function selected by the PMC. | R/W |
| 6 | PMR96 | Port 96 mode | | R/W |
| 5 | PMR95 | Port 95 mode | | R/W |
| 4 | PMR94 | Port 94 mode | | R/W |
| 3 | PMR93 | Port 93 mode | | R/W |
| 2 | PMR92 | Port 92 mode | | R/W |
| 1 | PMR91 | Port 91 mode | | R/W |
| 0 | PMR90 | Port 90 mode | | R/W |

chanhvinh.com



### 10.7.2 Port Control Register 9 (PCR9)

Address: H'FFFFF8

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PCR97 | PCR96 | PCR95 | PCR94 | PCR93 | PCR92 | PCR91 | PCR90 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PCR97 | Port 97 control | 0: When the corresponding pin is designated as a general I/O port, the pin functions as an input port.<br>1: When the corresponding pin is designated as a general I/O port, the pin functions as an output port.<br>PCR9 is a register that selects inputs/outputs in bit units for pins to be used as general I/O ports of port 9. | R/W |
| 6 | PCR96 | Port 96 control | | R/W |
| 5 | PCR95 | Port 95 control | | R/W |
| 4 | PCR94 | Port 94 control | | R/W |
| 3 | PCR93 | Port 93 control | | R/W |
| 2 | PCR92 | Port 92 control | | R/W |
| 1 | PCR91 | Port 91 control | | R/W |
| 0 | PCR90 | Port 90 control | | R/W |

- PCR97 bit to PCR90 bit (port 97 to 90 control)
  When the corresponding pin is designated in PMR9 as a general I/O pin, setting a PCR9 bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port.

chanhvinh.com



### 10.7.3 Port Data Register 9 (PDR9)

Address: H'FFFFE8

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PDR97 | PDR96 | PDR95 | PDR94 | PDR93 | PDR92 | PDR91 | PDR90 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PDR97 | Port 97 data | 0: Low level<br>1: High level<br>PDR9 is a register that stores output data for port 9 pins. When PCR9 bits are set to 1, the values stored in PDR9 are output.<br>When PDR9 is read while PCR9 bits are set to 1, the values stored in PDR9 are read. If PDR9 is read while PCR9 bits are cleared to 0, the pin states are read regardless of the value stored in PDR9. | R/W |
| 6 | PDR96 | Port 96 data | | R/W |
| 5 | PDR95 | Port 95 data | | R/W |
| 4 | PDR94 | Port 94 data | | R/W |
| 3 | PDR93 | Port 93 data | | R/W |
| 2 | PDR92 | Port 92 data | | R/W |
| 1 | PDR91 | Port 91 data | | R/W |
| 0 | PDR90 | Port 90 data | | R/W |

chanhvinh.com



### 10.7.4 Port Pull-Up Control Register 9 (PUCR9)

Address: H'FF0018

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PUCR97 | PUCR96 | PUCR95 | PUCR94 | PUCR93 | PUCR92 | PUCR91 | PUCR90 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PUCR97 | Port 97 pull-up control | 0: The pull-up MOS of corresponding pin is disabled.<br>1: The pull-up MOS of corresponding pin is enabled.<br>PUCR9 is a register that controls the pull-up MOS in bit units of the pins set as the input ports. | R/W |
| 6 | PUCR96 | Port 96 pull-up control | | R/W |
| 5 | PUCR95 | Port 95 pull-up control | | R/W |
| 4 | PUCR94 | Port 94 pull-up control | | R/W |
| 3 | PUCR93 | Port 93 pull-up control | | R/W |
| 2 | PUCR92 | Port 92 pull-up control | | R/W |
| 1 | PUCR91 | Port 91 pull-up control | | R/W |
| 0 | PUCR90 | Port 90 pull-up control | | R/W |

- PUCR97 bit to PUCR90 bit (port 97 to 90 pull-up control)

  This function is valid only for the pin set as general input, and for the input pin with a function selected by the PMC.

chanhvinh.com



### 10.7.5 Port Drive Control Register 9 (PDVR9)

Address: H'FF0038

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PDVR97 | PDVR96 | PDVR95 | PDVR94 | PDVR93 | PDVR92 | PDVR91 | PDVR90 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PDVR97 | Port 97 drive control | 0: Normal output<br>1: High-current drive output<br>PDVR9 is a register that controls drive capability of the output pins in a bit unit. | R/W |
| 6 | PDVR96 | Port 96 drive control | | R/W |
| 5 | PDVR95 | Port 95 drive control | | R/W |
| 4 | PDVR94 | Port 94 drive control | | R/W |
| 3 | PDVR93 | Port 93 drive control | | R/W |
| 2 | PDVR92 | Port 92 drive control | | R/W |
| 1 | PDVR91 | Port 91 drive control | | R/W |
| 0 | PDVR90 | Port 90 drive control | | R/W |

chanhvinh.com



## 10.8 Port A

Port A consists of general I/O pins that are also used as analog input pins for A/D converter unit 1 and unit 2 (only in the H8S/20223 and H8S/20235 Groups).

The functions of PA4 to PA7 can be selected with the peripheral function mapping register of the PMC (except for the H8S/20223 and H8S/20235 Groups).

For selection of functions by the peripheral function mapping controller, see section 9, Peripheral I/O Mapping Controller. Figure 10.8 shows the pin configuration of port A.

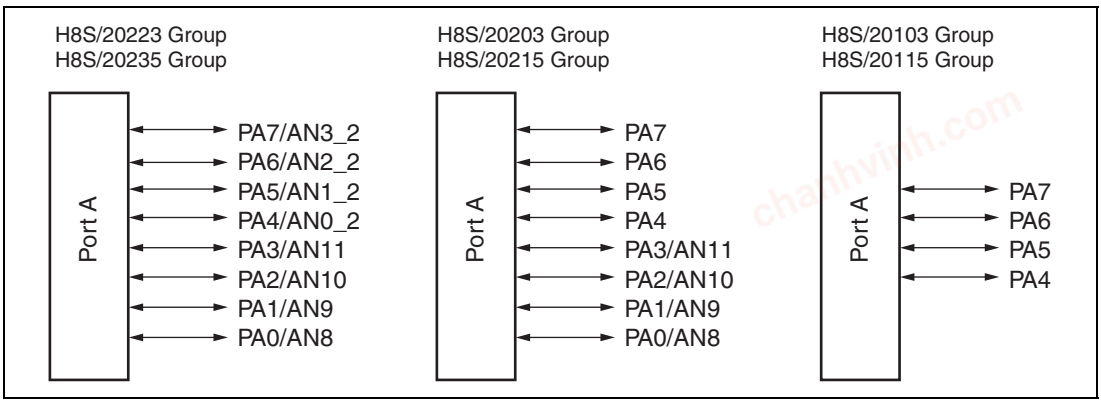



**Figure 10.8 Port A Pin Configuration**

Port A has the following registers.

- Port mode register A (PMRA)
- Port control register A (PCRA)
- Port data register A (PDRA)
- Port pull-up control register A (PUCRA)

chanhvinh.com



- H8S/20103 and H8S/20115 Groups

### 10.8.1 Port Mode Register A (PMRA)

Address: H'FF0009

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PMRA7 | PMRA6 | PMRA5 | PMRA4 | — | PMRA2 | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PMRA7 | Port A7 mode | 0: General I/O port<br>1: The function selected by the peripheral function mapping controller (PMC).<br>PMRA is a register that selects the function of the port A multiplexed pins: general I/O function or the function selected by the PMC. | R/W |
| 6 | PMRA6 | Port A6 mode | | R/W |
| 5 | PMRA5 | Port A5 mode | | R/W |
| 4 | PMRA4 | Port A4 mode | | R/W |
| 3 | — | Reserved | This bit is read as 0. The write value should be 0. | — |
| 2 | PMRA2 | Port B0 analog input select | 0: General I/O port<br>1: AN0 input pin | R/W |
| 1, 0 | — | Reserved | These bits read as 0. The write value should be 0. | — |

- PMRA7 bit to PMRA4 bit (port A7 to A4 mode)

  These bits select the function of the multiplexed pins PA7 to PA4: general I/O function or the function selected by the PMC.
- PMRA2 bit (port A2 mode)

  This bit selects general I/O function or the analog input function for PB0.

chanhvinh.com



### 10.8.2 Port Control Register A (PCRA)

Address: H'FFFFF9

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PCRA7 | PCRA6 | PCRA5 | PCRA4 | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PCRA7 | Port A7 control | 0: When the corresponding pin is designated as a general I/O port, the pin functions as an input port.<br>1: When the corresponding pin is designated as a general I/O port, the pin functions as an output port.<br>PCRA is a register that selects inputs/outputs in bit units for pins to be used as general I/O ports of port A. | R/W |
| 6 | PCRA6 | Port A6 control | | R/W |
| 5 | PCRA5 | Port A5 control | | R/W |
| 4 | PCRA4 | Port A4 control | | R/W |
| 3 to 0 | — | Reserved | These bits are read as 0. The write value should be 0. | — |

- PCRA7 bit to PCRA4 bit (port A7 to A4 control)

  When the corresponding pin is designated in PMRA as a general I/O pin, setting a PCRA bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port.

chanhvinh.com



### 10.8.3 Port Data Register A (PDRA)

Address: H'FFFFE9

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PDRA7 | PDRA6 | PDRA5 | PDRA4 | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PDRA7 | Port A7 data | 0: Low level<br>1: High level<br>PDRA is a register that stores output data for port A pins. When PCRA bits are set to 1, the values stored in PDRA are output.<br>When PDRA is read while PCRA bits are set to 1, the values stored in PDRA are read. If PDRA is read while PCRA bits are cleared to 0, the pin states are read regardless of the value stored in PDRA. | R/W |
| 6 | PDRA6 | Port A6 data | | R/W |
| 5 | PDRA5 | Port A5 data | | R/W |
| 4 | PDRA4 | Port A4 data | | R/W |
| 3 to 0 | — | Reserved | These bits are read as 0. The write value should be 0. | — |

chanhvinh.com



### 10.8.4 Port Pull-Up Control Register A (PUCRA)

Address: H'FF0019

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PUCRA7 | PUCRA6 | PUCRA5 | PUCRA4 | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PUCRA7* | Port A7 pull-up control | 0: The pull-up MOS of corresponding pin is disabled.<br>1: The pull-up MOS of corresponding pin is not enabled.<br>PUCRA is a register that controls the pull-up MOS in bit units of the pins set as the input ports. | R/W |
| 6 | PUCRA6* | Port A6 pull-up control | | R/W |
| 5 | PUCRA5* | Port A5 pull-up control | | R/W |
| 4 | PUCRA4* | Port A4 pull-up control | | R/W |
| 3 to 0 | — | Reserved | These bits are read as 0. The write value should be 0. | — |

Note: * When PA7 to PA4 are set as the analog input pin, clear the corresponding bits to 0.

- PUCRA7 bit to PUCRA4 bit (port A7 to A4 pull-up control)

  This function is valid only for the pin set as general input, and for the input pin with a function selected by the PMC.

chanhvinh.com



- H8S/20203 and H8S/20215 Groups

### 10.8.5 Port Mode Register A (PMRA)

Address: H'FF0009

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PMRA7 | PMRA6 | PMRA5 | PMRA4 | — | PMRA2 | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PMRA7 | Port A7 mode | 0: General I/O port<br>1: The function selected by the peripheral function mapping controller (PMC).<br>PMRA is a register that selects the function of the port A multiplexed pins: general I/O function or the function selected by the PMC. | R/W |
| 6 | PMRA6 | Port A6 mode | | R/W |
| 5 | PMRA5 | Port A5 mode | | R/W |
| 4 | PMRA4 | Port A4 mode | | R/W |
| 3 | — | Reserved | This bit is read as 0. The write value should be 0. | — |
| 2 | PMRA2 | Port B0 analog input select | 0: General I/O port<br>1: AN0 input pin | R/W |
| 1, 0 | — | Reserved | These bits are read as 0. The write value should be 0. | — |

- PMRA7 bit to PMRA4 bit (port A7 to A4 mode)

  These bits select the function of the multiplexed pins PA7 to PA4: general I/O function or the function selected by the PMC.

- PMRA2 bit (port A2 mode)

  This bit selects general I/O function or the analog input function for PB0.

chanhvinh.com



### 10.8.6 Port Control Register A (PCRA)

Address: H'FFFFF9

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PCRA7 | PCRA6 | PCRA5 | PCRA4 | PCRA3 | PCRA2 | PCRA1 | PCRA0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PCRA7 | Port A7 control | 0: When the corresponding pin is designated as a general I/O port, the pin functions as an input port.<br>1: When the corresponding pin is designated as a general I/O port, the pin functions as an output port.<br>PCRA is a register that selects inputs/outputs in bit units for pins to be used as general I/O ports of port A. | R/W |
| 6 | PCRA6 | Port A6 control | | R/W |
| 5 | PCRA5 | Port A5 control | | R/W |
| 4 | PCRA4 | Port A4 control | | R/W |
| 3 | PCRA3 | Port A3 control | | R/W |
| 2 | PCRA2 | Port A2 control | | R/W |
| 1 | PCRA1 | Port A1 control | | R/W |
| 0 | PCRA0 | Port A0 control | | R/W |

- PCRA7 bit to PCRA0 bit (port A7 to A0 control)

  When pins PA7 to PA4 are designated in PMRA as general I/O pins, setting a corresponding bit to 1 makes the pin an output port, while clearing the bit to 0 makes the pin an input port.

  When pins PA3 to PA0 are not selected for use as analog input channels by A/D converter settings, they are available for use as general I/O pins. Setting a PCRA bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin.

chanhvinh.com



### 10.8.7 Port Data Register A (PDRA)

Address: H'FFFFE9

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PDRA7 | PDRA6 | PDRA5 | PDRA4 | PDRA3 | PDRA2 | PDRA1 | PDRA0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PDRA7 | Port A7 data | 0: Low level | R/W |
| 6 | PDRA6 | Port A6 data | 1: High level | R/W |
| 5 | PDRA5 | Port A5 data | PDRA is a register that stores output data for port A pins. When PCRA bits are set to 1, the values stored in PDRA are output. | R/W |
| 4 | PDRA4 | Port A4 data | | R/W |
| 3 | PDRA3 | Port A3 data | When PDRA is read while PCRA bits are set to 1, the values stored in PDRA are read. If PDRA is read while PCRA bits are cleared to 0, the pin states are read regardless of the value stored in PDRA. | R/W |
| 2 | PDRA2 | Port A2 data | | R/W |
| 1 | PDRA1 | Port A1 data | | R/W |
| 0 | PDRA0 | Port A0 data | | R/W |
| | | | When pins PA3 to PA0 are set as analog input channels by ADCSR and ADCR of the A/D converter, however, the corresponding PDRA bits are always read as 1 even if the respective PCRA bits are cleared to 0. | |

chanhvinh.com



### 10.8.8 Port Pull-Up Control Register A (PUCRA)

Address: H'FF0019

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PUCRA7 | PUCRA6 | PUCRA5 | PUCRA4 | PUCRA3 | PUCRA2 | PUCRA1 | PUCRA0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PUCRA7* | Port A7 pull-up control | 0: The pull-up MOS of corresponding pin is disabled. | R/W |
| 6 | PUCRA6* | Port A6 pull-up control | 1: The pull-up MOS of corresponding pin is enabled. | R/W |
| 5 | PUCRA5* | Port A5 pull-up control | PUCRA is a register that controls the pull-up MOS in bit units of the pins set as the input ports. | R/W |
| 4 | PUCRA4* | Port A4 pull-up control | | R/W |
| 3 | PUCRA3* | Port A3 pull-up control | | R/W |
| 2 | PUCRA2* | Port A2 pull-up control | | R/W |
| 1 | PUCRA1* | Port A1 pull-up control | | R/W |
| 0 | PUCRA0* | Port A0 pull-up control | | R/W |

Note: * When PA7 to PA4 are set as the analog input pin, clear the corresponding bits to 0.

- PUCRA7 bit to PUCRA0 bit (port A7 to A0 pull-up control)

  This function is valid only for the pin set as general input, and for the input pin with a function selected by the PMC. However, this setting is invalid for the analog input pin.

chanhvinh.com



- H8S/20223 and H8S/20235 Groups

### 10.8.9 Port Mode Register A (PMRA)

Address: H'FF0009

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | PMRA3 | PMRA2 | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 to 4 | — | Reserved | These bits are read as 0. The write value should be 0. | — |
| 3 | PMRA3 | Port A4 analog input select | 0: General I/O port<br>1: AN0_2 input pin | R/W |
| 2 | PMRA2 | Port B0 analog input select | 0: General I/O port<br>1: AN0 input pin | R/W |
| 1, 0 | — | Reserved | These bits are read as 0. The write value should be 0. | — |

PMRA selects the analog input function for PA4 and PB0 functioning as general I/O.

- PMRA3 bit (Port A4 analog input select)
  This bit selects the analog input function for PA4.
- PMRA2 bit (Port B0 analog input select)
  This bit selects the analog input function for PB0.

chanhvinh.com



### 10.8.10 Port Control Register A (PCRA)

Address: H'FFFFF9

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PCRA7 | PCRA6 | PCRA5 | PCRA4 | PCRA3 | PCRA2 | PCRA1 | PCRA0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PCRA7 | Port A7 control | 0: When the corresponding pin is designated as a general I/O port, the pin functions as an input port.<br>1: When the corresponding pin is designated as a general I/O port, the pin functions as an output port.<br>PCRA is a register that selects inputs/outputs in bit units for pins to be used as general I/O ports of port A. | R/W |
| 6 | PCRA6 | Port A6 control | | R/W |
| 5 | PCRA5 | Port A5 control | | R/W |
| 4 | PCRA4 | Port A4 control | | R/W |
| 3 | PCRA3 | Port A3 control | | R/W |
| 2 | PCRA2 | Port A2 control | | R/W |
| 1 | PCRA1 | Port A1 control | | R/W |
| 0 | PCRA0 | Port A0 control | | R/W |

- PCRA7 bit to PCRA0 bit (port A7 to A0 control)

  When the corresponding pins are not selected for use as analog input channels by A/D converter settings, they are available for use as general I/O pins. Setting a PCRA bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin. However, for the PA4 pin, clearing the PMRA3 bit in PMRA to 0 is also required.

chanhvinh.com



### 10.8.11 Port Data Register A (PDRA)

Address: H'FFFFE9

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PDRA7 | PDRA6 | PDRA5 | PDRA4 | PDRA3 | PDRA2 | PDRA1 | PDRA0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PDRA7 | Port A7 data | 0: Low level | R/W |
| 6 | PDRA6 | Port A6 data | 1: High level | R/W |
| 5 | PDRA5 | Port A5 data | PDRA is a register that stores output data for port A pins. When PCRA bits are set to 1, the values stored in PDRA are output. | R/W |
| 4 | PDRA4 | Port A4 data | | R/W |
| 3 | PDRA3 | Port A3 data | When PDRA is read while PCRA bits are set to 1, the values stored in PDRA are read. If PDRA is read while PCRA bits are cleared to 0, the pin states are read regardless of the value stored in PDRA. | R/W |
| 2 | PDRA2 | Port A2 data | | R/W |
| 1 | PDRA1 | Port A1 data | | R/W |
| 0 | PDRA0 | Port A0 data | | R/W |
| | | | When the pins are set as analog input channels by ADCSR and ADCR of the A/D converter, however, the corresponding PDRA bits are always read as 1 even if the respective PCRA bits are cleared to 0. | |

chanhvinh.com



### 10.8.12 Port Pull-Up Control Register A (PUCRA)

Address: H'FF0019

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PUCRA7 | PUCRA6 | PUCRA5 | PUCRA4 | PUCRA3 | PUCRA2 | PUCRA1 | PUCRA0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PUCRA7* | Port A7 pull-up control | 0: The pull-up MOS of corresponding pin is disabled.<br>1: The pull-up MOS of corresponding pin is enabled.<br>PUCRA is a register that controls the pull-up MOS in bit units of the pins set as the input ports. | R/W |
| 6 | PUCRA6* | Port A6 pull-up control | | R/W |
| 5 | PUCRA5* | Port A5 pull-up control | | R/W |
| 4 | PUCRA4* | Port A4 pull-up control | | R/W |
| 3 | PUCRA3* | Port A3 pull-up control | | R/W |
| 2 | PUCRA2* | Port A2 pull-up control | | R/W |
| 1 | PUCRA1* | Port A1 pull-up control | | R/W |
| 0 | PUCRA0* | Port A0 pull-up control | | R/W |

Note: * When PA7 to PA0 are set as the analog input pin, clear the corresponding bits to 0.

- PUCRA7 bit to PUCRA0 bit (port A7 to A0 pull-up control)
  This function is valid only for the pin set as general input, and for the input pin with a function selected by the PMC. However, this setting is invalid for the analog input pin.

### 10.8.13 Notes on Using Port A

1. The PA4 pin is initially set as general I/O pin. If using this pin as the AN0_2 analog input pin for the A/D converter unit 2 in the H8S/20223 and H8S/20235 Groups, set the PMRA3 bit in PMRA to 1.
2. In the H8S/20223 and H8S/20235 Groups, pins PA7 to PA4 can be used as the general I/O pins or analog input pins. If using these pins as the general I/O pins, do not set bits CH3 to CH0 in ADCSR_2 of the A/D converter unit 2 to set these pins as the analog input pins.

chanhvinh.com



## 10.9 Port B

Port B consists of general I/O pins that are also used as analog input pins for the A/D converter unit 1, or as analog output pins for the D/A converter. Figure 10.9 shows the pin configuration of port B.

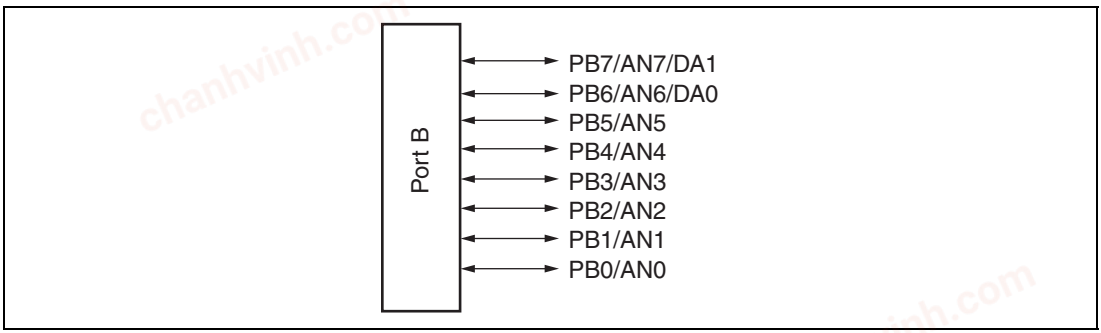



**Figure 10.9 Port B Pin Configuration**

Port B has the following registers.

- Port control register B (PCRB)
- Port data register B (PDRB)
- Port pull-up control register B (PUCRB)

chanhvinh.com



### 10.9.1 Port Control Register B (PCRB)

Address: H'FFFFFA

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PCRB7 | PCRB6 | PCRB5 | PCRB4 | PCRB3 | PCRB2 | PCRB1 | PCRB0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PCRB7 | Port B7 control | 0: When the corresponding pin is designated as a general I/O port, the pin functions as an input port.<br>1: When the corresponding pin is designated as a general I/O port, the pin functions as an output port.<br>PCRB is a register that selects inputs/outputs in bit units for pins to be used as general I/O ports of port B. | R/W |
| 6 | PCRB6 | Port B6 control | | R/W |
| 5 | PCRB5 | Port B5 control | | R/W |
| 4 | PCRB4 | Port B4 control | | R/W |
| 3 | PCRB3 | Port B3 control | | R/W |
| 2 | PCRB2 | Port B2 control | | R/W |
| 1 | PCRB1 | Port B1 control | | R/W |
| 0 | PCRB0 | Port B0 control | | R/W |

- PCRB7 bit to PCRB0 bit (port B7 to B0 control)

  When the corresponding pins are not selected for use as analog input or output channels by A/D converter or D/A converter settings, they are available for use as general I/O pins. Setting a PCRB bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin. However, for the PB0 pin, clearing the PMRA2 bit in PMRA to 0 is also required.

chanhvinh.com



### 10.9.2 Port Data Register B (PDRB)

Address: H'FFFFEA

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PDRB7 | PDRB6 | PDRB5 | PDRB4 | PDRB3 | PDRB2 | PDRB1 | PDRB0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PDRB7 | Port B7 data | 0: Low level | R/W |
| 6 | PDRB6 | Port B6 data | 1: High level | R/W |
| 5 | PDRB5 | Port B5 data | PDRB is a register that stores output data for port B pins. When PCRB bits are set to 1, the values stored in PDRB are output. | R/W |
| 4 | PDRB4 | Port B4 data | | R/W |
| 3 | PDRB3 | Port B3 data | When PDRB is read while PCRB bits are set to 1, the values stored in PDRB are read. If PDRB is read while PCRB bits are cleared to 0, the pin states are read regardless of the value stored in PDRB. | R/W |
| 2 | PDRB2 | Port B2 data | | R/W |
| 1 | PDRB1 | Port B1 data | | R/W |
| 0 | PDRB0 | Port B0 data | | R/W |
| | | | When the pins are set as analog input channels by ADCSR and ADCR of the A/D converter, however, the corresponding PDRB bits are always read as 1 even if the respective PCRB bits are cleared to 0. | |
| | | | Similarly, when pins PB6 and PB7 are set as analog output for the D/A converter by bit DAOE1 in DACR of the D/A converter, the corresponding PCRB bits are always read as 1 even if they are cleared to 0. | |

chanhvinh.com



### 10.9.3 Port Pull-Up Control Register B (PUCRB)

Address: H'FF001A

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PUCRB7 | PUCRB6 | PUCRB5 | PUCRB4 | PUCRB3 | PUCRB2 | PUCRB1 | PUCRB0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PUCRB7 | Port B7 pull-up control | 0: The pull-up MOS of corresponding pin is disabled. | R/W |
| 6 | PUCRB6 | Port B6 pull-up control | 1: The pull-up MOS of corresponding pin is enabled. | R/W |
| 5 | PUCRB5 | Port B5 pull-up control | PUCRB is a register that controls the pull-up MOS in bit units of the pins set as the input ports. | R/W |
| 4 | PUCRB4 | Port B4 pull-up control | | R/W |
| 3 | PUCRB3 | Port B3 pull-up control | | R/W |
| 2 | PUCRB2 | Port B2 pull-up control | | R/W |
| 1 | PUCRB1 | Port B1 pull-up control | | R/W |
| 0 | PUCRB0 | Port B0 pull-up control | | R/W |

- PUCRB7 bit to PUCRB0 bit (port B7 to B0 pull-up control)

  This function is valid only for the pin set as general input, and for the input pin with a function selected by the PMC. However, this setting is invalid for the analog input pin.

### 10.9.4 Notes on Using Port B

1. The PB0 pin is initially set as general I/O pin. If using this pin as the analog input pin for the A/D converter, set the PMRA2 bit in PMRA to 1.
2. Pins PB7 and PB6 can be used as analog input pins for the A/D converter or analog output pins for the D/A converter. Do not set these pins as analog input pins and analog output pins at the same time.

chanhvinh.com



## 10.10 Port J

Port J consists of pins PJ1 and PJ0. These pins can also be used as external oscillation pins and clock output pin. Figure 10.10 shows the pin configuration of port J. In selection of the function of these multiplexed pins, the PMRJ register setting is given priority.

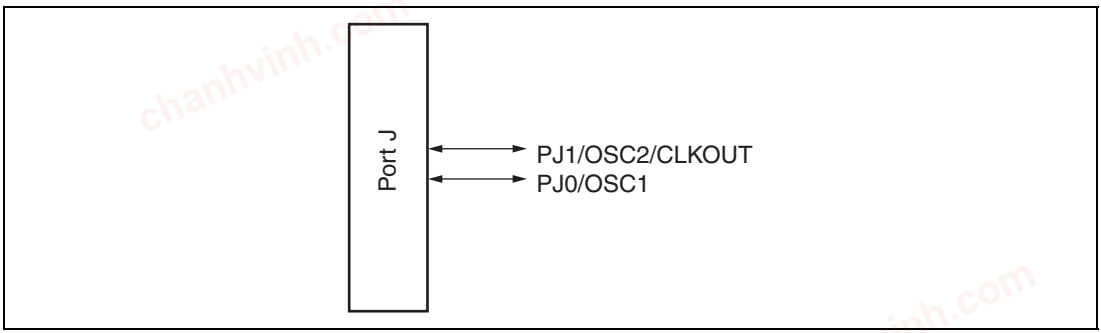



**Figure 10.10 Port J Pin Configuration**

Port J has the following registers.

- Port mode register J (PMRJ)
- Port control register J (PCRJ)
- Port data register J (PDRJ)
- Port pull-up control register J (PUCRJ)

chanhvinh.com



### 10.10.1 Port Mode Register J (PMRJ)

Address: H'FF000C

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | PMRJ[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 to 2 | — | Reserved | These bits are read as 0. The write value should be 0. | — |
| 1, 0 | PMRJ[1:0] | Port J[1:0] mode | Selects PJ1 and PJ0 pin functions. | R/W |

| PMRJ1 | PMRJ0 | PJ1 Pin | PJ0 Pin |
|---|---|---|---|
| 0 | 0 | PJ1 I/O | PJ0 I/O |
| 0 | 1 | PJ1 I/O | OSC1 input* (external clock input) |
| 1 | 0 | CLKOUT | PJ0 I/O |
| 1 | 1 | OSC2 | OSC1 |

Note: * Set the PMRJ1 and PMRJ0 bits to 01 to input the external clock on the OSC1 pin. Do not apply the external clock to the OSC1 pin while the PMRJ1 and PMRJ0 bits are set to 11.

chanhvinh.com



### 10.10.2 Port Control Register J (PCRJ)

Address: H'FFFFFC

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | PCRJ1 | PCRJ0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 to 2 | — | Reserved | These bits are read as 0. The write value should be 0. | — |
| 1 | PCRJ1 | Port J1 control | 0: When the corresponding pin is designated as a general I/O port, the pin functions as an input port.<br>1: When the corresponding pin is designated as a general I/O port, the pin functions as an output port.<br>PCRJ is a register that selects inputs/outputs in bit units for pins to be used as general I/O ports of port J. | R/W |
| 0 | PCRJ0 | Port J0 control | | R/W |

- PCRJ1 bit and PCRJ0 bit (port J1 and J0 control)

  When the general I/O port function is selected by PMRJ, setting a PCRJ bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port.

chanhvinh.com



### 10.10.3 Port Data Register J (PDRJ)

Address: H'FFFFEC

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | PDRJ1 | PDRJ0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 to 2 | — | Reserved | These bits are read as 0. The write value should be 0. | — |
| 1 | PDRJ1 | Port J1 data | 0: Low level | R/W |
| 0 | PDRJ0 | Port J0 data | 1: High level | R/W |
| | | | PDRJ is a register that stores output data for port J pins. When PCRJ bits are set to 1, the values stored in PDRJ are output. | |
| | | | When PDRJ is read while PCRJ bits are set to 1, the values stored in PDRJ are read. If PDRJ is read while PCRJ bits are cleared to 0, the pin states are read regardless of the value stored in PDRJ. | |

chanhvinh.com



### 10.10.4 Port Pull-Up Control Register J (PUCRJ)

Address: H'FF001C

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | PUCRJ1 | PUCRJ0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 to 2 | — | Reserved | These bits are read as 0. The write value should be 0. | — |
| 1 | PUCRJ1 | Port J1 pull-up control | 0: The pull-up MOS of corresponding pin is disabled.<br>1: The pull-up MOS of corresponding pin is enabled.<br>PUCRJ is a register that controls the pull-up MOS in bit units of the pins set as the input ports. | R/W |
| 0 | PUCRJ0 | Port J0 pull-up control | | R/W |

- PUCRJ1 bit and PUCRJ0 bit (port J1 and J0 pull-up control)

  This function is valid only for the pin set as general input, and for the input pin with a function selected by the PMC.

chanhvinh.com



# Section 11 Data Transfer Controller (DTC)

This LSI includes a data transfer controller (DTC). The DTC can be activated by an interrupt or software to transfer data.

Figure 11.1 shows a block diagram of the DTC.

## 11.1 Features

- Transfer possible over any number of channels
- Three transfer modes
  - — Normal mode
    One operation transfers one byte or one word of data.
    Memory address is incremented or decremented by 1 or 2.
    From 1 to 65,536 transfers can be specified.
  - — Repeat mode
    One operation transfers one byte or one word of data.
    Memory address is incremented or decremented by 1 or 2.
    Once the specified number of transfers (1 to 256) has ended, the initial state is restored, and transfer is repeated.
  - — Block transfer mode
    One operation transfers specified one block of data.
    The block size is 1 to 256 bytes or words.
    From 1 to 65,536 transfers can be specified.
    Either the transfer source or the transfer destination is designated as a block area.
- One activation source can trigger a number of data transfers (chain transfer)
- Direct specification of 16-Mbyte address space possible
- Activation by software is possible.
- Transfer can be set in byte or word units.
- A CPU interrupt can be requested for the interrupt that activated the DTC.
- Module standby mode can be set.

chanhvinh.com



The DTC's register information is stored in the on-chip RAM. A 32-bit bus connects the DTC to the on-chip RAM, enabling 32-bit/1-state reading and writing of the DTC register information.

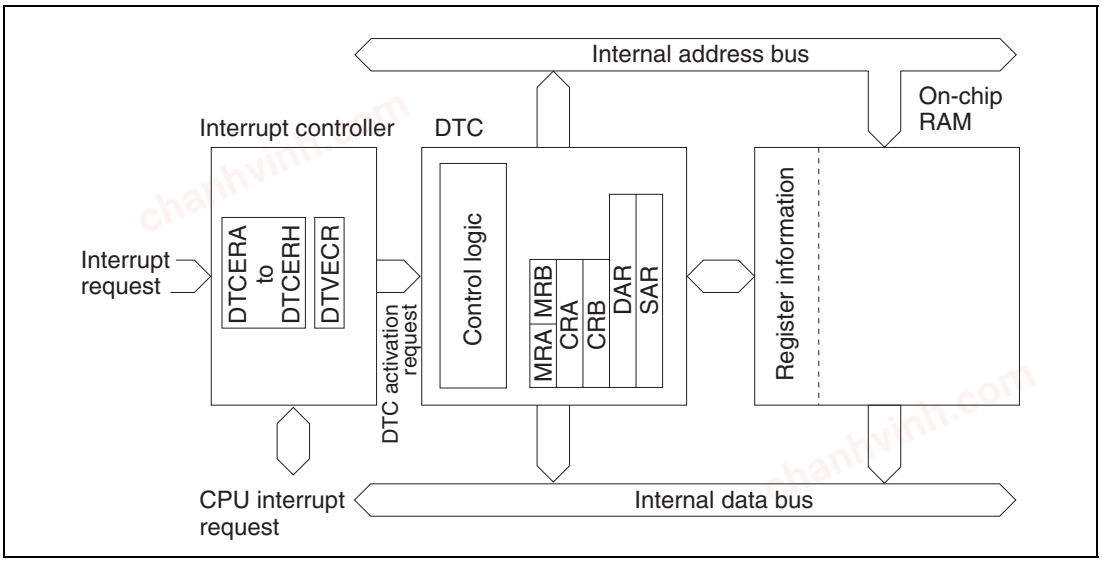



**Figure 11.1 Block Diagram of DTC**

chanhvinh.com



## 11.2 Register Descriptions

DTC has the following registers.

- DTC mode register A (MRA)
- DTC mode register B (MRB)
- DTC source address register (SAR)
- DTC destination address register (DAR)
- DTC transfer count register A (CRA)
- DTC transfer count register B (CRB)

The above six registers cannot be directly accessed from the CPU. When the DTC activation source is generated, the DTC reads from a set of register information that is stored in an on-chip RAM to the corresponding DTC register information and transfers data. After the data transfer, it writes a set of updated register information back to the RAM.

- DTC enable register A (DTCERA)
- DTC enable register B (DTCERB)
- DTC enable register C (DTCERC)
- DTC enable register D (DTCERD)
- DTC enable register E (DTCERE)
- DTC enable register F (DTCERF)
- DTC enable register G (DTCERG)
- DTC enable register H (DTCERH)
- DTC vector register (DTVECR)

chanhvinh.com



### 11.2.1 DTC Mode Register A (MRA)

Address: —

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | SM[1:0] | | DM[1:0] | | MD[1:0] | | DTS | Sz |
| Value after reset: | — | — | — | — | — | — | — | — |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7<br>6 | SM[1:0] | Source address mode 1 and 0 | 0×: SAR is fixed<br>10: SAR is incremented after a transfer (by +1 when Sz = 0; by +2 when Sz = 1)<br>11: SAR is decremented after a transfer (by –1 when Sz = 0; by –2 when Sz = 1) | — |
| 5<br>4 | DM[1:0] | Destination address mode 1 and 0 | 0×: DAR is fixed<br>10: DAR is incremented after a transfer (by +1 when Sz = 0; by +2 when Sz = 1)<br>11: DAR is decremented after a transfer (by –1 when Sz = 0; by –2 when Sz = 1) | — |
| 3<br>2 | MD[1:0] | DTC mode 1 and 0 | 00: Normal mode<br>01: Repeat mode<br>10: Block transfer mode<br>11: Setting prohibited | — |
| 1 | DTC | DTC transfer mode select | 0: Destination side is repeat area or block area.<br>1: Source side is repeat area or block area. | — |
| 0 | Sz | DTC data transfer size | 0: Byte-size transfer<br>1: Word-size transfer | — |

Legend:

×: Don't care

chanhvinh.com



MRA selects the DTC operating mode.

- SM[1:0] bits (source address mode 1 and 0)
  These bits specify an SAR operation after data transfer.
- DM[1:0] bits (destination address mode 1 and 0)
  These bits specify a DAR operation after data transfer.
- MD[1:0] bits (DTC mode 1 and 0)
  These bits specify the DTC transfer mode.
- DTS bit (DTC transfer mode select)
  This bit specifies whether the source side or the destination side is set to be a repeat area or block area, in repeat mode or block transfer mode.
- Sz bit (DTC data transfer size)
  This bit specifies the size of data to be transferred.

chanhvinh.com



### 11.2.2 DTC Mode Register B (MRB)

Address: —

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | CHNE | DISEL | CHNS | — | — | — | — | — |
| Value after reset: | — | — | — | — | — | — | — | — |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | CHNE | DTC chain transfer enable | 0: Disables chain transfer.<br>1: Enables chain transfer. | — |
| 6 | DISEL | DTC interrupt select | 0: Generates an interrupt request to the CPU only when the specified data transfer has been completed.<br>1: Generates an interrupt request to the CPU every time after the DTC transfer has been completed. | — |
| 5 | CHNS | Chain transfer select | 0: Performs chain transfer consecutively.<br>1: Performs chain transfer only when transfer counter = 0 | — |
| 4 to 0 | — | Reserved | These bits have no effect on DTC operation. The write value should be 0. | — |

MRB selects the DTC operating mode.

- CHNE bit (DTC chain transfer enable)

  When this bit is set to 1, a chain transfer will be performed. For details, see section 11.5.4, Chain Transfer.

  In the data transfer with CHNE set to 1, determination of the end of the specified number of transfers, clearing of the activation source flag, and clearing of DTCER are not performed.

- DISEL bit (DTC interrupt select)

  When this bit is set to 1, a CPU interrupt request is generated every time the DTC transfer is performed (the interrupt source flags as the activation source are not cleared to 0 by the DTC). When this bit is cleared to 0, a CPU interrupt request is generated at the time when the specified number of data transfers ends (the interrupt source flags as the activation source is cleared to 0 by the DTC).

chanhvinh.com



### 11.2.3 DTC Source Address Register (SAR)

Address: —

| Bit: | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|
| | bit23 | bit22 | bit21 | bit20 | bit19 | bit18 | bit17 | bit16 |
| Value after reset: | — | — | — | — | — | — | — | — |

| Bit: | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 |
|---|---|---|---|---|---|---|---|---|
| | bit15 | bit14 | bit13 | bit12 | bit11 | bit10 | bit9 | bit8 |
| Value after reset: | — | — | — | — | — | — | — | — |

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
| Value after reset: | — | — | — | — | — | — | — | — |

SAR designates the source address of data to be transferred by the DTC. For word-size transfer, specify an even source address.

### 11.2.4 DTC Destination Address Register (DAR)

Address: —

| Bit: | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|
| | bit23 | bit22 | bit21 | bit20 | bit19 | bit18 | bit17 | bit16 |
| Value after reset: | — | — | — | — | — | — | — | — |

| Bit: | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 |
|---|---|---|---|---|---|---|---|---|
| | bit15 | bit14 | bit13 | bit12 | bit11 | bit10 | bit9 | bit8 |
| Value after reset: | — | — | — | — | — | — | — | — |

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
| Value after reset: | — | — | — | — | — | — | — | — |

DAR designates the destination address of data to be transferred by the DTC. For word-size transfer, specify an even destination address.

chanhvinh.com



### 11.2.5 DTC Transfer Count Register A (CRA)

Address: —

| Bit: | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 |
|---|---|---|---|---|---|---|---|---|
| | bit15 | bit14 | bit13 | bit12 | bit11 | bit10 | bit9 | bit8 |
| Value after reset: | — | — | — | — | — | — | — | — |

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
| Value after reset: | — | — | — | — | — | — | — | — |

CRA designates the number of times that data is to be transferred by the DTC.

In normal mode, the entire CRA functions as a 16-bit transfer counter (1 to 65,536). It is decremented by 1 every time data is transferred, and transfer ends when the count reaches H'0000.

In repeat mode or block transfer mode, the CRA is divided into two parts: the upper 8 bits (CRAH) and the lower 8 bits (CRAL). In repeat mode, CRAH holds the number of transfers while CRAL functions as an 8-bit transfer counter (1 to 256). In block transfer mode, CRAH holds the size of blocks while CRAL functions as a block-size counter. CRAL is decremented by 1 every time data is transferred, and the contents of CRAH are sent when the count value reaches H'00.

### 11.2.6 DTC Transfer Count Register B (CRB)

Address: —

| Bit: | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 |
|---|---|---|---|---|---|---|---|---|
| | bit15 | bit14 | bit13 | bit12 | bit11 | bit10 | bit9 | bit8 |
| Value after reset: | — | — | — | — | — | — | — | — |

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
| Value after reset: | — | — | — | — | — | — | — | — |

CRB is a 16-bit register that designates the number of times block data is to be transferred by the DTC in block transfer mode. It functions as a 16-bit transfer counter (1 to 65,536) that is decremented by 1 every time data is transferred, and transfer ends when the count reaches H'0000. The CRB is not available in normal and repeat modes.

chanhvinh.com



### 11.2.7 DTC Enable Registers A to H (DTCERA to DTCERH)

Address: H'FF0534 to H'FF053B

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | DTCEn7 | DTCEn6 | DTCEn5 | DTCEn4 | DTCEn3 | DTCEn2 | DTCEn1 | DTCEn0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | DTCEn7 | DTC activation enable | 0: A relevant interrupt source is not selected as a DTC activation source.<br>1: A relevant interrupt source is selected as a DTC activation source.<br>[Setting condition]<br>• Setting this bit to 1 specifies a relevant interrupt source to a DTC activation source.<br>[Clearing conditions]<br>• When the DISEL bit in MRB is set to 1 and the data transfer has ended.<br>• When the specified number of data transfers has ended.<br>These bits are not automatically cleared when the DISEL bit is 0 and the specified number of data transfers has not ended.<br>• When 0 is written to DTCE after reading DTCE = 1. | R/W |
| 6 | DTCEn6 | | | R/W |
| 5 | DTCEn5 | | | R/W |
| 4 | DTCEn4 | | | R/W |
| 3 | DTCEn3 | | | R/W |
| 2 | DTCEn2 | | | R/W |
| 1 | DTCEn1 | | | R/W |
| 0 | DTCEn0 | | | R/W |

Notes: n = A to H
DTCE bits with no corresponding interrupt are reserved. The write value should always be 0.

DTCER, which is comprised of DTCERA to DTCERH, is a register that specifies DTC activation interrupt sources. The correspondence between interrupt sources and DTCE bits is shown in table 11.1. For DTCE bit setting, use bit manipulation instructions such as BSET and BCLR. If all interrupts are masked, multiple activation sources can be set at one time (only at the initial setting) by writing data after executing a dummy read on the relevant register.

chanhvinh.com



**Table 11.1 Correspondence between Interrupt Sources and DTCER**

| | Bit | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Register** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| DTCERA | IRQ0 | IRQ1 | IRQ2 | IRQ3 | IRQ4 | IRQ5 | IRQ6 | IRQ7 |
| DTCERB | IADEND_1 | IADCMP_1 | IADEND_2 *1 | IADCMP_2 *1 | ELC1FP | ELC2FP | SCI3_1_RXI | SCI3_1_TXI |
| DTCERC | SCI3_2_RXI | SCI3_2_TXI | SCI3_3_RXI | SCI3_3_TXI | — | — | — | — |
| DTCERD | IIC2/SSU_RXI | IIC2/SSU_TXI | — | — | ITCMA*2 | ITCMB*2 | ITCMC*2 | ITCMD*2 |
| DTCERE | ITDMA0_0 | ITDMB0_0 | ITDMC0_0 | ITDMD0_0 | ITDMA0_1 | ITDMB0_1 | ITDMC0_1 | ITDMD0_1 |
| DTCERF | ITDMA1_2 *3 | ITDMB1_2 *3 | ITDMC1_2 *3 | ITDMD1_2 *3 | ITDMA1_3 *3 | ITDMB1_3 *3 | ITDMC1_3 *3 | ITDMD1_3 *3 |
| DTCERG | — | — | — | ITESC | ITEMI | ITEHR | ITEDY | ITEWK |
| DTCERH | — | — | — | — | ITGMA | ITGMB | — | — |

Notes: —: Reserved bit

1. Supported only in the H8S/20223 and H8S/20235 Groups.
2. Supported only in the H8S/20103 and H8S/20115 Groups.
3. Not supported in the H8S/20103 and H8S/20115 Groups.

chanhvinh.com



### 11.2.8 DTC Vector Register (DTVECR)

Address: H'FF053D

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | SWDTE | DTVEC6 | DTVEC5 | DTVEC4 | DTVEC3 | DTVEC2 | DTVEC1 | DTVEC0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | SWDTE | DTC software activation enable | 0: Disables the DTC activation by software.<br>1: Enables the DTC activation by software.<br>Setting this bit to 1 activates DTC.<br>[Clearing conditions]<br>• When the DISEL bit is 0 and the specified number of data transfers has not ended.<br>• When 0 is written to the DISEL bit after a software-activated data transfer end interrupt (SWDTEND) request has been sent to the CPU.<br>When the DISEL bit is 1 and data transfer has ended or when the specified number of data transfers has ended, this bit will not be cleared. | R/W |
| 6 | DTVEC6 | DTC software activation vector 6 to 0 | These bits specify a vector number for DTC activation by software.<br>These bits specify a vector number for DTC software activation.<br>The vector address is expressed as H'0400 + (vector number × 2). For example, when DTVEC6 to DTVEC0 = H'10, the vector address is H'0420.<br>When the bit SWDTE is 0, these bits can be written. | R/W |
| 5 | DTVEC5 | | | R/W |
| 4 | DTVEC4 | | | R/W |
| 3 | DTVEC3 | | | R/W |
| 2 | DTVEC2 | | | R/W |
| 1 | DTVEC1 | | | R/W |
| 0 | DTVEC0 | | | R/W |

DTVECR enables or disables DTC activation by software, and sets a vector number for the software activation interrupt.

chanhvinh.com



## 11.3 Activation Sources

The DTC operates when activated by an interrupt request or by a write to DTVECR by software. An interrupt request can be designated by the DTCER bit. At the end of a data transfer (or the last consecutive transfer in the case of chain transfer), the activation source interrupt flag or corresponding bit to DTCER is cleared. For example, the activation source flag, in the case of SCI3_1_RXI, is the RDRF flag of SCI3_1.

When an interrupt has been designated a DTC activation source, existing CPU mask level and interrupt controller priorities have no effect. If there is more than one activation source at the same time, the DTC operates in accordance with the default priorities for the interrupt sources.

Table 11.2 shows a relationship between activation sources and DTCER clear conditions. Figure 11.2 shows a block diagram of DTC activation source control. For details, see section 4, Interrupt Controller.

**Table 11.2 Relationship between Activation Sources and DTCER Clearing**

| Activation Source | DISEL = 0 and Specified Number of Transfers Has Not Ended | DISEL = 1 or Specified Number of Transfers Has Ended |
|---|---|---|
| Activation by software | SWDTE bit is cleared to 0 | • SWDTE bit remains set to 1<br>• Interrupt request to CPU |
| Activation by an interrupt | • Corresponding DTCER bit remains set to 1.<br>• Activation source flag is cleared to 0. | • Corresponding DTCER bit is cleared to 0.<br>• Activation source flag remains set to 1.<br>• Interrupt that became the activation source is requested to the CPU. |

chanhvinh.com



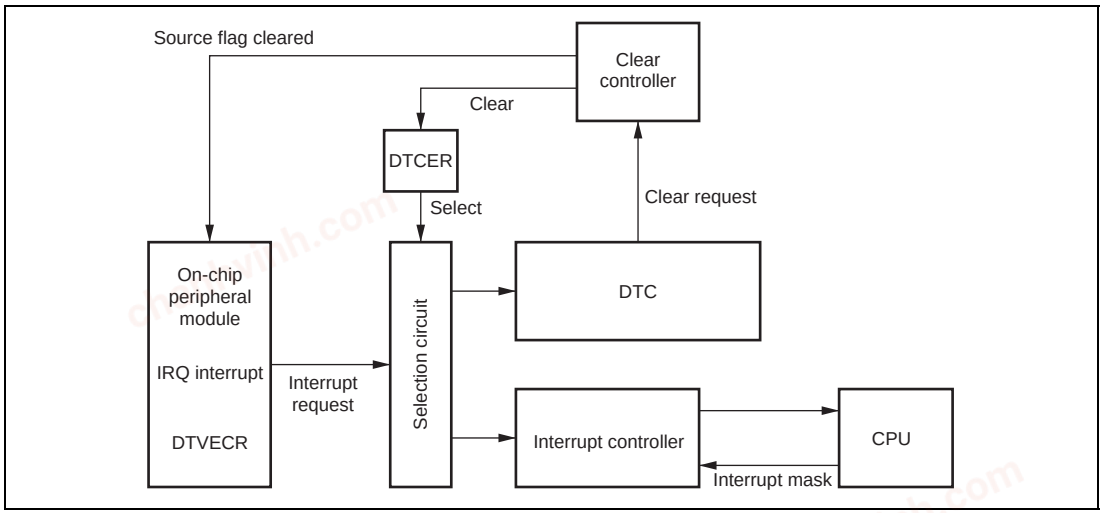



**Figure 11.2 Block Diagram of DTC Activation Source Control**

chanhvinh.com



## 11.4 Location of Register Information and DTC Vector Table

Locate the register information in the on-chip RAM. Register information should be located at the address that is multiple of four. Locating the register information in address space is shown in figure 11.3. Locate the MRA, SAR, MRB, DAR, CRA, and CRB registers, in that order, from the start address of the register information. In the case of chain transfer, register information should be located in consecutive areas as shown in figure 11.3 and the register information start address should be located at the corresponding vector address to the activation source. Figure 11.4 shows correspondences between the DTC vector address and register information. The DTC reads the start address of the register information from the vector address set for each activation source, and then reads the register information from that start address.

When the DTC is activated by software, the vector address is obtained from: H'0400 + (DTVECR[6:0] × 2). For example, if VOFR and DTVECR are H'0000 and H'18 respectively, the vector address is H'0430.

The configuration of the vector address is a 2-byte unit. These two bytes specify the lower bits of the start address. Whenever the DTC is used, set the VOFR to H'0000 (its default value).

chanhvinh.com



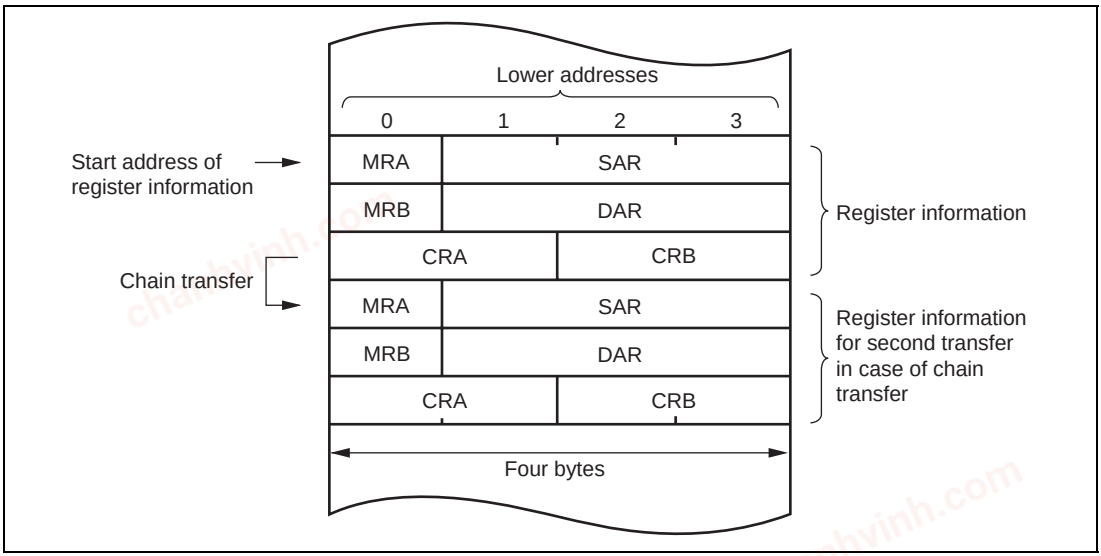



**Figure 11.3 Locating DTC Register Information in Address Space**

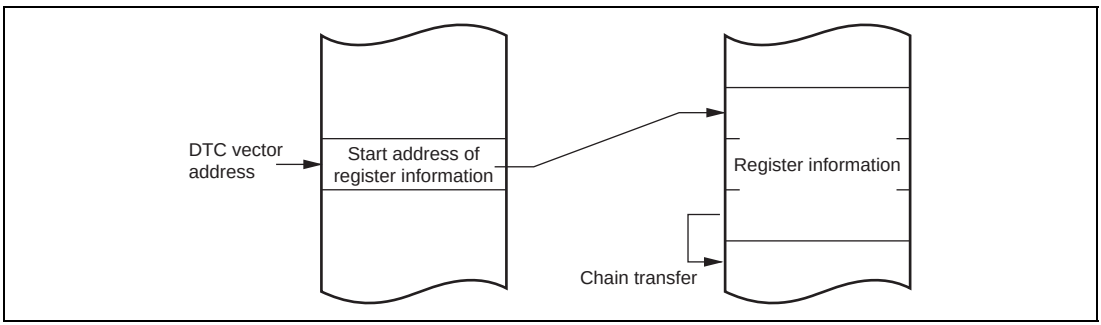



**Figure 11.4 Correspondence between DTC Vector Address and Register Information**

chanhvinh.com



**Table 11.3 Interrupt Sources, DTC Vector Addresses, and Corresponding DTCEs**

| Origin of Activation Source | Activation Source | Vector Number | Vector Address*1 | DTCE*5 | Priority |
|---|---|---|---|---|---|
| Software | Write to DTVECR | DTVECR | H'0400 + (DTVECR[6:0] × 2) | — | High |
| External pin | IRQ0 | 22 | H'42C to H'42D | DTCEA7 | |
| | IRQ1 | 23 | H'42E to H'42F | DTCEA6 | |
| | IRQ2 | 24 | H'430 to H'431 | DTCEA5 | |
| | IRQ3 | 25 | H'432 to H'433 | DTCEA4 | |
| | IRQ4 | 26 | H'434 to H'435 | DTCEA3 | |
| | IRQ5 | 27 | H'436 to H'437 | DTCEA2 | |
| | IRQ6 | 28 | H'438 to H'439 | DTCEA1 | |
| | IRQ7 | 29 | H'43A to H'43B | DTCEA0 | |
| A/D converter unit 1 | IADEND_1 (conversion completion) | 30 | H'43C to H'43D | DTCEB7 | |
| | IADCMP_1 (compare condition match) | 31 | H'43E to H'43F | DTCEB6 | |
| A/D converter unit 2*2 | IADEND_2 (conversion completion) | 32 | H'440 to H'441 | DTCEB5 | |
| | IADCMP_2 (compare condition match) | 33 | H'442 to H'443 | DTCEB4 | |
| ELC | ELC1FP (ELSR12 event occurrence) | 35 | H'446 to H'447 | DTCEB3 | |
| | ELC2FP (ELSR30 event occurrence) | 36 | H'448 to H'449 | DTCEB2 | |
| SCI3 channel 1 | SCI3_1 RXI | 38 | H'44C to H'44D | DTCEB1 | |
| | SCI3_1 TXI | 39 | H'44E to H'44F | DTCEB0 | |
| SCI3 channel 2 | SCI3_2 RXI | 42 | H'454 to H'455 | DTCEC7 | |
| | SCI3_2 TXI | 43 | H'456 to H'457 | DTCEC6 | |
| SCI3 channel 3 | SCI3_3 RXI | 46 | H'45C to H'45D | DTCEC5 | |
| | SCI3_3 TXI | 47 | H'45E to H'45F | DTCEC4 | |
| IIC2/SSU | IIC2/SSU_RXI | 60 | H'478 to H'479 | DTCED7 | |
| | IIC3/SSU_TXI | 61 | H'47A to H'47B | DTCED6 | Low |

chanhvinh.com



| Origin of Activation Source | Activation Source | Vector Number | Vector Address*1 | DTCE*5 | Priority |
|---|---|---|---|---|---|
| Timer RC*3 | ITCMA Input capture A/ compare match A | 71 | H'48E to H'48F | DTCED3 | High |
| | ITCMB Input capture B/ compare match B | 72 | H'490 to H'491 | DTCED2 | |
| | ITCMC Input capture C/ compare match C | 73 | H'492 to H'493 | DTCED1 | |
| | ITCMD Input capture D/ compare match D | 74 | H'494 to H'495 | DTCED0 | |
| Timer RD unit 0 channel 0 | ITDMA0_0 Input capture A/ compare match A | 76 | H'498 to H'499 | DTCEE7 | |
| | ITDMB0_0 Input capture B/ compare match B | 77 | H'49A to H'49B | DTCEE6 | |
| | ITDMC0_0 Input capture C/ compare match C | 78 | H'49C to H'49D | DTCEE5 | |
| | ITDMD0_0 Input capture D/ compare match D | 79 | H'49E to H'49F | DTCEE4 | |
| Timer RD unit 0 channel 1*4 | ITDMA0_1 Input capture A/ compare match A | 82 | H'4A4 to H'4A5 | DTCEE3 | |
| | ITDMB0_1 Input capture B/ compare match B | 83 | H'4A6 to H'4A7 | DTCEE2 | |
| | ITDMC0_1 Input capture C/ compare match C | 84 | H'4A8 to H'4A9 | DTCEE1 | |
| | ITDMD0_1 Input capture D/ compare match D | 85 | H'4AA to H'4AB | DTCEE0 | Low |

chanhvinh.com



| Origin of Activation Source | Activation Source | Vector Number | Vector Address*1 | DTCE*5 | Priority |
|---|---|---|---|---|---|
| Timer RD unit 1 channel 2*4 | ITDMA1_2 Input capture A/ compare match A | 87 | H'4AE to H'4AF | DTCEF7 | High |
| | ITDMB1_2 Input capture B/ compare match B | 88 | H'4B0 to H'4B1 | DTCEF6 | |
| | ITDMC1_2 Input capture C/ compare match C | 89 | H'4B2 to H'4B3 | DTCEF5 | |
| | ITDMD1_2 Input capture D/ compare match D | 90 | H'4B4 to H'4B5 | DTCEF4 | |
| Timer RD unit 1 channel 3*4 | ITDMA1_3 Input capture A/ compare match A | 93 | H'4BA to H'4BB | DTCEF3 | |
| | ITDMB1_3 Input capture B/ compare match B | 94 | H'4BC to H'4BD | DTCEF2 | |
| | ITDMC1_3 Input capture C/ compare match C | 95 | H'4BE to H'4BF | DTCEF1 | |
| | ITDMD1_3 Input capture D/ compare match D | 96 | H'4C0 to H'4C1 | DTCEF0 | |
| Timer RE | ITESC | 100 | H'4C8 to H'4C9 | DTCEG4 | |
| | ITEMI | 101 | H'4CA to H'4CB | DTCEG3 | |
| | ITEHR | 102 | H'4CC to H'4CD | DTCEG2 | |
| | ITEDY | 103 | H'4CE to H'4CF | DTCEG1 | |
| | ITEWK | 104 | H'4D0 to H'4D1 | DTCEG0 | |
| Timer RG | ITGMA Input capture A/ compare match A | 109 | H'4DA to H'4DB | DTCEH3 | |
| | ITGMB Input capture B/ compare match B | 110 | H'4DC to H'4DD | DTCEH2 | Low |

chanhvinh.com



Notes: 1. Vector address indicates the lower 11 bits of vector address when VOFR = H'0000.
2. Supported only in the H8S/20223 and H8S/20235 Groups and reserved in other products.
3. Supported only in the H8S/20103 and H8S/20115 Groups and reserved in other products.
4. Not supported in the H8S/20103 and H8S/20115 Groups, and reserved in these groups.
5. DTCE bits with no corresponding interrupt are reserved. The write value should always be 0.

## 11.5 Operation

The DTC stores register information in the on-chip RAM. When activated, the DTC reads register information in the on-chip RAM and transfers data. After the data transfer, it writes updated register information back to the on-chip RAM. Pre-storage of register information in the on-chip RAM makes it possible to transfer data over any required number of channels. There are three transfer modes: normal mode, repeat mode, and block transfer mode. Setting the CHNE bit to 1 allows a number of transfers with a single activation (chain transfer). Setting the CHNS bit to 1 enables chain transfer only when the transfer counter value is 0.

The 24-bit SAR designates the DTC transfer source address and the 24-bit DAR designates the transfer destination address. After each transfer, SAR and DAR are independently incremented, decremented, or left fixed according to the register information.

Figure 11.5 shows a flowchart of DTC operation, and table 11.4 summarizes the chain transfer conditions (for performing the first and second transfers).

chanhvinh.com



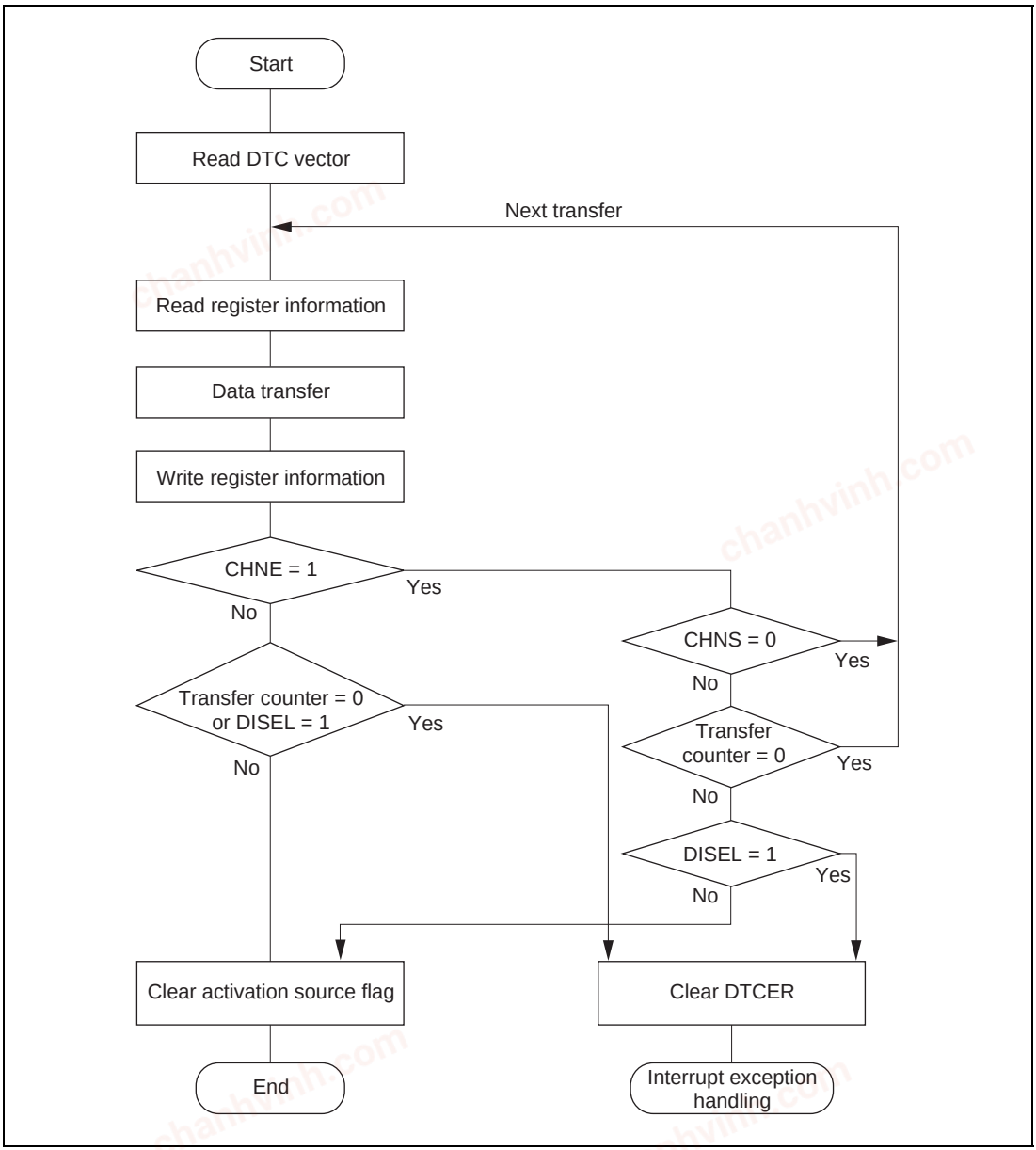



**Figure 11.5 Flowchart of DTC Operation**

chanhvinh.com



**Table 11.4 Chain Transfer Conditions**

| 1st Transfer | | | | 2nd Transfer | | | | |
|---|---|---|---|---|---|---|---|---|
| **CHNE** | **CHNS** | **DISEL** | **CR** | **CHNE** | **CHNS** | **DISEL** | **CR** | **DTC Transfer** |
| 0 | — | 0 | Except 0 | — | — | — | — | Ends at 1st transfer |
| 0 | — | 0 | 0 | — | — | — | — | Ends at 1st transfer |
| 0 | — | 1 | — | — | — | — | — | Interrupt request to CPU |
| 1 | 0 | — | — | 0 | — | 0 | Except 0 | Ends at 2nd transfer |
| | | | | 0 | — | 0 | 0 | Ends at 2nd transfer |
| | | | | 0 | — | 1 | — | Interrupt request to CPU |
| 1 | 1 | 0 | Except 0 | — | — | — | — | Ends at 1st transfer |
| 1 | 1 | — | 0 | 0 | — | 0 | Except 0 | Ends at 2nd transfer |
| | | | | 0 | — | 0 | 0 | Ends at 2nd transfer |
| | | | | 0 | — | 1 | — | Interrupt request to CPU |
| 1 | 1 | 1 | Except 0 | — | — | — | — | Ends at 1st transfer |
| | | | | | | | | Interrupt request to CPU |

### 11.5.1 Normal Mode

In normal mode, one operation transfers one byte or one word of data. Table 11.5 lists the register function in normal mode. From 1 to 65,536 transfers can be specified. Once the specified number of transfers has ended, a CPU interrupt can be requested.

**Table 11.5 Register Function in Normal Mode**

| Name | Abbreviation | Function |
|---|---|---|
| DTC source address register | SAR | Designates transfer source address |
| DTC destination address register | DAR | Designates transfer destination address |
| DTC transfer count register A | CRA | Designates transfer count |
| DTC transfer count register B | CRB | Not used |

chanhvinh.com



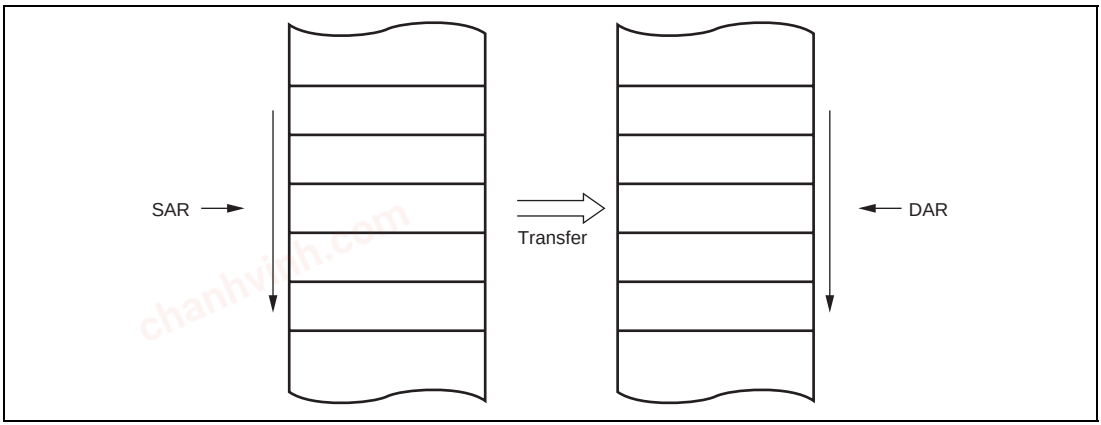



**Figure 11.6 Memory Mapping in Normal Mode**

### 11.5.2 Repeat Mode

In repeat mode, one operation transfers one byte or one word of data. Table 11.6 lists the register function in repeat mode. From 1 to 256 transfers can be specified. Once the specified number of transfers has ended, the initial state of the transfer counter and the address register specified as the repeat area is restored, and transfer is repeated. In repeat mode, the transfer counter value does not reach H'00, therefore CPU interrupts cannot be requested when DISEL = 0.

**Table 11.6 Register Function in Repeat Mode**

| Name | Abbreviation | Function |
|---|---|---|
| DTC source address register | SAR | Designates transfer source address |
| DTC destination address register | DAR | Designates transfer destination address |
| DTC transfer count register AH | CRAH | Holds number of transfers |
| DTC transfer count register AL | CRAL | Designates transfer count |
| DTC transfer count register B | CRB | Not used |

chanhvinh.com



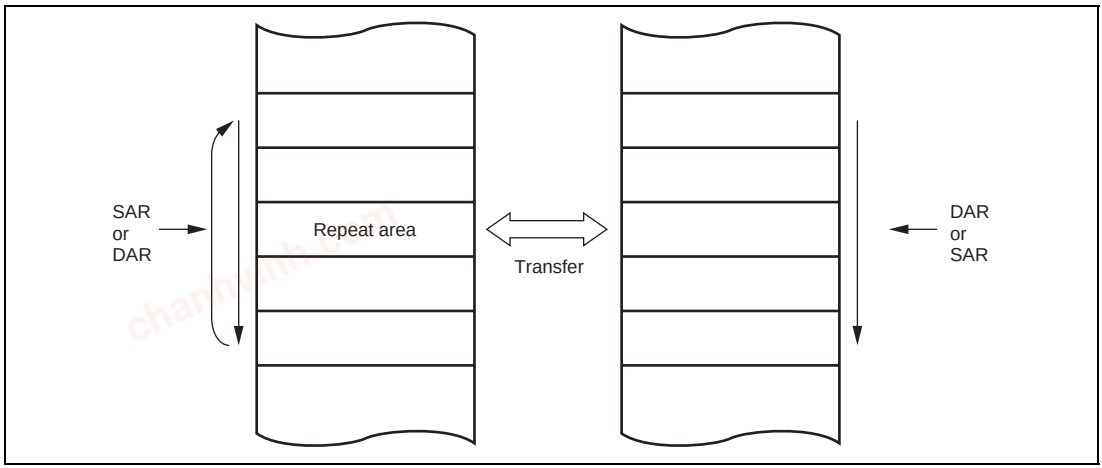



**Figure 11.7 Memory Mapping in Repeat Mode**

### 11.5.3 Block Transfer Mode

In block transfer mode, one operation transfers one block of data. Either the transfer source or the transfer destination is designated as a block area. Table 11.7 lists the register function in block transfer mode. The block size is 1 to 256. When the transfer of one block ends, the initial state of the block size counter and the address register specified as the block area is restored. The other address register is then incremented, decremented, or left fixed according to the register information. From 1 to 65,536 transfers can be specified. Once the specified number of transfers has ended, a CPU interrupt is requested.

**Table 11.7 Register Function in Block Transfer Mode**

| Name | Abbreviation | Function |
|---|---|---|
| DTC source address register | SAR | Designates source address |
| DTC destination address register | DAR | Designates destination address |
| DTC transfer count register AH | CRAH | Holds block size |
| DTC transfer count register AL | CRAL | Designates block size count |
| DTC transfer count register B | CRB | Designates transfer count |

chanhvinh.com



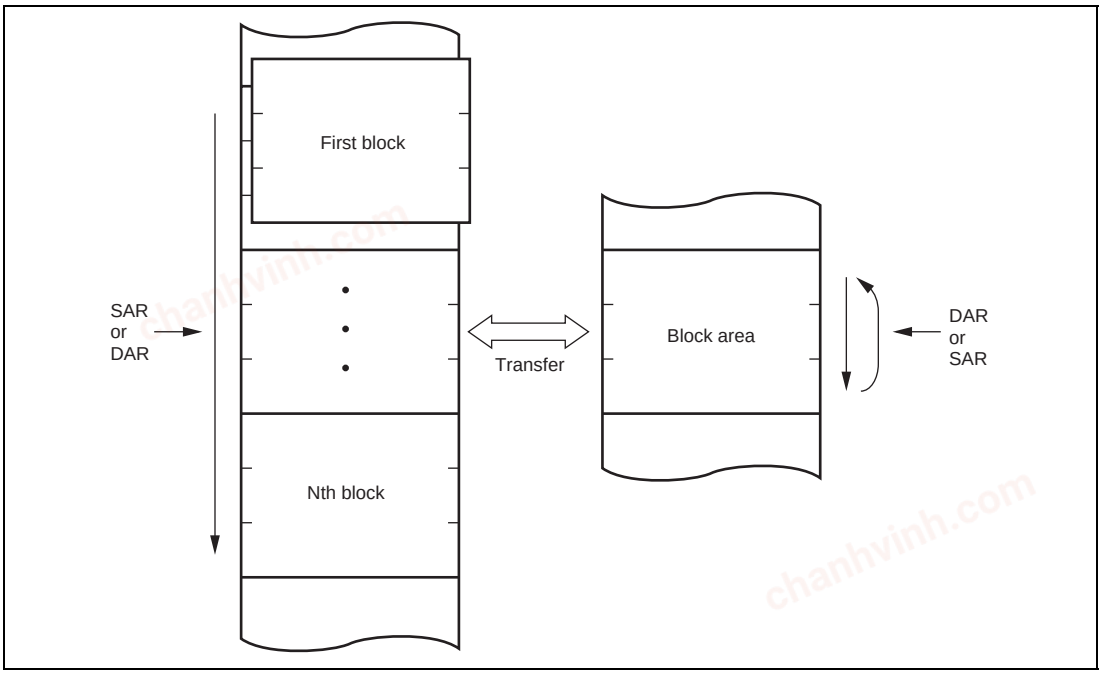



**Figure 11.8 Memory Mapping in Block Transfer Mode**

### 11.5.4 Chain Transfer

Setting the CHNE bit in MRB to 1 enables a number of data transfers to be performed consecutively in response to a single transfer request. SAR, DAR, CRA, CRB, MRA, and MRB can be set independently.

Figure 11.9 shows the operation of chain transfer. When activated, the DTC reads the register information start address stored at the vector address, and then reads the first register information at that start address. The CHNE bit in MRB is checked after the end of data transfer, if the value is 1, the next register information, which is located consecutively, is read and transfer is performed. This operation is repeated until the end of data transfer of register information with CHNE = 0. Setting both the CHNE bit and CHNS bit to 1 enables execution of chain transfer only when the transfer counter value is 0.

In the case of transfer with CHNE set to 1, an interrupt request to the CPU is not generated at the end of the specified number of transfers or by setting of the DISEL bit to 1, and the interrupt source flag for the activation source is not affected.

chanhvinh.com



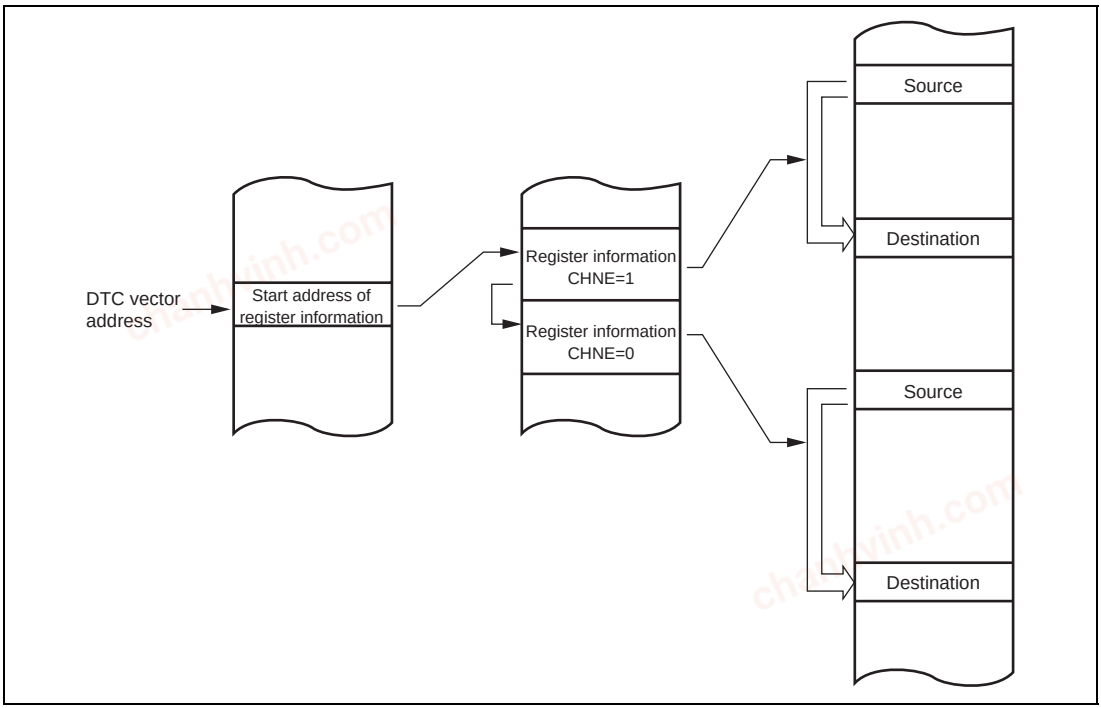



**Figure 11.9 Operation of Chain Transfer**

### 11.5.5 Interrupt Sources

An interrupt request is issued to the CPU when the DTC ends the specified number of data transfers, or when the DTC ends a data transfer for which the DISEL bit was set to 1. In the case of interrupt activation, the interrupt set as the activation source is generated. These interrupts to the CPU are subject to CPU mask level and interrupt controller priority level control.

In the case of activation by software, a software activated data transfer end interrupt (SWDTEND) is generated.

When the DISEL bit is 1 and one data transfer has ended or the specified number of transfers has ended, the SWDTE bit is held at 1 and an SWDTEND interrupt is generated after data transfer ends. The interrupt handling routine should clear the SWDTE bit to 0.

When the DTC is activated by software, an SWDTEND interrupt is not generated during a data transfer wait or during data transfer even if the SWDTE bit is set to 1.

chanhvinh.com



### 11.5.6 Operation Timing

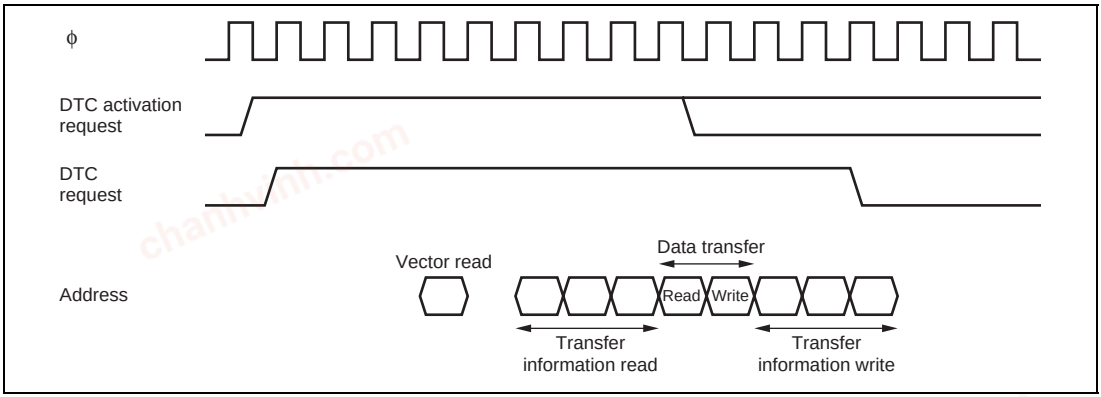



**Figure 11.10 DTC Operation Timing (Example in Normal Mode or Repeat Mode)**

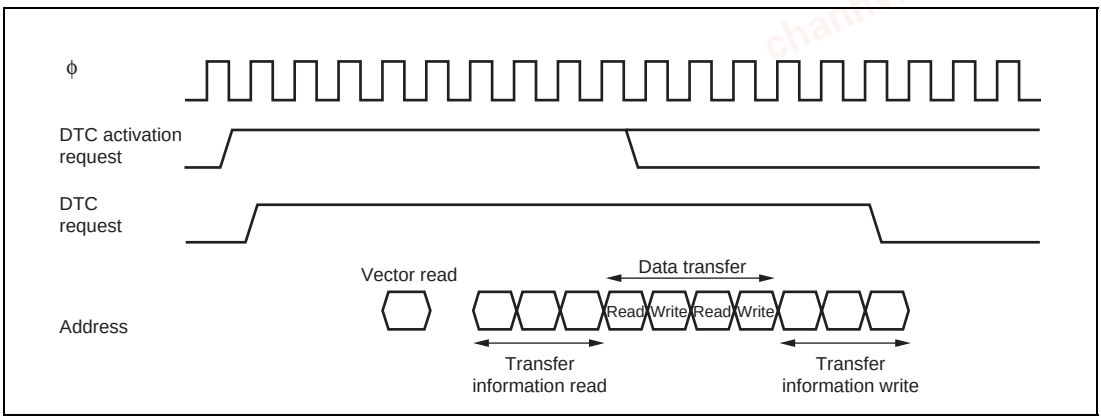



**Figure 11.11 DTC Operation Timing (Example of Block Transfer Mode, with Block Size of 2)**

chanhvinh.com



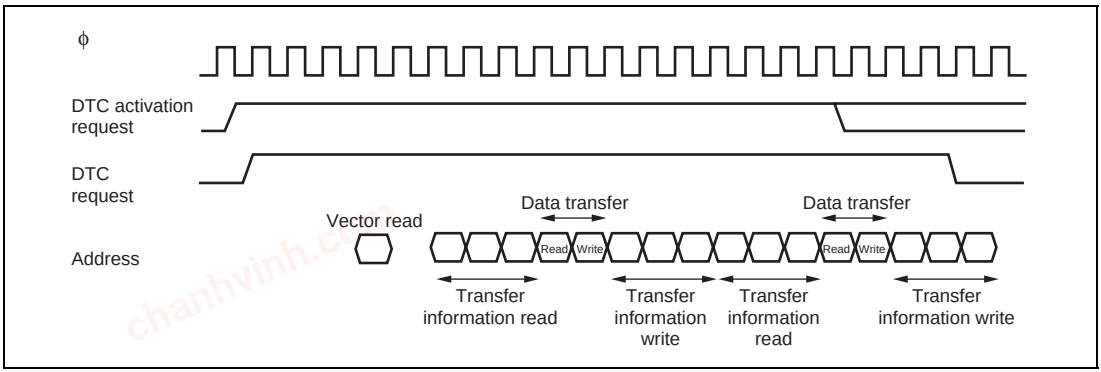



**Figure 11.12 DTC Operation Timing (Example of Chain Transfer)**

### 11.5.7 Number of DTC Execution States

Table 11.8 lists execution state for a single DTC data transfer, and table 11.9 shows the number of states required for each execution status.

**Table 11.8 DTC Execution State**

| Mode | Vector Read I | Register Information Read/Write J | Data Read K | Data Write L | Internal Operations M |
|---|---|---|---|---|---|
| Normal | 1 | 6 | 1 | 1 | 3 |
| Repeat | 1 | 6 | 1 | 1 | 3 |
| Block transfer | 1 | 6 | N | N | 3 |

Legend:

N: Block size (initial setting value of CRAH and CRAL)

chanhvinh.com



**Table 11.9 Number of States Required for Each Execution Status**

| Object to be Accessed | | On-Chip RAM | On-Chip ROM | Internal I/O Register | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Bus width | | 32 | 16 | 8 | | | 16 | | |
| Access states | | 1 | 1 | 2 | 3 | 4 | 2 | 3 | 4 |
| Execution state | Vector read $S_I$ | 1 | 1 | 2 | 3 | 4 | 2 | 3 | 4 |
| | Register information read/write $S_J$ | 1 | — | — | — | — | — | — | — |
| | Byte data read $S_K$ | 1 | 1 | 2 | 3 | 4 | 2 | 3 | 4 |
| | Word data read $S_K$ | 1 | 1 | 4 | 6 | 8 | 2 | 3 | 4 |
| | Byte data write $S_L$ | 1 | 1 | 2 | 3 | 4 | 2 | 3 | 4 |
| | Word data write $S_L$ | 1 | 1 | 4 | 6 | 8 | 2 | 3 | 4 |
| | Internal operation $S_M$ | 1 | | | | | | | |

The number of execution states is calculated from the formula below. Note that Σ means the sum of all transfers activated by one activation source (the number in which the CHNE bit is set to 1 + 1).

$$\text{Number of execution states} = I \cdot S_I + \Sigma (J \cdot S_J + K \cdot S_K + L \cdot S_L) + M \cdot S_M$$

For example, when the DTC vector address table is located in on-chip ROM and data is transferred from the on-chip ROM to an internal I/O register (two-state access) in normal mode, the time required for the DTC operation is 13 states. The time from activation to the end of the data write is 10 states.

chanhvinh.com



## 11.6 Procedures for Using DTC

### 11.6.1 Activation by Interrupt

The procedure for using the DTC with interrupt activation is as follows:

1. Set the MRA, MRB, SAR, DAR, CRA, and CRB register information in the on-chip RAM.
2. Set the start address of the register information in the DTC vector address.
3. Set the corresponding bit in DTCER to 1.
4. Set the enable bits for the interrupt sources to be used as the activation sources to 1. The DTC is activated when an interrupt used as an activation source is generated.
5. After the end of one data transfer, or after the specified number of data transfers have ended, the DTCE bit is cleared to 0 and a CPU interrupt is requested. If the DTC is to continue transferring data, set the DTCE bit to 1.

### 11.6.2 Activation by Software

The procedure for using the DTC with software activation is as follows:

1. Set the MRA, MRB, SAR, DAR, CRA, and CRB register information in the on-chip RAM.
2. Set the start address of the register information in the DTC vector address.
3. Check that the SWDTE bit is 0.
4. Write 1 to the SWDTE bit and the vector number to DTVECR.
5. Check the vector number written to DTVECR.
6. After the end of one data transfer, if the DISEL bit is 0 and a CPU interrupt is not requested, the SWDTE bit is cleared to 0. If the DTC is to continue transferring data, set the SWDTE bit to 1. When the DISEL bit is 1, or after the specified number of data transfers has ended, the SWDTE bit is held at 1 and a CPU interrupt is requested. Clear the SWDTE bit to 0 by an interrupt processing routine.

chanhvinh.com



## 11.7 Examples of Use of the DTC

### 11.7.1 Normal Mode

An example is shown in which the DTC is used to receive 128 bytes of data via the SCI3.

1. Set MRA to fixed source address (SM1 = SM0 = 0), incrementing destination address (DM1 = 1, DM0 = 0), normal mode (MD1 = MD0 = 0), and byte size (Sz = 0). The DTS bit can have any value. Set MRB for one data transfer by one interrupt (CHNE = 0, DISEL = 0). Set the RDR address in SCI3 of SAR, the start address of the RAM area where the data is stored in DAR, and 128 (H'0080) in CRA. CRB can be set to any value.
2. Set the start address of the register information at the DTC vector address.
3. Set the corresponding bit in DTCER to 1.
4. Set the SCI3 to the appropriate receive mode. Set the RIE bit in SCR to 1 to enable the reception complete (RXI) interrupt. Since the generation of a receive error during the SCI reception operation will disable subsequent reception, the CPU should be enabled to accept receive error interrupts.
5. Each time reception of one byte of data ends on the SCI3, the RDRF flag in SSR is set to 1, an RXI interrupt is generated, and the DTC is activated. The receive data is transferred from RDR to RAM by the DTC. DAR is incremented and CRA is decremented. The RDRF flag is automatically cleared to 0.
6. When CRA becomes 0 after the 128 data transfers have ended, the RDRF flag is held at 1, the DTCE bit is cleared to 0, and an RXI interrupt request is sent to the CPU. The interrupt handling routine should perform termination processing.

chanhvinh.com



### 11.7.2 Chain Transfer when Transfer Counter = 0

By executing the second data transfer, and performing re-setting of the first data transfer, only when the counter value for the first data transfer is 0, 256 or more repeat transfers can be performed.

An example is shown in which a 128-Kbyte input buffer is configured. The input buffer is assumed to have been set to start at lower address H'0000. Figure 11.13 shows overview of the chain transfer when the counter value is 0.

1. For the first transfer, set the normal mode for input data. Set fixed transfer source address (G/A, etc.), CRA = H'0000 (65,536 times), and CHNE = 1, CHNS = 1, and DISEL = 0.
2. Prepare the upper 8-bit addresses of the start addresses for each of the 65,536 transfer start addresses for the first data transfer in a separate area (in ROM, etc.). For example, if the input buffer comprises H'200000 to H'21FFFF, prepare H'21 and H'20.
3. For the second transfer, set repeat mode (with the source side as the repeat area) for re-setting the transfer destination address for the first data transfer. Use the upper 8 bits of DAR in the first register information area as the transfer destination. Set CHNE = DISEL = 0. If the above input buffer is specified as H'200000 to H'21FFFF, set the transfer counter to 2.
4. Execute the first data transfer 65,536 times by means of interrupts. When the transfer counter for the first data transfer reaches 0, the second data transfer is started. Set the upper 8 bits of the transfer source address for the first data transfer to H'21. The lower 16 bits of the transfer destination address of the first data transfer and the transfer counter are H'0000.
5. Next, execute the first data transfer the 65,536 times specified for the first data transfer by interrupts. When the transfer counter for the first data transfer reaches 0, the second data transfer is started. Set the upper 8 bits of the transfer source address for the first data transfer to H'20. The lower 16 bits of the transfer destination address of the first data transfer is H'0000.
6. Steps 4 and 5 are repeated endlessly. As repeat mode is specified for the second data transfer, an interrupt request is not sent to the CPU.

chanhvinh.com



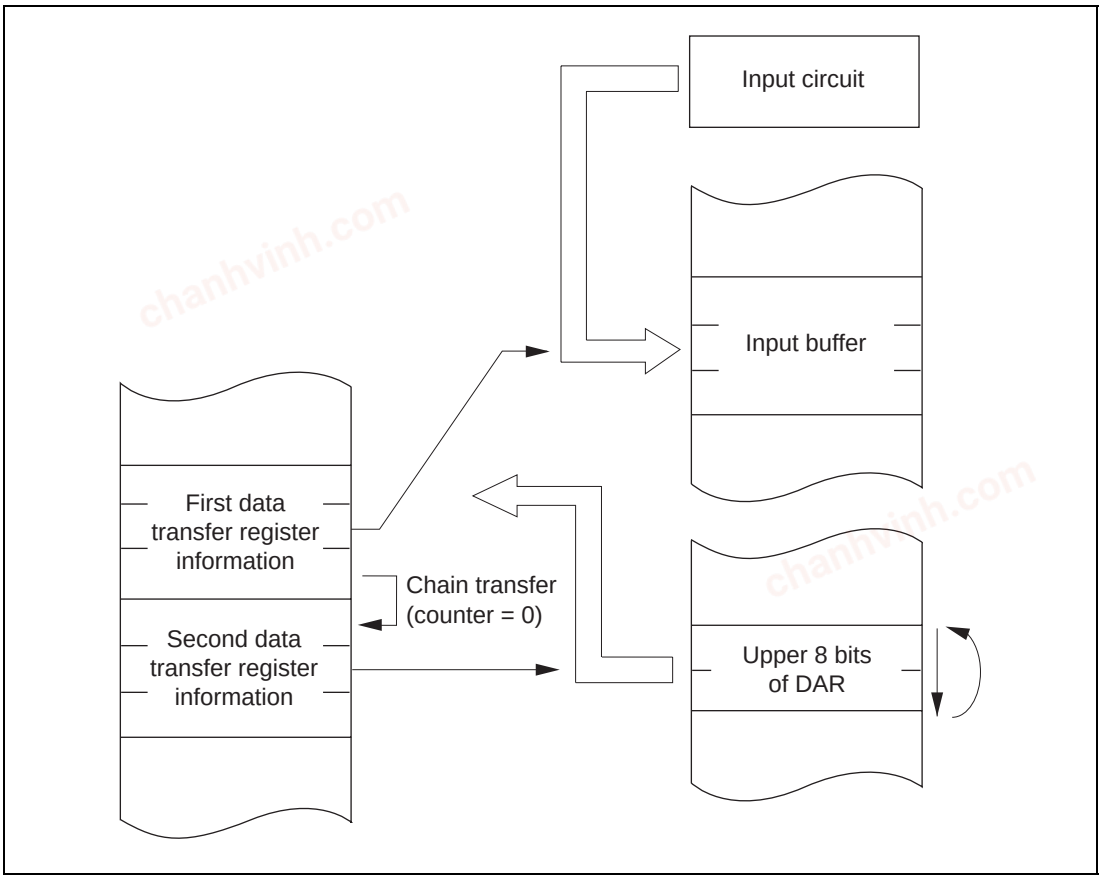



**Figure 11.13 Chain Transfer when Counter = 0**

chanhvinh.com



### 11.7.3 Software Activation

An example is shown in which the DTC is used to transfer a block of 128 bytes of data by software activation. The transfer source address is H'1000 and the destination address is H'2000. The vector number is H'60, so the vector address is H'04C0.

1. Set MRA to incrementing source address (SM1 = 1, SM0 = 0), incrementing destination address (DM1 = 1, DM0 = 0), block transfer mode (MD1 = 1, MD0 = 0), and byte size (Sz = 0). The DTS bit can have any value. Set MRB for one block transfer by one interrupt (CHNE = 0). Set the transfer source address (H'1000) in SAR, the transfer destination address (H'2000) in DAR, and 128 (H'8080) in CRA. Set 1 (H'0001) in CRB.
2. Set the start address of the register information at the DTC vector address (H'04C0).
3. Check that the SWDTE bit in DTVECR is 0. Check that there is currently no transfer activated by software.
4. Write 1 to the SWDTE bit and the vector number (H'60) to DTVECR. The write data is H'E0.
5. Read DTVECR again and check that H'60 is set to the vector number. If it is not, this indicates that the write has failed. This is because an interrupt occurred between steps 3 and 4 and led to a different software activation. To activate this transfer, go back to step 3.
6. If the write was successful, the DTC is activated and a block of 128 bytes of data is transferred.
7. After the transfer, an SWDTEND interrupt occurs. The interrupt handling routine should clear the SWDTE bit to 0 and perform other wrap-up processing.

chanhvinh.com



## 11.8 Usage Notes

### 11.8.1 Module Standby Mode Setting

DTC operation can be disabled or enabled using the module standby control register. The initial value is for DTC operation to be disabled. When the DTC is used, cancel module standby mode. Register access is disabled in module standby mode. Module standby mode cannot be set while the DTC is activated. For details, see section 6, Power-Down Modes.

### 11.8.2 DTCE Bit Setting

For DTCE bit setting, use bit manipulation instructions such as BSET and BCLR. If all interrupts are disabled, multiple activation sources can be set at one time (only at the initial setting) by writing data after executing a dummy read on the relevant register.

### 11.8.3 DTC Activation by SCI3, IIC2/SSU and A/D Converter Interrupt Sources

Interrupts and activation sources of the SCI3, IIC2/SSU, and A/D converter are cleared when the DTC reads or writes the prescribed register. Therefore, when the DTC is activated by an interrupt or activation source, the interrupt or activation source will be retained if a read/write of the relevant register is not included in the last chained data transfer.

The above operation is performed regardless of the DISEL bit setting.

### 11.8.4 Limitation on Usage of the Interrupt Vector Offset Register (VOFR)

When the DTC is used while the interrupt vector offset register (VOFR) holds an offset-address setting, the VOFR setting has no effect. The DTC refers to the vector definition in the vector address area defined by the initial value (H'0000) of VOFR, which leads to the execution of exception processing.

Whenever the DTC is used, set the VOFR to H'0000 (its default value).

chanhvinh.com



# Section 12 Event Link Controller

The event link controller (ELC) connects the events generated by the various peripheral modules to different modules. This function allows direct cooperation between the modules without CPU intervention. A block diagram of the ELC is shown in figure 12.1.

## 12.1 Overview

- Fifty-seven event signals can be directly connected to modules.
- The operation of timer modules can be selected when an event is input to the timer module.
- Events can be connected to ports 3 and 6.

  Single port-pin: An event link can be set for a single specific pin of a port.

  Port group: An event link can be set for a specific group of bits within an 8-bit port.

  In addition, in the specified single pin or group within a port, an event is generated by a change in the value of the linked signals.
- Four channels of events can be generated in arbitrary setting interval using the event-generation timer.

chanhvinh.com



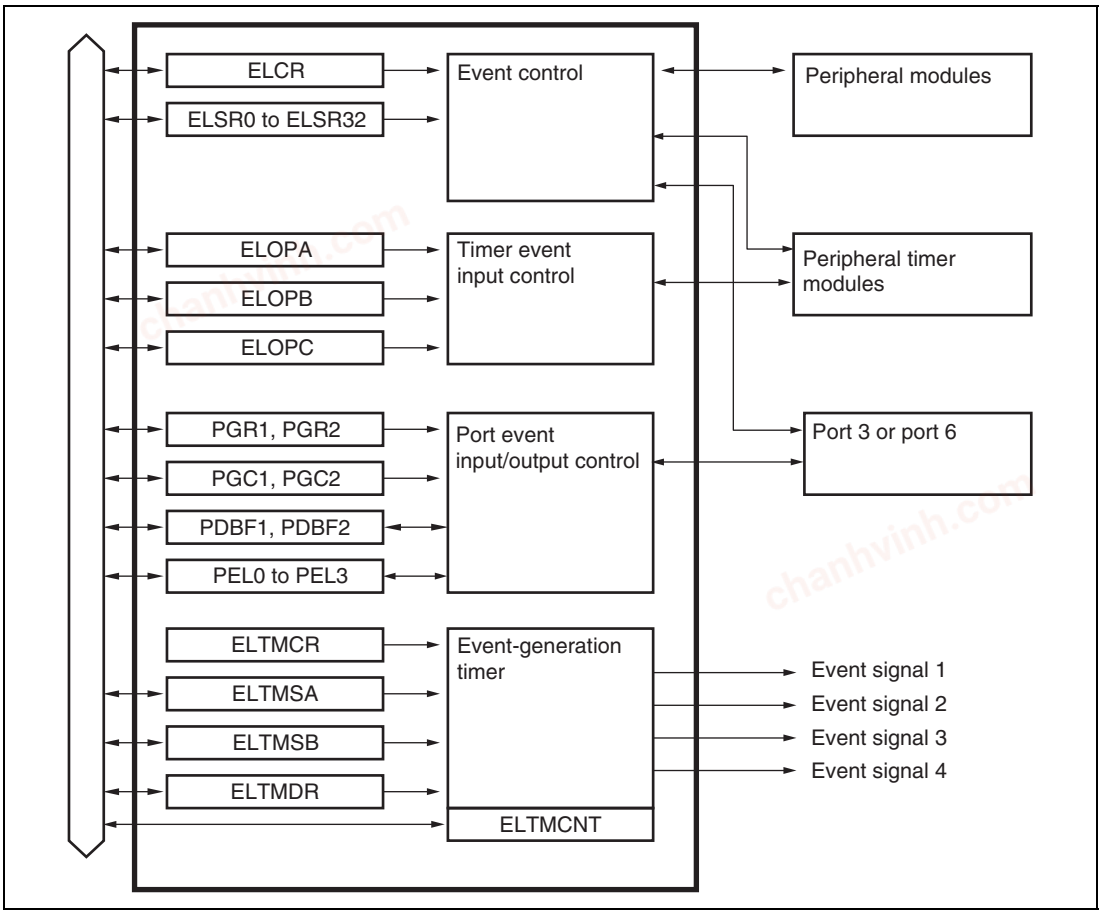



**Figure 12.1 Block Diagram of Event Link Controller**

chanhvinh.com



## 12.2 Register Descriptions

The ELC has the following registers.

- Event link control register (ELCR)
- Event link setting registers 0 to 32 (ELSR0 to ELSR32)
- Event link option setting register A (ELOPA)
- Event link option setting register B (ELOPB)
- Event link option setting register C (ELOPC)
- Port-group setting registers 1 and 2 (PGR1, PGR2)
- Port-group control registers 1 and 2 (PGC1, PGC2)
- Port buffer registers 1 and 2 (PDBF1 and PDBF2)
- Event link port setting registers 0 to 3 (PEL0 to PEL3)
- Event-generation timer control register (ELTMCR)
- Event-generation timer interval setting register A (ELTMSA)
- Event-generation timer interval setting register B (ELTMSB)
- Event-generation timer delay selection register (ELTMDR)
- ELC timer counter (ELTMCNT)

### 12.2.1 Event Link Control Register (ELCR)

Address: H'FF06BC

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | ELCON | — | — | — | — | — | — | — |
| Value after reset: | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | ELCON | All event link enable | 0: Linkage of all the events are disabled.<br>1: Linkage of all the events are enabled. | R/W |
| 6 to 0 | — | Reserved | These bits are read as 1. The write value should be 1. | — |

ELCR controls the operation of the event link controller (ELC) collectively.

chanhvinh.com



### 12.2.2 Event Link Setting Registers 0 to 32 (ELSR0 to ELSR32)

Address: H'FF0680 to H'FF0684, H'FF0688, H'FF068A to H'FF068C, H'FF068E, H'FF068F, H'FF0692, H'FF0693, H'FF0695 to H'FF0698, H'FF069D to H'FF06A0

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | ELSn7 | ELSn6 | ELSn5 | ELSn4 | ELSn3 | ELSn2 | ELSn1 | ELSn0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | ELSn7 | Event link select n7 | 00000000: Linkage of the event is disabled.<br>0000001 to 01100001: Set the number specific to the event signal to be linked.<br>Other than the above: Setting prohibited. | R/W |
| 6 | ELSn6 | Event link select n6 | | R/W |
| 5 | ELSn5 | Event link select n5 | | R/W |
| 4 | ELSn4 | Event link select n4 | | R/W |
| 3 | ELSn3 | Event link select n3 | | R/W |
| 2 | ELSn2 | Event link select n2 | | R/W |
| 1 | ELSn1 | Event link select n1 | | R/W |
| 0 | ELSn0 | Event link select n0 | | R/W |

[Legend]

n: 0 to 32 (except 5 to 7, 9, 13, 16, 17, 20, and 25 to 28)

Each of ELSR0 to ELSR32 specifies an event signal to be linked for the peripheral module. Table 12.1 shows the correspondence between ELSR0 to ELSR32 and the peripheral modules. Table 12.2 shows the correspondence between the event signal names and the numbers specific to the signals.

chanhvinh.com



**Table 12.1 Correspondence between ELSR and Peripheral Modules**

| Register Name | Peripheral Module (Functions) |
|---|---|
| ELSR0 | Timer RA |
| ELSR1 | Timer RB |
| ELSR2*[1] | Timer RC |
| ELSR3 | Timer RD_0 channel 0 |
| ELSR4 | Timer RD_0 channel 1 |
| ELSR8 | Timer RG |
| ELSR10 | AD converter unit 1 |
| ELSR11*[2] | AD converter unit 2 |
| ELSR12*[3] | Interrupts 1 |
| ELSR14 | Output port-group 1 |
| ELSR15 | Output port-group 2 |
| ELSR18 | Input port-group 1 |
| ELSR19 | Input port-group 2 |
| ELSR21 | Single-port 0 |
| ELSR22 | Single-port 1 |
| ELSR23 | Single-port 2 |
| ELSR24 | Single-port 3 |
| ELSR29 | Clock oscillator |
| ELSR30*[3] | Interrupts 2 |
| ELSR31 | DA converter channel 0 |
| ELSR32 | DA converter channel 1 |

Notes: 1. Supported only in the H8S/20103 and H8S/20115 Groups.
2. Supported only in the H8S/20223 and H8S/20235 Groups.
3. When a setting of DTC-transfer-end signals (signal number: H'3D) is added to interrupt functions, make the setting for Interrupts 1.

chanhvinh.com



**Table 12.2 Correspondence between Event Signal Names and ELSn Bit Values**

| ELSn7 to ELSn0 Bit Value (Signal Number) | Name of Event Signal to Set ELSR |
|---|---|
| 00000001 (H'01) | Timer RA underflow |
| 00000010 (H'02) | Timer RB underflow |
| 00000011 (H'03)*[1] | Timer RC overflow |
| 00000100 (H'04)*[1] | Timer RC compare-match A |
| 00000101 (H'05)*[1] | Timer RC compare-match B |
| 00000110 (H'06)*[1] | Timer RC compare-match C |
| 00000111 (H'07)*[1] | Timer RC compare-match D |
| 00001000 (H'08) | Timer RD_0 channel 0 overflow |
| 00001001 (H'09) | Timer RD_0 channel 0 compare-match A |
| 00001010 (H'0A) | Timer RD_0 channel 0 compare-match B |
| 00001011 (H'0B) | Timer RD_0 channel 0 compare-match C |
| 00001100 (H'0C) | Timer RD_0 channel 0 compare-match D |
| 00001101 (H'0D) | Timer RD_0 channel 1 overflow |
| 00001110 (H'0E) | Timer RD_0 channel 1 underflow |
| 00001111 (H'0F) | Timer RD_0 channel 1 compare-match A |
| 00010000 (H'10) | Timer RD_0 channel 1 compare-match B |
| 00010001 (H'11) | Timer RD_0 channel 1 compare-match C |
| 00010010 (H'12) | Timer RD_0 channel 1 compare-match D |
| 00100001 (H'21) | Timer RG overflow |
| 00100010 (H'22) | Timer RG underflow |
| 00100011 (H'23) | Timer RG compare-match A |
| 00100100 (H'24) | Timer RG compare-match B |
| 00101001 (H'29) | AD conversion end in AD converter unit 1 |
| 00101010 (H'2A)*[2] | AD conversion end in AD converter unit 2 |
| 00101100 (H'2C) | Input edge detection on input port-group 1 |
| 00101101 (H'2D) | Input edge detection on input port-group 2 |
| 00101111 (H'2F) | Input edge detection on single input port 0 |
| 00110000 (H'30) | Input edge detection on single input port 1 |
| 00110001 (H'31) | Input edge detection on single input port 2 |
| 00110010 (H'32) | Input edge detection on single input port 3 |
| 00110111 (H'37) | Voltage-drop detection in LVD |

chanhvinh.com



| ELSn7 to ELSn0 Bit Value (Signal Number) | Name of Event Signal to Set ELSR |
|---|---|
| 00111001 (H'39) | CPG backup start |
| 00111010 (H'3A) | WDT increment |
| 00111100 (H'3C) | Timer RE interval (week, day, hour, minute, or second) |
| 00111101 (H'3D) | DTC transfer end |
| 00111110 (H'3E) | Transmit-buffer empty in IIC2/SSU |
| 00111111 (H'3F) | Transmit end in IIC2/SSU |
| 01000000 (H'40) | Receive-buffer full in IIC2/SSU |
| 01000001 (H'41) | Stop-condition detection in IIC2/SSU |
| 01000010 (H'42) | Arbitration loss/overrun error in IIC2/SSU |
| 01000011 (H'43) | NACK detection/conflict error in IIC2/SSU |
| 01001010 (H'4A) | SCI3_1 transmit-buffer empty |
| 01001011 (H'4B) | SCI3_1 transmit end |
| 01001100 (H'4C) | SCI3_1 receive-buffer full |
| 01001101 (H'4D) | SCI3_1 transfer error |
| 01001110 (H'4E) | SCI3_2 transmit-buffer empty |
| 01001111 (H'4F) | SCI3_2 transmit end |
| 01010000 (H'50) | SCI3_2 receive-buffer full |
| 01010001 (H'51) | SCI3_2 transfer error |
| 01010010 (H'52) | SCI3_3 transmit-buffer empty |
| 01010011 (H'53) | SCI3_3 transmit end |
| 01010100 (H'54) | SCI3_3 receive-buffer full |
| 01010101 (H'55) | SCI3_3 transfer error |
| 01011110 (H'5E) | Timer ELC event 0 |
| 01011111 (H'5F) | Timer ELC event 1 |
| 01100000 (H'60) | Timer ELC event 2 |
| 01100001 (H'61) | Timer ELC event 3 |
| Other than the above: Setting prohibited | |

Notes: 1. Selected for the H8S/20103 and H8S/20115 Groups.
2. Selected for the H8S/20223 and H8S/20235 Groups.

chanhvinh.com



### 12.2.3 Event Link Option Setting Register A (ELOPA)

Address: H'FF06B5

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TMRAM[2:1] | | TMRBM[2:1] | | TMRCM[2:1] | | TMRD1M[2:1] | |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7<br>6 | TMRAM [2:1] | Timer RA operation select | 00: Timer starts counting.<br>01: Timer counts events.<br>10: Setting prohibited.<br>11: Events disabled. | R/W |
| 5<br>4 | TMRBM [2:1] | Timer RB operation select | 00: Timer starts counting.<br>01: Timer counts events.<br>10: Setting prohibited.<br>11: Events disabled. | R/W |
| 3<br>2 | TMRCM [2:1]*1 | Timer RC operation select | 00: Timer starts counting.<br>01: Timer counts events.<br>10: Timer performs input-capture operation.*2<br>11: Events disabled. | R/W |
| 1<br>0 | TMRD1M [2:1] | Timer RD_0 channel 0 operation select | 00: Timer starts counting.<br>01: Timer counts events.<br>10: Timer performs input-capture operation.*3<br>11: Events disabled. | R/W |

Notes: 1. Selected only for the H8S/20103 and H8S/20115 Groups and reserved in other products. When writing, b'11 should be written.
2. The TRCCNT value is captured by GRD.
3. The TRDCNT_0 value is captured by GRD_0.

ELOPA determines the operation of timer RA, timer RB, timer RC, and timer RD_0 when an event is input to the timer.

chanhvinh.com



### 12.2.4 Event Link Option Setting Register B (ELOPB)

Address: H'FF06B6

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TMRD2M[2:1] | | — | — | — | — | — | — |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7, 6 | TMRD2M [2:1] | Timer RD_0 channel 1 operation select | 00: Timer starts counting.<br>01: Timer counts events.<br>10: Timer performs input-capture operation.*<br>11: Events disabled. | R/W |
| 5 to 0 | — | Reserved | These bits are read as 1. The write value should be 1. | — |

Note: * The TRDCNT_1 value is captured by GRD_1.

ELOPB determines the operation of timer RD_0 when an event is input to the timer.

### 12.2.5 Event Link Option Setting Register C (ELOPC)

Address: H'FF06B7

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TMRGM[2:1] | | — | — | — | — | — | — |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7, 6 | TMRGM [2:1] | Timer RG operation select | 00: Timer starts counting.<br>01: Timer counts events.<br>10: Timer performs input-capture operation.*<br>11: Events disabled. | R/W |
| 5 to 0 | — | Reserved | These bits are read as 1. The write value should be 1. | — |

Note: * The TRGCNT value is captured by GRB.

ELOPC determines the operation of timer RG when an event is input to the timer.

chanhvinh.com



### 12.2.6 Port-Group Setting Registers 1 and 2 (PGR1 and PGR2)

Address: H'FF06A2, H'FF06A3

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PGRn7 | PGRn6 | PGRn5 | PGRn4 | PGRn3 | PGRn2 | PGRn1 | PGRn0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PGRn7 | Port-group setting n7 | 0: The port bit is not specified as the member of the same group. | R/W |
| 6 | PGRn6 | Port-group setting n6 | 1: The port bit is specified as the member of the same group. | R/W |
| 5 | PGRn5 | Port-group setting n5 | | R/W |
| 4 | PGRn4 | Port-group setting n4 | | R/W |
| 3 | PGRn3 | Port-group setting n3 | | R/W |
| 2 | PGRn2 | Port-group setting n2 | | R/W |
| 1 | PGRn1 | Port-group setting n1 | | R/W |
| 0 | PGRn0 | Port-group setting n0 | | R/W |

[Legend]
n: 1 or 2

PGR specifies each port bit in the same 8-bit I/O port as the member of a group. One to eight port bits can be specified as the members of the same group as required. The correspondence between PGR and ports is shown in table 12.3.

chanhvinh.com



### 12.2.7 Port-Group Control Registers 1 and 2 (PGC1 and PGC2)

Address: H'FF06A6, H'FF06A7

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | PGCOn[2:0] | | | — | PGCOVEn | PGCIn[1:0] | |
| Value after reset: | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is read as 1. The write value should be 1. | — |
| 6 to 4 | PGCOn[2:0] | Port group operation select | 000: 0 is output when the event is input.<br>001: 1 is output when the event is input.<br>010: The toggled (inverted) value is output when the event is input.<br>011: The buffer value is output when the event is input.<br>1XX: The bit value is sifted out in the group (from MSB to LSB) when the event is input. | R/W |
| 3 | — | Reserved | This bit is read as 1. The write value should be 1. | — |
| 2 | PGCOVEn | PDBF overwrite | 0: Overwriting PDBF is disabled.<br>1: Overwriting PDBF is enabled. | R/W |
| 1, 0 | PGCIn[1:0] | Event output edge select | 00: Event is generated upon detection of the rising edge of the external input signal.<br>01: Event is generated upon detection of the falling edge of the external input signal.<br>1X: Event is generated upon detection of both the rising and falling edge of the external input signal. | R/W |

[Legend]
n: 1 or 2
X: Don't care.

For the output port-group, PGC specifies the form of outputting the signal externally via the port when the event signal is input. For the input port-group, PGC enables/disables overwriting of PDBF and specifies the conditions of event generation (edge of the externally input signal).

The correspondence between PGC and ports is shown in table 12.3.

chanhvinh.com



### 12.2.8 Port Buffer Registers 1 and 2 (PDBF1 and PDBF2)

Address: H'FF06AA, H'FF06AB

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PDBFn7 | PDBFn6 | PDBFn5 | PDBFn4 | PDBFn3 | PDBFn2 | PDBFn1 | PDBFn0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PDBFn7 | Port buffer n7 | Data is transferred between PDR and PDBF when an event is input. Write access to the bit specified as a member of the input port-group by the CPU is invalid. For details, see section 12.3, Operation. | R/W |
| 6 | PDBFn6 | Port buffer n6 | | R/W |
| 5 | PDBFn5 | Port buffer n5 | | R/W |
| 4 | PDBFn4 | Port buffer n4 | | R/W |
| 3 | PDBFn3 | Port buffer n3 | | R/W |
| 2 | PDBFn2 | Port buffer n2 | | R/W |
| 1 | PDBFn1 | Port buffer n1 | | R/W |
| 0 | PDBFn0 | Port buffer n0 | | R/W |

[Legend]
n: 1, 2

PDBF is an 8-bit readable/writable register used in combination with PGR. For PDBF operations, see section 12.3, Operation. The correspondence of PDBF and PDR is shown in table 12.3.

**Table 12.3 Registers Related to Port-Groups and Corresponding Port Numbers**

| Port Group Setting Register (PGR) | Port Group Control Register (PGC) | Port Buffer Register (PDBF) | Port Number |
|---|---|---|---|
| PGR1 | PGC1 | PDBF1 | Port 3 |
| PGR2 | PGC2 | PDBF2 | Port 6 |

chanhvinh.com



### 12.2.9 Event Link Port Setting Registers 0 to 3 (PEL0 to PEL3)

Address: H'FF06AD to H'FF06B0

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | PSMn[1:0] | | PSPn[4:3] | | PSPn[2:0] | | |
| Value after reset: | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is read as 1. The write value should be 1. | — |
| 6<br>5 | PSMn[1:0] | Event link specification | • For the output port, data to be output from the port is specified.<br>00: 0 is output when the event is input.<br>01: 1 is output when the event is input.<br>1X: The toggled (inverted) value is output when the event is input.<br>• For the input port, the edge on which the event is to be output is specified.<br>00: Event is output upon detection of the rising edge.<br>01: Event is output upon detection of the falling edge.<br>1X: Event is output upon detection of both the rising and falling edge. | R/W |
| 4<br>3 | PSPn[4:3] | Port number specification | 00: Do not set this value.<br>01: Port 3 (corresponding to PGR1)<br>10: Port 6 (corresponding to PGR2)<br>11: Do not set this value. | R/W |
| 2 | PSPn2 | Bit number specification | A bit number in an 8-bit port is specified. | R/W |
| 1 | PSPn1 | | | R/W |
| 0 | PSPn0 | | | R/W |

[Legend]
n: 0 to 3
X: Don't care.

PEL specifies the 1-bit port (hereinafter referred to as a single-port) to which an event is to be linked, the port operation upon the event signal input, and the conditions of event generation. With this LSI, a total of four bits in either port 3 or port 6 (8-bit ports) can be specified as single-ports.

chanhvinh.com



### 12.2.10 Event-Generation Timer Control Register (ELTMCR)

Address: H'FF06B8

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TMRSTR | — | — | — | CLSRS[3:0] | | | |
| Value after reset: | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | TMRSTR | Timer count start | 0: Counter is stopped.<br>1: Counter is incremented. | R/W |
| 6 to 4 | — | Reserved | These bits are read as 1. The write value should be 1. | — |
| 3 to 0 | CLSRS[3:0] | Clock source (φELC) select | 0000: φ<br>0001: φ/2<br>0010: φ/4<br>0011: φ/8<br>0100: φ/16<br>0101: φ/32<br>0110: φ/64<br>0111: φ/128<br>1000: φ/256<br>1001: φ/512<br>1010: φ/1024<br>1011: φ/2048<br>1100: φ/4096<br>1101: φ/8192<br>1110: Reserved (Counter is stopped.)<br>1111: Reserved (Counter is stopped.) | R/W |

Note: Be sure to stop the counter before changing the clock source.

ELTMCR controls the ELTMCNT operation and selects the clock source.

chanhvinh.com



### 12.2.11 Event-Generation Timer Interval Setting Register A (ELTMSA)

Address: H'FF06B9

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | C1CLS[3:0] | | | | C0CLS[3:0] | | | |
| Value after reset: | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 to 4 | C1CLS[3:0]* | Channel 1 event-generation interval select | 0000: Clock source ϕELC/1<br>0001: Clock source ϕELC/2<br>0010: Clock source ϕELC/4<br>0011: Clock source ϕELC/8<br>0100: Clock source ϕELC/16<br>0101: Clock source ϕELC/32<br>0110: Clock source ϕELC/64<br>0111: Clock source ϕELC/128<br>1000: Clock source ϕELC/256 (initial value)<br>1001: Clock source ϕELC/512<br>1010: Clock source ϕELC/1024<br>1011: Clock source ϕELC/2048<br>1100: Clock source ϕELC/4096<br>1101: Clock source ϕELC/8192<br>1110: Clock source ϕELC/16384<br>1111: Clock source ϕELC/32768 | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 3 to 0 | C0CLS[3:0]* | Channel 0 event-generation interval select | 0000: Clock source φELC/1 | R/W |
| | | | 0001: Clock source φELC/2 | |
| | | | 0010: Clock source φELC/4 | |
| | | | 0011: Clock source φELC/8 | |
| | | | 0100: Clock source φELC/16 | |
| | | | 0101: Clock source φELC/32 | |
| | | | 0110: Clock source φELC/64 | |
| | | | 0111: Clock source φELC/128 | |
| | | | 1000: Clock source φELC/256 (initial value) | |
| | | | 1001: Clock source φELC/512 | |
| | | | 1010: Clock source φELC/1024 | |
| | | | 1011: Clock source φELC/2048 | |
| | | | 1100: Clock source φELC/4096 | |
| | | | 1101: Clock source φELC/8192 | |
| | | | 1110: Clock source φELC/16384 | |
| | | | 1111: Clock source φELC/32768 | |

Notes: Be sure to stop the counter (the TMRSTR bit in ELTMCR is 0) before switching an event generation interval.

* While system operation clock φ has been selected as a clock source (the CLSRS[3:0] bits in ELTMCR are B'0000), do not set to B'0000 to the C1CLS[3:0] and C0CLS[3:0] bits.

ELTMSA determines the event-generation interval for channels 0 and 1, and sets the division ratio for the clock source specified by ELTMCR.

chanhvinh.com



### 12.2.12 Event-Generation Timer Interval Setting Register B (ELTMSB)

Address: H'FF06BA

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | C3CLS[3:0] | | | | C2CLS[3:0] | | | |
| Value after reset: | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 to 4 | C3CLS[3:0]* | Channel 3 event-generation interval select | 0000: Clock source ϕELC/1 | R/W |
| | | | 0001: Clock source ϕELC/2 | |
| | | | 0010: Clock source ϕELC/4 | |
| | | | 0011: Clock source ϕELC/8 | |
| | | | 0100: Clock source ϕELC/16 | |
| | | | 0101: Clock source ϕELC/32 | |
| | | | 0110: Clock source ϕELC/64 | |
| | | | 0111: Clock source ϕELC/128 | |
| | | | 1000: Clock source ϕELC/256 (initial value) | |
| | | | 1001: Clock source ϕELC/512 | |
| | | | 1010: Clock source ϕELC/1024 | |
| | | | 1011: Clock source ϕELC/2048 | |
| | | | 1100: Clock source ϕELC/4096 | |
| | | | 1101: Clock source ϕELC/8192 | |
| | | | 1110: Clock source ϕELC/16384 | |
| | | | 1111: Clock source ϕELC/32768 | |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 3 to 0 | C2CLS[3:0]* | Channel 2 event-generation interval select | 0000: Clock source ϕELC/1 | R/W |
| | | | 0001: Clock source ϕELC/2 | |
| | | | 0010: Clock source ϕELC/4 | |
| | | | 0011: Clock source ϕELC/8 | |
| | | | 0100: Clock source ϕELC/16 | |
| | | | 0101: Clock source ϕELC/32 | |
| | | | 0110: Clock source ϕELC/64 | |
| | | | 0111: Clock source ϕELC/128 | |
| | | | 1000: Clock source ϕELC/256 (initial value) | |
| | | | 1001: Clock source ϕELC/512 | |
| | | | 1010: Clock source ϕELC/1024 | |
| | | | 1011: Clock source ϕELC/2048 | |
| | | | 1100: Clock source ϕELC/4096 | |
| | | | 1101: Clock source ϕELC/8192 | |
| | | | 1110: Clock source ϕELC/16384 | |
| | | | 1111: Clock source ϕELC/32768 | |

Notes: Be sure to stop the counter (the TMRSTR bit in ELTMCR is 0) before switching an event generation interval.

* While system operation clock ϕ has been selected as a clock source (the CLSRS[3:0] bits in ELTMCR are B'0000), do not set to B'0000 to the C3CLS[3:0] and C2CLS[3:0] bits.

ELTMSB determines the event-generation interval for channels 2 and 3, and sets the division ratio for the clock source specified by ELTMCR.

chanhvinh.com



### 12.2.13 Event-Generation Timer Delay Selection Register (ELTMDR)

Address: H'FF06BB

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | C3DLY[1:0] | | C2DLY[1:0] | | C1DLY[1:0] | | C0DLY[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7, 6 | C3DLY[1:0] | Channel 3 delay select | 00: No delay<br>01: 1 clock cycle<br>10: 2 clock cycles<br>11: 3 clock cycles | R/W |
| 5, 4 | C2DLY[1:0] | Channel 2 delay select | 00: No delay<br>01: 1 clock cycle<br>10: 2 clock cycles<br>11: 3 clock cycles | R/W |
| 3, 2 | C1DLY[1:0] | Channel 1 delay select | 00: No delay<br>01: 1 clock cycle<br>10: 2 clock cycles<br>11: 3 clock cycles | R/W |
| 1, 0 | C0DLY[1:0] | Channel 0 delay select | 00: No delay<br>01: 1 clock cycle<br>10: 2 clock cycles<br>11: 3 clock cycles | R/W |

Note: When clock source ϕELC/1 is selected as an event-generation interval in ELTMSA or ELTMSB, there is no delay regardless of settings of the above bits.

ELTMDR determines the necessary delay time, which is the time from the specified event-generation timing (= interval) to the actual generation timing of the event in terms of the cycles of the selected clock source.

chanhvinh.com



### 12.2.14 ELC Timer Counter (ELTMCNT)

Address: H'FF06C0

| Bit: | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

ELTMCNT is a 16-bit readable/writable up-counter. To select the input clock signal to be supplied to the counter, use the CLSRS[3:0] bits in ELTMCR. ELTMCNT cannot be accessed in 8-bit units; it must always be accessed in 16-bit units. The initial value of ELTMCNT is H'0000.

To set the event-generation interval to the time from starting of the timer to generation of the first event, set the counter to 0.

chanhvinh.com



## 12.3 Operation

### 12.3.1 Relation between Interrupt Processing and Event Linking

The modules incorporated in this LSI are provided with the interrupt request status flags and the bits to enable/disable these interrupt requests. When an interrupt request is generated in a module, the corresponding interrupt request status flag is set. If the corresponding interrupt request is enabled then, the interrupt requested is issued to the CPU.

In contrast, the ELC uses interrupt requests (hereinafter referred to as events) generated in modules as event signals that directly activate other modules. This means that the event signal can be used whether or not the interrupt signal is enabled. Figure 12.2 shows the relation between the interrupt processing and ELC.

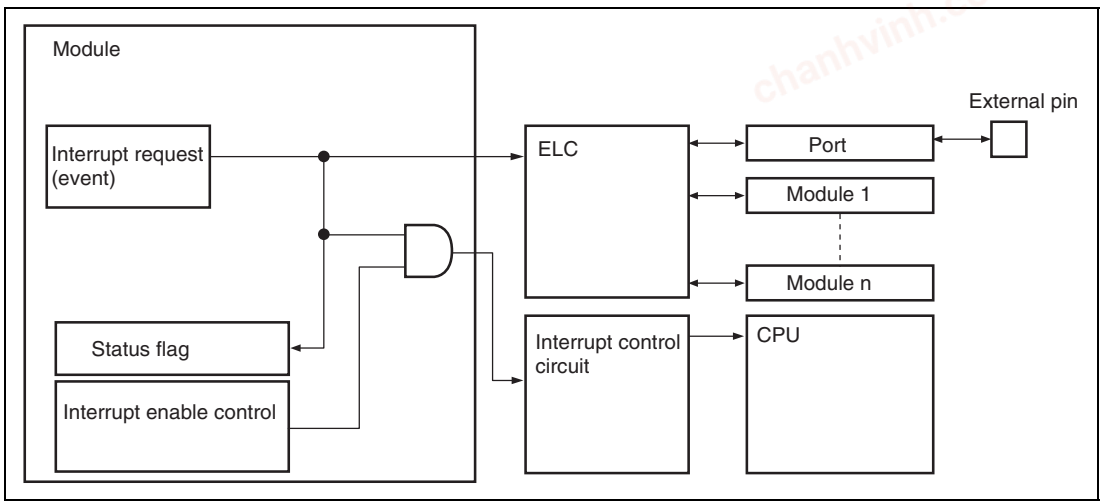



**Figure 12.2 Relation between Interrupt Processing and ELC**

### 12.3.2 Event Linkage

When an event has been set as a trigger in the event-link setting registers (ELSR0 to ELSR32) and then occurs, that event is linked with the corresponding module (activate the module). Only one type of event can be connected with one module. When a module is to be activated by the event-link controller, the operation of the module must be set up in advance. Table 12.4 lists the operations of modules when an event is input.

chanhvinh.com



**Table 12.4 Operations of Modules when Event is Input**

| Module | Operations when Event is Input | | |
|---|---|---|---|
| Timer RA<br>Timer RB<br>Timer RC<br>Timer RD<br>Timer RG | Each timer operates differently depending on the setting of the relevant event link option setting register as below.<br>• Starts counting when an event signal is input.<br>• Counts the input events.<br>• Performs input-capture operation when an event is input. (except timer RA and timer RB) | | |
| A/D converter | Starts A/D conversion when an event signal is input. | | |
| D/A converter | Starts D/A conversion when an event signal is input. | | |
| Output ports | The value of PDR (port data register) changes when an event signal is input. (The value of the signal to be output from the relevant external pin changes.) | Port-groups | The port-group operates differently depending on the settings as below.<br>• Changes the PDR value to the specified value.<br>• Transfers the PDBF values to the PDR.<br>• Shifts out the bit value. |
| | | Single-ports | Changes the PDR value to the specified value. |
| Input ports | When the signal value of the input pin changes. | Port-groups | Generates an event. |
| | | Single-ports | |
| | When an event is input | Port-groups | Transfers the signal value of the external pin to PDBF. |
| | | Single-ports | Event connection is impossible. |
| Clock oscillator | Switches the clock source to the low-speed on-chip oscillator operation. | | |
| Interrupt controller | Issues an interrupt request to the CPU, and the DTC starts to transfer data. | | |

chanhvinh.com



### 12.3.3 Operation of Peripheral Timer Modules When Event is Input

Three different operations are performed depending on the ELOP settings when an event is input.

- Counting-Start Operation
  When an event is input, the timer starts counting, which sets the count start bit* in each timer control register to 1. An event that is input while the count start bit is 1 is invalid.
- Event-Counter Operation
  Event-input is selected as the timer clock source and the timer counts events.
- Input-Capture Operation
  When an event is input, the timer performs input-capture operation.

Note: * See the descriptions on the bit in the relevant timer section.

### 12.3.4 Operation of A/D and D/A Converters When Event is Input

The A/D and D/A converter start A/D and D/A conversion, respectively, which sets the start bits* in the A/D control register and the output enable bits* in the D/A control register to 1.

Note: * See the descriptions on the bit in the A/D and D/A converter sections.

chanhvinh.com



### 12.3.5 Port Operation upon Event Input and Event Generation

The port operation to be performed upon event input to the port can be set and the operation causing the port to generate an event can be set.

#### (1) Single-Ports and Port-Groups

There are two event link modes: event link to single-ports and event link to port-groups. In the former mode, events can be connected to single-ports in an 8-bit port. In the latter mode, events can be connected to port-groups consisting of any two or more bits in the same 8-bit port.

A single-port can be set by specifying any one bit in the port* to which an event can be connected using the PEL register. A port-group can be set by specifying any two or more bits in the port* to which an event can be connected using the PGC register. One input port-group and one output port-group can be set in the same port.

If the port bit is specified as both a single-port and a member of a port-group, both functions are effective when the relevant port is input, whereas only the group-port function is effective when the relevant port is output.

The input or output direction of ports can be selected using the PCR register.

Note: Port 3 and port 6

#### (2) Event Generation by Input Single-Ports

An input single-port generates an event when the signal value of the external pin connected to the relevant port changes. The event-generation condition is specified using the PEL0 to PEL3 registers. An example of operation is shown in figure 12.3.

#### (3) Output Single-Port Operation upon Event Input

When an event is input to an output single-port, the PDR value of the relevant port changes. The specific change of the PDR value is specified using the PEL0 to PEL3 registers. Thus, the change of the PDR value changes the signal value of the external pin connected to the relevant port. An example of operation is shown in figure 12.3.

chanhvinh.com



### (4) Input Port-Group Operation upon Event Input and Event Generation

An input port-group generates an event when the signal value of any one of the external pins connected to the relevant port-group changes. The event-generation condition is specified using the PGC1 and PGC2 registers. When an event is input to an input port-group, the signal value of the external pin upon event input is transferred to PDBF. In this case, only the values of the bits specified as members of the input port-group are transferred. An example of operation is shown in figure 12.4.

### (5) Output Port-Group Operation upon Event Input

When an event is input to an output port-group, the PDR values change to the values according to the PGC1 or PGC2 settings. An example of operation is shown in figure 12.5.

chanhvinh.com



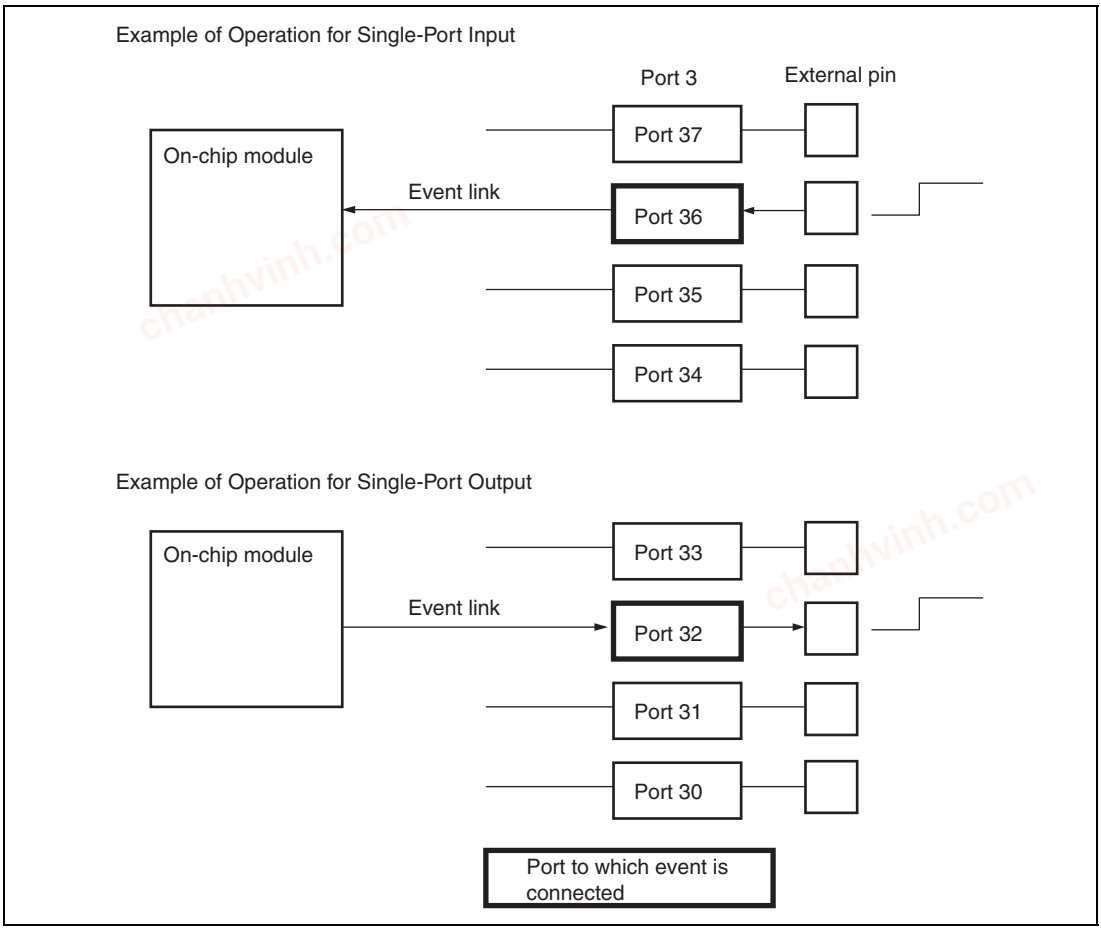



**Figure 12.3 Event Linkage related to Single-Ports**

chanhvinh.com



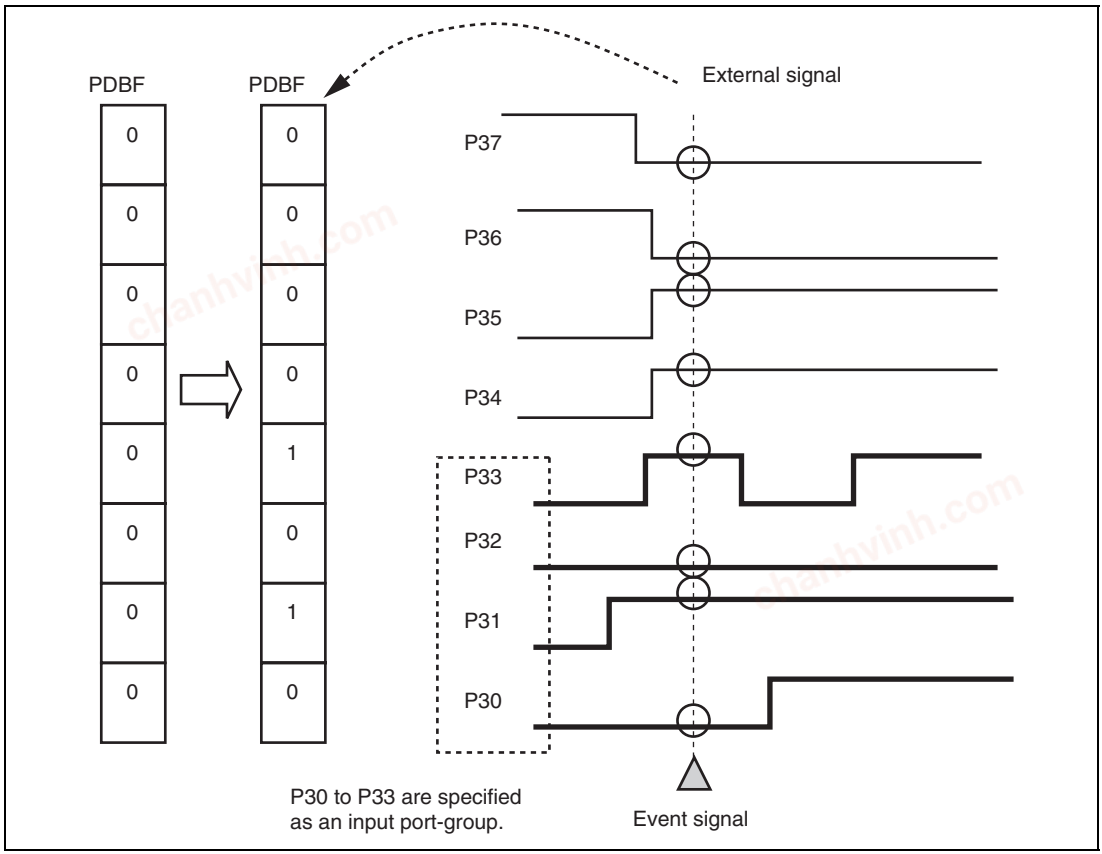



**Figure 12.4 Event Linkage related to Input Port-Groups**

chanhvinh.com



### (6) Operation of Port Buffer Registers

#### (a) Input Port-Groups

When an event is input to an input port-group, the signal value of the external pin of the bit specified as the members of the input port-group is transferred to PDBF. If another event is input to the input port-group in this state, the current PDR value is transferred or not depending on the PGCOVE bit setting in PGC as described below.

- PGCOVE = 0 (overwriting PDBF is disabled)
  If the PDBF value that has been transferred upon the latest event input has already been read by the CPU (or transferred by the DTC), the signal value of the external pin is transferred to PDBF. If not read, the signal value of the external pin is not transferred and the input event is invalid.
- PGCOVE = 1 (overwriting PDBF is enabled)
  When another event is input to an input port-group, the signal value of the external pin is transferred to PDBF.

#### (b) Output Port-Groups

If an output port-group is specified so that it should output the PDBF value, the PDBF value is transferred to PDR when an event is input to the output port-group. In this case, only the values of the bits specified as the members of the output port-group are transferred

If an output port-group is specified so that it should shift out the bit values in the group (PGCO bits = 1xx in PGC), the PDBF data is transferred to PDR, and then the PDR value is shifted bit by bit from MSB to LSB. The initial value to be output to the port-group should be provided in PDBF.

Examples of operation are shown in figures 12.5 and 12.6.

### (7) Restrictions on Writing to PDR or PDBF by CPU

When the ELCON bit in ELCR is set to 1, write access to the following registers is invalid.

- If bits are specified as members of the input port-group and the event-linkage is set for the port-group, write access to the relevant bits in PDBF by the CPU is invalid.
- If port bits are specified as members of the output port-group, write access to the relevant bits in PDR by the CPU is invalid.
- If a port bit is specified as an output single-port and the event-linkage is set (by ELSR) for the port, write access to the relevant bit in PDR by the CPU is invalid.

chanhvinh.com



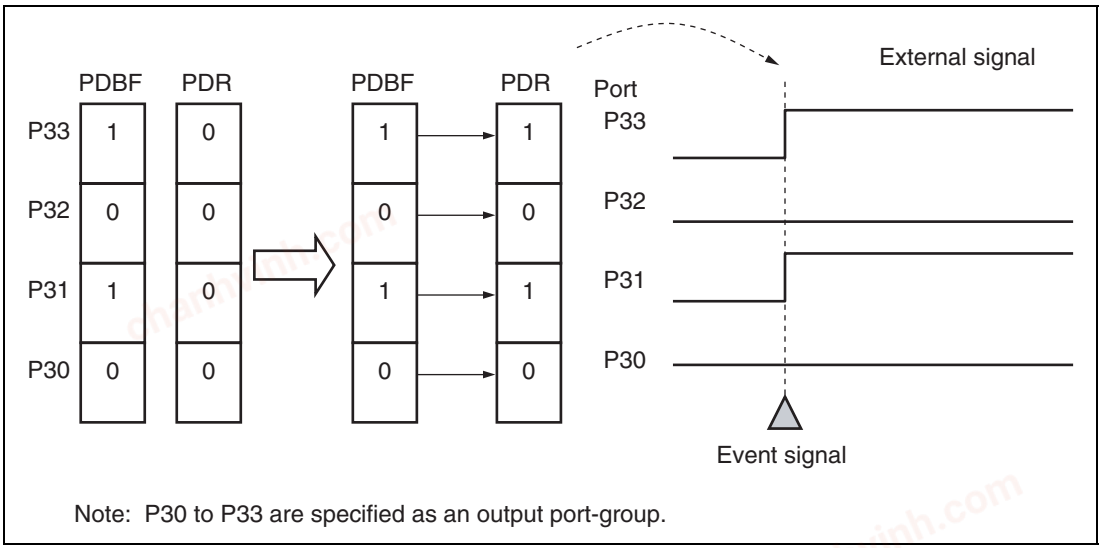



**Figure 12.5 Event Linkage related to Output Port-Groups**

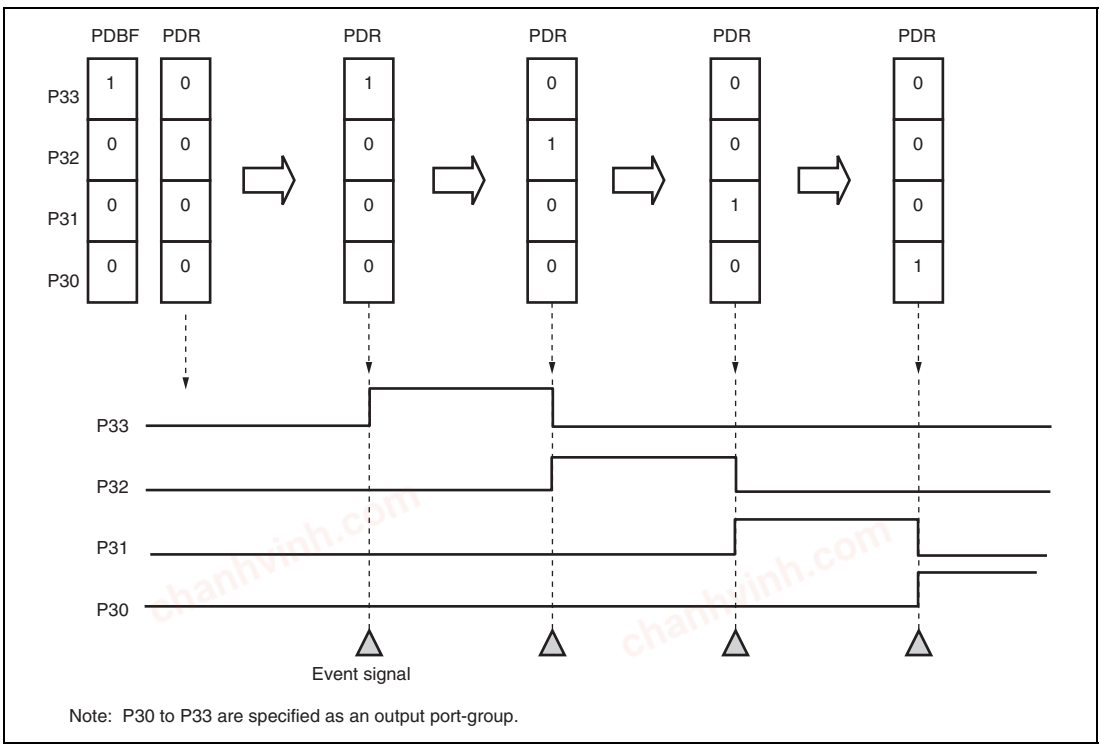



**Figure 12.6 Bit-Shifting Operation of Output Port-Groups**

chanhvinh.com



### 12.3.6 Event-Generation Timer

The event-generation timer can generate an event at specified interval. The generated event can be connected to another module. The features of the timer are given below.

- The interval can be generated using the 16-bit free-running counter.
- The delay time (of 0 to 3 counter clock cycles) can be set, which is the time from the set event-generation timing (= interval) to actual generation of the event.
- Four-channel event output is available (figure 12.8).

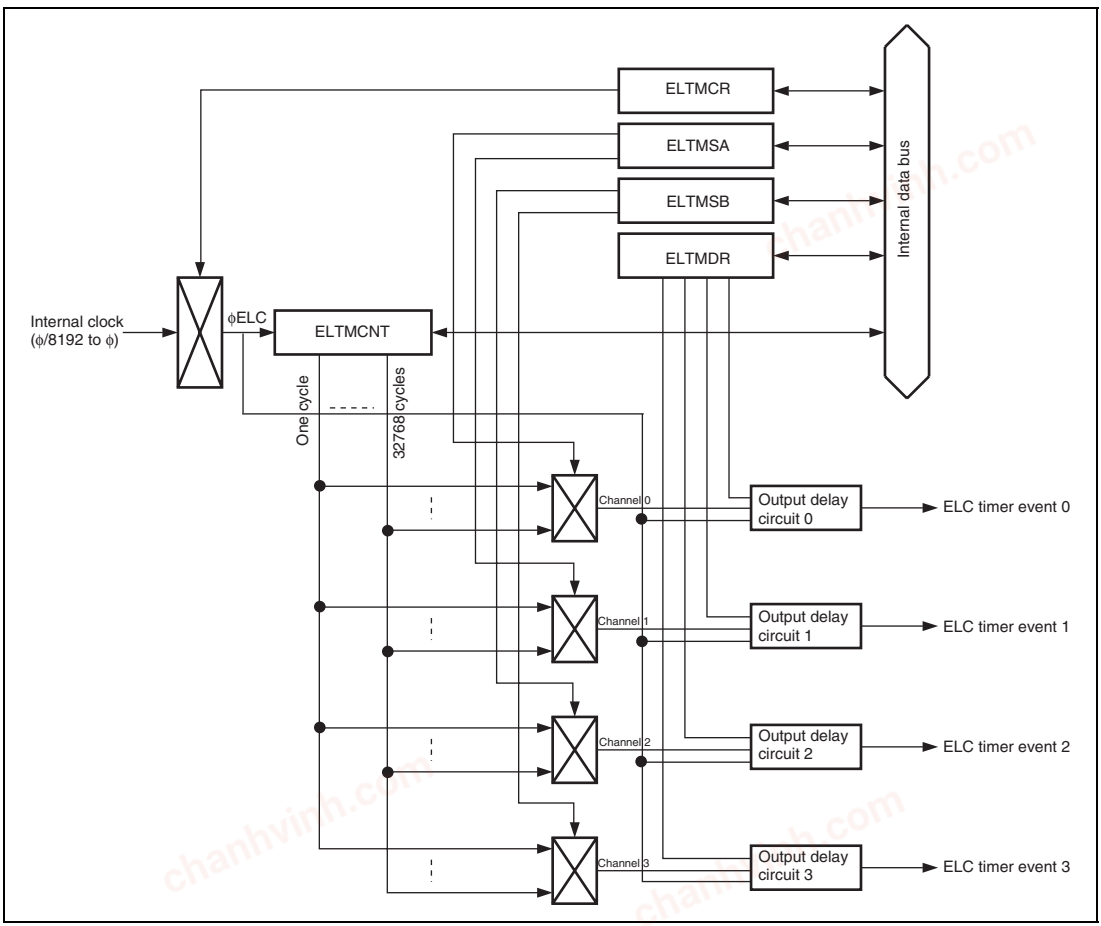



**Figure 12.7 Block Diagram of Event-Generation Timer**

chanhvinh.com



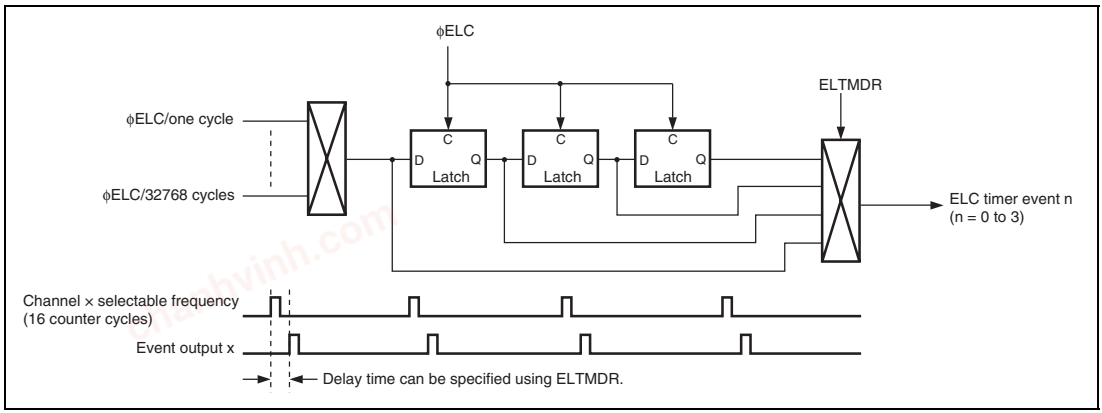



**Figure 12.8 Operation of Event-Generation Timer**

chanhvinh.com



### 12.3.7 Procedure for Linking Events

The following describes the procedure for linking events.

1. Set the operation of the module to which an event is to be linked.
2. To the ELSRn register corresponding to the module to which an event signal is to be linked, set the number of the event signal.
3. If events are to be linked to timers, set the ELOPA to ELOPC registers corresponding to the timers as required.
4. Set the ELCON bit in ELCR to 1, which enables linkage of all the events.
5. Set the operation of the module from which an event is output, and start the module. This allows the event output from the module to start the module to which an event is linked as specified.
6. To stop event linkage of some independent modules, set B'00000000 to the ELSn7 to ELSn0 bits in the ELSRn corresponding to the modules. To stop linkage of all the events, clear the ELCON bit in ELCR to 0.

   If events are linked to ports, set the registers corresponding to the ports as below.

   — PDR: Set the initial values of the output ports.
   — PCR: Set the I/O direction of the ports.
   — PGR: If ports are used as a port-group, set the ports (in bit units) to be grouped.
   — PGC: Set the operation of the port-group.
   — PEL: If ports are used as single-ports, set the ports, the operation of the ports when an event is input, and the condition when an event is generated.

chanhvinh.com



# Section 13 Timer RA

The timer RA is an 8-bit reload timer with a prescaler. The prescaler and the timer are comprised of a reload register and a counter, respectively.

## 13.1 Overview

- Operating mode: 5 modes

  Timer mode: Counts internal count sources.

  Pulse output mode: Counts internal count sources and produces a toggle output in timer underflow.

  Event counter mode: Counts external events.

  Pulse width measurement mode: Measures the pulse width of external pulses.

  Pulse cycle measurement mode: Measures the pulse cycle of external pulses.
- Selection of eight count sources

  ϕ, ϕ/2, ϕ/8, ϕ/32, ϕ/64, ϕ/128, ϕsub, or an external event input to the TRAIO pin.
- An interrupt generated on an underflow of the counter

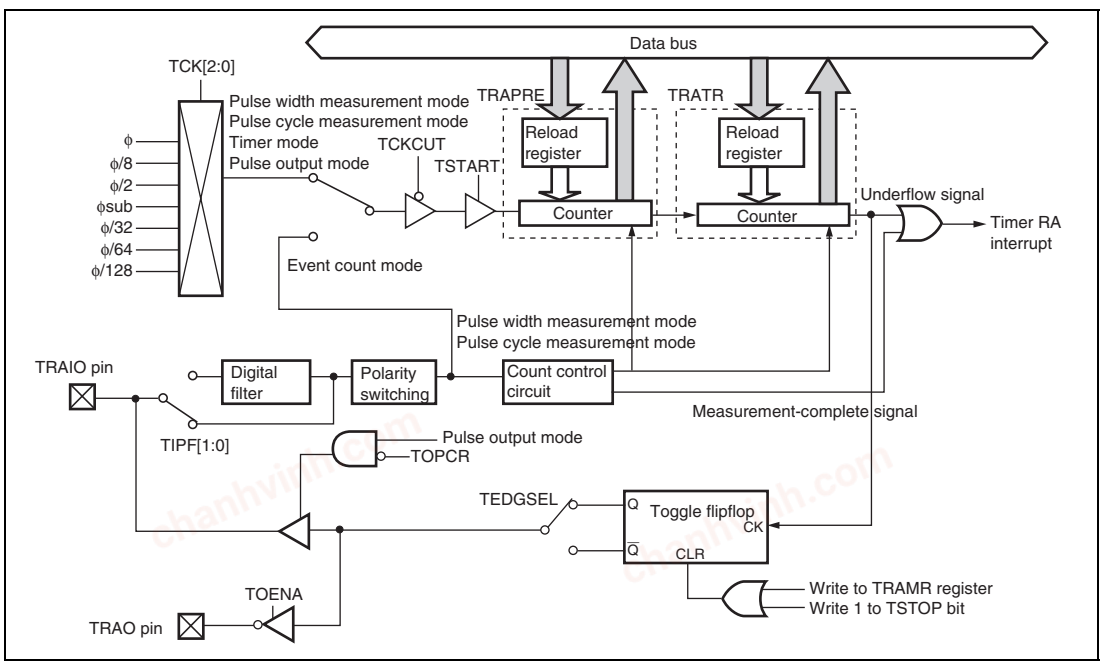



**Figure 13.1 Block Diagram of Timer RA**

chanhvinh.com



Table 13.1 shows the timer RA input/output pins.

**Table 13.1 Pin Configuration**

| Name | Abbreviation | I/O | Function |
|---|---|---|---|
| Timer RA input/output | TRAIO | I/O | External event input and pulse input/output |
| Timer RA output | TRAO | Output | Inverted pulse output of TRAIO output |

## 13.2 Register Descriptions

The timer RA has the following registers:

- Timer RA control register (TRACR)
- Timer RA I/O control register (TRAIOC)
- Timer RA mode register (TRAMR)
- Timer RA prescaler register (TRAPRE)
- Timer RA timer register (TRATR)
- Timer RA interrupt request status register (TRAIR)

chanhvinh.com



### 13.2.1 Timer RA Control Register (TRACR)

Address: H'FF06F0

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | TUNDF | TEDGF | — | TSTOP | TCSTF | TSTART |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7, 6 | — | Reserved | These bits are read as 0. The write valued should be 0. | — |
| 5 | TUNDF | Timer RA underflow flag | [Setting condition]<br>• When timer RA underflows from H'00 to H'FF.<br>[Clearing condition]<br>• When 0 is written to this bit* | R/W |
| 4 | TEDGF | Valid edge detection flag | [Setting conditions]<br>• When the pulse width measurement is completed with TSTART in TRACR = 1, in pulse width measurement mode.<br>• When the timer RA prescaler underflows at the second time after a valid edge of the measurement pulse is input, in pulse cycle measurement mode.<br>[Clearing condition]<br>• When 0 is written to this bit* | R/W |
| 3 | — | Reserved | This bit is read as 0. The write value should be 0. | — |
| 2 | TSTOP | Timer RA count forced stop | 0: Timer RA counting is continued.<br>1: Timer RA counting is forcedly stopped. | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 1 | TCSTF | Timer RA count status flag | 0: Timer RA counting has been stopped.<br>1: Timer RA counting is in progress.<br>[Setting condition]<br>• When 1 is written to TSTART and counting is started.<br>• The start of counting after ELOPA of the event link controller is selected counting by timer RA, the specified event is occurred, and the TSTART bit is set to 1.<br>[Clearing condition]<br>• When 0 is written to TSTART and counting is stopped.<br>• When 1 is written to TSTOP and counting is stopped. | R |
| 0 | TSTART | Timer RA count start | 0: Timer RA counting is stopped.<br>1: Timer RA counting is started. | R/W |

Notes: 1. A MOV instruction should be used to write 0 to this register.
2. The timer RA registers should not be accessed until the TCSTF bit changes after the TSTART bit is set, apart from TRACR which can be read at any time during timer operation.

TRACR controls the timer RA counter and indicates the timer RA state.

- TSTOP bit (timer RA count forced stop)

  Setting this bit to 1 initializes the counter of the timer and the prescaler, bits TSTART and TCSTF, and timer outputs. This bit is always read as 0.

chanhvinh.com



### 13.2.2 Timer RA I/O Control Register (TRAIOC)

Address: H'FF06F1

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TIOGT[1:0] | | TIPF[1:0] | | TIOSEL | TOENA | TOPCR | TEDGSEL |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7, 6 | TIOGT[1:0] | TRAIO event input control | 00: Input control is not performed. (Events are always enabled.)<br>01: Input control is performed. (Events are enabled when IRQ2 input is high.)<br>10: Setting prohibited<br>11: Setting prohibited | R/W |
| 5, 4 | TIPF[1:0] | TRAIO input filter select | 00: No filter operation<br>01: Filtered (Sampled at $\phi$)<br>10: Filtered (Sampled at $\phi$/8)<br>11: Filtered (Sampled at $\phi$/32)<br>These bits should be set to B'00 in timer mode and pulse output mode. | R/W |
| 3 | TIOSEL | TRAIO input select | 0: Input from the TRAIO pin<br>1: Input from the LIN | R/W |
| 2 | TOENA | TRAO output enable | 0: TRAO outputs are disabled.<br>1: TRAO outputs are enabled.<br>This bit should be set to 0 except in event counter mode and pulse output mode. | R/W |
| 1 | TOPCR | TRAIO output control | 0: TRAIO outputs are enabled.<br>1: TRAIO outputs are disabled.<br>This bit should be set to 0 except in pulse output mode. | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 0 | TEDGSEL | Input/output polarity switch | • Timer mode<br>This bit should be set to 0.<br>• Pulse output mode<br>0: The initial value of TRAIO output is set at a high level.<br>1: The initial value of TRAIO output is set at a low level.<br>• Event count mode<br>0: Counter incremented at the TRAIO input rising edge. The initial value of TRAIO output is set at a low level.<br>1: Counter incremented at the TRAIO input falling edge. The initial value of TRAIO output is set at a high level.<br>• Pulse width measurement mode<br>0: Measures the low-level width of TRAIO input.<br>1: Measures the high-level width of TRAIO input.<br>• Pulse cycle measurement mode<br>0: Measures from the rising edge of the measurement pulse to the next rising edge.<br>1: Measures from the falling edge of the measurement pulse to the next falling edge. | R/W |

Note: When TCSTF = 1, do not rewrite this register.

- TIOGT1 bit and TIOGT0 bit (TRAIO event input control 1 and 0)
  These bits control input events in event counter mode.

- TIPF1 bit and TIPF0 bit (TRAIO input filter select 1 and 0)
  If filtered operation is selected, the input is determined when the same value is sampled three times in succession from the TRAIO pin.

chanhvinh.com



### 13.2.3 Timer RA Mode Register (TRAMR)

Address: H'FF06F2

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TCKCUT | TCK[2:0] | | | — | TMOD[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | TCKCUT | Timer RA count source cutoff | 0: Count source is supplied.<br>1: Count source is cut off. | R/W |
| 6 to 4 | TCK[2:0] | Timer RA count source select | 000: $\phi$<br>001: $\phi$/8<br>010: Setting prohibited<br>011: $\phi$/2<br>100: $\phi$sub<br>101: $\phi$/32<br>110: $\phi$/64<br>111: $\phi$/128 | R/W |
| 3 | — | Reserved | This bit is read as 0. The write value should be 0. | — |
| 2 to 0 | TMOD[2:0] | Timer RA operating mode select | 000: Timer mode<br>001: Pulse output mode<br>010: Event count mode<br>011: Pulse width measurement mode<br>100: Pulse cycle measurement mode<br>101: Setting prohibited<br>110: Setting prohibited<br>111: Setting prohibited | R/W |

Note: The counting should be stopped (when both the TSTART and TCSTF bits in TRACR are 0) when this register is modified.

- TCK2 bit and TCK0 bit (timer RA count source select)
  A count source is selected if the mode is not the event count mode.
- TMOD2 bit to TMOD0 bit (timer RA operating mode select)
  Writing to TRAMR initializes the output level.

chanhvinh.com



### 13.2.4 Timer RA Interrupt Enable Status Register (TRAIR)

Address: H'FF06F5

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TRAIE | TRAIF | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | TRAIE | Timer RA interrupt request enable | 0: Timer RA interrupt requests are disabled.<br>1: Timer RA interrupt requests are enabled. | R/W |
| 6 | TRAIF | Timer RA interrupt request flag | [Setting conditions]<br>• When the timer RA underflows.<br>• When the input pulse measurement is completed in pulse width measurement mode.<br>• When the timer RA prescaler underflows at the second time after a valid edge of measurement pulse is input, in pulse cycle measurement mode.<br>[Clearing condition]<br>• When 1 is read from the bit and then 0 is written to. | R/W |
| 5 to 0 | — | Reserved | This bit is read as 0. The write value should be 0. | — |

chanhvinh.com



### 13.2.5 Timer RA Prescaler Register (TRAPRE)

Address: H'FF06F3

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

TRAPRE consists of a reload register and an 8-bit counter, each with an initial value of H'FF.

If a down-count is performed using the count source selected with TRAMR and an underflow occurs, the value of the reload register is loaded to the counter. The underflow becomes a count source for TRATR.

The reload register and the counter are assigned to the same address. On write, a value is written to the reload register, and on read, a counter value is read. During a write to TRAPRE the load timing from the reload register to the counter differs between counting in progress and counting stopped. Writing to TRAPRE when counting is stopped causes the data to be written to both the reload register and the counter. Writing to TRAPRE during counting causes the new value to be written to the reload register after four cycles of count source, and to be loaded to the counter in synchronization with the next count source.

chanhvinh.com



### 13.2.6 Timer RA Timer Register (TRATR)

Address: H'FF06F4

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

TRATR consists of a reload register and an 8-bit counter, each with an initial value of H'FF. TRATR performs a down-count of the prescaler underflows. When an underflow occurs in TRATR, the value of the reload register is loaded to the counter and a timer RA interrupt request is generated at the same time.

The reload register and the counter are assigned to the same address. On write, a value is written to the reload register, and on read, a counter value is read. However, on read in pulse cycle measurement mode, a value in the read buffer is read. During a write to TRATR the load timing from the reload register to the counter differs between counting in progress and counting stopped. Writing to TRATR when counting is stopped causes the data to be written to both the reload register and the counter. Writing to TRATR during counting causes the new value to be written to the reload register in synchronization with an underflow of the prescaler first after four counts of the count source, and to be loaded to the counter in synchronization with the next underflow of the prescaler.

TRAPRE and TRATR should not be set to H'00 at the same time.

chanhvinh.com



## 13.3 Operation

### 13.3.1 Operations Common to Various Modes

#### (1) Starting and Stopping Operation

Writing the value 1 to the TSTART bit in TRACR starts counting in a set operating mode; writing the value 0 to the TSTART bit stops the counting. The prescaler counts down in the counter clock cycle to be input into the prescaler. The timer counts down using the underflow of the prescaler as a count source.

#### (2) Forced Termination of Operation

Writing 1 to the TSTOP bit in TRACR stops the counting forcibly. When the counting is stopped, the timer counter, the prescaler counter, and any associated flags are initialized while the reload registers of the prescaler and the timer counter are retained.

#### (3) Interrupt Request

An interrupt request is generated on the underflow of the timer RA counter.

#### (4) Reading and Writing Count Value

Reading registers TRAPRE and TRATR reads count values from each register. If a write is performed to TRAPRE or TRATR when the counting is stopped, a specified value is written to both the reload register and the counter.

If a write is performed to the TRAPRE register during counting, first a set value is written to the reload register in synchronization with the count source after four cycles of count source, and the set value is then transferred to the prescaler counter in synchronization with the next count source. If a write is performed to the TRATR register, a set value is written to the reload register in synchronization with the underflow of the prescaler after four cycles of count source, and the set value is transferred to the timer counter in synchronization the next underflow of the prescaler. For this reason, if a write is performed to TRAPRE or TRATR during counting, the value of the counter is not updated immediately after the execution of the write command. Figure 13.2 shows an example operation where a count value is rewritten when the timer RA is counting.

chanhvinh.com



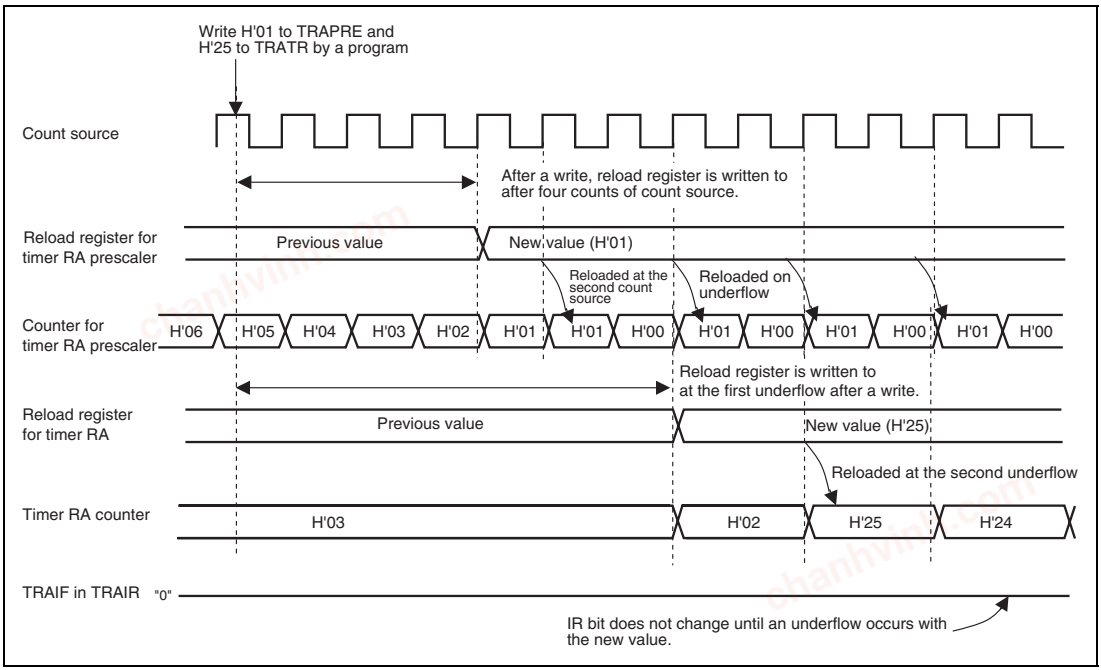



**Figure 13.2 Rewriting Count Value When Timer RA Counting is in Progress**

### 13.3.2 Timer Mode

This mode counts internal clocks as a count source. Setting the TMOD[2:0] bits in TRAMR to B'000 activates the timer mode operation. A count source is selected in terms of the TCK[2:0] bits in TRAMR.

### 13.3.3 Pulse Output Mode

This mode counts internal clocks as a count source, and toggle-outputs pulses from the TRAIO pin each time the counter underflows. Setting the TMOD[2:0] bits in TRAMR to B'001 activates pulse the output mode operation. A count source is selected using the TCK[2:0] bits in TRAMR. The initial output value of the pin is set using the TEDGSEL bit in TRAIOC. By setting the TOENA bit in TRAIOC, a reverse output can be output from the TRAO pin to the TRAIO pin.

chanhvinh.com



### 13.3.4 Event Counter Mode

This mode counts external events that are input from the TRAIO pin as a count source. Setting the TMOD[2:0] bits in TRAMR to B'010 activates the event-counter mode operation. By setting the TEDGSEL bit in TRAIOC, it is possible to specify whether counting is to be performed on the rising or falling edge of an input event from the TRAIO pin. Also, by setting the TIOGT[1:0] bits in TRAIOC, a function enables external event input when the IRQ2 pin is at a high level. Setting the TIPF[1:0] bits in TRAIOC allows applying a filter to external event input. Similar to the pulse output operation mode, a toggle can be output from the TRAO pin in synchronization with an underflow of the timer counter. In event counter mode, even if 1 is written to the TSTART bit, the value of the TCSTF bit will not become 1 unless the corresponding event signal is input. If the event signal is input while TCSTF = 0, the counter value will be the number of times the event has occurred minus 3. If the event signal is input while TCSTF=1, the number time the event has occurred = counter value.

### 13.3.5 Pulse Width Measurement Mode

This mode measures the pulse width of external signals that are input from the TRAIO pin. Setting the TMOD[2:0] bits in TRAMR to B'011 activates the pulse width measurement mode operation. A count source is selected in terms of the TCK[2:0] bits in TRAMR. The TEDGSEL bit in TRAIOC can be used to specify whether the low-level width or the high-level width of input pulses is to be measured. Setting the TIPF[1:0] bits in TRAIOC allows applying a filter to external pulse input. Figure 13.3 shows an operation example of pulse width measurement mode.

chanhvinh.com



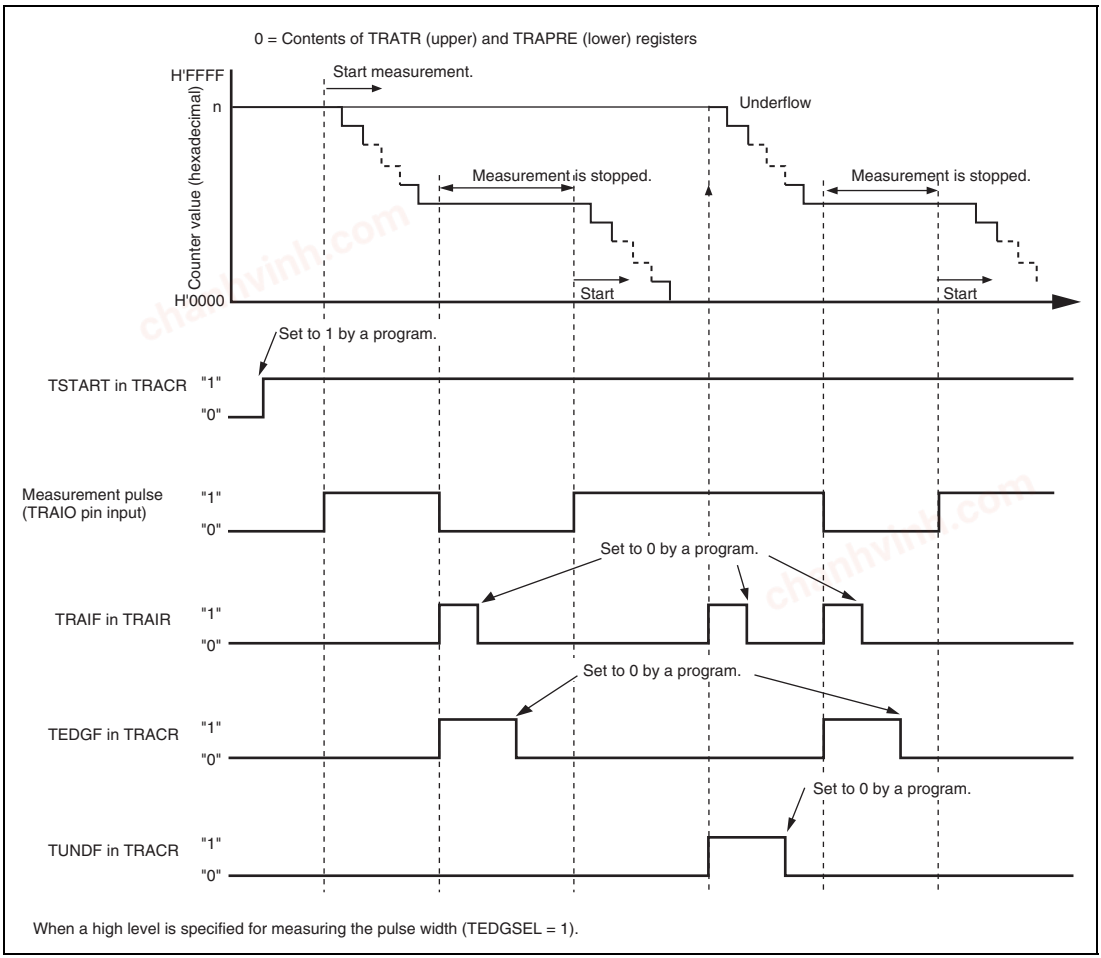



**Figure 13.3 Operation Example of Pulse Width Measurement Mode**

chanhvinh.com



### 13.3.6 Pulse Cycle Measurement Mode

This mode measures the cycle of external pulses that are input from the TRAIO pin. Setting the TMOD[2:0] bits in TRAMR to B'100 activates the pulse cycle measurement operation. The TEDGSEL in TRAIOC can be used to specify whether the period from the falling edge to another falling edge of the input pulse of the TRAIO pin is to be measured or the period from the rising edge to another rising edge is to be measured. Setting the TIPF[1:0] bits in TRAIOC also enables to apply a filter to external pulse input. Count sources are selected using the TCK[2:0] bits in TRAMR.

After the start of timer counting, each time a valid input edge is input from the TRAIO pin, a value is transferred from the counter of the timer RA to the read buffer in synchronization with the underflow of the timer RA prescaler. The value in the read buffer is retained until the timer RA register is read. Also, after a value is transferred to the read buffer, a value is transferred from the reload register to the counter in synchronization with the next underflow of the timer RA prescaler. Reading of the read buffer should not be performed until the TEDGF bit in TRACR is set to 1. An interrupt request is generated either when the TEDGF bit in TRACR is set to 1 or when the timer RA counter underflows.

For pulse input to the TRAIO pin, pulses with a cycle greater than double the cycle of the timer RA prescaler should be input. Also, input pulses for which the high pulse width and the low pulse width are greater than the cycle of the timer RA prescaler. If pulses with a short cycle are input, the input is ignored in some cases.

Figure 13.4 shows an operation example of pulse cycle measurement mode.

chanhvinh.com



### 13.3.7 Operation through an Event Link

Using the event link controller (ELC), timer RA can be made to operate in the following ways in relation to events occurring in other modules.

#### (1) Starting Counter Operation

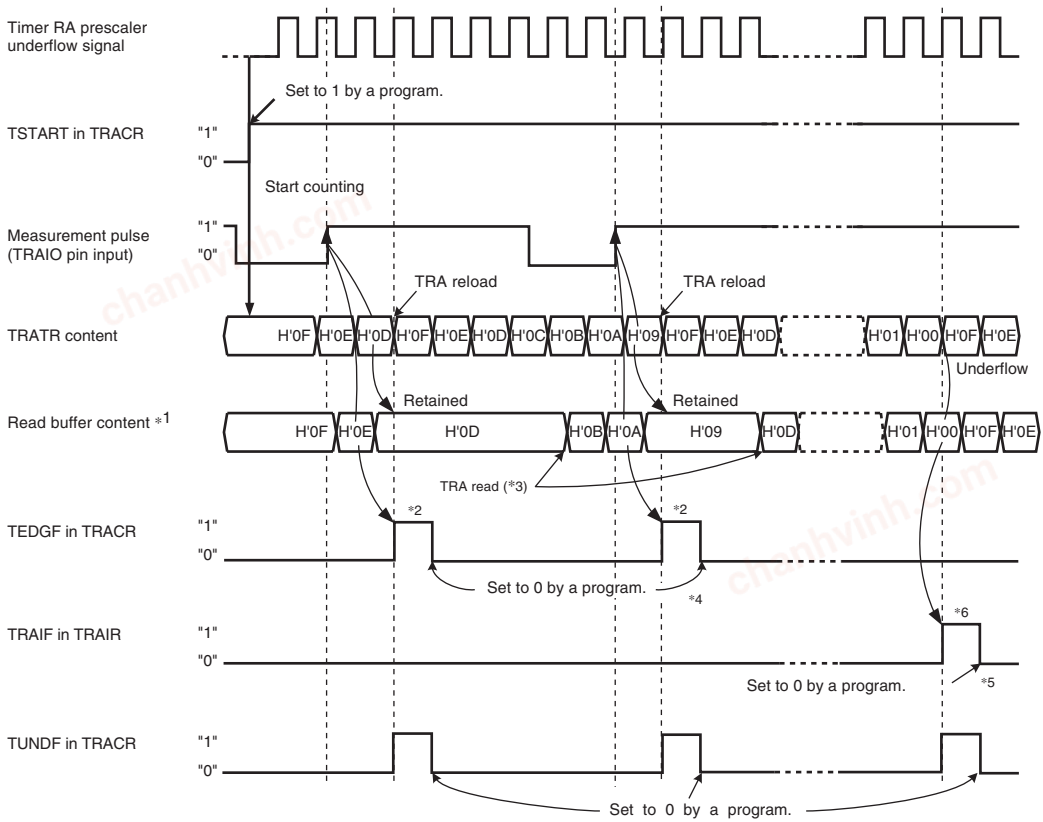
The start of counting operations by timer RA can be selected by the ELOPA register of the ELC. When the event specified in ELSR0 occurs, the TSTART bit in the TRACR is set to 1, which starts counting by timer RA. However, if the specified event occurs when the TCSTF flag has already been set to 1, that event is not effective.

#### (2) Counting Events

The counting of events by timer RA can be selected by the ELOPA register of the ELC. When the event specified in ELSR0 occurs, event-counter operation proceeds with that event as the source to drive counting, regardless of the setting in the TCK[2:0] bits in TRAMR. When event-counter operation is to be employed, set the TSTART bit in TRACR to 1 beforehand. When the value of the counter is read, the value read out is the actual number of input events minus three.

chanhvinh.com






Measurement condition: The initial value of TRATR is H'0F and the width from the rising edge to the next rising edge of measurement pulse is measured (TEDGSEL = 0).

Notes:
1. When TRATR is read in pulse cycle measurement mode, the value in the read buffer can be read.
2. The TEDGF bit in TRACR is set to 1 (valid edge exists) on the second underflow of the timer RA prescaler after a valid edge of the measurement pulses is input.
3. TRATR should be read between the TEDGF bit setting to 1 (valid edge exists) and the next valid edge input.
   The value in the read buffer is retained until TRATR is read. Therefore, if the value is not read before the next valid edge input, the measurement result of the previous cycle remains.
4. The MOV instruction should be used to write 0 to the TEDGF bit in TRACR using a program.
   Here, 1 should be written to the TUNDF bit .
5. The MOV instruction should be used to write 0 to the TUNDF bit in TRACR using a program.
   Here, 1 should be written to the TEDGF bit.
6. If an underflow of timer RA and reloading of timer RA by valid edge input occur at the same time, both the TUNDF and TEDGF bits are set to 1. In this case, the effectiveness of the TUNDF bit should be determined according to the read buffer contents.

**Figure 13.4 Operation Example of Pulse Cycle Measurement Mode**

chanhvinh.com



## 13.4 Usage Notes

1. The prescaler and timer are read out per byte inside the microcomputer even when they are read out in 16-byte unit. Therefore, the timer value can be updated while those two registers are read out.
2. The TEDGF and TUNDF bits in TRACR used in pulse width and pulse cycle measurement modes assume the value 0 when 0 is written by a program and do not change if 1 is written. If one flag is set to 0 by a program, use the MOV instruction to write 1 to the other flag. In this manner, unintended flag changes can be prevented.
3. When a transition is made to pulse width or pulse cycle measurement mode from another mode, the TEDGF and TUNDF bits are undefined. Timer RA counting should be started by writing 0 to the TEDGF and TUNDF bits.
4. In some cases, the TEDGF bit becomes 1 on the first timer RA prescaler underflow signal that is generated after the start of counting.
5. When using the pulse cycle measurement mode, set the TEDGF bit to 0 by allowing a length of time 2 cycles or greater of the timer RA prescaler after the counting process is started.
6. After 1 is written to the TSTART bit when counting is stopped, the TCSTF bit remains 0 for the number of cycle of count source. Registers associated with the timer RA except the TRACR for reading should not be accessed until the TCSTF bit becomes 1. Counting starts from a valid edge of the first count source after the TCSTF bit becomes 1.

   After 0 is written to the TSTART bit when counting is in progress, the TCSTF bit remains 1 for the number of cycle of count source. Registers associated with the timer RA except the TRACR for reading should not be accessed until the TCSTF bit becomes 0. Counting stops when the TCSTF bit becomes 0.
7. When writing successively to TRAPRE during counting (TCSTF=1), allow at least four cycles of the clock source for counting as the minimum interval for writing.
8. When writing successively to TRATR during counting (TCSTF=1), allow at least four cycles of the clock source for counting as the minimum interval for writing.
9. When values for TRAPRE and TRATR are successively read out from the same register, allow at least two cycles of the clock source for counting as the minimum interval for reading.

chanhvinh.com



# Section 14 Timer RB

The timer RB is an 8-bit reload timer with an 8-bit prescaler. The prescaler and the timer are each comprised of a reload register and a counter. The timer RB has two reload registers: timer RB primary register and timer RB secondary register.

## 14.1 Overview

- Four operating modes
  Timer mode: Counts either internal count sources or timer RA underflows.
  Programmable waveform generation mode: Outputs any pulse widths continuously.
  Programmable one-shot generation mode: Outputs one-shot pulses.
  Programmable wait one-shot generation mode: Outputs delayed one-shot pulses.
- Selection of eight count sources
  ϕ, ϕ/2, ϕ/4, ϕ/8, ϕ/32, ϕ/64, ϕ/128, or an underflow of timer RA
- An interrupt generated on an underflow of the timer RB counter

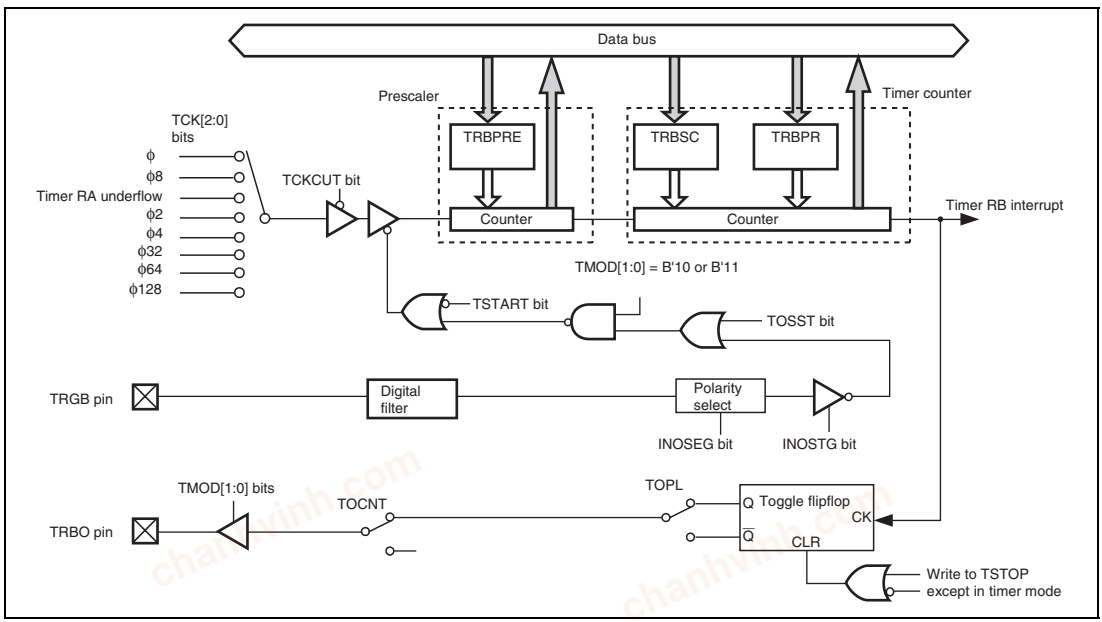



**Figure 14.1 Block Diagram of Timer RB**

chanhvinh.com



Table 14.1 shows the timer RB input/output pins.

**Table 14.1 Pin Configuration**

| Name | I/O | Function |
|---|---|---|
| TRGB | Input | External trigger input |
| TRBO | Output | Successive pulse output or one-shot pulse output |

## 14.2 Register Descriptions

The timer RB has the following registers:

- Timer RB control register (TRBCR)
- Timer RB one-shot control register (TRBOCR)
- Timer RB I/O control register (TRBIOC)
- Timer RB mode register (TRBMR)
- Timer RB interrupt request status register (TRBIR)
- Timer RB prescaler register (TRBPRE)
- Timer RB secondary register (TRBSC)
- Timer RB primary register (TRBPR)

chanhvinh.com



### 14.2.1 Timer RB Control Register (TRBCR)

Address: H'FFFFA0

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | TSTOP | TCSTF | TSTART |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 to 3 | — | Reserved | These bits are read as 0. The write value should be 0. | — |
| 2 | TSTOP | Count forced stop | 0: Timer RB counting is continued.<br>1: Timer RB counting is forcedly stopped. | R/W |
| 1 | TCSTF | Count status flag | 0: Timer RB counting is stopped.<br>1: Timer RB counting is in progress.<br>[Setting conditions]<br>• When 1 is written to TSTART and counting is started.<br>• The start of counting after ELOPA of the event link controller is selected counting by timer RB, the specified event is occurred, and the TSTART bit is set to 1.<br>[Clearing conditions]<br>• When 0 is written to TSTART and counting is stopped.<br>• When 1 is written to TSTOP and counting is stopped. | R |
| 0 | TSTART | Count start | 0: Timer RB counting is stopped.<br>1: Timer RB counting is started. | R/W |

Notes: 1. The timer RB registers should not be accessed until the TCSTF bit changes after the TSTART bit is set, apart from TRBCR which can be read at any time during timer operation.
2. A MOV instruction should be used to write to this register.

- TSTOP bit (count forced stop)

  Setting this bit to 1 stops counting forcibly. At this time, the counter of the timer RB prescaler and the timer RB counter are initialized. Also, bits TSTART and TCSTF in TRBCR, bits TOSSTF, TOSSP, TOSST in TRBOCR, and TRBO outputs are initialized. The reload register of the prescaler and the timer RB counter are hold. This bit is always read as 0.

chanhvinh.com



### 14.2.2 Timer RB One-Shot Control Register (TRBOCR)

Address: H'FFFFA1

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | TOSSTF | TOSSP | TOSST |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 to 3 | — | Reserved | These bits are read as 0. The write value should be 0. | — |
| 2 | TOSSTF | One-shot status flag | 0: Timer RB one-shot function has been stopped.<br>1: Timer RB one-shot function is active (including wait time).<br>[Setting conditions]<br>• When 1 is written to the TOSST bit.<br>• When trigger inputs to the TRGB pin are enabled.<br>[Clearing conditions]<br>• When 1 is written to the TOSSP bit.<br>• When 0 is written to the TSTART bit in TRBCR.<br>• When 1 is written to the TSTOP bit in TRBCR.<br>[In programmable one-shot generation mode]<br>• When the timer counter reaches H'00 and the reloading is performed.<br>[In programmable wait on-shot generation mode]<br>When the counter value reaches H'00 during the secondary counting and the reloading is performed. | R |
| 1 | TOSSP*[1] | One-shot stop | 0: Timer RB counting is not stopped.<br>1: Timer RB counting is stopped. | R/W |
| 0 | TOSST*[2] | One-shot start | 0: Timer RB counting is stopped.<br>1: Timer RB counting is started. | R/W |

Notes: 1. The TOSSP bit should be modified to 1 when the TOSSTF bit is 1.
2. The TOSST bit should be modified to 1 when the TOSSTF bit is 1.

chanhvinh.com



- TOSSP bit (one-shot stop)

  Writing 1 to this bit stops the timer counting. This bit is always read as 0

- TOSST bit (one-shot start)

  In programmable one-shot generation mode or programmable wait one-shot generation mode, writing 1 to this bit starts the timer counting and one-shot pulse output in synchronization with the count source. This bit is always read as 0.

### 14.2.3 Timer RB I/O Control Register (TRBIOC)

Address: H'FFFFA2

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | TIPF[1:0] | | INOSEG | INOSTG | TOCNT | TOPL |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7, 6 | — | Reserved | These bits are read as 0. The write value should be 0. | — |
| 5, 4 | TIPF[1:0] | TRGB input filter select | 00: No filter operation<br>01: Filtered (Sampled at ϕ)<br>10: Filtered (Sampled at ϕ/8)<br>11: Filtered (Sampled at ϕ/32)<br>These bits should be set to B'00 for timer mode or pulse output mode. | R/W |
| 3 | INOSEG | One-shot trigger polarity select | 0: Triggered at a falling edge.<br>1: Triggered at a rising edge. | R/W |
| 2 | INOSTG | One-shot trigger control | 0: The one-shot trigger function for the TRGB pin is disabled.<br>1: The one-shot trigger function for TRGB pin is enabled. | R/W |
| 1 | TOCNT | Timer RB output switch | 0: Waveform is output from timer RB.<br>1: Waveform output is disabled. | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 0 | TOPL | Timer RB output level select | Programmable Waveform Generation Mode<br>0: A high-level signal is output in primary period, a low-level signal in secondary period and a low-level signal when the timer stops.<br>1: A low-level signal is output in primary period, a high-level signal in secondary period, and a high-level signal when the timer stops.<br>Programmable one-shot generation mode<br>0: A high-level signal is output for one-shot pulse output and a low-level signal when the timer stops.<br>1: A low-level signal is output for one-shot pulse output and a high-level signal when the timer stops.<br>Programmable wait one-shot generation mode<br>0: A high-level signal is output for one-shot pulse output and a low-level signal during the wait time or the time when the timer stops.<br>1: A low-level signal is output for one-shot pulse output and a high-level signal during the wait time or the time when the timer stops.<br>This bit should be 0 in timer mode. | R/W |

- INOSEG bit (one-shot trigger polarity select)

  Selects an edge for the one-shot trigger signal input from the TRGB pin in programmable one-shot generation mode or programmable wait one-shot generation mode. This bit should be 0 in timer mode or programmable waveform generation mode.

- INOSTG bit (one-shot trigger control)

  Enables or disables one-shot trigger signal input from the TRGB pin. This bit should be 0 in timer mode or programmable waveform generation mode.

- TOCNT bit (timer RB output switch)

  For TRBO output state or output change conditions in each mode, see section 14.3.6, TOCNT Settings and Pin State Update Conditions.

- TOPL bit (timer RB output level select)

  This bit should be 0 in timer mode.

chanhvinh.com



### 14.2.4 Timer RB Mode Register (TRBMR)

Address: H'FFFFA3

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TCKCUT | TCK[2:0] | | | TWRC | — | TMOD[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | TCKCUT*[1] | Count source cutoff | 0: Timer RB clock source is supplied.<br>1: Timer RB clock source is cut off. | R/W |
| 6 to 4 | TCK[2:0]*[1] | Count source select | 000: ϕ<br>001: ϕ/8<br>010: Underflow of timer RA<br>011: ϕ/2<br>100: ϕ4<br>101: ϕ/32<br>110: ϕ/64<br>111: ϕ/128 | R/W |
| 3 | TWRC | Write control | 0: Both the reload register and counter are written to.<br>1: Only the reload register is written to. | |
| 2 | — | Reserved | This bit is read as 0. The write value should be 0. | — |
| 1, 0 | TMOD[1:0]*[2] | Operating mode select | 00: Timer mode<br>01: Programmable waveform generation mode<br>10: Programmable one-shot generation mode<br>11: Programmable wait one-shot generation mode | R/W |

Notes: 1. A count source should not be switched or cut off during counting. The count source should be switched or cut off when both the TSTART and TCSTF bits in TRBCR are 0 (when the timer counting is stopped).
2. An operating mode should be selected when the counting is stopped (when both the TSTART and TCSTF bits in TRBCR are 0).

- TWRC bit (write control)

  Controls the timing when the counter reflects the value of the reload register. This bit should be 1 except in timer mode.

chanhvinh.com



### 14.2.5 Timer RB Interrupt Request Status Register (TRBIR)

Address: H'FFFFA7

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TRBIE | TRBIF | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | TRBIE | Interrupt enable | 0: Timer RB interrupt requests are disabled.<br>1: Timer RB interrupt requests are enabled. | R/W |
| 6 | TRBIF | Interrupt request flag | [Setting conditions]<br>Timer mode<br>• When the timer RA underflows.<br>Programmable waveform generation mode<br>• A half cycle of the count source after the counter underflow in the secondary period<br>Programmable one-shot generation mode.<br>• A half cycle of the count source after the counter underflow<br>Programmable wait one-shot generation mode<br>• A half cycle of the counter source after the counter underflow in the secondary period<br>[Clearing condition]<br>• When 1 is read from the bit and then 0 is written to. | R/W |
| 5 to 0 | — | Reserved | These bits are read as 0. The write value should be 0. | — |

chanhvinh.com



### 14.2.6 Timer RB Prescaler Register (TRBPRE)

Address: H'FFFFA4

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

TRBPRE is a reload register for the timer RB prescaler. The timer RB prescaler consists of a reload register and an 8-bit counter.

If a down-count is performed using the count source selected on TRBMR and an underflow occurs, the value of the reload register is loaded to the counter. The underflow becomes a count source for TRBTR.

TRBPRE and the counter are assigned to the same address. On write, a value is written to the reload register, and on read, a counter value is read. During a write to TRBPRE, the load timing from the reload register to the counter differs between counting in progress and counting stopped by the setting of the TWRC bit in TRBMR. For details, see descriptions of each operating mode.

The initial values of TRBPRE and the counter are H'FF.

### 14.2.7 Timer RB Secondary Register (TRBSC)

Address: H'FFFFA5

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

TRBSC is an 8-bit write-only register that sets the secondary period for the timer RB counter. This register is used only in programmable waveform generation mode and programmable wait one-shot generation mode. This register is not used in timer mode or programmable one-shot generation mode.

When TRBSC is written to in any operating mode where TRBSC is used, both the TRBSC and TRBPR should be written to in this order. Even if only TRBSC is to be modified, TRBPR should also be set to the previous value. The initial value is H'FF.

chanhvinh.com



### 14.2.8 Timer RB Primary Register (TRBPR)

Address: H'FFFFA6

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

TRBPR is an 8-bit reload register that sets the cycle or primary period for the timer RB counter. The timer RB counter consists of two registers, primary and secondary registers, and a counter. The primary register and counter are assigned to the same address. On write to TRBPR, a value is written to the reload register, and on read from TRBPR, a counter value is read.

During a write to TRBPR the load timing from the reload register to the counter differs between counting in progress and counting stopped. For details, see descriptions of each operating mode.

The initial values of TRBPR and the counter are H'FF.

chanhvinh.com



## 14.3 Operation

### 14.3.1 Timer Mode

The internal clock pulses or timer RA underflows are counted as a count source in timer mode. When an underflow occurs on the timer RB counter, the value of TRBPR is reloaded and counting is continued. TRBOCR and TRBSC are not used in timer mode. A count source is selected with the TCK[2:0] bits in TRBMR.

**(1) Starting and Stopping Operation**

Writing the value 1 to the TSTART bit in TRBCR starts counting; writing the value 0 to the TSTART bit stops the counting.

**(2) Forced Termination of Operation**

Writing 1 to the TSTOP bit in TRBCR stops the counting forcedly. When the counting is forcedly stopped, the timer RB counter, the prescaler counter, and any associated flags are initialized.

**(3) Interrupt Request**

An interrupt request is generated on the underflow of the timer RB counter.

**(4) Reading and Writing Count Value**

Reading TRBPRE and TRBTR reads count values from each register.

If a write is performed to TRBPRE or TRBTR when the counting is stopped, a specified value is written to both the reload register and the counter.

If a write is performed to TRBPRE during counting when TWRC in TRBMR is 0, first a set value is written to the reload register, and the set value is then transferred to the prescaler counter in synchronization with the count source. If a write is performed to TRBPR, a set value is written to the reload register in synchronization with the underflow of the prescaler after four cycles of the count source of the prescaler, and the set value is transferred to the timer counter in synchronization with the next underflow of the prescaler.

For this reason, if a write is performed to TRBPRE or TRBPR during counting when TWRC is 1, the value is written only to the reload register. Loading to the counter is performed in synchronization with the underflow of the prescaler or timer counter.

chanhvinh.com



### 14.3.2 Programmable Waveform Generation Mode

This mode alternately reloads and counts values of TRBPR and TRBSC, and produces toggle output from the TRBO pin each time the counter underflows. At the start of counting, this mode counts beginning with the value assigned to TRBPR. TRBOCR is not used when programmable waveform generation mode is used.

#### (1) Starting and Stopping Operation

Writing the value 1 to the TSTART bit in TRBCR starts counting; writing the value 0 to the TSTART bit stops the counting.

#### (2) Forced Termination of Operation

Writing 1 to the TSTOP bit in TRBCR stops the counting forcedly. When the counting is forcedly stopped, the timer RB counter, the prescaler counter, and any associated flags are initialized.

#### (3) Interrupt Request

An interrupt request is generated on the underflow of the timer RB counter during the secondary period counting.

#### (4) Reading and Writing Count Value

Reading TRBPRE and TRBTR reads count values from each register.

If a write is performed to TRBPRE, TRBPR, or TRBSC when counting is stopped, set values are written to both the reload register and the counter.

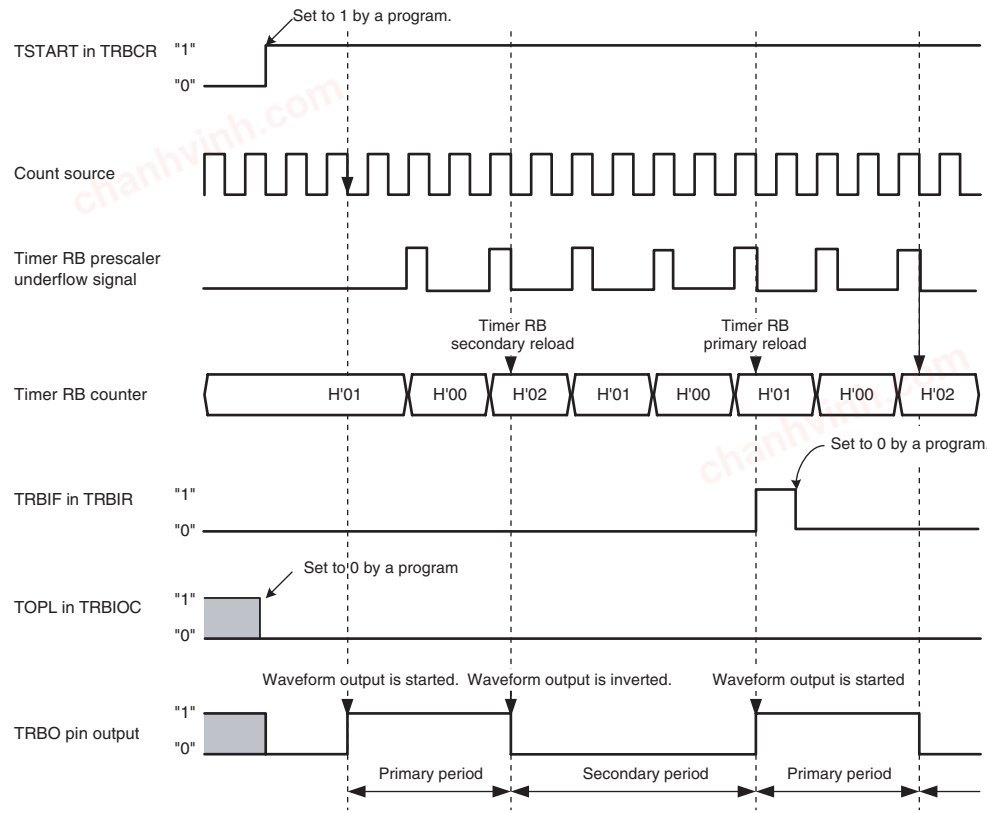
If a write is performed to TRBPRE, TRBPR, or TRBSC when counting is in progress, data is written only to the respective reload registers. The output of a waveform reflects a set value beginning with the next primary period after data is written to TRBPR.

However, if writing to TRBSC or TRBPR proceeds when the value of the counter is H'00, updating of the waveform will be suspended for one cycle.

chanhvinh.com



Figure 14.2 shows an operation example of the timer RB in programmable waveform generation mode.




Conditions for the above operation:

TRBPRE=H'01, TRBPR=H'01, TRBSC=H'02
TOCNT = 0 in TRBIOC (Timer RB waveform is output from the TRBO pin.)

**Figure 14.2 Operation in Programmable Waveform Generation Mode**

chanhvinh.com



### 14.3.3 Programmable One-Shot Generation Mode

This mode outputs one-shot pulses from the TRBO pin, based on either program or external trigger input. When a trigger is generated, beginning with that point in time the timer operates only once for any length of time specified in TRBPR. TRBSC is not used in this mode. In this mode, TRBPRE or TRBPR should not be set to H'00.

#### (1) Starting and Stopping Operation

The counting is started when 1 is written to the TOSST bit in TRBOCR or a valid trigger signal is input to the TRGB pin after the TSTART bit in TRBCR is set to 1 and the TCSTF flag is set to 1. For a trigger input, the pulse must be longer than one cycle of the clock source for counting.

The counting is stopped when reloading is performed with an underflow of the counter, when 1 is written to the TOSSP bit in TRBOCR, or when 0 is written to the TSTART bit in TRBCR.

#### (2) Forced Termination of Operation

Writing 1 to the TSTOP bit in TRBCR stops the counting forcedly. When the counting is forcedly stopped, the timer RB counter, the prescaler counter, and any associated flags are initialized.

#### (3) Interrupt Request

An interrupt request is generated on the underflow of the timer RB counter.

#### (4) Reading and Writing Count Value

Reading TRBPRE and TRBTR reads count values from each register.

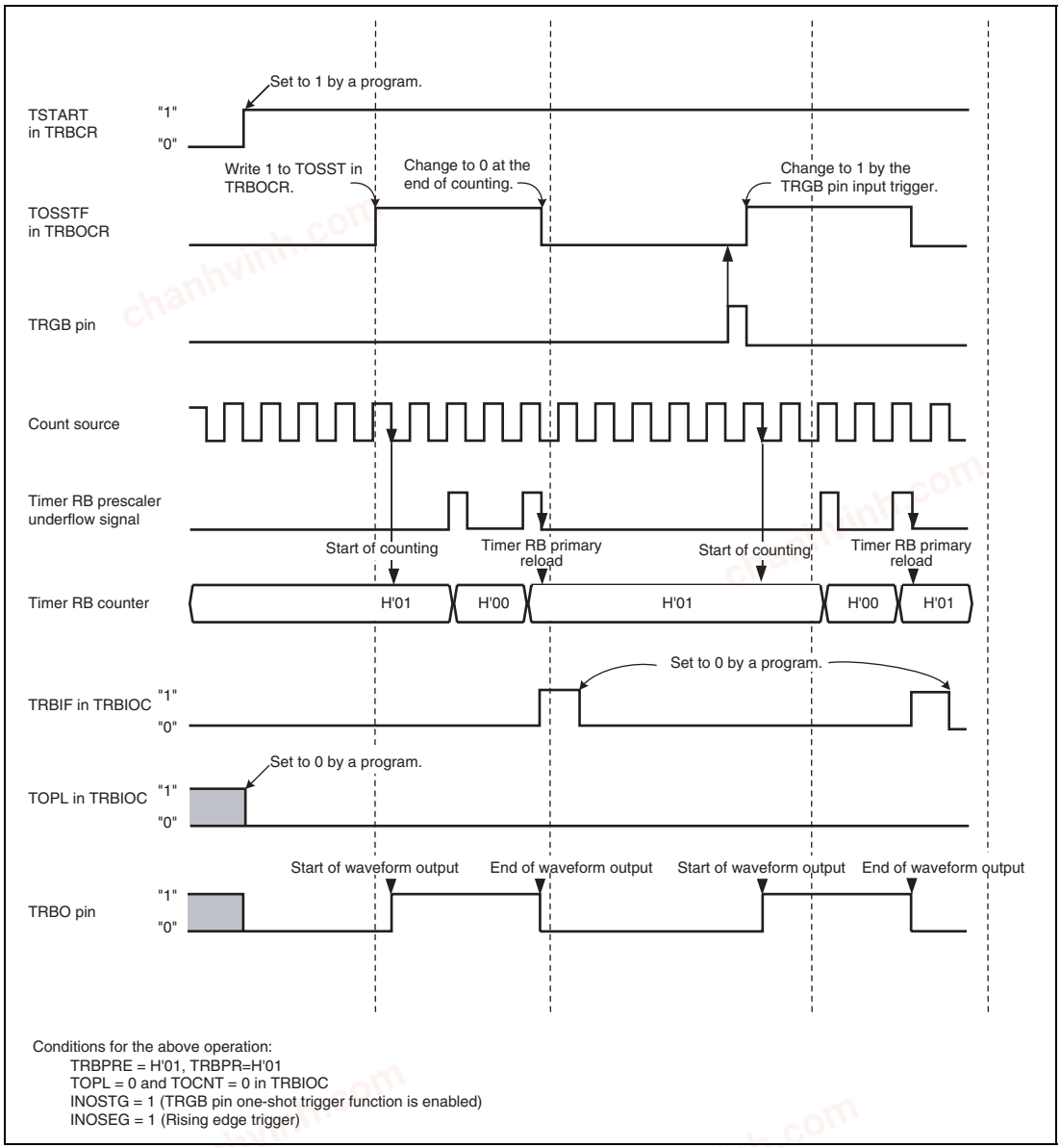
If a write is performed to TRBPRE or TRBPR when counting is stopped, set values are written to both the reload register and the counter.

If a write is performed to TRBPRE or TRBPR during counting, data is written only to the respective reload registers. The value written to TRBPRE takes effect in synchronization with the underflow of the prescaler. The value written to TRBPR takes effect during the next one-shot pulse.

Figure 14.3 shows an operation example of the timer RB in programmable one-shot generation mode.

chanhvinh.com






Conditions for the above operation:
TRBPRE = H'01, TRBPR=H'01
TOPL = 0 and TOCNT = 0 in TRBIOC
INOSTG = 1 (TRGB pin one-shot trigger function is enabled)
INOSEG = 1 (Rising edge trigger)

**Figure 14.3 Operation in Programmable One-Shot Generation Mode**

chanhvinh.com



### 14.3.4 Programmable Wait One-Shot Generation Mode

This mode outputs one-shot pulses from the TRBO pin after a fixed amount of time based on either program or external trigger input. When a trigger is generated, beginning with that point in time, pulses are output only once for any length of time set in TRBSC, after any length of time set in TRBPR.

#### (1) Starting and Stopping Operation

The counting is started when 1 is written to the TOSST bit in TRBOCR or a valid trigger signal is input to the TRGB pin after the TSTART bit in TRBCR is set to 1 and the TCSTF flag is set to 1. For a trigger input, the pulse must be longer than one cycle of the clock source for counting.

The counting is stopped when reloading is performed with an underflow of the timer RB counter during the secondary period counting, when 1 is written to the TOSSP bit in TRBOCR, or when 0 is written to the TSTART bit in TRBCR.

#### (2) Forced Termination of Operation

Writing 1 to the TSTOP bit in TRBCR stops the counting forcedly. When the counting is forcedly stopped, the timer RB counter, the prescaler counter, and any associated flags are initialized.

#### (3) Interrupt Request

An interrupt request is generated on the underflow of the timer RB counter during the secondary period counting.

#### (4) Reading and Writing Count Value

Reading TRBPRE and TRBTR reads count values from each register.

If a write is performed to TRBPRE, TRBPR, or TRBSC when counting is stopped, set values are written to both the reload register and the counter.

If a write is performed to TRBPRE, TRBPR, or TRBSC during counting, data is written only to the respective reload registers. The value written to TRBPRE takes effect in synchronization with the underflow of the prescaler. The value written to TRBPR takes effect during the next one-shot pulse.

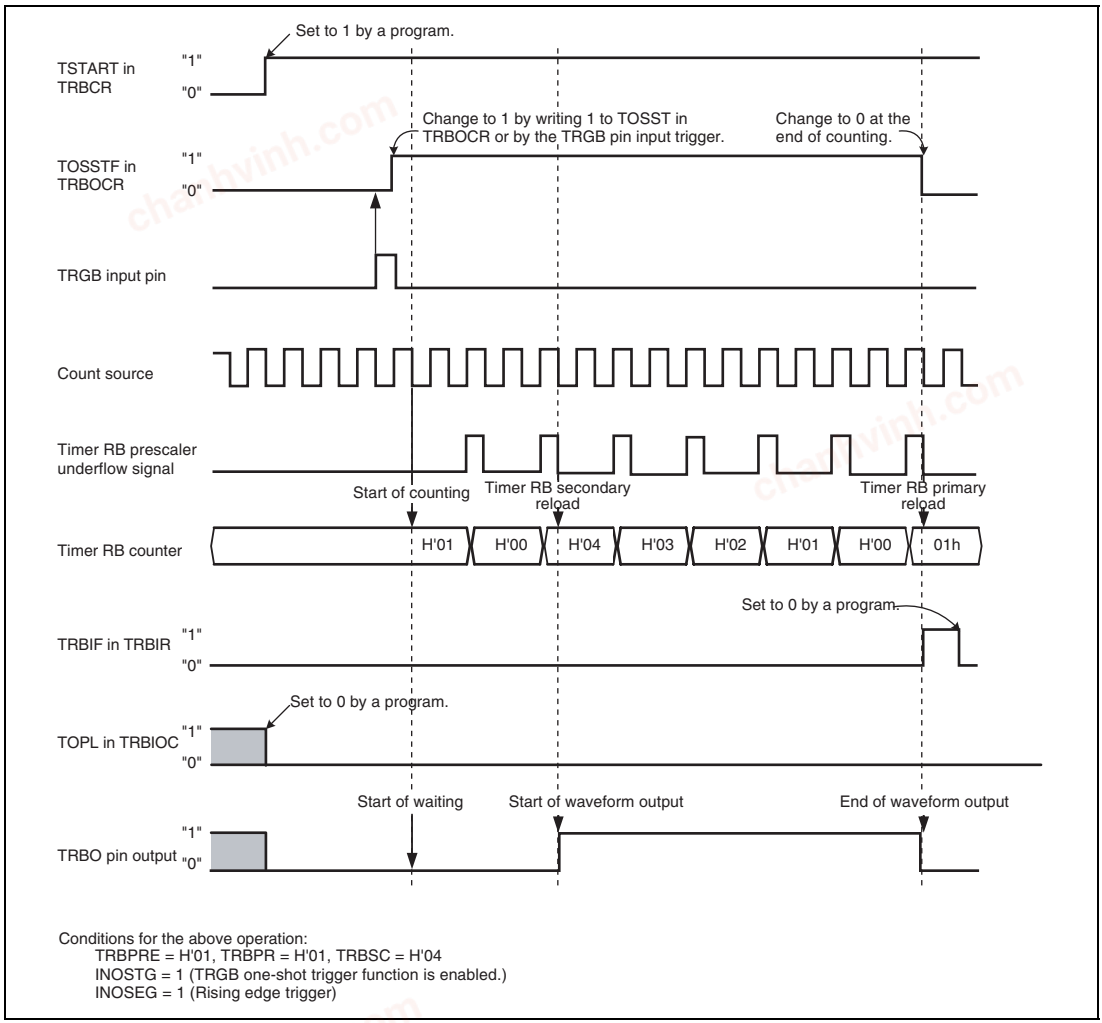
After writing to TRBSC and TRBPR when TCSTF = 1 or TOSSTF = 0, if a write is successively performed to TRBSC and then to TRBPR, allow an interval of 5 cycles of the clock source for counting before writing 1 to the TOSST bit.

In this mode, TRBPRE or TRBPR should not be set to H'00.

chanhvinh.com



Figure 14.4 shows an operation example of the timer RB in programmable wait one-shot generation mode.




Conditions for the above operation:
TRBPRE = H'01, TRBPR = H'01, TRBSC = H'04
INOSTG = 1 (TRGB one-shot trigger function is enabled.)
INOSEG = 1 (Rising edge trigger)

**Figure 14.4 Operation in Programmable Wait One-Shot Generation Mode**

chanhvinh.com



### 14.3.5 Timing at Which Values Take Effect in Prescaler or Counter Depending on TWRC Bit

Depending on the value assigned to the TWRC bit in TRBMR, the timing at which the value written to TRBPRE, TRBPR, or TRBSC during timer operation takes effect in the counter can vary.

If TWRC is set to 1 and value is written only to the register, the counter value is updated between cycles, thus preventing the occurrence of fractional cycles. In modes other than the timer mode, TWRC should be set to 1.

Figure 14.5 shows operation examples on the prescaler and the counter when the value of TWRC is 0 and 1.

If TCSTF is 1, even when TWRC is cleared to 0, any transfer to the prescaler or the counter is performed in synchronization with the count source; therefore, the counter value is not updated immediately after the execution of a write instruction.

chanhvinh.com



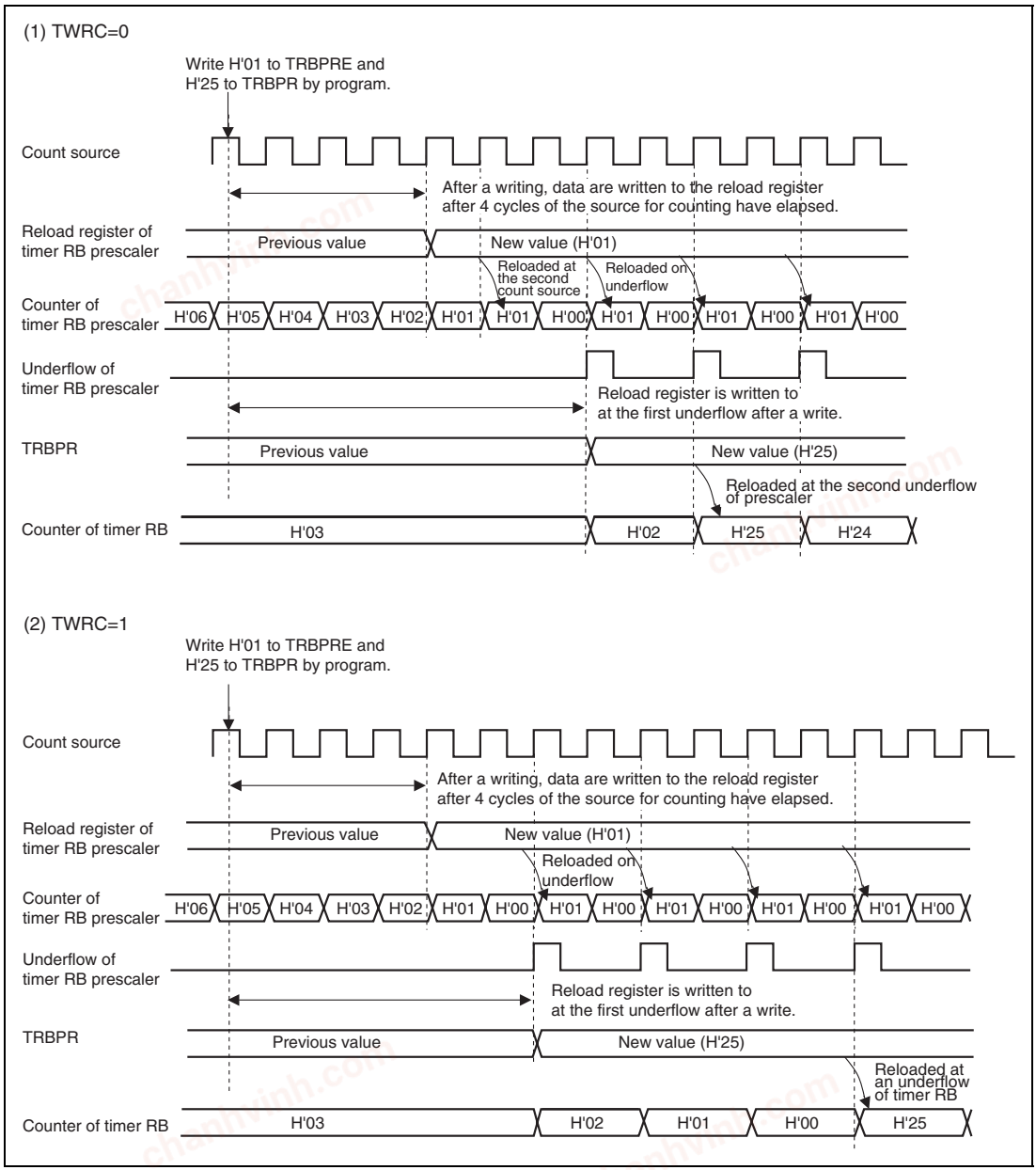



**Figure 14.5 TWRC Settings and Operation of Prescaler and Counter**

chanhvinh.com



### 14.3.6 TOCNT Settings and Pin State Update Conditions

Depending on the TOCNT bit in TRBIOC and the corresponding bit in PMR, the user can select whether the pin is used as a general I/O port or as a specific timer waveform output. In the case of timer mode, however, the pin operates as a general I/O port, irrespective of TOCNT bit settings.

When the TOCNT bit is rewritten, the pin state is not updated immediately; the change takes effect when either of the following conditions occurs:

Pin state update conditions:

- When the TSTART bit in TRBCR is changed from 0 to 1
- When TRBPR is reloaded to the counter

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

chanhvinh.com



### 14.3.7 Operation through an Event Link

Using the event link controller (ELC), timer RB can be made to operate in the following ways in relation to events occurring in other modules.

#### (1) Starting Counter Operation

The start of counting operations by timer RB can be selected by the ELOPA register of the ELC. When the event specified in ELSR1 occurs, the TSTART bit in the TRBCR is set to 1, which starts counting by timer RB. However, if the specified event occurs when the TCSTF flag has already been set to 1, that event is not effective.

#### (2) Counting Events

The counting of events by timer RB can be selected by the ELOPA register of the ELC. When the event specified in ELSR1 occurs, event counter operation proceeds with that event as the source to drive counting, regardless of the setting in the TCK[2:0] bits in TRBMR. When event-counter operation is to be employed, set the TSTART bit in TRBCR to 1 beforehand. When the value of the counter is read, the value read out is the actual number of input events minus three.

## 14.4 Interrupt Request

This module provides a timer RB interrupt enable bit (the TRBIE bit in TRBIR) and a timer RB interrupt request flag (the TRBIF bit in TRBIR). An interrupt request is issued to the CPU when the TRBIE bit is set to 1 while the TRBIF bit is 1, or when the TRBIE bit changes from 0 to 1 while the TRBIF bit is 1. Since the condition under which the TRBIF bit is set varies with operation modes, see the explanation on the TRBIF bit and the description of the various operation modes.

chanhvinh.com



## 14.5 Usage Notes

1. In programmable one-shot generation mode and programmable wait one-shot generation mode, if the counting is stopped by clearing the TSTART bit in TRBCR to 0, the timer counter holds a count value, and then stops.
2. After 1 is written to the TSTART bit when the counting is stopped, the TCSTF bit remains 0 for the number of cycles of the count source. The timer RB related registers*, with the exception of the TRBCR for reading should not be accessed until the TCSTF bit is set to 1.
   After 0 is written to the TSTART bit during counting, the TCSTF bit remains 1 for the number of cycles of the count source. The timer RB related registers*, with the exception of the TRBCR for reading should not be accessed until the TCSTF bit is cleared to 0.

Note: Timer RB-related registers refer to registers TRBCR, TRBOCR, TRBIOC, TRBMR, TRBPRE, TRBSC, and TRBPR.

3. TRBPRE and TRBPR should not be set to H'00 at the same time.
4. When rewriting the bits TRBPRE, TRBPR, and TRBSC at TSTART = 0, set TSTART to 1 after the passage of at least 2 cycles of the system clock $\phi$.
5. When TSTART = 1 or TCSTF = 1, TRBIOC, or TRBMR should not be rewritten.
6. When writing 1 to the TOSST bit, read the TCSTF bit and write by verifying the value 1.
7. In programmable waveform generation mode or programmable wait one-shot mode, make sure another write to TRBSC does not occur between writing to TRBPR and reloading to the counter.
8. When writing successively to TRBPRE during counting (TCSTF=1), allow at least four cycles of the clock source for counting as the minimum interval for writing
9. When writing successively to TRBPR and TRBSC during counting (TCSTF=1), allow at least four cycles of the clock source for counting as the minimum interval for writing
10. When 1 is written to the TOSST or TOSSP bit in TRBOCR, the value of the TOSSTF bit changes accordingly after 1 to 2 cycles of the source for counting. If 1 is written to the TOSSP bit during the period between the TOSST bit having been set to 1 and the value of the TOSSTF bit becoming 1, the value of the TOSSTF bit will become 0 in some cases and 1 in others, depending on the internal state. In the same way, if 1 is written to the TOSST bit during the period between the TOSSP bit having been set to 1 and the value of the TOSSTF bit becoming 0, whether the value of the TOSSTF bit will become 0 or 1 is not defined.
11. When values for TRBPRE and TRBPR are successively read out from the same register, allow at least two cycles of the clock source for counting as the minimum interval for reading
12. When timer RB selects the timer RA underflow as a count source, be sure to set timer RA to timer mode, pulse output mode, or event counter mode.

chanhvinh.com



# Section 15 Timer RC

Timer RC is a 16-bit timer having output compare and input capture functions. Timer RC can count external events and output pulses with a desired duty cycle using the compare match function between the timer counter and four general registers. Thus, it can be applied to various systems.

Note: Timer RC is not supported in H8S/20223, H8S/20203, H8S/20215, and H8S/20235 Groups.

## 15.1 Features

- Selection of six counter clock sources
  Five internal clocks (φ, φ/2, φ/4, φ/8, and φ/32) and an external clock (for counting external events)
- Capability to process up to four pulse outputs or four pulse inputs
- Four general registers
  Can be used as output compare or input capture registers independently
  Can be used as buffer registers for the output compare or input capture registers
- Timer inputs and outputs
  — Timer mode
    Output compare function (Selection of 0 output, 1 output, or toggle output)
    Input capture function (Rising edge, falling edge, or both edges can be detected.)
    Counter clearing function (Counter cycle can be set.)
  — PWM mode
    Generates up to three-phase PWM output.
  — PWM2 mode
    Generates pulses with a desired period and duty cycle.
- Any initial timer output value can be set
- Five interrupt sources
  Four compare match/input capture interrupts and an overflow interrupt.

chanhvinh.com



Table 15.1 summarizes the timer RC functions, and figure 15.1 shows a block diagram of timer RC.

**Table 15.1 Timer RC Functions**

| Item | | Counter | Input/Output Pins | | | |
|---|---|---|---|---|---|---|
| | | | FTIOA | FTIOB | FTIOC | FTIOD |
| Count clock | | Internal clocks: φ, φ/2, φ/4, φ/8, and φ/32<br>External clock: FTCI | | | | |
| General registers (output compare/input capture registers) | | Period specified in GRA | GRA | GRB | GRC (buffer register for GRA in buffer mode) | GRD (buffer register for GRB in buffer mode) |
| Counter clearing function | | GRA input capture/ compare match | GRA input capture/ compare match | — | — | — |
| | | TRGC input | — | — | — | — |
| Initial output value setting function | | — | Yes | Yes | Yes | Yes |
| Buffer function | | — | Yes | Yes | — | — |
| Compare match output | 0 output | — | Yes | Yes | Yes | Yes |
| | 1 output | — | Yes | Yes | Yes | Yes |
| | Toggle output | — | Yes | Yes | Yes | Yes |
| Input capture function | | — | Yes | Yes | Yes | Yes |
| PWM mode | | — | — | Yes | Yes | Yes |
| PWM2 mode | | — | — | Yes | — | — |
| Interrupt sources | | Overflow | Compare match/input capture | Compare match/input capture | Compare match/input capture | Compare match/input capture |

chanhvinh.com



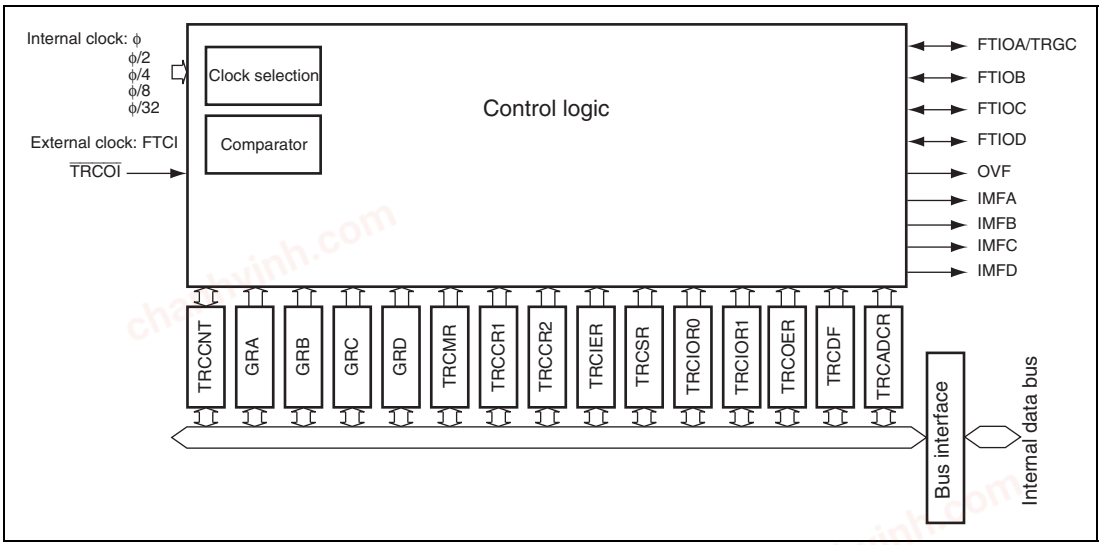



**Figure 15.1 Timer RC Block Diagram**

Table 15.2 summarizes the timer RC pins.

**Table 15.2 Pin Configuration**

| Pin Name | Input/ Output | Function |
|---|---|---|
| FTCI | Input | External clock input pin |
| FTIOA/TRGC | I/O | Output pin for GRA output compare/input pin for GRA input capture/ external trigger input pin (TRGC) |
| FTIOB | I/O | Output pin for GRB output compare/input pin for GRB input capture/ PWM output pin in PWM mode |
| FTIOC | I/O | Output pin for GRC output compare/input pin for GRC input capture/ PWM output pin in PWM mode |
| FTIOD | I/O | Output pin for GRD output compare/input pin for GRD input capture/ PWM output pin in PWM mode |
| $\overline{\text{TRCOI}}$ | Input | Input pin for timer output disabling signal |

chanhvinh.com



## 15.2 Register Descriptions

The timer RC has the following registers.

- Timer RC mode register (TRCMR)
- Timer RC control register 1 (TRCCR1)
- Timer RC control register 2 (TRCCR2)
- Timer RC interrupt enable register (TRCIER)
- Timer RC status register (TRCSR)
- Timer RC I/O control register 0 (TRCIOR0)
- Timer RC I/O control register 1 (TRCIOR1)
- Timer RC output enable register (TRCOER)
- Timer RC digital filtering function select register (TRCDF)
- Timer RC A/D conversion start trigger control register (TRCADCR)
- Timer RC counter (TRCCNT)
- General register A (GRA)
- General register B (GRB)
- General register C (GRC)
- General register D (GRD)

chanhvinh.com



### 15.2.1 Timer RC Mode Register (TRCMR)

Address: H'FFFF8A

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | CTS | — | BUFEB | BUFEA | PWM2 | PWMD | PWMC | PWMB |
| Value after reset: | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | CTS | Counter start | 0: TRCCNT stops counting.<br>1: TRCCNT starts counting.<br>[Setting conditions]<br>• When 1 is written in CTS<br>• When the specified event is occurred after ELOPA of the event link controller is selected counting by timer RC.<br>[Clearing conditions]<br>• When 0 is written in CTS<br>• In PWM2 mode, when the CSTP bit in TRCCR2 is set to 1 and a compare match signal is generated. | R/W |
| 6 | — | Reserved | This bit is read as 1. The write value should be 1. | — |
| 5 | BUFEB | Buffer operation B | 0: GRD functions as an input capture/output compare register<br>1: GRD functions as the buffer register for GRB | R/W |
| 4 | BUFEA | Buffer operation A | 0: GRC functions as an input capture/output compare register<br>1: GRC functions as the buffer register for GRA | R/W |
| 3 | PWM2 | PWM2 mode | 0: Timer RC functions in PWM2 mode.<br>The following settings are invalid: TRCIOR0, TRCIOR1, and the PWMB, PWMC, and PWMD bits in TRCMR.<br>1: Timer RC functions in timer mode or PWM mode.<br>The following settings are valid: TRCIOR0, TRCIOR1, and the PWMB, PWMC, and PWMD bits in TRCMR. | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 2 | PWMD | PWM mode D | Selects the output mode of the FTIOD pin.<br>0: Functions in timer mode<br>1: Functions in PWM mode | R/W |
| 1 | PWMC | PWM mode C | Selects the output mode of the FTIOC pin.<br>0: Functions in timer mode<br>1: Functions in PWM mode | R/W |
| 0 | PWMB | PWM mode B | Selects the output mode of the FTIOB pin.<br>0: Functions in timer mode<br>1: Functions in PWM mode | R/W |

### 15.2.2 Timer RC Control Register 1 (TRCCR1)

Address: H'FFFF8B

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | CCLR | CKS[2:0] | | | TOD | TOC | TOB | TOA |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | CCLR | Counter clear | 0: TRCCNT functions as a free-running counter.<br>1: The TRCCNT value is cleared by input capture A/compare match A. | R/W |
| 6 to 4 | CKS[2:0]*2 | Clock select 2 to 0 | Select the source of the clock input to TRCCNT.<br>000: TRCCNT counts the internal clock $\phi$.<br>001: TRCCNT counts the internal clock $\phi/2$.<br>010: TRCCNT counts the internal clock $\phi/4$.<br>011: TRCCNT counts the internal clock $\phi/8$.<br>100: TRCCNT counts the internal clock $\phi/32$.<br>101: TRCCNT counts the rising edge of the external event (FTCI).<br>110: Setting prohibited<br>111: Reserved (setting prohibited) | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 3 | TOD | Timer output level setting D | 0: Output value is 0*[1].<br>1: Output value is 1*[1]. | R/W |
| 2 | TOC | Timer output level setting C | 0: Output value is 0*[1].<br>1: Output value is 1*[1]. | R/W |
| 1 | TOB | Timer output level setting B | 0: Output value is 0*[1].<br>1: Output value is 1*[1]. | R/W |
| 0 | TOA | Timer output level setting A | 0: Output value is 0*[1].<br>1: Output value is 1*[1]. | R/W |

Notes: 1. The change of the setting is immediately reflected in the output value.
2. When the counter clock is switched over, the counter should be halted.

- TOD bit (timer output level setting D)

  Sets the output value of the FTIOD pin until the first compare match D is generated. In PWM mode, controls the output polarity of the FTIOD pin.

- TOC bit (timer output level setting C)

  Sets the output value of the FTIOC pin until the first compare match C is generated. In PWM mode, controls the output polarity of the FTIOC pin.

- TOB bit (timer output level setting B)

  Sets the output value of the FTIOB pin until the first compare match B is generated. In PWM mode, controls the output polarity of the FTIOB pin.

- TOA bit (timer output level setting A)

  Sets the output value of the FTIOA pin until the first compare match A is generated. In PWM mode, controls the output polarity of the FTIOA pin.

chanhvinh.com



### 15.2.3 Timer RC Control Register 2 (TRCCR2)

Address: H'FFFF90

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TCEG[1:0] | | CSTP | — | — | POLD | POLC | POLB |
| Value after reset: | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7, 6 | TCEG[1:0] | TRGC input edge select | 00: A trigger input on TRGC is disabled.<br>01: The rising edge is selected.<br>10: The falling edge is selected.<br>11: Both edges are selected. | R/W |
| 5 | CSTP | Count stop | 0: TRCCNT counting up continues.<br>1: TRCCNT counting up is halted. | R/W |
| 4, 3 | — | Reserved | These bits are read as 1. The write value should be 1. | — |
| 2 | POLD | PWM mode output level control D | 0: The TRCIOD output is active low.<br>1: The TRCIOD output is active high. | R/W |
| 1 | POLC | PWM mode output level control C | 0: The TRCIOC output is active low.<br>1: The TRCIOC output is active high. | R/W |
| 0 | POLB | PWM mode output level control B | 0: The TRCIOB output is active low.<br>1: The TRCIOB output is active high. | R/W |

- TCEG[1:0] bits (TRGC input edge select)

  These bits select the input edge of the TRGC signal. This function is only enabled when the PWM2 bit in TRCMR is set to 0.

- CSTP bit (count stop)

  Specifies whether TRCCNT counting up is halted by the compare match A signal. This function is enabled in all operating modes. To resume counting after counting has been stopped on a compare match, set the CTS bit in the timer RC mode register (TRCMR) to 1.

chanhvinh.com



### 15.2.4 Timer RC Interrupt Enable Register (TRCIER)

Address: H'FFFF8C

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | OVIE | — | — | — | IMIED | IMIEC | IMIEB | IMIEA |
| Value after reset: | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | OVIE | Timer overflow interrupt enable | 0: An interrupt (FOVI) requested by the OVF flag in TRCSR is disabled.<br>1: An interrupt (FOVI) requested by the OVF flag in TRCSR is enabled. | R/W |
| 6 to 4 | — | Reserved | These bits are read as 1. The write value should be 1. | — |
| 3 | IMIED | Input capture/ compare match interrupt enable D | 0: An interrupt (IMID) requested by the IMFD flag in TRCSR is disabled.<br>1: An interrupt (IMID) requested by the IMFD flag in TRCSR is enabled. | R/W |
| 2 | IMIEC | Input capture/ compare match interrupt enable C | 0: An interrupt (IMIC) requested by the IMFC flag in TRCSR is disabled.<br>1: An interrupt (IMIC) requested by the IMFC flag in TRCSR is enabled. | R/W |
| 1 | IMIEB | Input capture/ compare match interrupt enable B | 0: An interrupt (IMIB) requested by the IMFB flag in TRCSR is disabled.<br>1: An interrupt (IMIB) requested by the IMFB flag in TRCSR is enabled. | R/W |
| 0 | IMIEA | Input capture/ compare match interrupt enable A | 0: An interrupt (IMIA) requested by the IMFA flag in TRCSR is disabled.<br>1: An interrupt (IMIA) requested by the IMFA flag in TRCSR is enabled. | R/W |

chanhvinh.com



### 15.2.5 Timer RC Status Register (TRCSR)

Address: H'FFFF8D

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | OVF | — | — | — | IMFD | IMFC | IMFB | IMFA |
| Value after reset: | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | OVF | Timer overflow flag | 0: TRCCNT has not overflowed.<br>1: TRCCNT has overflowed.<br>[Setting condition]<br>• When TRCCNT overflows from H'FFFF to H'0000.<br>[Clearing condition]<br>• Read OVF when OVF = 1, then write 0 in OVF. | R/W |
| 6 to 4 | — | Reserved | These bits are read as 1. The write value should be 1. | — |
| 3 | IMFD | Input capture/ compare match flag D | [Setting conditions]<br>• TRCCNT = GRD when GRD functions as an output compare register.<br>• The TRCCNT value is transferred to GRD by an input capture signal when GRD functions as an input capture register.<br>• TRCCNT = GRD when the PWMD bit is set to 1 or the PWM2 bit to 0 in TRCMR.<br>[Clearing conditions]<br>• Read IMFD when IMFD = 1, then write 0 in IMFD.<br>• The DTC is activated by an IMFD interrupt and the DISEL bit in MRB of DTC is 0. | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 2 | IMFC | Input capture/ compare match flag C | [Setting conditions]<br>• TRCCNT = GRC when GRC functions as an output compare register.<br>• The TRCCNT value is transferred to GRC by an input capture signal when GRC functions as an input capture register.<br>• TRCCNT = GRC when the PWMC bit is set to 1 or the PWM2 bit to 0 in TRCMR.<br>[Clearing conditions]<br>• Read IMFC when IMFC = 1, then write 0 in IMFC.<br>• The DTC is activated by an IMFC interrupt when the DISEL bit in MRB of DTC is 0. | R/W |
| 1 | IMFB | Input capture/ compare match flag B | [Setting conditions]<br>• TRCCNT = GRB when GRB functions as an output compare register.<br>• The TRCCNT value is transferred to GRB by an input capture signal when GRB functions as an input capture register.<br>• TRCCNT = GRB when the PWMB bit is set to 1 or the PWM2 bit to 0 in TRCMR.<br>[Clearing conditions]<br>• Read IMFB when IMFB = 1, then write 0 in IMFB.<br>• The DTC is activated by an IMFB interrupt when the DISEL bit in MRB of DTC is 0. | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 0 | IMFA | Input capture/ compare match flag A | [Setting conditions]<br>• TRCCNT = GRA when GRA functions as an output compare register.<br>• The TRCCNT value is transferred to GRA by an input capture signal when GRA functions as an input capture register.<br>[Clearing condition]<br>• Read IMFA when IMFA = 1, then write 0 in IMFA.<br>The DTC is activated by an IMFA interrupt when the DISEL bit in MRB of DTC is 0. | R/W |

chanhvinh.com



### 15.2.6 Timer RC I/O Control Register 0 (TRCIOR0)

Address: H'FFFF8E

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | IOB2 | IOB[1:0] | | — | IOA2 | IOA[1:0] | |
| Value after reset: | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is read as 1. The write value should be 1. | — |
| 6 | IOB2 | I/O control B2 | Selects the GRB function.<br>0: GRB functions as an output compare register<br>1: GRB functions as an input capture register | R/W |
| 5, 4 | IOB[1:0] | I/O control B1 and B0 | When IOB2 = 0,<br>00: No output on compare match<br>01: 0 output to the FTIOB pin on compare match of GRB<br>10: 1 output to the FTIOB pin on compare match of GRB<br>11: Toggle output to the FTIOB pin on compare match of GRB<br>When IOB2 = 1,<br>00: Input capture to GRB at rising edge at the FTIOB pin<br>01: Input capture to GRB at falling edge at the FTIOB pin<br>1X: Input capture to GRB at rising and falling edges of the FTIOB pin | R/W |
| 3 | — | Reserved | This bit is read as 1. The write value should be 1. | — |
| 2 | IOA2 | I/O control A2 | Selects the GRA function.<br>0: GRA functions as an output compare register<br>1: GRA functions as an input capture register | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 1, 0 | IOA[1:0] | I/O control A1 and A0 | When IOA2 = 0,<br>00: No output on compare match<br>01: 0 output to the FTIOA pin on compare match of GRA<br>10: 1 output to the FTIOA pin on compare match of GRA<br>11: Toggle output to the FTIOA pin on compare match of GRA<br>When IOA2 = 1,<br>00: Input capture to GRA at rising edge of the FTIOA pin<br>01: Input capture to GRA at falling edge of the FTIOA pin<br>1X: Input capture to GRA at rising and falling edges of the FTIOA pin | R/W |

[Legend]

X: Don't care.

Notes: 1. When a GR register functions as a buffer register for a paired GR register, the settings in the IOA2 and IOB2 bits in TRCIOR0 and the IOC2 and IOD2 bits in TRCIOR1 of both registers should be the same.

2. The setting of TRCIOR is invalid in PWM mode and PWM2 mode.

TRCIOR0 selects the functions of GRA and GRB, and specifies the functions of the FTIOA and FTIOB pins.

chanhvinh.com



### 15.2.7 Timer RC I/O Control Register 1 (TRCIOR1)

Address: H'FFFF8F

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | IOD3 | IOD2 | IOD[1:0] | | IOC3 | IOC2 | IOC[1:0] | |
| Value after reset: | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | IOD3 | I/O control D3 | 0: GRD is used as GR for the FTIOB pin<br>1: GRD is used as GR for the FTIOD pin | R/W |
| 6 | IOD2 | I/O control D2 | 0: GRD functions as an output compare register<br>1: GRD functions as an input capture register | R/W |
| 5, 4 | IOD[1:0] | I/O control D1 and D0 | When IOD3 = 0,<br>00: No output on compare match<br>01: 0 output to the FTIOB pin on compare match of GRD<br>10: 1 output to the FTIOB pin on compare match of GRD<br>11: Toggle output to the FTIOB pin on compare match of GRD<br>When IOD3 = 1 and IOD2 = 0,<br>00: No output on compare match<br>01: 0 output to the FTIOD pin on compare match of GRD<br>10: 1 output to the FTIOD pin on compare match of GRD<br>11: Toggle output to the FTIOD pin on compare match of GRD<br>When IOD3 = 1 and IOD2 = 1,<br>00: Input capture to GRD at rising edge of the FTIOD pin<br>01: Input capture to GRD at falling edge of the FTIOD pin<br>1X: Input capture to GRD at rising and falling edges of the FTIOD pin | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 3 | IOC3 | I/O control C3 | 0: GRC is used as GR for the FTIOA pin<br>1: GRC is used as GR for the FTIOC pin | R/W |
| 2 | IOC2 | I/O control C2 | 0: GRC functions as an output compare register<br>1: GRC functions as an input capture register | R/W |
| 1, 0 | IOC[1:0] | I/O control C1 and C0 | When IOC3 = 0,<br>00: No output on compare match<br>01: 0 output to the FTIOA pin on compare match of GRC<br>10: 1 output to the FTIOA pin on compare match of GRC<br>11: Toggle output to the FTIOA pin on compare match of GRC<br>When IOC3 = 1 and IOC2 = 0,<br>00: No output on compare match<br>01: 0 output to the FTIOC pin on compare match of GRC<br>10: 1 output to the FTIOC pin on compare match of GRC<br>11: Toggle output to the FTIOC pin on compare match of GRC<br>When IOC3 = 1 and IOC2 = 1,<br>00: Input capture to GRC at rising edge of the FTIOC pin<br>01: Input capture to GRC at falling edge of the FTIOC pin<br>1X: Input capture to GRC at rising and falling edges of the FTIOC pin | R/W |

[Legend]

X: Don't care.

Notes: 1. When a GR register functions as a buffer register for a paired GR register, the settings in the IOA2 and IOB2 bits in TRCIOR0 and the IOC2 and IOD2 bits in TRCIOR1 of both registers should be the same.

2. The setting of TRCIOR1 is invalid in PWM mode and PWM2 mode.

chanhvinh.com



### 15.2.8 Timer RC Output Enable Register (TRCOER)

Address: H'FFFF92

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PTO | — | — | — | ED | EC | EB | EA |
| Value after reset: | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PTO | Timer output disabled mode | 0: The ED, EC, EB and EA bits are not set to 1 by the low level input of the $\overline{\text{TRCOI}}$ signal.<br>1: The ED, EC, EB and EA bits are set to 1 by the low level input of the $\overline{\text{TRCOI}}$ signal. | R/W |
| 6 to 4 | — | Reserved | These bits are read as 1. The write value should be 1. | — |
| 3 | ED | Master enable D | 0: The FTIOD output is enabled according to the TRCMR and TRCIOR1 settings<br>1: The FTIOD output is disabled regardless of the TRCMR and TRCIOR1 settings. | R/W |
| 2 | EC | Master enable C | 0: The FTIOC output is enabled according to the TRCMR and TRCIOR1 settings.<br>1: The FTIOC output is disabled regardless of the TRCMR and TRCIOR1 settings. | R/W |
| 1 | EB | Master enable B | 0: The FTIOB output is enabled according to the TRCMR and TRCIOR0 settings<br>1: The FTIOB output is disabled regardless of the TRCMR and TRCIOR0 settings. | R/W |
| 0 | EA | Master enable A | 0: The FTIOA output is enabled according to the TRCIOR0 settings<br>1: The FTIOA output is disabled regardless of the TRCIOR0 settings. | R/W |

TRCOER enables or disables the timer outputs. When setting the PTO bit to 1 and driving the TRCOI signal low, the ED, EC, EB and EA bits are set to 1 and timer RC outputs are disabled.

chanhvinh.com



### 15.2.9 Timer RC Digital Filtering Function Select Register (TRCDF)

Address: H'FFFF91

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | DFCK[1:0] | | — | DFTRG | DFD | DFC | DFB | DFA |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7, 6 | DFCK[1:0] | Digital filter clock select | These bits select the clock to be used by the digital filter.<br>00: $\phi$/32<br>01: $\phi$/8<br>10: $\phi$<br>11: Clock specified by bits CKS2 to CKS0 in TRCCR1 | R/W |
| 5 | — | Reserved | This bit is read as 0. The write value should be 0. | — |
| 4 | DFTRG | Digital filter function trigger pin | 0: Disables the digital filter for the TRGC pin<br>1: Enables the digital filter for the TRGC pin | R/W |
| 3 | DFD | Digital filter function D | 0: Disables the digital filter for the FTIOD pin<br>1: Enables the digital filter for the FTIOD pin | R/W |
| 2 | DFC | Digital filter function C | 0: Disables the digital filter for the FTIOC pin<br>1: Enables the digital filter for the FTIOC pin | R/W |
| 1 | DFB | Digital filter function B | 0: Disables the digital filter for the FTIOB pin<br>1: Enables the digital filter for the FTIOB pin | R/W |
| 0 | DFA | Digital filter function A | 0: Disables the digital filter for the FTIOA pin<br>1: Enables the digital filter for the FTIOA pin | R/W |

Note: The setting in this register is valid on the corresponding pin when the FTIOA to FTIOD inputs are enabled by TRCIOR0 and TRCIOR1 and the TRGC input is selected by bits TCEG1 and TCEG0 in TRCCR2.

chanhvinh.com



### 15.2.10 Timer RC A/D Conversion Start Trigger Control Register (TRCADCR)

Address: H'FFFF93

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | ADTRGAE | ADTRGBE | ADTRGCE | ADTRGDE |
| Value after reset: | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 to 4 | — | Reserved | These bits are read as 1. The write value should be 1. | — |
| 3 | ADTRGAE | A/D conversion start trigger A enable | 0: A/D conversion start trigger is not generated by compare match of GRA<br>1: A/D conversion start trigger is generated by compare match of GRA | R/W |
| 2 | ADTRGBE | A/D conversion start trigger B enable | 0: A/D conversion start trigger is not generated by compare match of GRB<br>1: A/D conversion start trigger is generated by compare match of GRB | R/W |
| 1 | ADTRGCE | A/D conversion start trigger C enable | 0: A/D conversion start trigger is not generated by compare match of GRC<br>1: A/D conversion start trigger is generated by compare match of GRC | R/W |
| 0 | ADTRGDE | A/D conversion start trigger D enable | 0: A/D conversion start trigger is not generated by compare match of GRD<br>1: A/D conversion start trigger is generated by compare match of GRD | R/W |

TRCADCR selects the trigger source to start A/D conversion. A/D conversion start trigger is generated by a corresponding compare match.

chanhvinh.com



### 15.2.11 Timer RC Counter (TRCCNT)

Address: H'FFFF80

| Bit: | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

TRCCNT is a 16-bit readable/writable up-counter. The input clock is selected by bits CKS2 to CKS0 in TRCCR1. TRCCNT can be cleared to H'0000 through a compare match of GRA by setting the CCLR bit in TRCCR1 to 1. When TRCCNT overflows from H'FFFF to H'0000, the OVF flag in TRCSR is set to 1. If the OVIE bit in TRCIER is set to 1 at this time, an interrupt request is generated. TRCCNT must always be read from or written to in units of 16 bits; 8-bit accesses are not allowed. The initial value of TRCCNT is H'0000.

chanhvinh.com



### 15.2.12 General Registers A, B, C, and D (GRA, GRB, GRC, and GRD)

GRA

Address: H'FFFF82

| Bit: | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

GRB

Address: H'FFFF84

| Bit: | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

GRC

Address: H'FFFF86

| Bit: | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

GRD

Address: H'FFFF88

| Bit: | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

chanhvinh.com



Each general register is a 16-bit readable/writable register that can function as either an output-compare register or an input-capture register. The function is selected by settings in TRCIOR0 and TRCIOR1.

When a general register is used as an input-compare register, its value is constantly compared with the TRCCNT value. When the two values match (a compare match), the corresponding flag (the IMFA, IMFB, IMFC, or IMFD bit) in TRCSR is set to 1. An interrupt request is generated at this time, when the IMIEA, IMIEB, IMIEC, or IMIED bit in TRCIER is set to 1. A compare match output can be selected in TRCIOR.

When a general register is used as an input-capture register, an external input-capture signal is detected and the current TRCCNT value is stored in the general register. The corresponding flag (the IMFA, IMFB, IMFC, or IMFD bit) in TRCSR is set to 1. If the corresponding interrupt-enable bit (the IMIEA, IMIEB, IMIEC, or IMIED bit) in TRCIER is set to 1 at this time, an interrupt request is generated. The edge of the input-capture signal is selected in TRCIOR.

GRC and GRD can be used as buffer registers of GRA and GRB, respectively, by setting BUFEA and BUFEB in TRCMR.

For example, when GRA is set as an output-compare register and GRC is set as the buffer register for GRA, the value in the buffer register GRC is sent to GRA whenever compare match A is generated.

When GRA is set as an input-capture register and GRC is set as the buffer register for GRA, the value in TRCCNT is transferred to GRA and the value in the buffer register GRA is transferred to GRC whenever an input capture is generated.

GRA to GRD must be written or read in 16-bit units; 8-bit access is not allowed. GRA to GRD are initialized to H'FFFF by a reset.

chanhvinh.com



## 15.3 Operation

Timer RC has the following operating modes.

- Timer mode operation
  Enables output compare and input capture functions by setting the IOA2 to IOA0 and IOB2 to IOB0 bits in TRCIOR0 and the IOC3 to IOC0 and IOD3 to IOD0 bits in TRCIOR1.
- PWM mode operation
  Enables PWM mode operation by setting the PWMD, PWMC, and PWMB bits in TRCMR.
- PWM2 mode operation
  Enables PWM2 mode operation by setting the PWM2 bit in TRCMR.

The FTIOA to FTIOD pins indicate the timer output mode by each register setting. Set 1 to the PMCR and PMR bits corresponding to the pins selected by the PMC.

**Table 15.3 FTIOA Pin Functions**

| Register Name | TRCOER | TRCMR | TRCIOR0 | |
|---|---|---|---|---|
| **Bit Name** | **EA** | **PWM2** | **IOA2 to IOA0** | **Function** |
| Setting values | 0 | 1 | 001, 01X | Timer mode waveform output (output compare function) |
| | X | 1 | 1XX | Timer mode (input capture function) |
| | X | 1 | 000 | General input port (when PCR = 0 on the corresponding pin) |
| | Other than above | | | Setting prohibited |

[Legend]
X: Don't care.

chanhvinh.com



**Table 15.4 FTIOB Pin Functions**

| Register Name | TRCOER | TRCMR | | TRCIOR0 | |
|---|---|---|---|---|---|
| **Bit Name** | **EB** | **PWM2** | **PWMB** | **IOB2 to IOB0** | **Function** |
| Setting values | 0 | 0 | X | XXX | PWM2 mode waveform output |
| | 0 | 1 | 1 | XXX | PWM mode waveform output |
| | 0 | 1 | 0 | 001, 01X | Timer mode waveform output (output compare function) |
| | X | 1 | 0 | 1XX | Timer mode (input capture function) |
| | X | 1 | 0 | 000 | General input port (when PCR = 0 on the corresponding pin) |
| | Other than above | | | | Setting prohibited |

[Legend]
X: Don't care.

**Table 15.5 FTIOC Pin Functions**

| Register Name | TRCOER | TRCMR | | TRCIOR1 | |
|---|---|---|---|---|---|
| **Bit Name** | **EC** | **PWM2** | **PWMC** | **IOC2 to IOC0** | **Function** |
| Setting values | 0 | 1 | 1 | XXX | PWM mode waveform output |
| | 0 | 1 | 0 | 001, 01X | Timer mode waveform output (output compare function) |
| | X | 1 | 0 | 1XX | Timer mode (input capture function) |
| | X | 1 | 0 | 000 | General input port (when PCR = 0 on the corresponding pin) |
| | Other than above | | | | Setting prohibited |

[Legend]
X: Don't care.

chanhvinh.com



**Table 15.6 FTIOD Pin Functions**

| Register Name | TRCOER | TRCMR | | TRCIOR1 | |
|---|---|---|---|---|---|
| **Bit Name** | **ED** | **PWM2** | **PWMD** | **IOD2 to IOD0** | **Function** |
| Setting values | 0 | 1 | 1 | XXX | PWM mode waveform output |
| | 0 | 1 | 0 | 001, 01X | Timer mode waveform output (output compare function) |
| | X | 1 | 0 | 1XX | Timer mode (input capture function) |
| | X | 1 | 0 | 000 | General input port (when PCR = 0 on the corresponding pin) |
| | Other than above | | | | Setting prohibited |

[Legend]
X: Don't care.

### 15.3.1 Timer Mode Operation

TRCCNT performs free-running or periodic counting operations. After a reset, TRCCNT is set as a free-running counter. When the CTS bit in TRCMR is set to 1, TRCCNT starts counting. When the TRCCNT value overflows from H'FFFF to H'0000, the OVF flag in TRCSR is set to 1. If the OVIE in TRCIER is set to 1, an interrupt request is generated. Figure 15.2 shows an example of free-running counting.

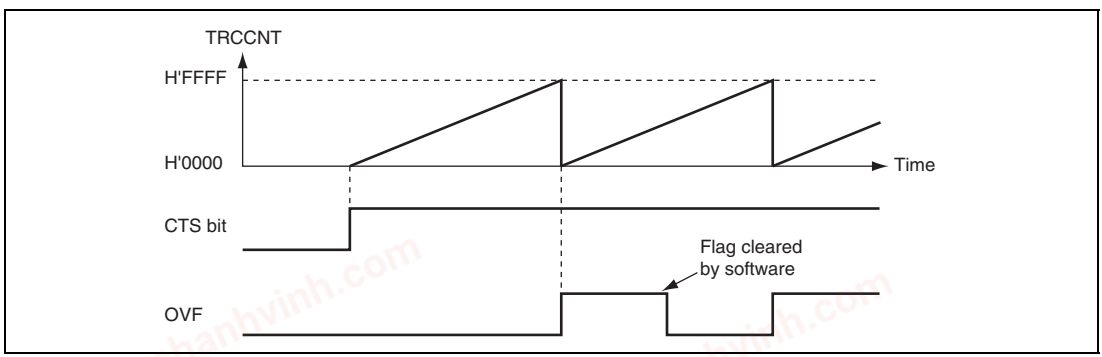



**Figure 15.2 Free-Running Counter Operation**

chanhvinh.com



Periodic counting operation can be performed when GRA is set as an output compare register and the CCLR bit in TRCCR1 is set to 1. When the counter value matches GRA, TRCCNT is cleared to H'0000, and the IMFA flag in TRCSR is set to 1. If the corresponding IMIEA bit in TRCIER is set to 1, an interrupt request is generated. TRCCNT continues counting from H'0000. Figure 15.3 shows an example of periodic counting.

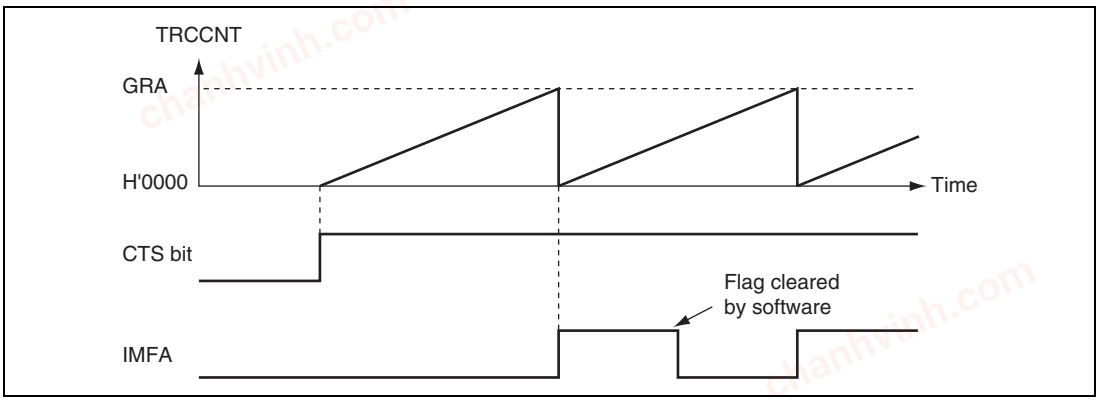



**Figure 15.3 Periodic Counter Operation**

By setting a general register as an output compare register, the specified level of a signal can be output on the FTIOA, FTIOB, FTIOC, or FTIOD pin on compare match A, B, C, or D. The output level can be selected from 0, 1, or toggle. Figure 15.4 shows an example of TRCCNT functioning as a free-running counter. In this example, 1 is output on compare match A and 0 is output on compare match B. When the signal level is already at the selected output level, it is not changed on a compare match.

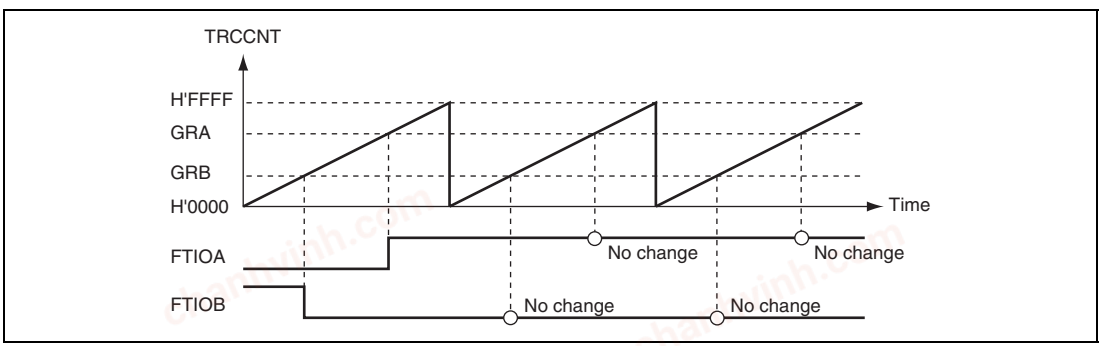



**Figure 15.4 0 and 1 Output Example (TOA = 0, TOB = 1)**

chanhvinh.com



Figure 15.5 shows an example of toggled output when TRCCNT functions as a free-running counter, and the toggled output is selected for both compare matches A and B.

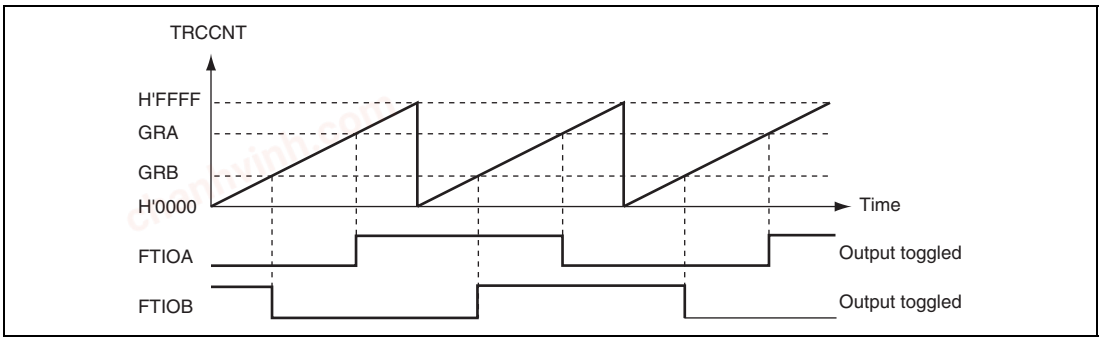



**Figure 15.5 Toggle Output Example (TOA = 0, TOB = 1)**

Figure 15.6 shows another example of toggled output when TRCCNT functions as a periodic counter on both compare matches A and B.

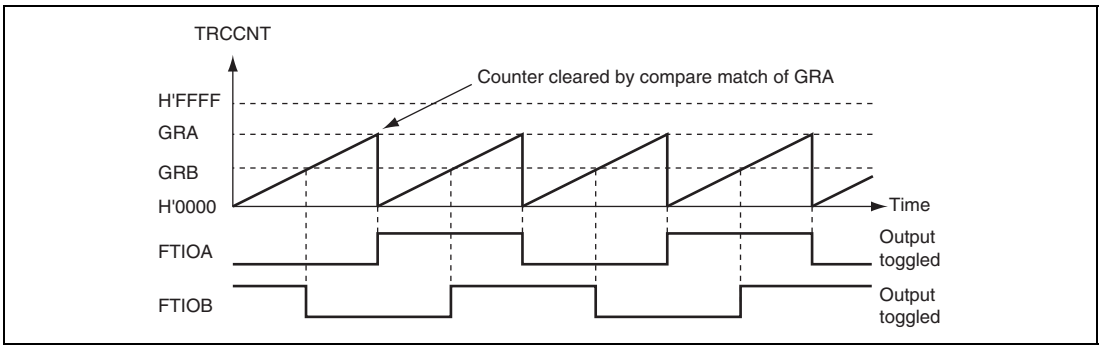



**Figure 15.6 Toggle Output Example (TOA = 0, TOB = 1)**

chanhvinh.com



The TRCCNT value can be captured into a general register (GRA, GRB, GRC, or GRD) when signal levels are changed on an input-capture pin (FTIOA, FTIOB, FTIOC, or FTIOD) by specifying the general register as an input capture register. The capture timing can be selected from the rising, falling, or both edges. By using the input-capture function, the width or cycle of a pulse can be measured. Figure 15.7 shows an example of an input capture when both edges of the FTIOA signal and the falling edge of the FTIOB signal are selected as capture timings. TRCCNT functions as a free-running counter.

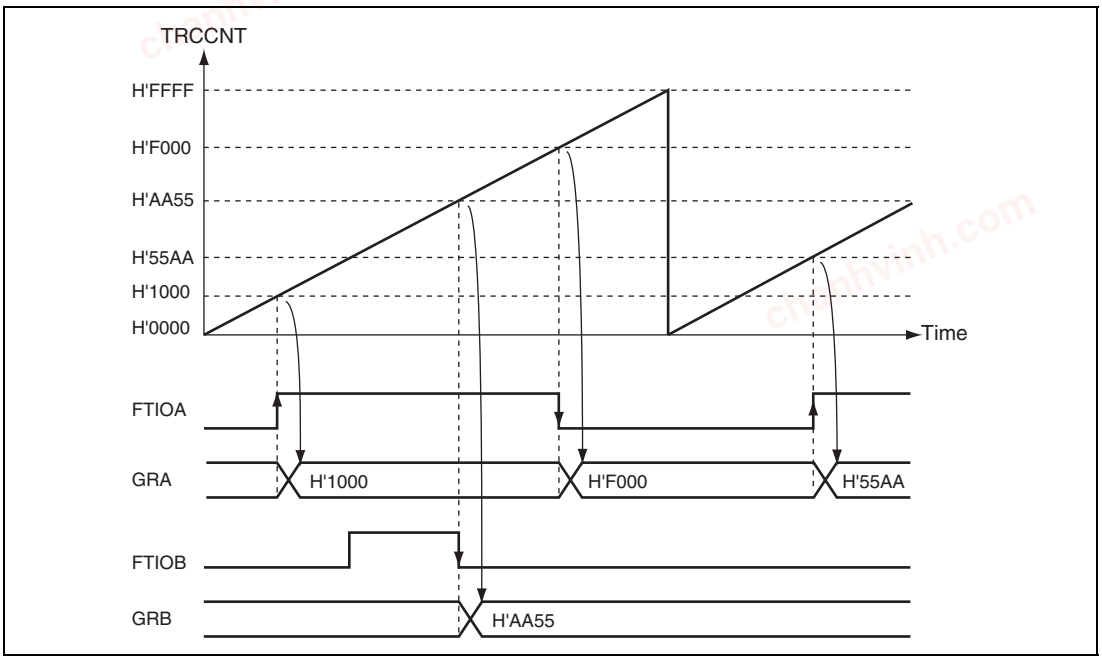



**Figure 15.7 Input Capture Operating Example**

chanhvinh.com



Figure 15.8 shows an example of buffer operation when GRA is set as an input-capture register and GRC is set as the buffer register for GRA. TRCCNT functions as a free-running counter and is captured at both rising and falling edges of the FTIOA signal. Due to the buffer operation, the GRA value is transferred to GRC on an input-capture A and the TRCCNT value is stored in GRA.

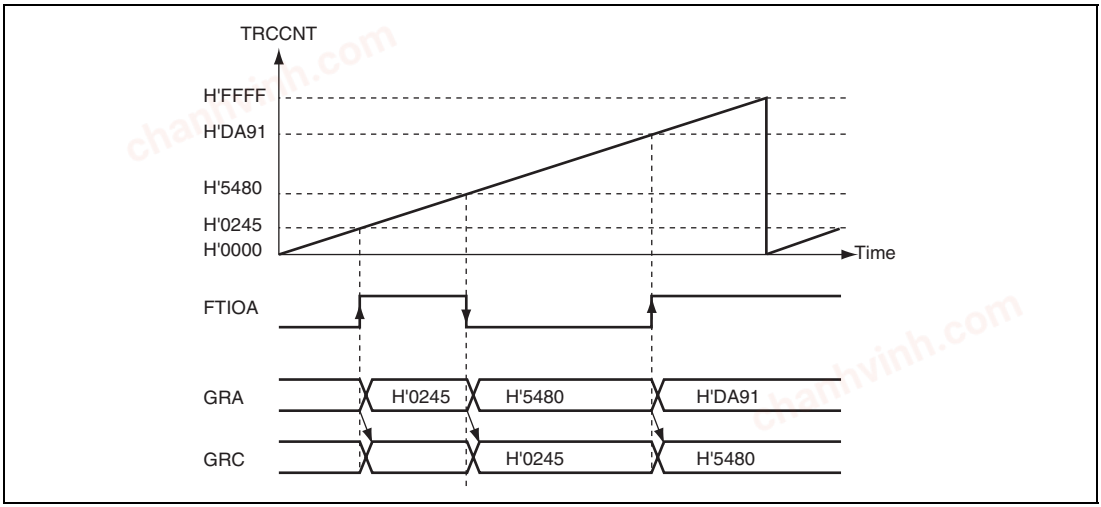



**Figure 15.8 Buffer Operation Example (Input Capture)**

chanhvinh.com



### 15.3.2 PWM Mode Operation

In PWM mode, PWM waveforms are generated by using GRA as the cycle register and GRB, GRC, and GRD as duty cycle registers. PWM waveforms are output from the FTIOB, FTIOC, and FTIOD pins. Up to three-phase PWM waveforms can be output. In PWM mode, a general register functions as an output compare register automatically. The initial output level of each pin depends on the settings in TRCCR1 and TRCCR2. Table 15.7 shows an example of the initial output level of the FTIOB pin.

**Table 15.7 Initial Output Level of FTIOB Pin**

| Bit TOB (TRCCR1) | Bit POLB (TRCCR2) | Initial Output Level |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

The output level of each pin is determined by the value of the corresponding PWM mode output level control bit (POLB, POLC, or POLD) in TRCCR2. When POLB is 0, the FTIOB output pin is set to 0 on compare match B, and set to 1 on compare match A, whereas when POLB is 1, the FTIOB output pin is set to 1 on compare match B, and set to 0 on compare match A. When an output pin is set to PWM mode, the settings in TRCIOR0 and TRCIOR1 are ignored. If the same value is set in the cycle register and duty cycle register, output levels are not changed when a compare match occurs.

chanhvinh.com



Figure 15.9 shows an example of operation in PWM mode. The output signals go 1 and TRCCNT is cleared on compare match A, and the output signals go 0 on compare match B, C, and D.

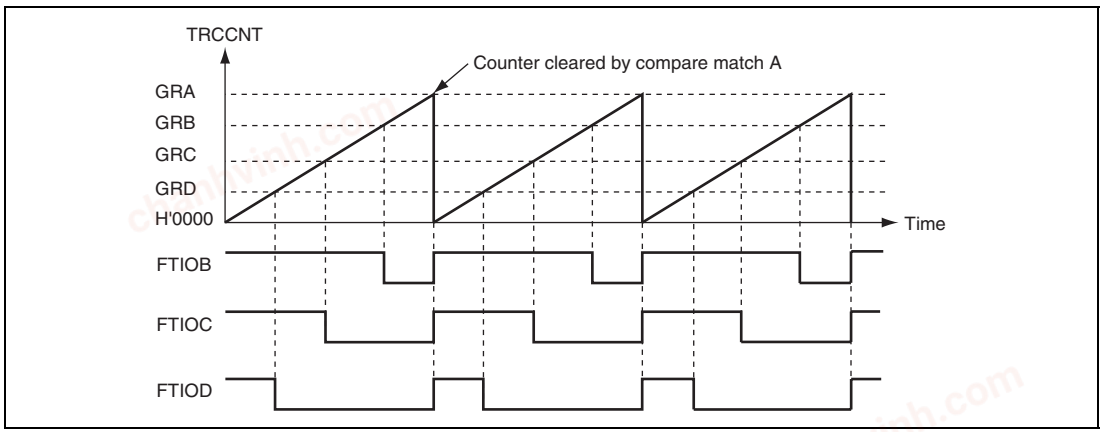



**Figure 15.9 PWM Mode Example (1)**

Figure 15.10 shows another example of operation in PWM mode. The output signals go 0 and TRCCNT is cleared on compare match A, and the output signals go 1 on compare match B, C, and D.

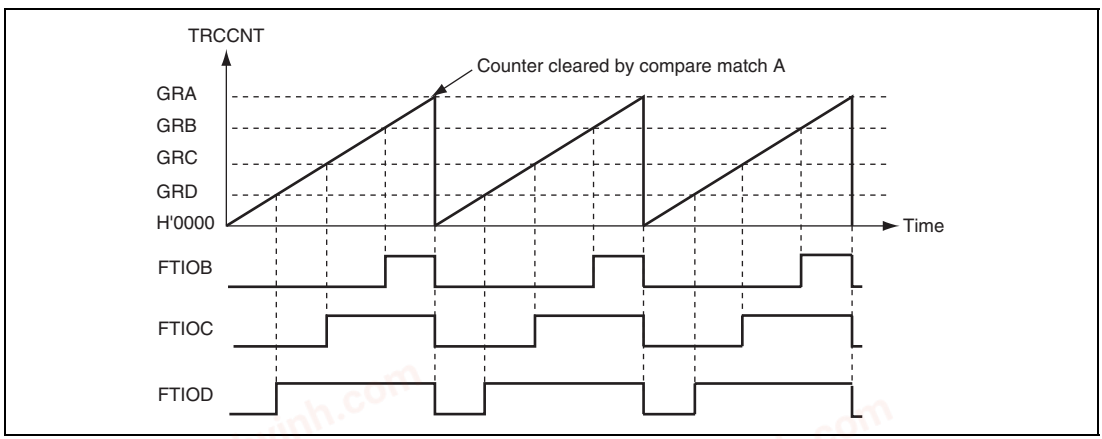



**Figure 15.10 PWM Mode Example (2)**

chanhvinh.com



Figure 15.11 shows an example of buffer operation when the FTIOB pin is set to PWM mode and GRD is set as the buffer register for GRB. TRCCNT is cleared on compare match A, and the FTIOB pin outputs 1 on compare match B and 0 on compare match A.

Due to the buffer operation, the FTIOB output levels are changed and the value of buffer register GRD is transferred to GRB whenever compare match B occurs. This procedure is repeated every time compare match B occurs.

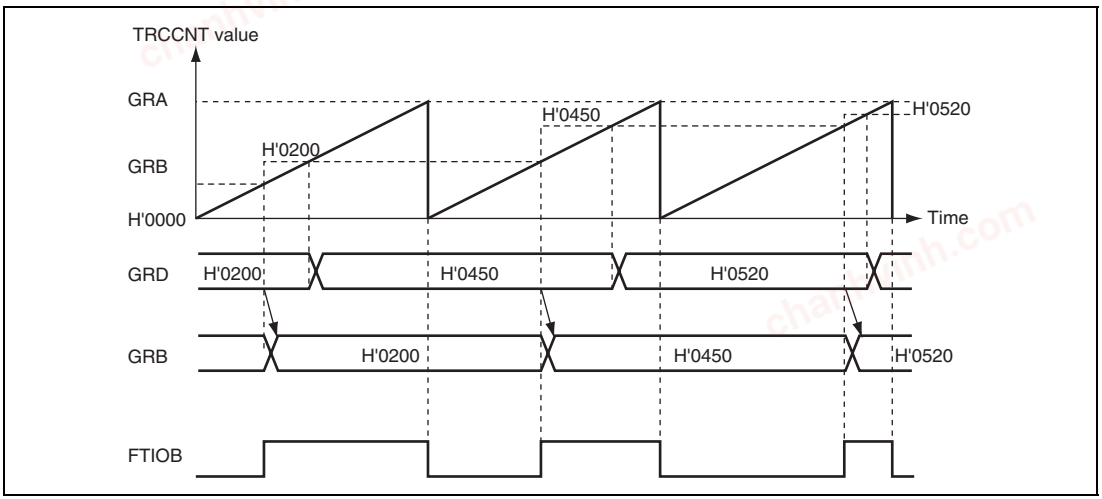



**Figure 15.11 Buffer Operation Example (Output Compare)**

chanhvinh.com



Figures 15.12 and 15.13 show examples of the output of PWM waveforms with duty cycles of 0% and 100%.

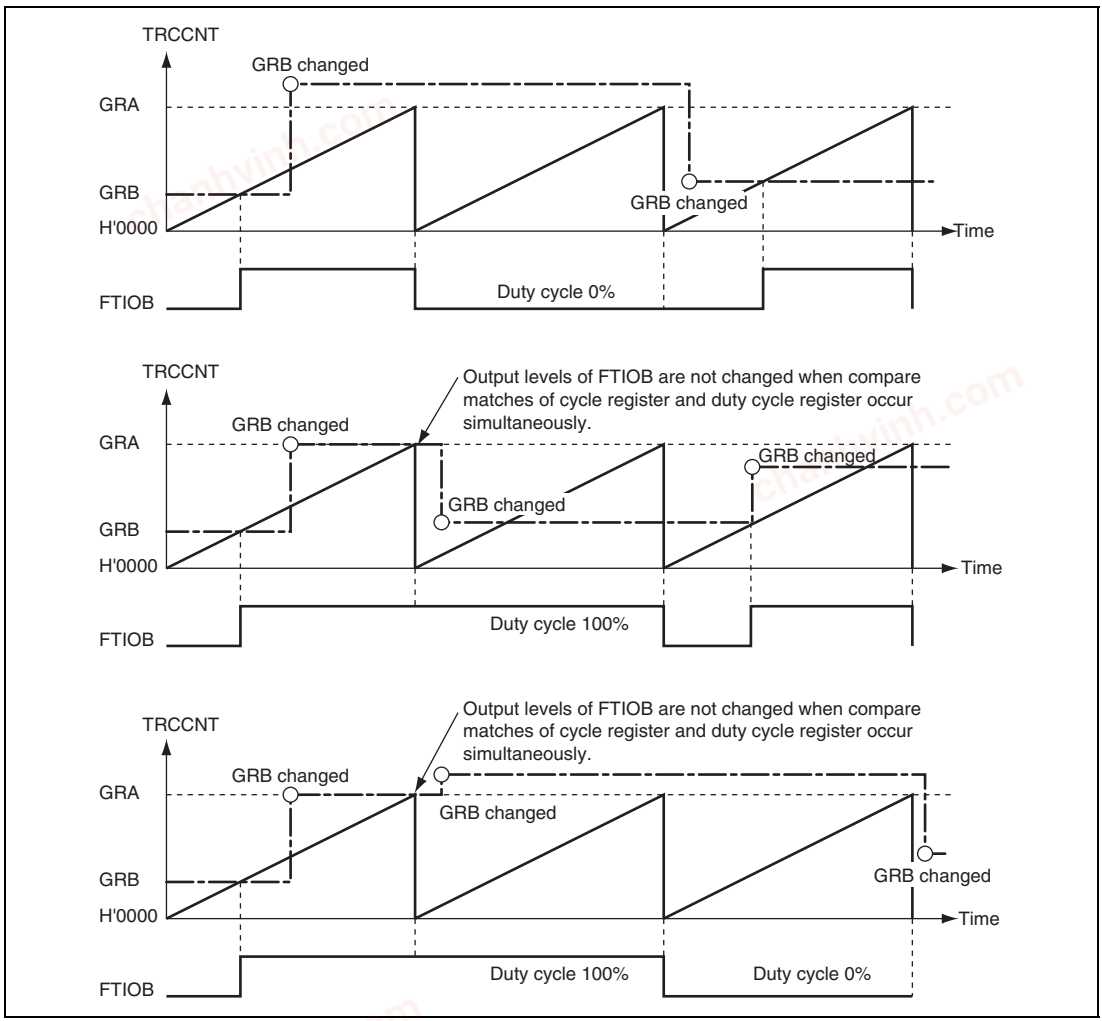



**Figure 15.12 PWM Mode Example (Initial Output Set to 0)**

chanhvinh.com



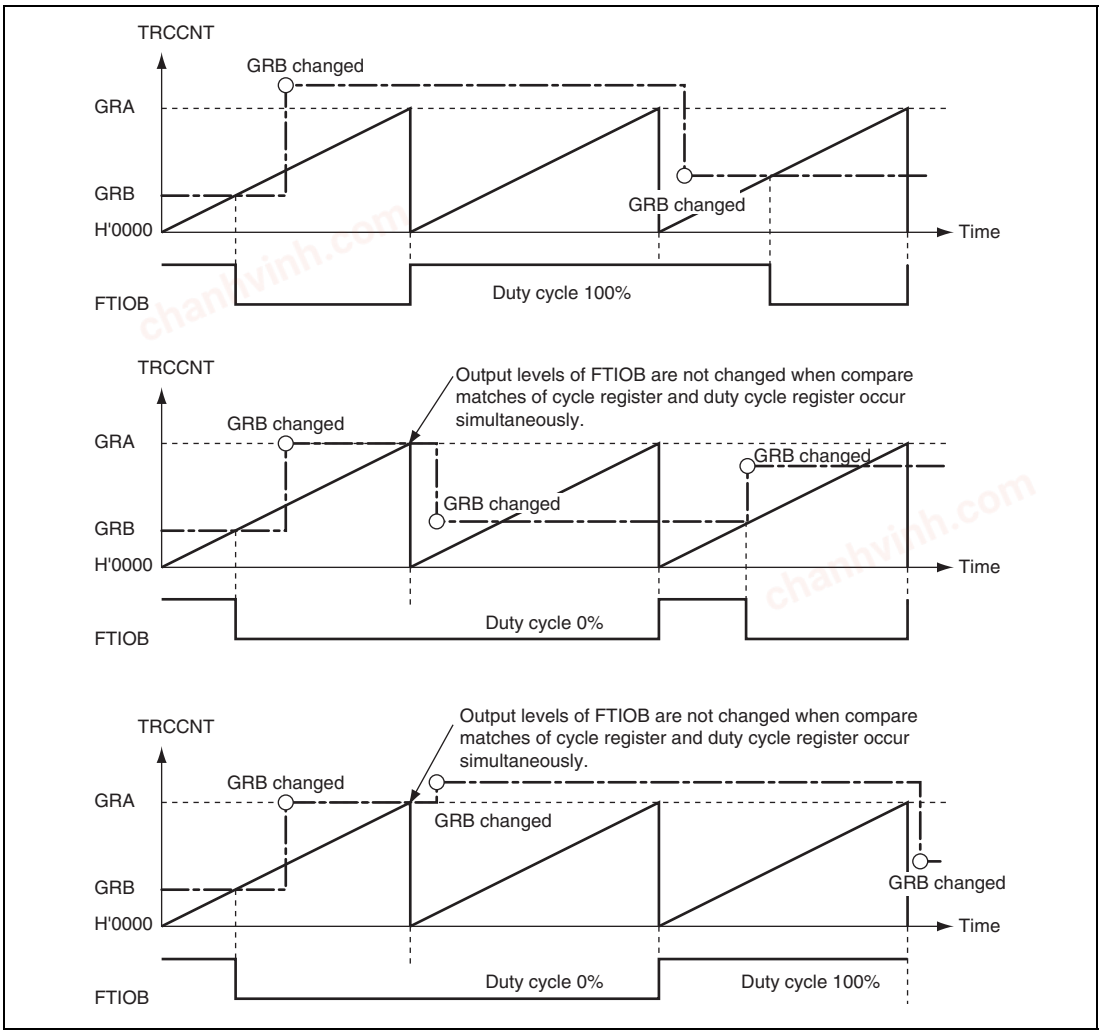



**Figure 15.13 PWM Mode Example (Initial Output Set to 1)**

chanhvinh.com



### 15.3.3 PWM2 Mode Operation

In PWM2 mode, waveforms are output on the FTIOB pin when a compare match occurs on GRB or GRC. GRD functions as a buffer register for GRB by setting the BUFEB bit in TRCMR to 1. The output level of the FTIOB signal is specified by the TOB bit in TRCCR1. When TOB = 0, 1 is output on a compare match of GRC and 0 is output on a compare match of GRB. When TOB = 1, 0 is output on a compare match of GRC and 1 is output on a compare match of GRB.

Table 15.8 shows the correspondence between the pin configuration and GR registers and figure 15.14 is a block diagram in PWM2 mode.

Figures 15.15 and 15.16 show the GRD and GRB buffer operating timing in PWM2 mode.

In PWM2 mode, the value of GRD is transferred to GRB on a compare match of GRA and the counter is cleared. Note, however, that the counter is only cleared when the CCLR bit in TRCCR1 is set to 1. Moreover, when the trigger input is enabled by the TCEG1 and TCEG0 bits in TRCCR2, the value of GRD is transferred to GRB by the trigger signal and the counter is cleared. The input/output pins of timers which do not operate in PWM2 mode are only used as general I/O ports.

**Table 15.8 Pin Configuration in PWM2 Mode and GR Registers**

| Pin Name | Input/Output | Compare Match Register | Buffer Register |
|---|---|---|---|
| FTIOA | I/O | Port*/TRGC | Port*/TRGC |
| FTIOB | Output | GRB | GRD |
| | | GRC | — |
| FTIOC | I/O | Port* | Port* |
| FTIOD | I/O | Port* | Port* |

Note: * When the port functions, clear the PMR bit on the corresponding pin to 0.

chanhvinh.com



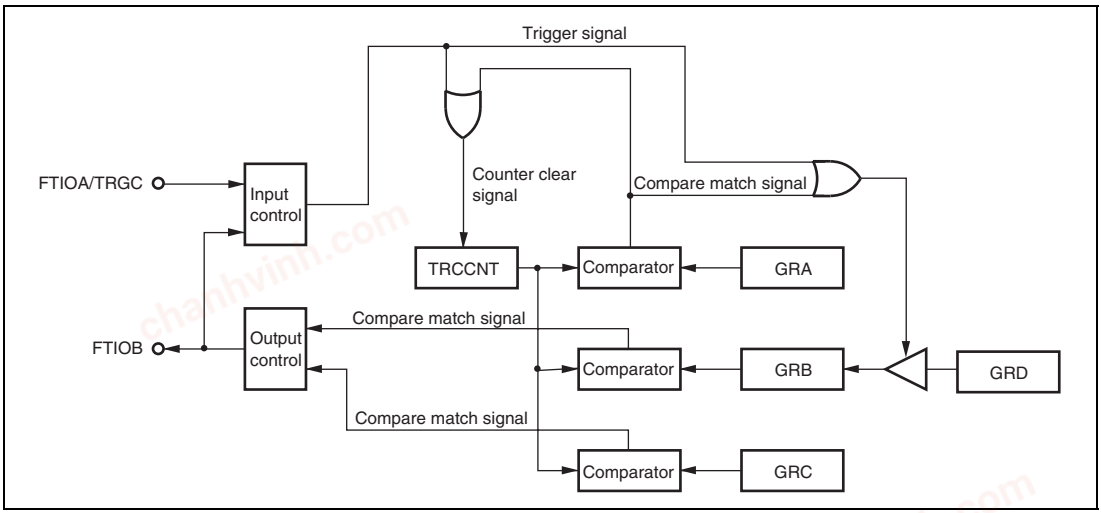



**Figure 15.14 Block Diagram in PWM2 Mode**

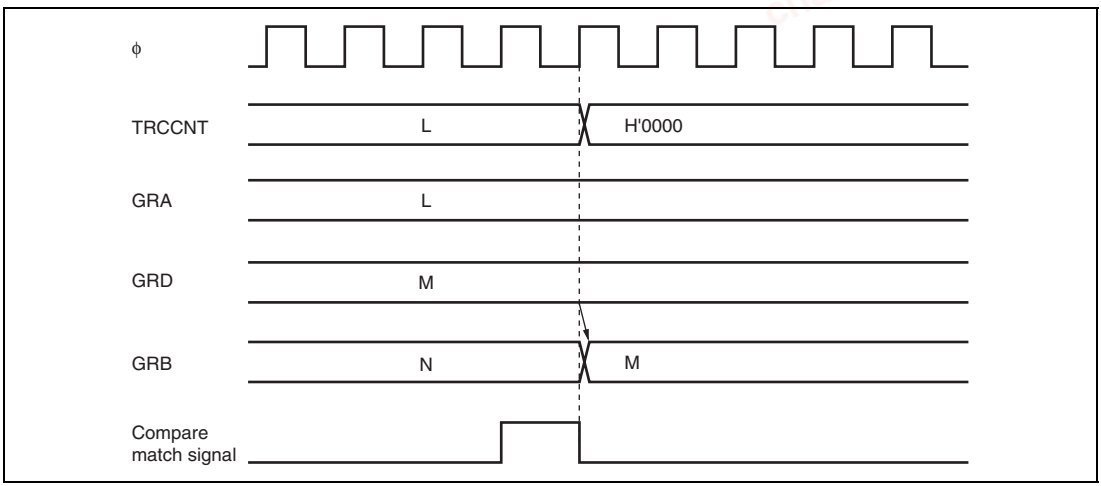



**Figure 15.15 GRD and GRB Buffer Operating Timing in PWM2 Mode (1)**

chanhvinh.com



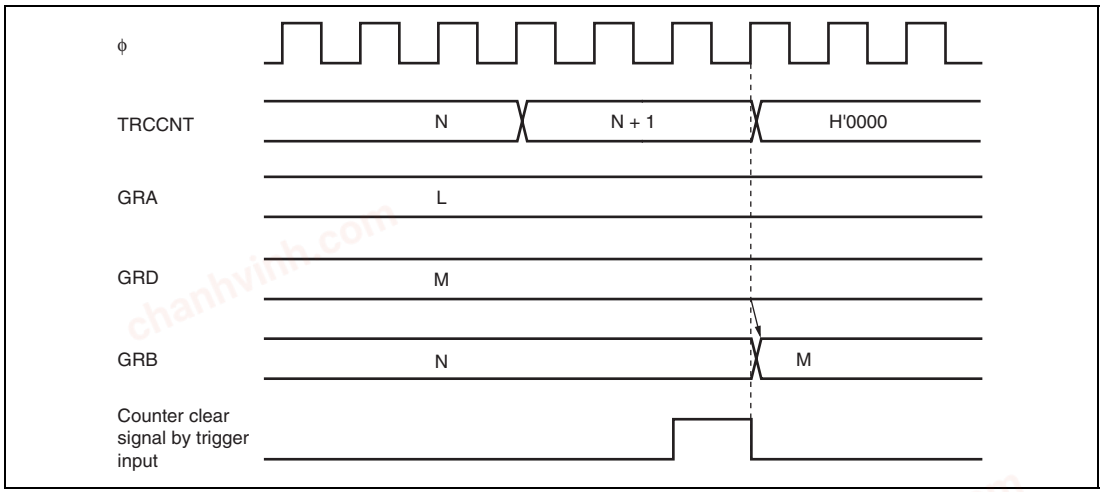



**Figure 15.16 GRD and GRB Buffer Operating Timing in PWM2 Mode (2)**

In PWM2 mode, a pulse with arbitrary pulse width and delay time to the TRGC input can be output from the FTIOB pin

Figures 15.17 and 15.18 show these examples in PWM2 mode. In these examples, the falling edge of the TRGC input is selected by TRCCR2 (setting the TCEG1 bit to 1 and clearing the TCEG0 bit to 0), TRCCNT continues counting-up on compare match A of GRA (clearing the CSTP bit in TRCCR2 to 0), and GRD is set as the buffer register (setting the BUFEB bit in TRCMR to 1). The initial value of the output signal is set to either 0 or 1 by TRCCR1 (clearing the TOB bit to 0 or setting the TOB bit to 1), TRCCNT is cleared on compare match A (setting the CCLR bit in TRCCR1 to 1), and the waveform is output from the FTIOB pin (clearing the PWM2 bit in TRCMR to 0).

When the TOB bit in TRCCR1 is cleared to 0 with the PWM2 mode function, the input edge is ignored while the FTIOB pin is driven high. Whereas, when the TOB bit is set to 1, the input edge is ignored while the FTIOB pin is driven low. The transfer from GRD to GRB is carried out on a compare match of GRA and the TRGC input. However, if the TRGC input is canceled due to the change of the FTIOB level, the transfer from GRD to GRB is not carried out.

chanhvinh.com



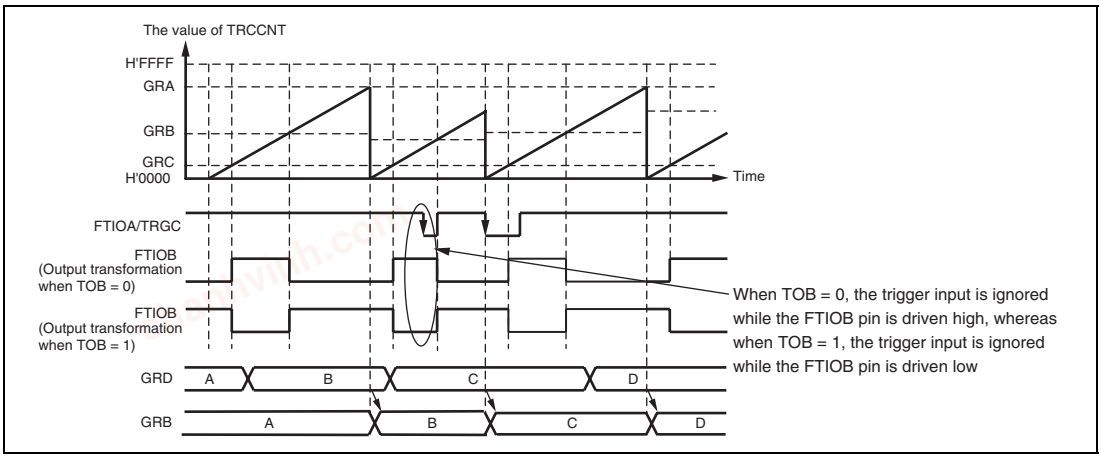



**Figure 15.17 Example (1) of TRGC Synchronous Operation in PWM2 Mode**

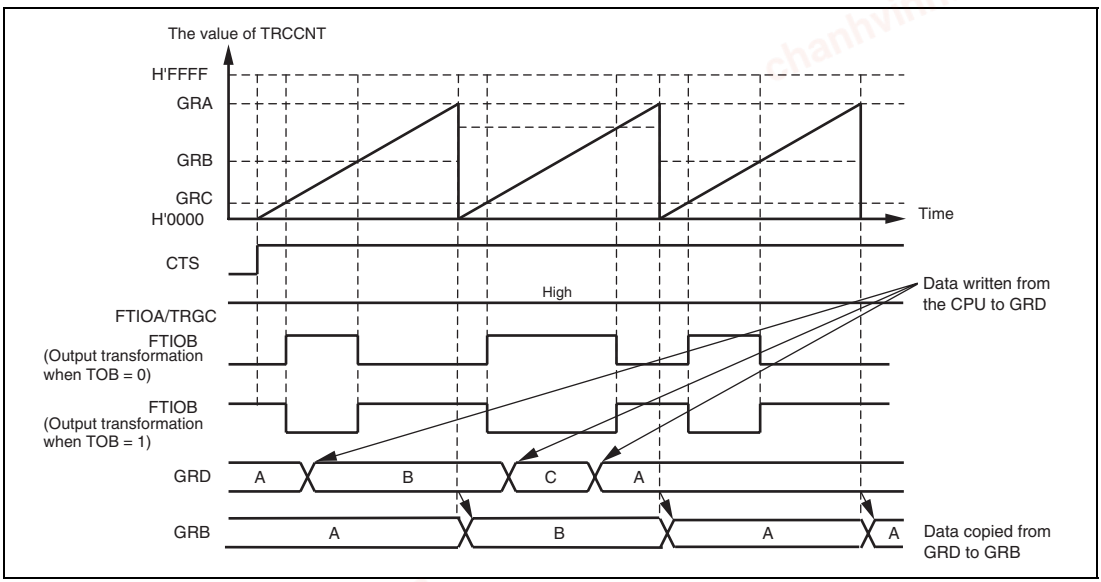



**Figure 15.18 Example (2) of TRGC Synchronous Operation in PWM2 Mode**

The following is an example of stopping operation of the counter in PWM2 mode. When the CSTP bit in TRCCR2 is set to 1 and the CCLR bit in TRCCR1 is set to 1, TRCCNT is cleared to H'0000 on a compare match with GRA and stops counting. Moreover, TRCCNT is forcibly stopped and cleared to the initial value when the CTS bit in TRCMR is cleared to 0. Figure 15.19 shows such an example when the TOB bit in TRCCR1 is cleared to 0 and set to 1.

chanhvinh.com



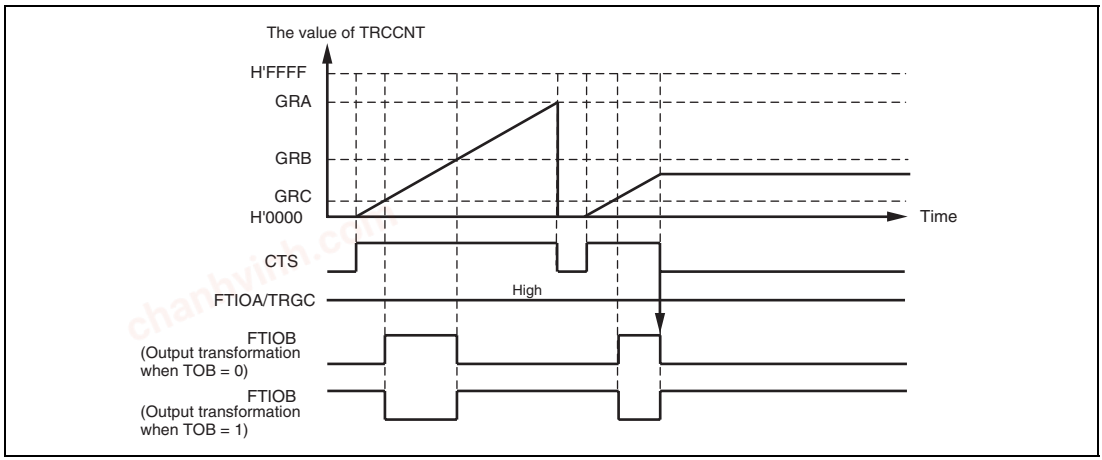



**Figure 15.19 Example of Stopping Operation of the Counter in PWM2 Mode**

The following is an example of output operation of the one-shot pulse waveform in PWM2 mode. When the TRGC input is disabled by TRCCR2 (clearing the TCEG1 and TCEG0 bits to 0), TRCCNT is set to stop counting-up on compare match A with GRA (setting the CSTP bit in TRCCR2 to 1), TRCCNT is cleared on compare match A (setting the CCRL bit in TRCCR1 to 1), and the initial value of the output signal is set to 0 by TRCCR1 (clearing the TOB bit to 0), TRCCNT starts counting when the CTS bit in TRCMR is set to 1. Then, TRCCNT is cleared to H'0000 on a compare match with GRA and stops counting, and the one-shot pulse waveform is output. Figure 15.20 shows such an example.

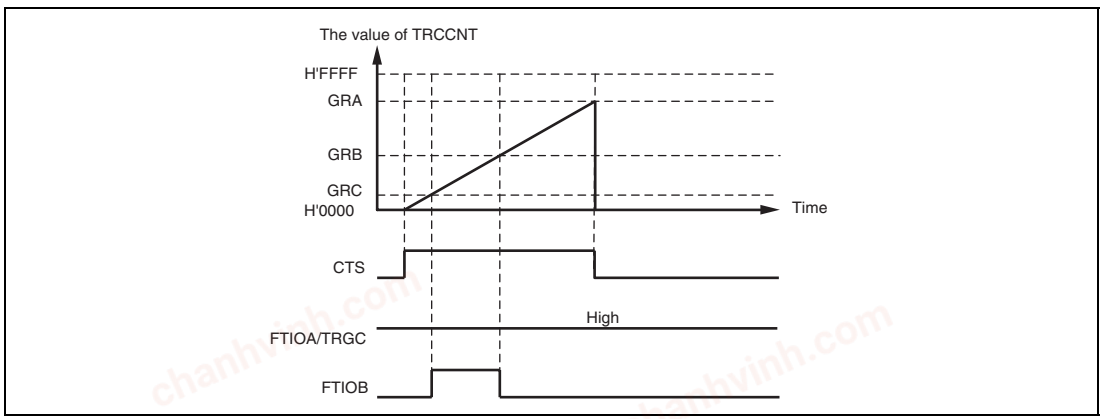



**Figure 15.20 Example (1) of Output Operation of One-Shot Pulse Waveform in PWM2 Mode**

chanhvinh.com



The following is an example of operation when TRCCNT starts counting by the TRGC input and the one-shot pulse waveform is output in PWM2 mode. When the falling edge of the TRGC input is selected by TRCCR2 (setting the TCEG1 bit to 1 and clearing the TCEG0 bit to 0), TRCCNT is set to counting-up on compare match A with GRA (setting the CSTP bit in TRCCR2 to 1), TRCCNT is cleared on compare match A (setting the CCRL bit in TRCCR1 to 1), and the initial value of the output signal is set to 0 by TRCCR1 (clearing the TOB bit to 0), TRCCNT starts counting at the falling edge of FTIOA/TRGC after the CTS bit in TRCMR has been set to 1. Then, TRCCNT is cleared to H'0000 on a compare match with GRA and stops counting, and the one-shot pulse waveform is output. Figure 15.21 shows such an example.

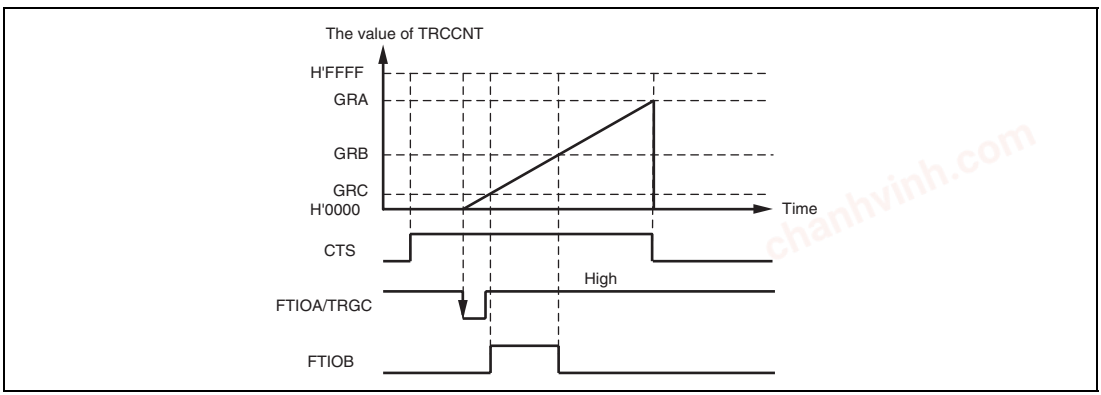



**Figure 15.21 Example (2) of Output Operation of One-Shot Pulse Waveform in PWM2 Mode**

chanhvinh.com



### 15.3.4 Digital Filtering Function for Input Capture Inputs

Input signals on the FTIOA to FTIOD and TRGC pin can be input via the digital filters. The digital filter includes three latches connected in series and a match detector circuit. The input signals on the FTIOA to FTIOD or TRGC pins are using on the sampling clock specified by the DFCK1 and DFCK0 bits in TRCDF. When outputs of the three latches match, the match detector circuit outputs the signal level of the input. Otherwise, the output remains unchanged. That is, when a pulse width is equal to or greater than three sampling clock cycles, the pulse is input as a signal. When a pulse width is less than three sampling clock cycles, the pulse is considered as noise to be removed.

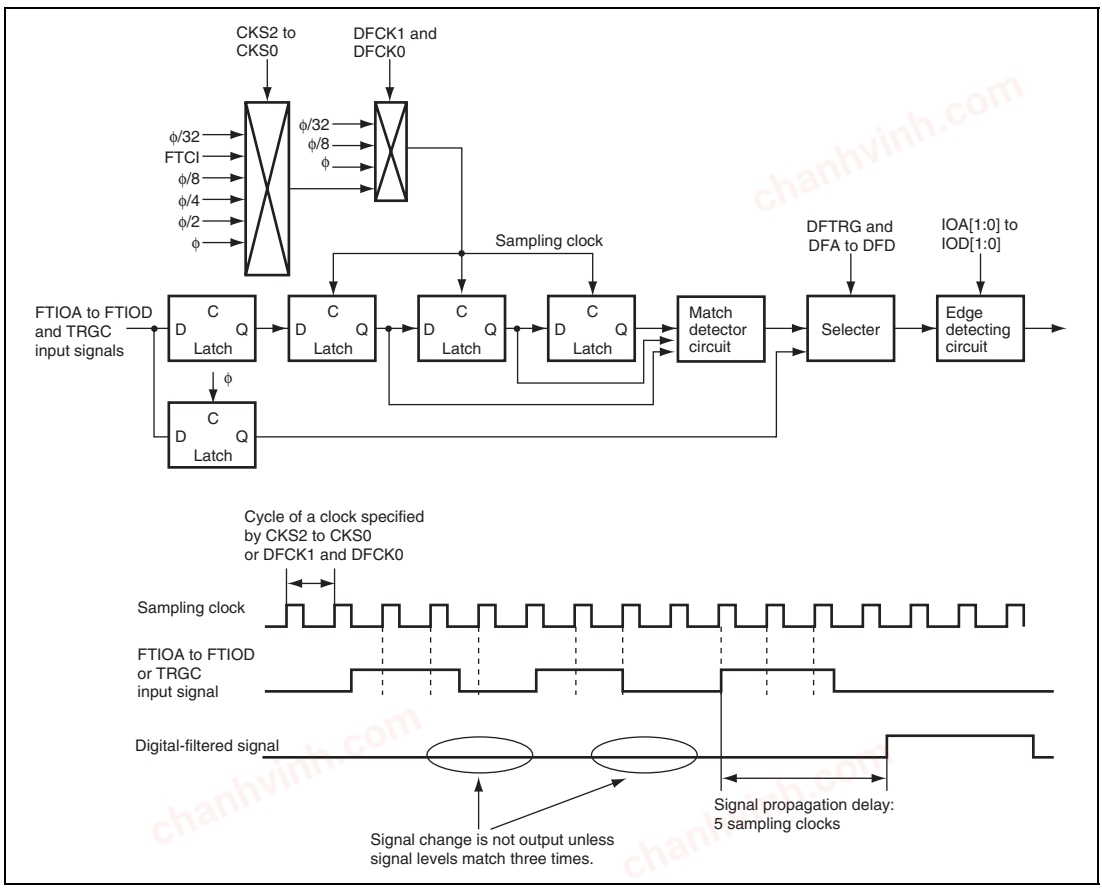



**Figure 15.22 Block Diagram of Digital Filter**

chanhvinh.com



### 15.3.5 A/D Conversion Start Trigger Setting Function

Timer RC can generate the A/D conversion start trigger signal on compare matches A, B, C, and D by setting the timer RC A/D conversion start trigger control register (TRCADCR). Figure 15.23 shows an example where the A/D conversion start trigger signal is set to be output on compare matches B and C.

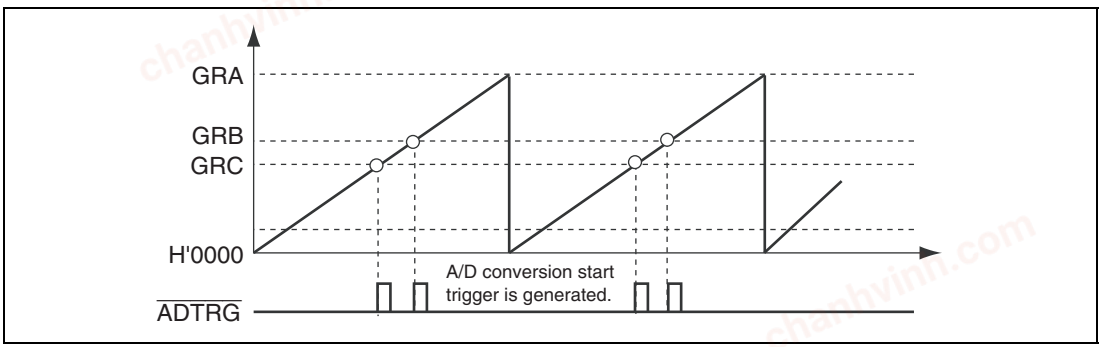



**Figure 15.23 Example of Compare Match**

In buffer operation, a buffer register cannot be used to generate the A/D conversion start trigger. Moreover, GRC cannot serve as a buffer register for GRA in PWM2 mode. Table 15.9 shows the A/D conversion start trigger source in each operating mode.

chanhvinh.com



**Table 15.9 A/D Conversion Start Trigger Generation in Each Operating Mode**

| | | A/D Conversion Start Trigger Generation | | | |
|---|---|---|---|---|---|
| **Operating Mode** | **Buffer Operation** | **GRA** | **GRB** | **GRC** | **GRD** |
| Input capture | Enabled | × | × | × | × |
| | Disabled | × | × | × | × |
| Compare match | Enabled | O | O | × | × |
| | Disabled | O | O | O | O |
| PWM mode | Enabled | O | O | × | × |
| | Disabled | O | O | O | O |
| PWM2 mode | Enabled | O | O | O | × |
| | Disabled | O | O | O | O |

[Legend]

O: The A/D conversion start trigger signal is generated.

×: The A/D conversion start trigger signal is not generated.

chanhvinh.com



### 15.3.6 Function of Changing Output Pins for GR

With the settings of bits IOC3 and IOD3 in TRCIOR1, pins for outputs of compare match signals for GRC and GRD can be changed from the FTIOC and FTIOD pins to the FTIOA and FTIOB pins. This means that the compare match A signal with the compare match C signal can be output on the FTIOA pin. The compare match B with the compare match D signal can be output on the FTIOB pin. Figure 15.24 is a block diagram of this function. Channel 0 and channel 1 can be set independently.

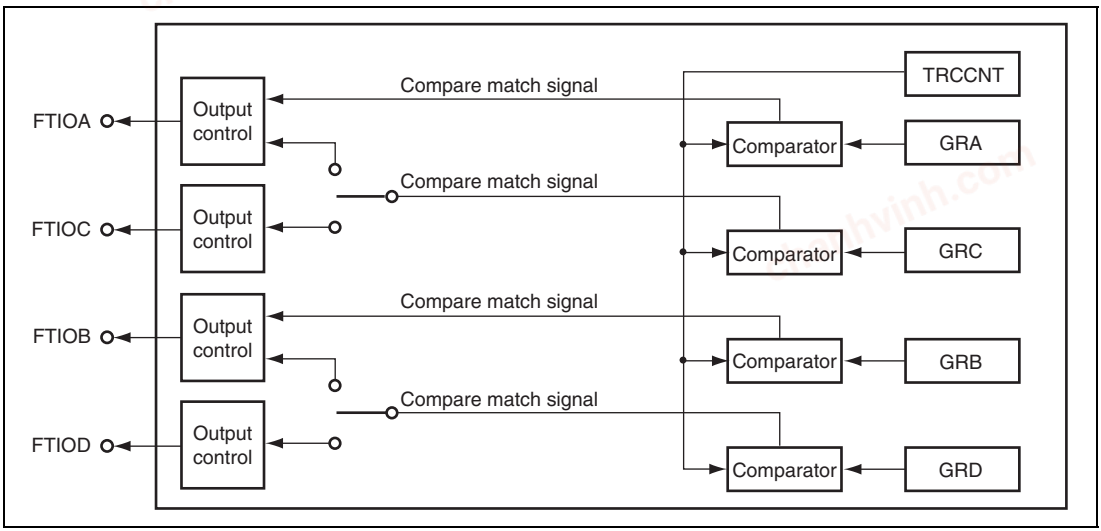



**Figure 15.24 Block Diagram of Output Pins for GR**

chanhvinh.com



Figure 15.25 is an example when non-overlapped pulses are output on pins FTIOA and FTIOB. In this example, TRCCNT functions as a periodic counter which is cleared on compare match A (bit CCLR in TRCCR1 is set to 1), an output signal is toggled on compare match A (bits IOA2 to IOA0 in TRCIOR0 are set to B'011), the output signal on the FTIOA pin is toggled on compare match C (GRC) (bits IOC3 to IOC0 in TRCIOR1are set to B'0X11), an output signal is toggled on compare match B (GRB) (bits IOB2 to IOB0 in TRCIOR0 are set to B'011), and the output signal on the FTIOB pin is toggled on compare match D (GRD) (bits IOD3 to IOD0 in TRCIOR1 are set to B'0X11). The cycle of the pulse is arbitrary.

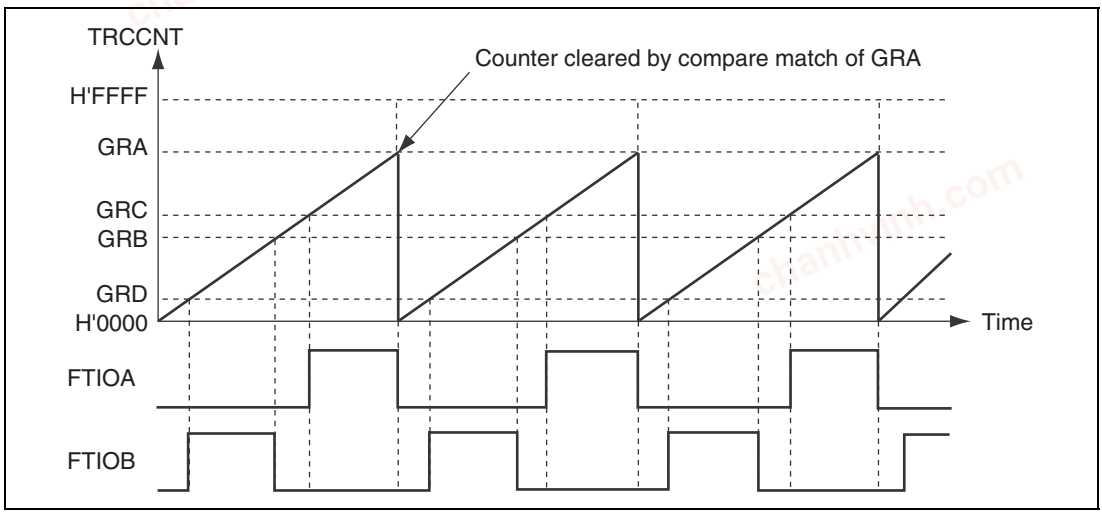



**Figure 15.25 Example of Non-Overlapped Pulses Output on Pins FTIOA and FTIOB (TRCCNT Used)**

chanhvinh.com



### 15.3.7 Operation through an Event Link

Using the event link controller (ELC), timer RC can be made to operate in the following ways in relation to events occurring in other modules.

#### (1) Staring Counter Operation

The start of counting operations by timer RC can be selected by ELOPA of the ELC. When the event specified by ELSR2 occur, the CTS bit in TRCMR is set to 1, which stars counting by timer RC. However, if the specified event occurs when the CTS bit has already been set to 1, the event is not effective.

#### (2) Counting Event

The counting of events by timer RC can be selected by ELOPA of the ELC. When the event specified in ELSR2 occurs, event counter operation proceeds with that event as the source to drive counting, regardless of the setting of the CKS[2:0] bits in TRCCR1 and the CTS bit in TRCMR. When the value of the counter is read, the value read out is the actual number of input events.

#### (3) Input Capture

Input capture operation of timer RC can be selected by ELOPA of the ELC. When the event specified in ELSR2 occurs, GRD captures the value of TRCCNT. When input capture operation initiated by an event link is in use, set the IOD[3:0] bits = b'1101 in TRCIOR1 of timer RC, set the CTS bit in TRCMR to 1, and then start the counter. Since input on the FTIOD pin becomes valid at the same time, fix the input to the FTIOD pin or take other measures such as not allocating the FTIOD pin to the port in the PMC, etc.

chanhvinh.com



## 15.4 Operation Timing

### 15.4.1 TRCCNT Counting Timing

Figure 15.26 shows the TRCCNT count timing when the internal clock source is selected. Figure 15.27 shows the timing when the external clock source is selected.

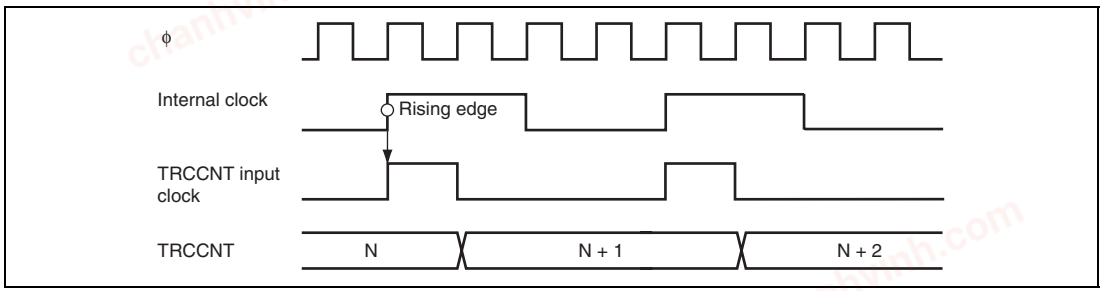



**Figure 15.26 Count Timing for Internal Clock Source**

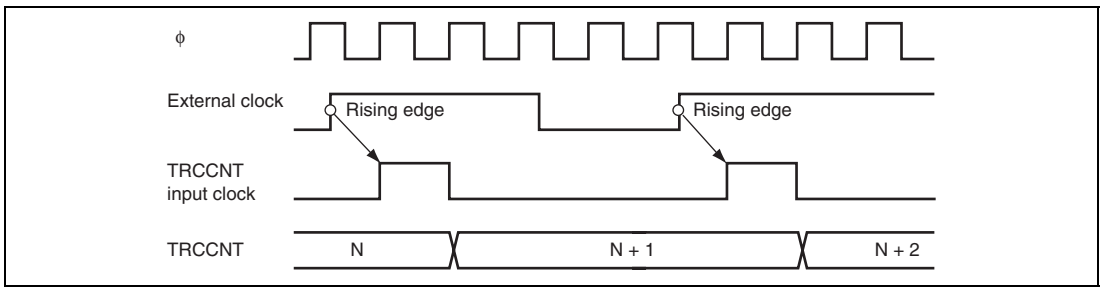



**Figure 15.27 Count Timing for External Clock Source**

chanhvinh.com



### 15.4.2 Output Compare Output Timing

The compare match signal is generated in the last state in which TRCCNT and GR match (when TRCCNT changes from the matching value to the next value). When the compare match signal is generated, the output value selected in TRCIOR is output on the compare match output pin (FTIOA, FTIOB, FTIOC, or FTIOD).

When TRCCNT matches GR, the compare match signal is generated only after the next counter clock pulse is input.

Figure 15.28 shows the output compare timing.

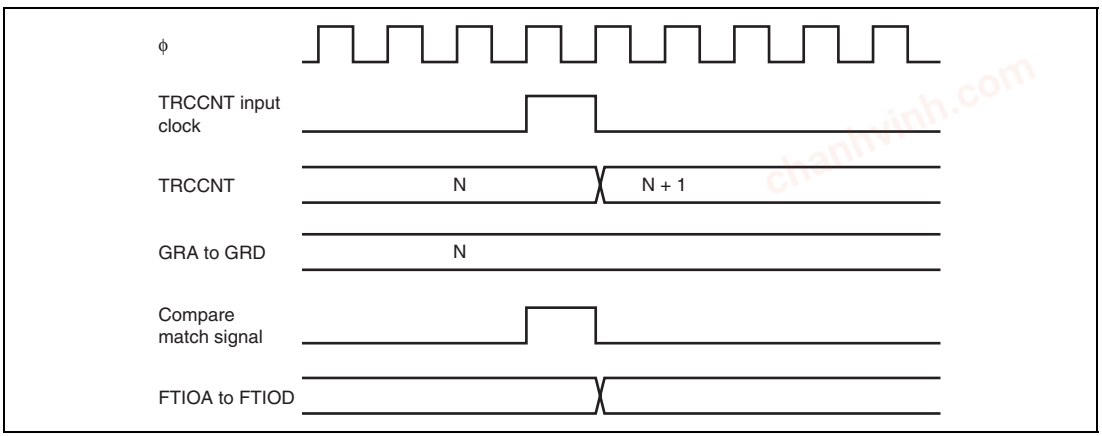



**Figure 15.28 Output Compare Output Timing**

chanhvinh.com



### 15.4.3 Input Capture Timing

Input capture on the rising edge, falling edge, or both edges can be selected through settings in TRCIOR0 and TRCIOR1. Figure 15.29 shows the timing when the falling edge is selected.

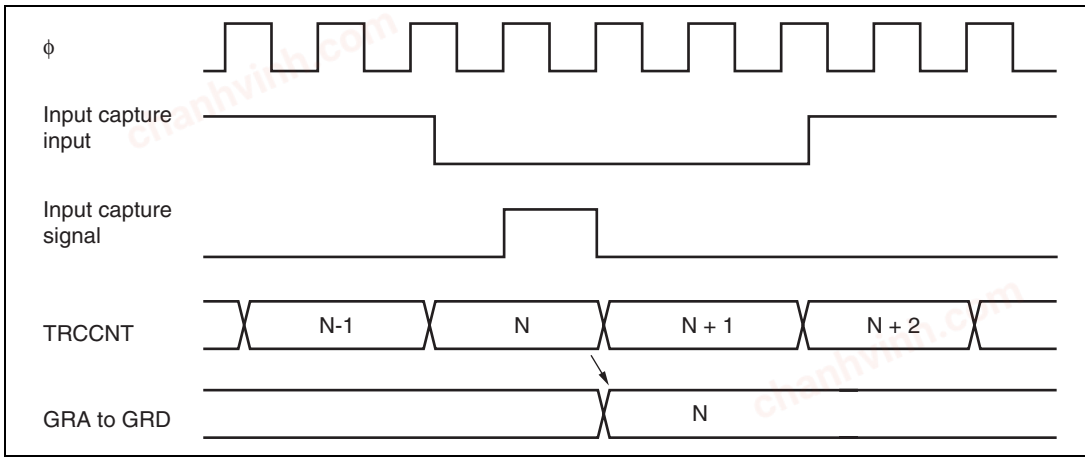



**Figure 15.29 Input Capture Input Signal Timing**

### 15.4.4 Timing of Counter Clearing by Compare Match

Figure 15.30 shows the timing when the counter is cleared by compare match A. When the GRA value is N, the counter counts from 0 to N, and its cycle is N + 1.

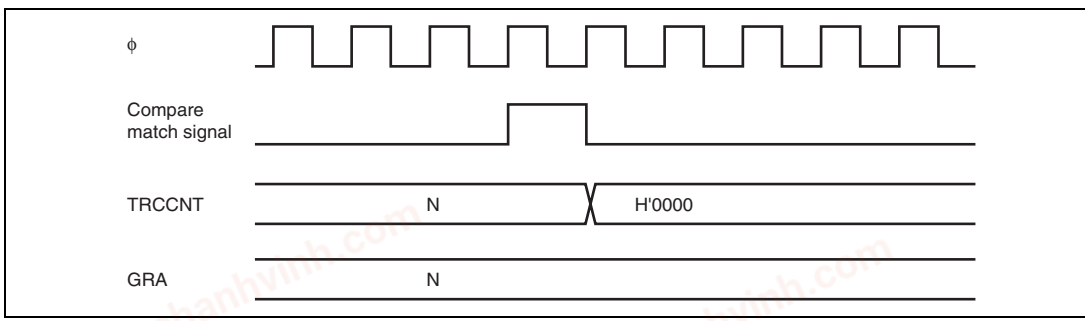



**Figure 15.30 Timing of Counter Clearing by Compare Match**

chanhvinh.com



### 15.4.5 Buffer Operation Timing

Figures 15.31 and 15.32 show the buffer operation timing.

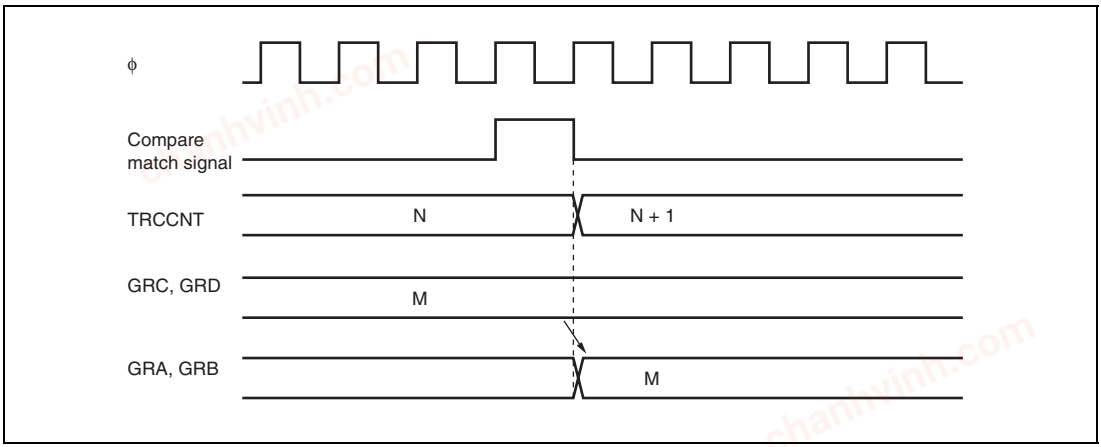



**Figure 15.31 Buffer Operation Timing (Compare Match)**

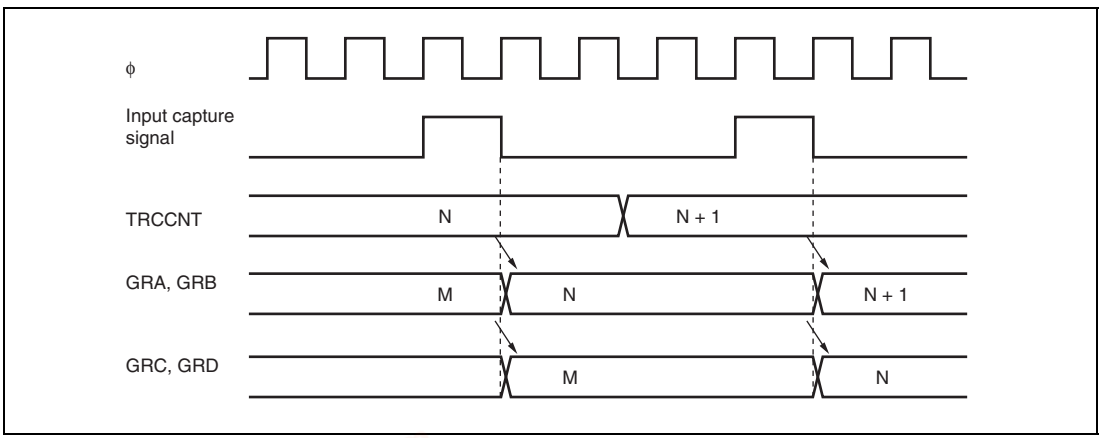



**Figure 15.32 Buffer Operation Timing (Input Capture)**

chanhvinh.com



### 15.4.6 Timing of IMFA to IMFD Flag Setting at Compare Match

If a general register (GRA, GRB, GRC, or GRD) matches TRCCNT, the corresponding IMFA to IMFD flag which is used as output compare register is set to 1.

The compare match signal is generated in the last state in which the values match (when TRCCNT is updated from the matching count to the next count). Therefore, when TRCCNT matches a general register (GRA, GRB, GRC, or GRD), the compare match signal is generated only after the next TRCCNT clock pulse is input.

Figure 15.33 shows the timing of the IMFA to IMFD flag setting at compare match.

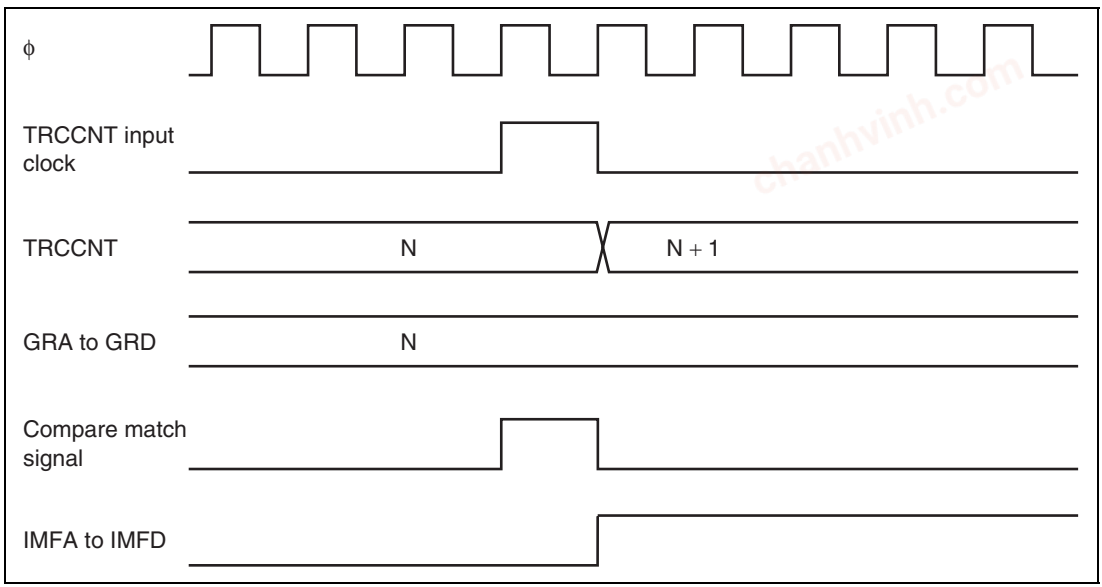



**Figure 15.33 Timing of IMFA to IMFD Flag Setting at Compare Match**

chanhvinh.com



### 15.4.7 Timing of IMFA to IMFD Setting at Input Capture

The corresponding IMFA, IMFB, IMFC, or IMFD flag which functions as a general register is set to 1 when an input capture occurs. Figure 15.34 shows the timing of the IMFA to IMFD flag setting at input capture.

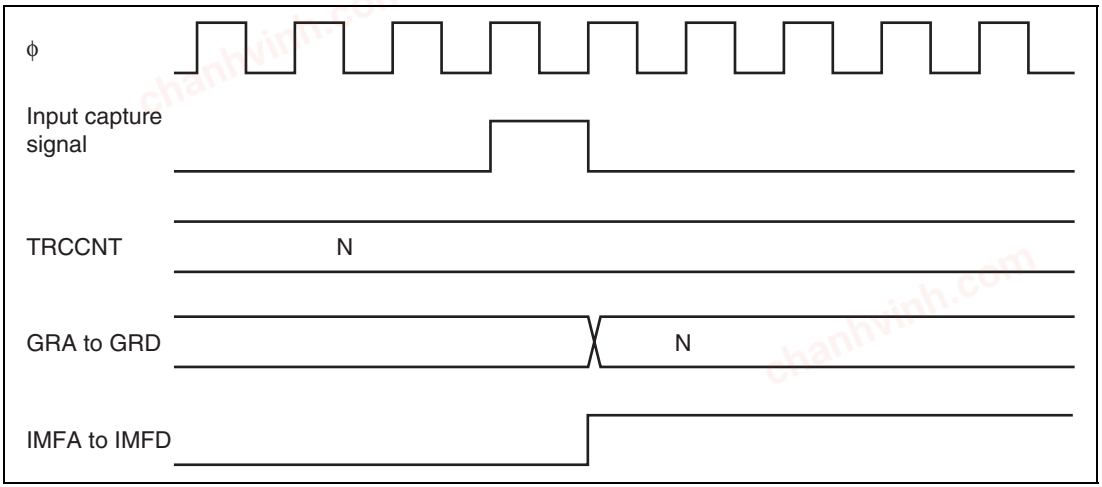



**Figure 15.34 Timing of IMFA to IMFD Flag Setting at Input Capture**

chanhvinh.com



### 15.4.8 Timing of Status Flag Clearing

When the CPU reads a status flag while it is set to 1, then writes 0 in the status flag, the status flag is cleared. Figure 15.35 shows the status flag clearing timing.

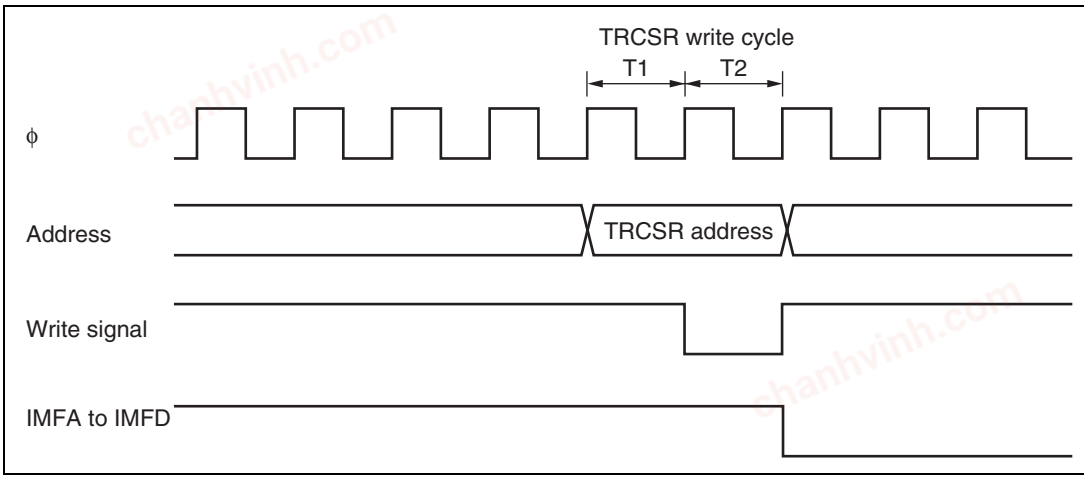



**Figure 15.35 Timing of Status Flag Clearing by CPU**

chanhvinh.com



### 15.4.9 Timing of A/D Conversion Start Trigger Generation on Compare Match

Figure 15.36 shows the timing of the A/D conversion start trigger generation on compare match.

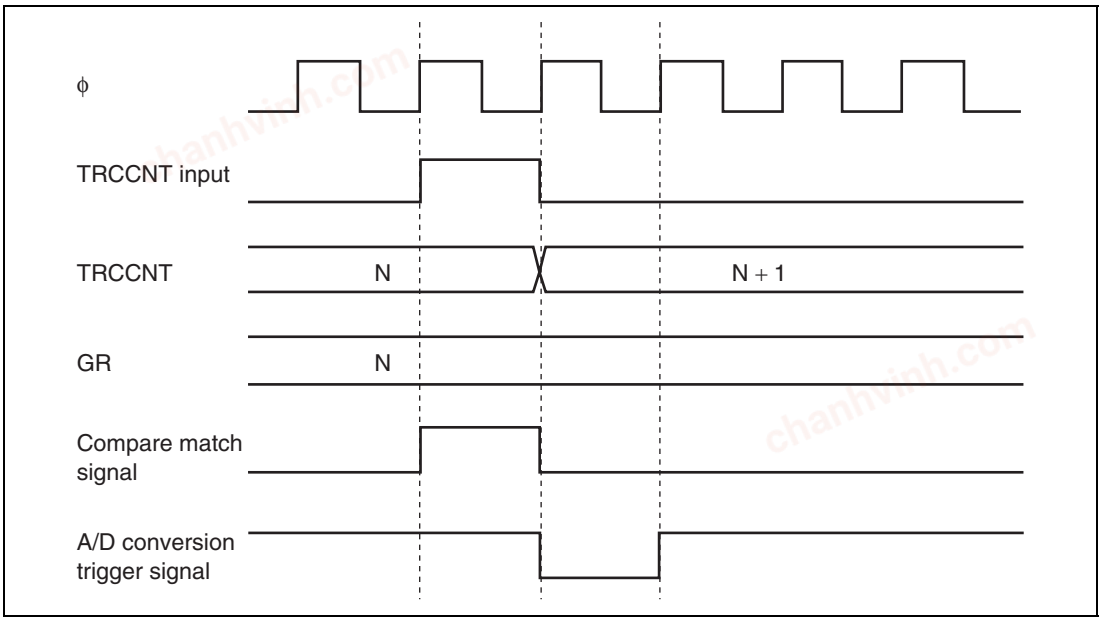



**Figure 15.36 Timing of A/D Conversion Start Trigger Generation on Compare Match**

chanhvinh.com



## 15.5 Usage Notes

The following types of contention or operation can occur in timer RC operation.

1. When the digital filtering function for input is not in use, the pulse width of the input clock signal and the input capture signal must be at least three system clock (ϕ) cycles when the CKS2 to CKS0 bits in TRCCR1 = B'0XX or B'10X; shorter pulses will not be detected correctly.
2. Writing to registers is performed in the T2 state of a TRCCNT write cycle.
   If counter clear signal occurs in the T2 state of a TRCCNT write cycle, clearing of the counter takes priority and the write is not performed, as shown in figure 15.37. If the TRCCNT write cycle contends with the TRCCNT counting-up, writing takes precedence.
3. TRCCNT may erroneously count up depends on the timing of switching internal clocks. The count clock is generated by detecting the rising edge of the divided system clock (ϕ) when the internal clock is selected. If clocks are switched as shown in figure 15.38, the change from the low level of the previous clock to the high level of the new clock is considered as the rising edge. In this case, TRCCNT counts up the clock erroneously.
4. If timer RC enters the module standby mode while an interrupt is being requested, the interrupt request cannot be cleared. Before entering the module standby mode, disable interrupt requests.

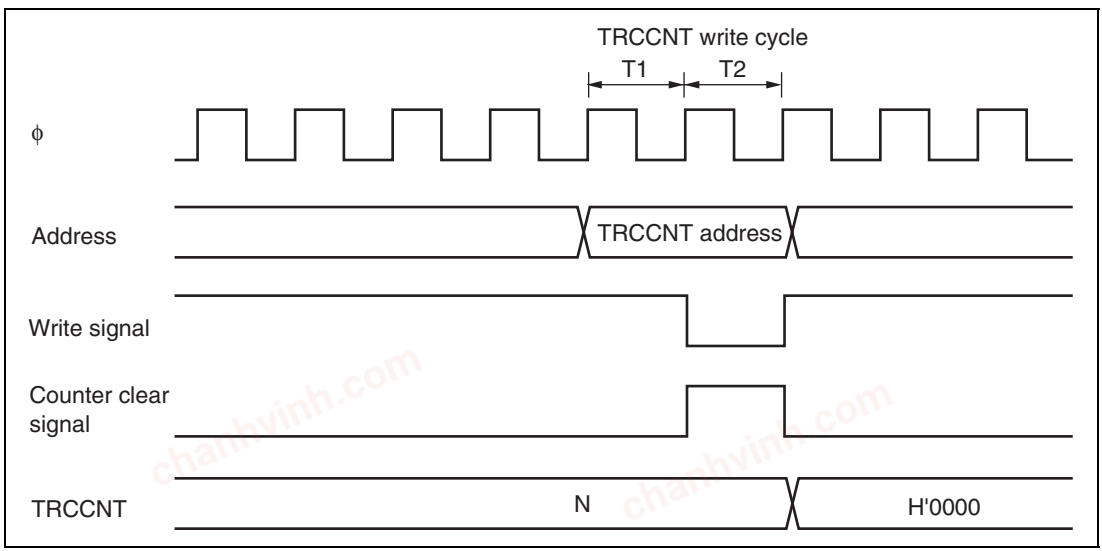



**Figure 15.37 Contention between TRCCNT Write and Clear**

chanhvinh.com



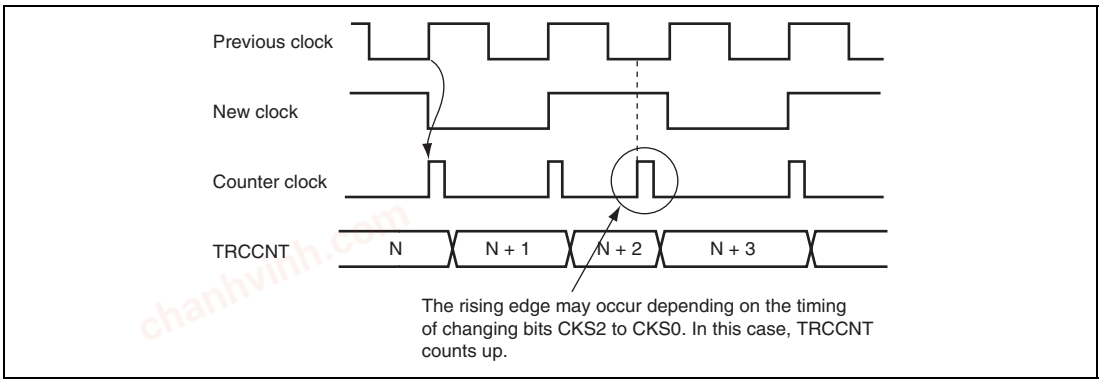



**Figure 15.38 Internal Clock Switching and TRCCNT Operation**

5. The TOA to TOD bits in TRCCR1 decide the output value of the FTIO pin until the first compare match occurs. Once a compare match occurs and this compare match changes the values of FTIOA to FTIOD output, the values of the FTIOA to FTIOD pin output and the values read from the TOA to TOD bits may differ. Moreover, when the writing to TRCCR1 and the generation of the compare match A to D occur at the same timing, the writing to TRCCR1 has the priority. Thus, output change due to the compare match is not reflected to the FTIOA to FTIOD pins. Therefore, when bit manipulation instruction is used to write to TRCCR1, the values of the FTIOA to FTIOD pin output may result in an unexpected result. When TRCCR1 is to be written to while compare match is operating, stop the counter once before accessing to TRCCR1, read the port H state to reflect the values of FTIOA to FTIOD output, to TOA to TOD, and then restart the counter. Figure 15.39 shows an example when the compare match and the bit manipulation instruction to TRCCR1 occur at the same timing.

chanhvinh.com



TRCCR1 has been set to H'06. Compare match B and compare match C are used. The FTIOB pin output 1, and is set to the toggle output or the 0 output on compare match B.
When the TOC bit is cleared (the FTIOC signal is low) by execution of BCLR #2,@TRCCR1 and compare match B occurs at the same timing as shown below, writing H'02 to TRCCR1 has priority and the FTIOB signal is not driven low on compare match B; the FTIOB signal remains high.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TRCCR1 | CCLR | CKS2 | CKS1 | CKS0 | TOD | TOC | TOB | TOA |
| Setting | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |

BCLR #2,@TRCCR1
(1) TRCCR1 is read as H'06.
(2) TRCCR1 is modified from H'06 to H'02.
(3) H'02 is written to TRCCR1.

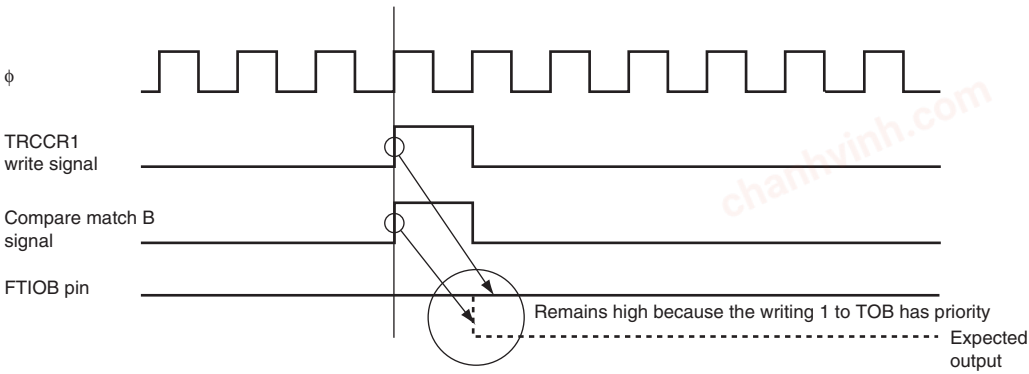



**Figure 15.39 When Compare Match and Bit Manipulation Instruction to TRCCR1 Occur at the Same Timing**

chanhvinh.com



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

chanhvinh.com



# Section 16 Timer RD

This LSI has two units of 16-bit timers (timer RD_0 and timer RD_1), each of which has two channels (one unit for the H8S/20103 and H8S/20115 Groups). Table 16.1 lists the timer RD functions, table 16.2 lists the channel configuration of timer RD, and figure 16.1 is a block diagram of the entire timer RD. Block diagrams of channels 0 and 1 are shown in figures 16.2 and 16.3.

Timer RD_0 has the same functions as timer RD_1. Therefore, the unit number (_0 or _1) is not explicitly mentioned in this section unless otherwise noted.

## 16.1 Features

- Capability to process up to eight inputs/outputs
- Eight general registers (GR): four registers for each channel
  Independently assignable output compare or input capture functions
- Selection of six counter clock sources: five internal clocks (ϕ, ϕ/2, ϕ/4, ϕ/8, and ϕ/32) and an external clock
- Seven selectable operating modes
  - — Timer mode
    Output compare function (Selection of 0 output, 1 output, or toggle output)
    Input capture function (Rising edge, falling edge, or both edges)
  - — Synchronous operation
    Timer counters_0 and _1 (TRDCNT_0 and TRDCNT_1) can be written simultaneously.
    Simultaneous clearing by compare match or input capture is possible.
  - — PWM mode
    Up to six-phase PWM output can be provided with desired duty ratio.
  - — PWM3 mode
    One-phase PWM output for non-overlapped normal and counter phases
  - — Reset synchronous PWM mode
    Three-phase PWM output for normal and counter phases
  - — Complementary PWM mode
    Three-phase PWM output for non-overlapped normal and counter phases
    The A/D conversion start trigger can be set for PWM cycles.
  - — Buffer operation
    The input capture register can be consisted of double buffers.
    The output compare register can automatically be modified.

chanhvinh.com



- High-speed access by the internal 16-bit bus
  16-bit TRDCNT and GR registers can be accessed in high speed by a 16-bit bus interface
- Any initial timer output value can be set
- Output of the timer is disabled by external trigger
- Eleven interrupt sources
  Four compare match/input capture interrupts and an overflow interrupt are available for each channel. An underflow interrupt can be set for channel 1.

chanhvinh.com



**Table 16.1 Timer RD Functions (One Unit)**

| Item | | Channel 0 | Channel 1 |
|---|---|---|---|
| Count clock | | Internal clocks: φ, φ/2, φ/4, φ/8, and φ/32<br>External clock: FTIOA0 (TCLK) | |
| General registers (output compare/input capture registers) | | GRA_0, GRB_0, GRC_0, GRD_0 | GRA_1, GRB_1, GRC_1, GRD_1 |
| Buffer register | | GRC_0, GRD_0 | GRC_1, GRD_1 |
| I/O pins | | FTIOA0, FTIOB0, FTIOC0, FTIOD0 | FTIOA1, FTIOB1, FTIOC1, FTIOD1 |
| Counter clearing function | | Compare match/input capture of GRA_0, GRB_0, GRC_0, or GRD_0 | Compare match/input capture of GRA_1, GRB_1, GRC_1, or GRD_1 |
| Compare match output | 0 output | Yes | Yes |
| | 1 output | Yes | Yes |
| | Toggle output | Yes | Yes |
| Input capture function | | Yes | Yes |
| Synchronous operation | | Yes | Yes |
| PWM mode | | Yes | Yes |
| PWM3 mode | | | Yes |
| Reset synchronous PWM mode | | | Yes |
| Complementary PWM mode | | | Yes |
| Buffer function | | Yes | Yes |
| Interrupt sources | | Compare match/ input capture A0 to D0<br>Overflow | Compare match/ input capture A1 to D1<br>Overflow<br>Underflow |

chanhvinh.com



**Table 16.2 Channel Configuration of Timer RD**

| Unit | Channel | Pin |
|---|---|---|
| Timer RD_0 | 0 | FTIOA0 |
| (Unit 0) | | FTIOB0 |
| | | FTIOC0 |
| | | FTIOD0 |
| | 1 | FTIOA1 |
| | | FTIOB1 |
| | | FTIOC1 |
| | | FTIOD1 |
| | Shared by channels 0 and 1 | $\overline{\text{TRDOI_0}}$ |
| Timer RD_1 | 2 | FTIOA2 |
| (Unit 1) | | FTIOB2 |
| | | FTIOC2 |
| | | FTIOD2 |
| | 3 | FTIOA3 |
| | | FTIOB3 |
| | | FTIOC3 |
| | | FTIOD3 |
| | Shared by channels 2 and 3 | $\overline{\text{TRDOI_1}}$ |

chanhvinh.com



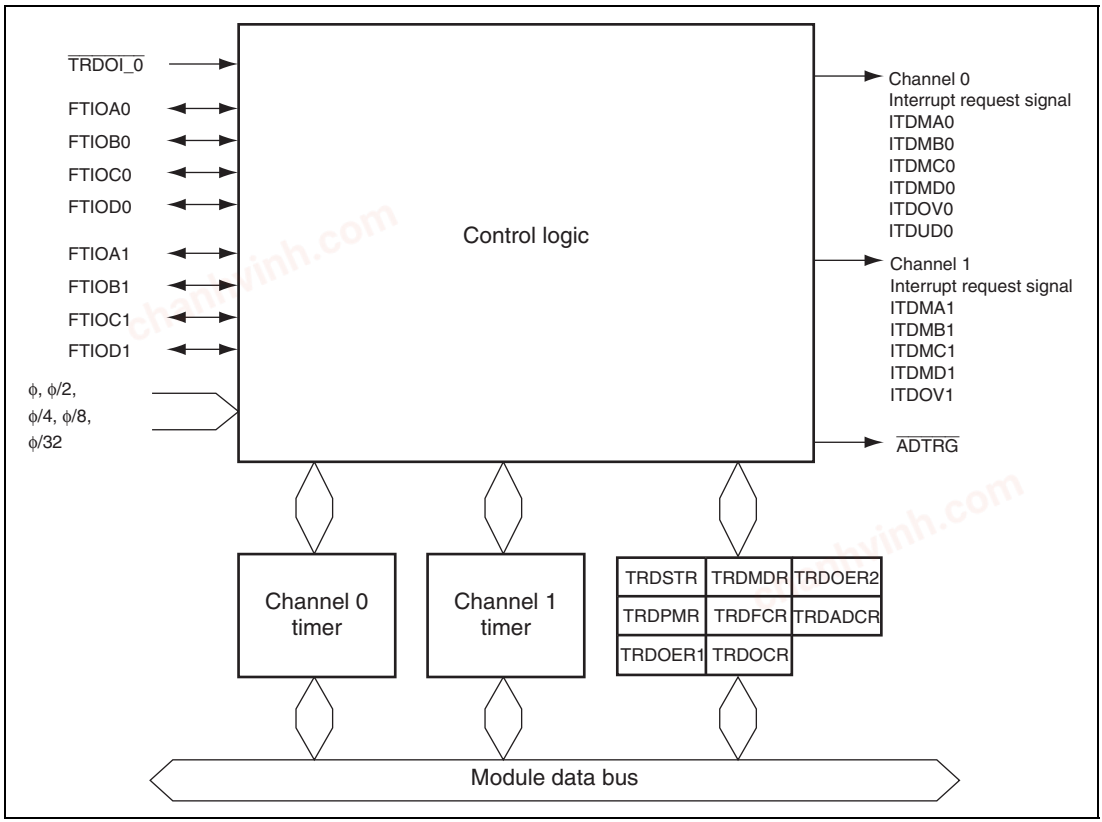



**Figure 16.1 Timer RD (One Unit) Block Diagram**

chanhvinh.com



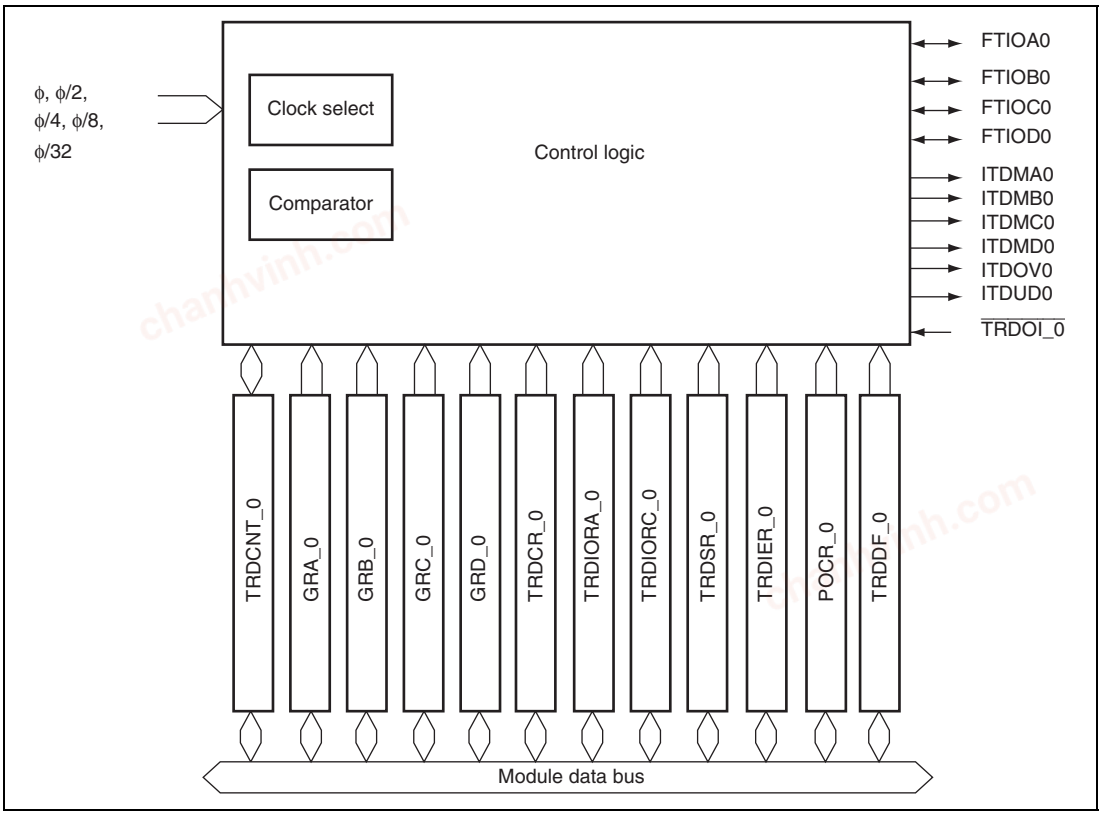



**Figure 16.2 Timer RD (Channel 0) Block Diagram**

chanhvinh.com



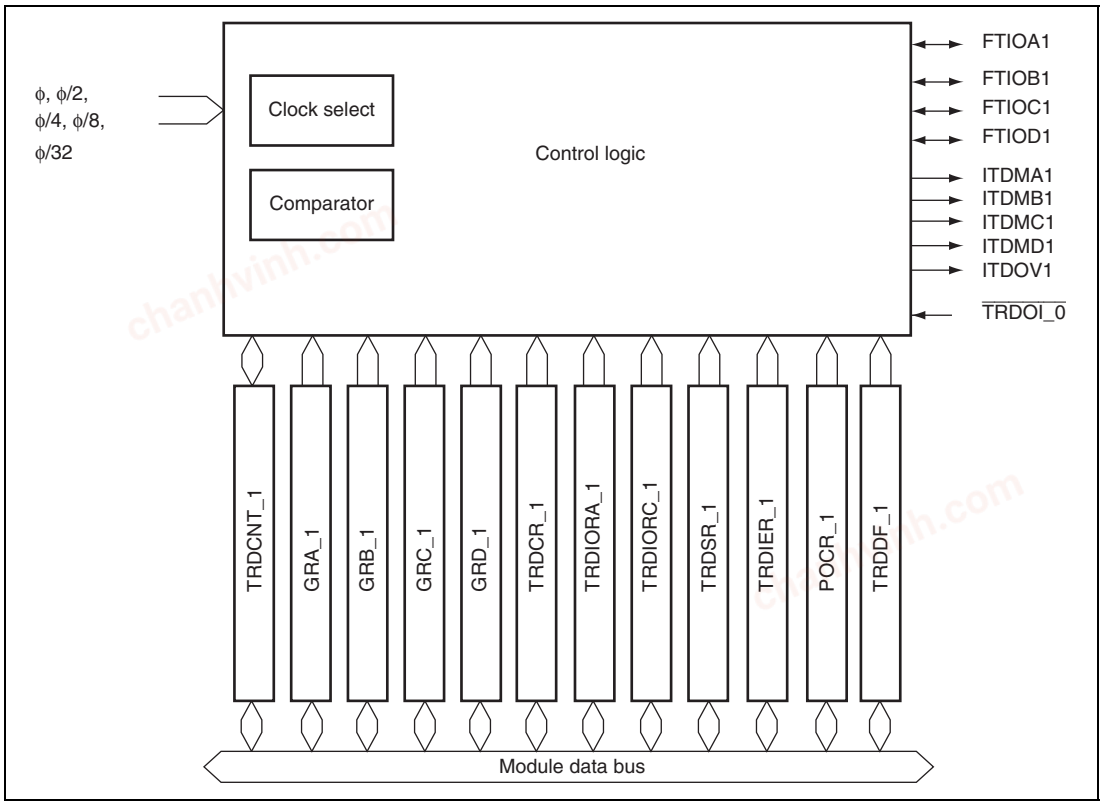



**Figure 16.3 Timer RD (Channel 1) Block Diagram**

chanhvinh.com



Table 16.3 summarizes the timer RD pins.

**Table 16.3 Pin Configuration (One Unit)**

| Pin Name | Input/Output | Function |
|---|---|---|
| FTIOA0 | I/O | GRA_0 output compare output, GRA_0 input capture input, or external clock input (TCLK) |
| FTIOB0 | I/O | GRB_0 output compare output, GRB_0 input capture input, or PWM output |
| FTIOC0 | I/O | GRC_0 output compare output, GRC_0 input capture input, or PWM synchronous output (in reset synchronous PWM and complementary PWM modes) |
| FTIOD0 | I/O | GRD_0 output compare output, GRD_0 input capture input, or PWM output |
| FTIOA1 | I/O | GRA_1 output compare output, GRA_1 input capture input, or PWM output (in reset synchronous PWM and complementary PWM modes) |
| FTIOB1 | I/O | GRB_1 output compare output, GRB_1 input capture input, or PWM output |
| FTIOC1 | I/O | GRC_1 output compare output, GRC_1 input capture input, or PWM output |
| FTIOD1 | I/O | GRD_1 output compare output, GRD_1 input capture input, or PWM output |
| $\overline{\text{TRDOI_0}}$ | Input | Input pin for timer output disabling signal |

chanhvinh.com



## 16.2 Register Descriptions

Timer RD has the following registers.

Common

- Timer RD start register (TRDSTR)
- Timer RD mode register (TRDMDR)
- Timer RD PWM mode register (TRDPMR)
- Timer RD function control register (TRDFCR)
- Timer RD output master enable register 1 (TRDOER1)
- Timer RD output master enable register 2 (TRDOER2)
- Timer RD output control register (TRDOCR)
- Timer RD A/D conversion start trigger control register (TRDADCR)

Channel 0

- Timer RD control register_0 (TRDCR_0)
- Timer RD I/O control register A_0 (TRDIORA_0)
- Timer RD I/O control register C_0 (TRDIORC_0)
- Timer RD status register_0 (TRDSR_0)
- Timer RD interrupt enable register_0 (TRDIER_0)
- PWM mode output level control register_0 (POCR_0)
- Timer RD digital filtering function select register_0 (TRDDF_0)
- Timer RD counter_0 (TRDCNT_0)
- General register A_0 (GRA_0)
- General register B_0 (GRB_0)
- General register C_0 (GRC_0)
- General register D_0 (GRD_0)

chanhvinh.com



Channel 1

- Timer RD control register_1 (TRDCR_1)
- Timer RD I/O control register A_1 (TRDIORA_1)
- Timer RD I/O control register C_1 (TRDIORC_1)
- Timer RD status register_1 (TRDSR_1)
- Timer RD interrupt enable register_1 (TRDIER_1)
- PWM mode output level control register_1 (POCR_1)
- Timer RD digital filtering function select register_1 (TRDDF_1)
- Timer RD counter_1 (TRDCNT_1)
- General register A_1 (GRA_1)
- General register B_1 (GRB_1)
- General register C_1 (GRC_1)
- General register D_1 (GRD_1)

chanhvinh.com chanhvinh.com chanhvinh.com

chanhvinh.com



### 16.2.1 Timer RD Start Register (TRDSTR)

Address: H'FFFFD2

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | CSTPN1 | CSTPN0 | STR1 | STR0 |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 to 4 | — | Reserved | These bits are read as 1. The write value should be 1. | — |
| 3 | CSTPN1 | Channel 1 counter stop | 0: Counting is stopped on a compare match of TRDCNT_1 and GRA_1<br>1: Counting is continued on a compare match of TRDCNT_1 and GRA_1<br>Set this bit to 1 to restart counting after the counting has been stopped on a compare match. | R/W |
| 2 | CSTPN0 | Channel 0 counter stop | 0: Counting is stopped on a compare match of TRDCNT_0 and GRA_0<br>1: Counting is continued on a compare match of TRDCNT_0 and GRA_0<br>Set this bit to 1 to restart counting after the counting has been stopped on a compare match. | R/W |
| 1 | STR1 | Channel 1 counter start | 0: TRDCNT_1 stops counting.<br>1: TRDCNT_1 starts counting.<br>[Setting conditions]<br>• When 1 is written in STR1<br>• When the specified event is occurred after ELOPB of the event link controller is selected counting by timer RD_0 for channel 1.<br>[Clearing conditions]<br>• When 0 is written in STR1 while CSTPN1 = 1<br>• When the compare match A1 signal is generated while CSTPN1 = 0 | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 0 | STR0 | Channel 0 counter start | 0: TRDCNT_0 stops counting.<br>1: TRDCNT_0 starts counting.<br>[Setting conditions]<br>• When 1 is written in STR0<br>• When the specified event is occurred after ELOPA of the event link controller is selected counting by timer RD_0 for channel 0.<br>[Clearing conditions]<br>• When 0 is written in STR0 while CSTPN0 = 1<br>• When the compare match A1 signal is generated while CSTPN0 = 0 | R/W |

Note: Use a MOV instruction to modify this register.

chanhvinh.com



### 16.2.2 Timer RD Mode Register (TRDMDR)

Address: H'FFFFD3

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | BFD1 | BFC1 | BFD0 | BFC0 | — | — | — | SYNC |
| Value after reset: | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | BFD1 | Buffer operation D1 | 0: GRD_1 operates normally<br>1: GRB_1 and GRD_1 are used together for buffer operation | R/W |
| 6 | BFC1 | Buffer operation C1 | 0: GRC_1 operates normally<br>1: GRA_1 and GRC_1 are used together for buffer operation | R/W |
| 5 | BFD0 | Buffer operation D0 | 0: GRD_0 operates normally<br>1: GRB_0 and GRD_0 are used together for buffer operation | R/W |
| 4 | BFC0 | Buffer operation C0 | 0: GRC_0 operates normally<br>1: GRA_0 and GRC_0 are used together for buffer operation | R/W |
| 3 to 1 | — | Reserved | These bits are read as 1. The write value should be 1. | — |
| 0 | SYNC | Timer synchronization | 0: TRDCNT_1 and TRDCNT_0 operate as independent timer counters<br>1: TRDCNT_1 and TRDCNT_0 operate synchronously<br>TRDCNT_1 and TRDCNT_0 can be pre-set or cleared synchronously. | R/W |

chanhvinh.com



### 16.2.3 Timer RD PWM Mode Register (TRDPMR)

Address: H'FFFFD4

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | PWMD1 | PWMC1 | PWMB1 | — | PWMD0 | PWMC0 | PWMB0 |
| Value after reset: | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is read as 1. The write value should be 1. | — |
| 6 | PWMD1 | PWM mode D1 | 0: FTIOD1 operates normally<br>1: FTIOD1 operates in PWM mode | R/W |
| 5 | PWMC1 | PWM mode C1 | 0: FTIOC1 operates normally<br>1: FTIOC1 operates in PWM mode | R/W |
| 4 | PWMB1 | PWM mode B1 | 0: FTIOB1 operates normally<br>1: FTIOB1 operates in PWM mode | R/W |
| 3 | — | Reserved | This bit is read as 1. The write value should be 1. | — |
| 2 | PWMD0 | PWM mode D0 | 0: FTIOD0 operates normally<br>1: FTIOD0 operates in PWM mode | R/W |
| 1 | PWMC0 | PWM mode C0 | 0: FTIOC0 operates normally<br>1: FTIOC0 operates in PWM mode | R/W |
| 0 | PWMB0 | PWM mode B0 | 0: FTIOB0 operates normally<br>1: FTIOB0 operates in PWM mode | R/W |

chanhvinh.com



### 16.2.4 Timer RD Function Control Register (TRDFCR)

Address: H'FFFFD5

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PWM3 | STCLK | ADEG | ADTRG | OLS1 | OLS0 | CMD[1:0] | |
| Value after reset: | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PWM3 | PWM3 mode select | 0: PWM3 mode is selected<br>1: PWM3 mode is not selected*[1] | R/W |
| 6 | STCLK | External clock input select | 0: External clock input is disabled<br>1: External clock input is enabled | R/W |
| 5 | ADEG | A/D trigger edge select | 0: The A/D trigger signal is asserted when TRDCNT_0 matches GRA_0 in complementary PWM mode<br>1: The A/D trigger signal is asserted when TRDCNT_1 underflows in complementary PWM mode | R/W |
| 4 | ADTRG | External trigger disable | 0: A/D trigger for PWM cycles is disabled in complementary PWM mode<br>1: A/D trigger for PWM cycles is enabled in complementary PWM mode*[2] | R/W |
| 3 | OLS1 | Output level select 1 | 0: Initial output is high and the active level is low.<br>1: Initial output is low and the active level is high. | R/W |
| 2 | OLS0 | Output level select 0 | 0: Initial output is high and the active level is low.<br>1: Initial output is low and the active level is high. | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 1, 0 | CMD[1:0] | Combination mode 1 and 0 | 00: Channel 0 and channel 1 operate normally<br>01: Channel 0 and channel 1 are used together to operate in reset synchronous PWM mode<br>10: Channel 0 and channel 1 are used together to operate in complementary PWM mode (transferred when TRDCNT_0 matches GRA_0)<br>11: Channel 0 and channel 1 are used together to operate in complementary PWM mode (transferred when TRDCNT_1 underflows)<br>Note: When the reset synchronous PWM mode or complementary PWM mode is selected by these bits, this setting has the priority to the settings for PWM mode by each bit in TRDPMR. Stop TRDCNT_0 and TRDCNT_1 before making settings for reset synchronous PWM mode or complementary PWM mode. | R/W |

Notes: 1. This bit is valid when both bits CMD1 and CMD0 are cleared to 0. When PWM3 mode is selected, TRDPMR, TRDIORA, and TRDIORC are invalid.
2. The A/D converter registers should be set so that A/D conversion is started by an external trigger.

- OLS1 bit (output level select 1)

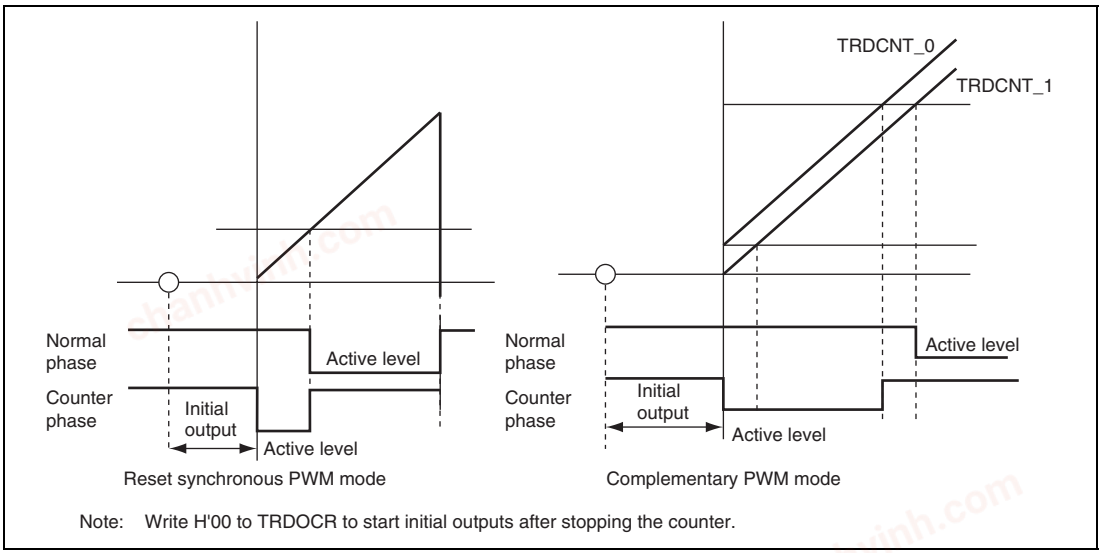
  This bit selects the output level for counter phase in reset synchronous PWM mode and complementary PWM mode.

- OLS0 bit (output level select 0)

  This bit selects the output level for normal phase in reset synchronous PWM mode and complementary PWM mode.

chanhvinh.com






Note: Write H'00 to TRDOCR to start initial outputs after stopping the counter.

**Figure 16.4 Example of Outputs in Reset Synchronous PWM Mode and Complementary PWM Mode**

### 16.2.5 Timer RD Output Master Enable Register 1 (TRDOER1)

Address: H'FFFFD6

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | ED1 | EC1 | EB1 | EA1 | ED0 | EC0 | EB0 | EA0 |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | ED1 | Master enable D1 | 0: FTIOD1 pin output is enabled according to the TRDPMR, TRDFCR, and TRDIORC_1 settings<br>1: FTIOD1 pin output is disabled regardless of the TRDPMR, TRDFCR, and TRDIORC_1 settings (FTIOD1 pin is operated as an I/O port). | R/W |
| 6 | EC1 | Master enable C1 | 0: FTIOC1 pin output is enabled according to the TRDPMR, TRDFCR, and TRDIORC_1 settings<br>1: FTIOC1 pin output is disabled regardless of the TRDPMR, TRDFCR, and TRDIORC_1 settings (FTIOC1 pin is operated as an I/O port). | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 5 | EB1 | Master enable B1 | 0: FTIOB1 pin output is enabled according to the TRDPMR, TRDFCR, and TRDIORA_1 settings<br>1: FTIOB1 pin output is disabled regardless of the TRDPMR, TRDFCR, and TRDIORA_1 settings (FTIOB1 pin is operated as an I/O port). | R/W |
| 4 | EA1 | Master enable A1 | 0: FTIOA1 pin output is enabled according to the TRDPMR, TRDFCR, and TRDIORA_1 settings<br>1: FTIOA1 pin output is disabled regardless of the TRDPMR, TRDFCR, and TRDIORA_1 settings (FTIOA1 pin is operated as an I/O port). | R/W |
| 3 | ED0 | Master enable D0 | 0: FTIOD0 pin output is enabled according to the TRDPMR, TRDFCR, and TRDIORC_0 settings<br>1: FTIOD0 pin output is disabled regardless of the TRDPMR, TRDFCR, and TRDIORC_0 settings (FTIOD0 pin is operated as an I/O port). | R/W |
| 2 | EC0 | Master enable C0 | 0: FTIOC0 pin output is enabled according to the TRDPMR, TRDFCR, and TRDIORC_0 settings<br>1: FTIOC0 pin output is disabled regardless of the TRDPMR, TRDFCR, and TRDIORC_0 settings (FTIOC0 pin is operated as an I/O port). | R/W |
| 1 | EB0 | Master enable B0 | 0: FTIOB0 pin output is enabled according to the TRDPMR, TRDFCR, and TRDIORA_0 settings<br>1: FTIOB0 pin output is disabled regardless of the TRDPMR, TRDFCR, and TRDIORA_0 settings (FTIOB0 pin is operated as an I/O port). | R/W |
| 0 | EA0 | Master enable A0 | 0: FTIOA0 pin output is enabled according to the TRDPMR, TRDFCR, and TRDIORA_0 settings<br>1: FTIOA0 pin output is disabled regardless of the TRDPMR, TRDFCR, and TRDIORA_0 settings (FTIOA0 pin is operated as an I/O port). | R/W |

TRDOER1 enables/disables the outputs for channel 0 and channel 1. When $\overline{\text{TRDOI}}$ is selected for inputs, if a low level signal is input to $\overline{\text{TRDOI}}$, the bits in TRDOER1 are set to 1 to disable the output for timer RD.

chanhvinh.com



### 16.2.6 Timer RD Output Master Enable Register 2 (TRDOER2)

Address: H'FFFFD7

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PTO | — | — | — | — | — | — | — |
| Value after reset: | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | PTO | Timer output disabled mode | 0: The corresponding bit in TRDOER1 is not set to 1 when the low level is input to the $\overline{\text{TRDOI}}$ pin<br>1: The corresponding bit in TRDOER1 is set to 1 when the low level is input to the $\overline{\text{TRDOI}}$ pin | R/W |
| 6 to 0 | — | Reserved | These bits are read as 1. The write value should be 1. | — |

### 16.2.7 Timer RD Output Control Register (TRDOCR)

Address: H'FFFFD8

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TOD1 | TOC1 | TOB1 | TOA1 | TOD0 | TOC0 | TOB0 | TOA0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | TOD1 | Output level select D1 | 0: 0 output at the FTIOD1 pin*<br>1: 1 output at the FTIOD1 pin* | R/W |
| 6 | TOC1 | Output level select C1 | 0: 0 output at the FTIOC1 pin*<br>1: 1 output at the FTIOC1 pin* | R/W |
| 5 | TOB1 | Output Level Select B1 | 0: 0 output at the FTIOB1 pin*<br>1: 1 output at the FTIOB1 pin* | R/W |
| 4 | TOA1 | Output level select A1 | 0: 0 output at the FTIOA1 pin*<br>1: 1 output at the FTIOA1 pin* | R/W |
| 3 | TOD0 | Output level select D0 | 0: 0 output at the FTIOD0 pin*<br>1: 1 output at the FTIOD0 pin* | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 2 | TOC0 | Output level select C0 | 0: 0 output at the FTIOC0 pin*<br>1: 1 output at the FTIOC0 pin* | R/W |
| 1 | TOB0 | Output level select B0 | • In modes other than PWM3 mode<br>0: 0 output at the FTIOB0 pin*<br>1: 1 output at the FTIOB0 pin*<br>• In PWM3 mode<br>0: 1 output at the FTIOB0 pin on GRB_1 compare match and 0 output at the FTIOB0 pin on GRB_0 compare match<br>1: 0 output at the FTIOB0 pin on GRB_1 compare match and 1 output at the FTIOB0 pin on GRB_0 compare match | R/W |
| 0 | TOA0 | Output level select A0 | • In modes other than PWM3 mode<br>0: 0 output at the FTIOA0 pin*<br>1: 1 output at the FTIOA0 pin*<br>• In PWM3 mode<br>0: 1 output at the FTIOB0 pin on GRA_1 compare match and 0 output at the FTIOB0 pin on GRA_0 compare match<br>1: 0 output at the FTIOB0 pin on GRA_1 compare match and 1 output at the FTIOB0 pin on GRA_0 compare match | R/W |

Note: * The change of the setting is immediately reflected in the output value.

TRDOCR selects the initial outputs before the first occurrence of a compare match. Note that bits OLS1 and OLS0 in TRDFCR set these initial outputs in reset synchronous PWM mode and complementary PWM mode.

In PWM3 mode, TRDOCR selects the output level of the FTIOA0 and FTIOB0 pins.

chanhvinh.com



### 16.2.8 Timer RD A/D Conversion Start Trigger Control Register (TRDADCR)

Address: H'FFFFD9

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | ADTRGD1E | ADTRGC1E | ADTRGB1E | ADTRGA1E | ADTRGD0E | ADTRGC0E | ADTRGB0E | ADTRGA0E |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | ADTRGD1E | A/D conversion start trigger D1 enable | 0: A/D conversion start trigger is not generated by compare match of GRD_1<br>1: A/D conversion start trigger is generated by compare match of GRD_1 | R/W |
| 6 | ADTRGC1E | A/D conversion start trigger C1 enable | 0: A/D conversion start trigger is not generated by compare match of GRC_1<br>1: A/D conversion start trigger is generated by compare match of GRC_1 | R/W |
| 5 | ADTRGB1E | A/D conversion start trigger B1 enable | 0: A/D conversion start trigger is not generated by compare match of GRB_1<br>1: A/D conversion start trigger is generated by compare match of GRB_1 | R/W |
| 4 | ADTRGA1E | A/D conversion start trigger A1 enable | 0: A/D conversion start trigger is not generated by compare match of GRA_1<br>1: A/D conversion start trigger is generated by compare match of GRA_1 | R/W |
| 3 | ADTRGD0E | A/D conversion start trigger D0 enable | 0: A/D conversion start trigger is not generated by compare match of GRD_0<br>1: A/D conversion start trigger is generated by compare match of GRD_0 | R/W |
| 2 | ADTRGC0E | A/D conversion start trigger C0 enable | 0: A/D conversion start trigger is not generated by compare match of GRC_0<br>1: A/D conversion start trigger is generated by compare match of GRC_0 | R/W |
| 1 | ADTRGB0E | A/D conversion start trigger B0 enable | 0: A/D conversion start trigger is not generated by compare match of GRB_0<br>1: A/D conversion start trigger is generated by compare match of GRB_0 | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 0 | ADTRGA0E | A/D conversion start trigger A0 enable | 0: A/D conversion start trigger is not generated by compare match of GRA_0<br>1: A/D conversion start trigger is generated by compare match of GRA_0 | R/W |

TRDADCR selects the trigger source to start A/D conversion. A/D conversion start trigger is generated by a corresponding compare match.

### 16.2.9 Timer RD Counter (TRDCNT)

Address: H'FFFFB0, H'FFFFBA

| Bit: | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Timer RD has two TRDCNT counters (TRDCNT_0 and TRDCNT_1), one for each channel. The TRDCNT counters are 16-bit readable/writable registers that increment/decrement according to input clocks. Input clocks can be selected by bits TPSC2 to TPSC0 in TRDCR. TRDCNT_0 and TRDCNT_1 increment/decrement in complementary PWM mode while they only increment in other modes.

The TRDCNT counters are initialized to H'0000 by compare matches with corresponding GRA, GRB, GRC, or GRD, or input captures to GRA, GRB, GRC, or GRD (counter clearing function). When the TRDCNT counters overflow, an OVF flag in TRDSR for the corresponding channel is set to 1. When TRDCNT_1 underflows, an UDF flag in TRDSR is set to 1. The TRDCNT counters cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit.

chanhvinh.com



### 16.2.10 General Registers A, B, C, and D (GRA, GRB, GRC, and GRD)

GRA

Address: H'FFFFB2, H'FFFFBC

| Bit: | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

GRB

Address: H'FFFFB4, H'FFFFBE

| Bit: | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

GRC

Address: H'FFFFB6, H'FFFFC0

| Bit: | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

GRD

Address: H'FFFFB8, H'FFFFC2

| Bit: | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

chanhvinh.com



GR are 16-bit registers. Timer RD has eight general registers (GR), four for each channel. The GR registers are dual function 16-bit readable/writable registers, functioning as either output compare or input capture registers. Functions can be switched by TRDIORA and TRDIORC.

The values in GR and TRDCNT are constantly compared with each other when the GR registers are used as output compare registers. When the both values match, the IMFA to IMFD flags in TRDSR are set to 1. Compare match outputs can be selected by TRDIORA and TRDIORC.

When the GR registers are used as input capture registers, the TRDCNT value is stored after detecting external signals. At this point, IMFA to IMFD flags in the corresponding TRDSR are set to 1. Detection edges for input capture signals can be selected by TRDIORA and TRDIORC.

When PWM mode, complementary PWM mode, or reset synchronous PWM mode is selected, the values in TRDIORA and TRDIORC are ignored. Upon reset, the GR registers are set as output compare registers (no output) and initialized to H'FFFF. The GR registers cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit.

chanhvinh.com



### 16.2.11 Timer RD Control Register (TRDCR)

Address: H'FFFFCA, H'FFFFD1

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | CCLR[2:0] | | | CKEG[1:0] | | TPSC[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 to 5 | CCLR[2:0] | Counter clear 2 to 0 | 000: Disables TRDCNT clearing<br>001: Clears TRDCNT by GRA compare match/input capture*[1]<br>010: Clears TRDCNT by GRB compare match/input capture*[1]<br>011: Synchronization clear; Clears TRDCNT in synchronous with counter clearing of the other channel's timer*[2]<br>100: Disables TRDCNT clearing<br>101: Clears TRDCNT by GRC compare match/input capture*[1]<br>110: Clears TRDCNT by GRD compare match/input capture*[1]<br>111: Synchronization clear; Clears TRDCNT in synchronous with counter clearing of the other channel's timer*[2] | R/W |
| 4, 3 | CKEG[1:0] | Clock edge 1 and 0 | 00: Count at rising edge<br>01: Count at falling edge<br>1X: Count at both edges | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 2 to 0 | TPSC[2:0] | Time prescaler 2 to 0 | 000: Internal clock: count by φ | R/W |
| | | | 001: Internal clock: count by φ/2 | |
| | | | 010: Internal clock: count by φ/4 | |
| | | | 011: Internal clock: count by φ/8 | |
| | | | 100: Internal clock: count by φ/32 | |
| | | | 101: External clock: count by FTIOA0 (TCLK) pin input | |
| | | | 110: Setting prohibited | |
| | | | 111: Reserved (setting prohibited) | |

[Legend]

X: Don't care

Notes: 1. When GR functions as an output compare register, TRDCNT is cleared by compare match. When GR functions as input capture, TRDCNT is cleared by input capture.

2. Synchronous operation is set by TRDMDR.

TRDCR selects a TRDCNT counter clock, an edge when an external clock is selected, and counter clearing sources. Timer RD has a total of two TRDCR registers, one for each channel.

chanhvinh.com



### 16.2.12 Timer RD I/O Control Registers (TRDIORA and TRDIORC)

• TRDIORA

Address: H'FFFFC5, H'FFFFCC

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | IOB2 | IOB[1:0] | | — | IOA2 | IOA[1:0] | |
| Value after reset: | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

• TRDIORC

Address: H'FFFFD6, H'FFFFCD

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | IOD3 | IOD2 | IOD[1:0] | | IOC3 | IOC2 | IOC[1:0] | |
| Value after reset: | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

- TRDIORA

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is read as 1. The write value should be 1. | — |
| 6 | IOB2 | I/O control B2 | Selects the GRB function.<br>0: GRB functions as an output compare register<br>1: GRB functions as an input capture register | R/W |
| 5, 4 | IOB[1:0] | I/O control B1 and B0 | When IOB2 = 0,<br>00: No output at compare match<br>01: 0 output to the FTIOB pin at GRB compare match<br>10: 1 output to the FTIOB pin at GRB compare match<br>11: Output toggles to the FTIOB pin at GRB compare match<br>When IOB2 = 1,<br>00: Input capture to GRB at rising edge at the FTIOB pin<br>01: Input capture to GRB at falling edge at the FTIOB pin<br>1X: Input capture to GRB at rising and falling edges at the FTIOB pin | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 3 | — | Reserved | This bit is read as 1. The write value should be 1. | — |
| 2 | IOA2 | I/O control A2 | Selects the GRA function.<br>0: GRA functions as an output compare register<br>1: GRA functions as an input capture register | R/W |
| 1, 0 | IOA[1:0] | I/O control A1 and A0 | When IOA2 = 0,<br>00: No output at compare match<br>01: 0 output to the FTIOA pin at GRA compare match<br>10: 1 output to the FTIOA pin at GRA compare match<br>11: Output toggles to the FTIOA pin at GRA compare match<br>When IOA2 = 1,<br>00: Input capture to GRA at rising edge at the FTIOA pin<br>01: Input capture to GRA at falling edge at the FTIOA pin<br>1X: Input capture to GRA at rising and falling edges at the FTIOA pin | R/W |

[Legend]

X: Don't care.

Notes: 1. When a GR register functions as a buffer register for a paired GR register, the settings in the IOA2 and IOB2 bits in TRDIORA and the IOC2 and IOD2 bits in TRDIORC of both registers should be the same.

2. In PWM mode, PWM3 mode, complementary PWM mode, and reset synchronous PWM mode, the settings of TRDIORA are invalid.

TRDIORA selects whether GRA or GRB is used as an output compare register or an input capture register. When an output compare register is selected, the output setting is selected. When an input capture register is selected, an input edge of an input capture signal is selected. TRDIORA also selects the function of FTIOA or FTIOB pin.

chanhvinh.com



- TRDIORC

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | IOD3 | I/O control D3 | Specifies GRD to be used as GR for the FTIOB or FTIOD pin.<br>0: GRD is used as GR for the FTIOB pin<br>1: GRD is used as GR for the FTIOD pin | R/W |
| 6 | IOD2 | I/O control D2 | Selects the GRD function.<br>0: GRD functions as an output compare register<br>1: GRD functions as an input capture register | R/W |
| 5, 4 | IOD[1:0] | I/O control D1 and D0 | When IOD3 = 0,<br>00: No output at compare match<br>01: 0 output to the FTIOB pin at GRD compare match<br>10: 1 output to the FTIOB pin at GRD compare match<br>11: Output toggles to the FTIOB pin at GRD compare match<br>When IOD3 = 1 and IOD2 = 0,<br>00: No output at compare match<br>01: 0 output to the FTIOD pin at GRD compare match<br>10: 1 output to the FTIOD pin at GRD compare match<br>11: Output toggles to the FTIOD pin at GRD compare match<br>When IOD3 = 1 and IOD2 = 1,<br>00: Input capture to GRD at rising edge at the FTIOD pin<br>01: Input capture to GRD at falling edge at the FTIOD pin<br>1X: Input capture to GRD at rising and falling edges at the FTIOD pin | R/W |
| 3 | IOC3 | I/O control C3 | Specifies GRC to be used as GR for the FTIOA or FTIOC pin.<br>0: GRC is used as GR for the FTIOA pin<br>1: GRC is used as GR for the FTIOC pin | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 2 | IOC2 | I/O control C2 | Selects the GRC function.<br>0: GRC functions as an output compare register<br>1: GRC functions as an input capture register | R/W |
| 1, 0 | IOC[1:0] | I/O control C1 and C0 | When IOC3 = 0,<br>00: No output at compare match<br>01: 0 output to the FTIOA pin at GRC compare match<br>10: 1 output to the FTIOA pin at GRC compare match<br>11: Output toggles to the FTIOA pin at GRC compare match<br>When IOC3 = 1 and IOC2 = 0,<br>00: No output at compare match<br>01: 0 output to the FTIOC pin at GRC compare match<br>10: 1 output to the FTIOC pin at GRC compare match<br>11: Output toggles to the FTIOC pin at GRC compare match<br>When IOC3 = 1 and IOC2 = 1,<br>00: Input capture to GRC at rising edge at the FTIOC pin<br>01: Input capture to GRC at falling edge at the FTIOC pin<br>1X: Input capture to GRC at rising and falling edges at the FTIOC pin | R/W |

[Legend]

X: Don't care.

Notes: 1. When a GR register functions as a buffer register for a paired GR register, the settings in the IOA2 and IOB2 bits in TRDIORA and the IOC2 and IOD2 bits in TRDIORC of both registers should be the same.

2. In PWM mode, PWM3 mode, complementary PWM mode, and reset synchronous PWM mode, the settings of TRDIORC are invalid.

TRDIORC selects whether GRC or GRD is used as an output compare register or an input capture register. When an output compare register is selected, the output setting is selected. When an input capture register is selected, an input edge of an input capture signal is selected. TRDIORC also selects the function of the FTIOA to FTIOD pins.

chanhvinh.com



### 16.2.13 Timer RD Status Register (TRDSR)

Address: H'FFFFC7, H'FFFFCE

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | UDF | OVF | IMFD | IMFC | IMFB | IMFA |
| Value after reset: | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7, 6 | — | Reserved | These bits are read as 1. The write value should be 1. | — |
| 5 | UDF* | Underflow flag | 0: TRDCNT_1 has not underflowed.<br>1: TRDCNT_1 has underflowed.<br>[Setting condition]<br>• When TRDCNT underflows<br>[Clearing condition]<br>• When 0 is written to UDF after reading UDF = 1 | R/W |
| 4 | OVF | Overflow flag | 0: TRDCNT has not overflowed.<br>1: TRDCNT has overflowed.<br>[Setting condition]<br>• When TRDCNT value is underflowed<br>[Clearing condition]<br>When 0 is written to OVF after reading OVF = 1 | R/W |
| 3 | IMFD | Input capture/ compare match flag D | [Setting conditions]<br>• When TRDCNT = GRD and GRD is functioning as output compare register<br>• When TRDCNT = GRD while the FTIOD pin operates in PWM mode<br>• When TRDCNT = GRD in PWM3 mode, reset synchronous PWM mode, or complementary PWM mode<br>• When TRDCNT value is transferred to GRD by input capture signal and GRD is functioning as input capture register<br>[Clearing conditions]<br>• When the DTC is activated by an IMFD interrupt and the DISEL bit in MRB of the DTC is 0<br>• When 0 is written to IMFD after reading IMFD = 1 | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 2 | IMFC | Input capture/ compare match flag C | [Setting conditions]<br>• When TRDCNT = GRC and GRC is functioning as output compare register<br>• When TRDCNT = GRC while the FTIOC pin operates in PWM mode<br>• When TRDCNT = GRC in PWM3 mode, reset synchronous PWM mode, or complementary PWM mode<br>• When TRDCNT value is transferred to GRC by input capture signal and GRC is functioning as input capture register<br>[Clearing conditions]<br>• When the DTC is activated by an IMFC interrupt and the DISEL bit in MRB of the DTC is 0<br>• When 0 is written to IMFC after reading IMFC = 1 | R/W |
| 1 | IMFB | Input capture/ compare match flag B | [Setting conditions]<br>• When TRDCNT = GRB and GRB is functioning as output compare register<br>• When TRDCNT = GRB while the FTIOB pin operates in PWM mode<br>• When TRDCNT = GRB in PWM mode, PWM3 mode, reset synchronous PWM mode, or complementary PWM mode (in reset synchronous PWM mode, however, while TRDCNT_0 = GRB_1 and TRDCNT_0 = GRB_0)<br>• When TRDCNT value is transferred to GRB by input capture signal and GRB is functioning as input capture register<br>[Clearing conditions]<br>• When the DTC is activated by an IMFB interrupt and the DISEL bit in MRB of the DTC is 0<br>• When 0 is written to IMFB after reading IMFB = 1 | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 0 | IMFA | Input capture/ compare match flag A | [Setting conditions]<br>• When TRDCNT = GRA and GRA is functioning as output compare register<br>• When TRDCNT = GRA in PWM mode, PWM3 mode, reset synchronous PWM mode, or complementary PWM mode (in reset synchronous PWM mode, however, while TRDCNT_0 = GRA_1 and TRDCNT_0 = GRA_0)<br>• When TRDCNT value is transferred to GRA by input capture signal and GRA is functioning as input capture register<br>[Clearing conditions]<br>• When the DTC is activated by an IMFA interrupt and the DISEL bit in MRB of the DTC is 0<br>• When 0 is written to IMFA after reading IMFA = 1 | R/W |

Note: * Bit 5 is not the UDF flag in TRDSR_0. It is a reserved bit. It is always read as 1.

TRDSR is each interrupt request flag of the timer RD. If an interrupt is enabled by a corresponding bit in TRDIER, TRDSR requests an interrupt for the CPU. Timer RD has two TRDSR registers, one for each channel.

chanhvinh.com



### 16.2.14 Timer RD Interrupt Enable Register (TRDIER)

Address: H'FFFFC8, H'FFFFCF

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | OVIE | IMIED | IMIEC | IMIEB | IMIEA |
| Value after reset: | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 to 5 | — | Reserved | These bits are read as 1. The write value should be 1. | — |
| 4 | OVIE | Overflow interrupt enable | 0: Interrupt requests (OVI) by OVF or UDF flag are disabled.<br>1: Interrupt requests (OVI) by OVF or UDF flag are enabled. | R/W |
| 3 | IMIED | Input capture/ compare match interrupt enable D | 0: Interrupt requests (IMID) by IMFD flag are disabled.<br>1: Interrupt requests (IMID) by IMFD flag are enabled. | R/W |
| 2 | IMIEC | Input capture/ compare match interrupt enable C | 0: Interrupt requests (IMIC) by IMFC flag are disabled.<br>1: Interrupt requests (IMIC) by IMFC flag are enabled. | R/W |
| 1 | IMIEB | Input capture/ compare match interrupt enable B | 0: Interrupt requests (IMIB) by IMFB flag are disabled.<br>1: Interrupt requests (IMIB) by IMFB flag are enabled. | R/W |
| 0 | IMIEA | Input capture/ compare match interrupt enable A | 0: Interrupt requests (IMIA) by IMFA flag are disabled.<br>1: Interrupt requests (IMIA) by IMFA flag are enabled. | R/W |

Timer RD has two TRDIER registers, one for each channel.

chanhvinh.com



### 16.2.15 PWM Mode Output Level Control Register (POCR)

Address: H'FFFFC9, H'FFFFD0

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | POLD | POLC | POLB |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 to 3 | — | Reserved | These bits are read as 1. The write value should be 1. | — |
| 2 | POLD | PWM mode output level control D | 0: The output level of FTIOD is active low.<br>1: The output level of FTIOD is active high. | R/W |
| 1 | POLC | PWM mode output level control C | 0: The output level of FTIOC is active low.<br>1: The output level of FTIOC is active high. | R/W |
| 0 | POLB | PWM mode output level control B | 0: The output level of FTIOB is active low.<br>1: The output level of FTIOB is active high. | R/W |

Timer RD has two POCR registers, one for each channel.

chanhvinh.com



### 16.2.16 Timer RD Digital Filtering Function Select Register (TRDDF)

Address: H'FFFFCA, H'FFFFD1

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | DFCK[1:0] | | — | — | DFD | DFC | DFB | DFA |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7, 6 | DFCK[1:0] | Digital filter clock select | 00: φ/32<br>01: φ/8<br>10: φ<br>11: Clock specified by bits TPSC2 to TPSC0 in TRDCR | R/W |
| 5, 4 | — | Reserved | These bits are read as 0. The write value should be 0. | — |
| 3 | DFD | Digital filter function D | 0: Disables the digital filter for the FTIOD pin<br>1: Enables the digital filter for the FTIOD pin | R/W |
| 2 | DFC | Digital filter function C | 0: Disables the digital filter for the FTIOC pin<br>1: Enables the digital filter for the FTIOC pin | R/W |
| 1 | DFB | Digital filter function B | 0: Disables the digital filter for the FTIOB pin<br>1: Enables the digital filter for the FTIOB pin | R/W |
| 0 | DFA | Digital filter function A | 0: Disables the digital filter for the FTIOA pin<br>1: Enables the digital filter for the FTIOA pin | R/W |

Note: The setting in this register is valid on the corresponding pin when the FTIOA to FTIOD inputs are enabled by TRDIORA and TRDIORC.
Timer RD has two TRDDF registers, one for each channel.

chanhvinh.com



### 16.2.17 Interface with CPU

#### (1) 16-Bit Register

TRDCNT and GR are 16-bit registers. Reading/writing in a 16-bit unit is enabled but disabled in an 8-bit unit since the data bus with the CPU is 16-bit width. These registers must always be accessed in a 16-bit unit. Figure 16.5 shows an example of accessing the 16-bit registers.

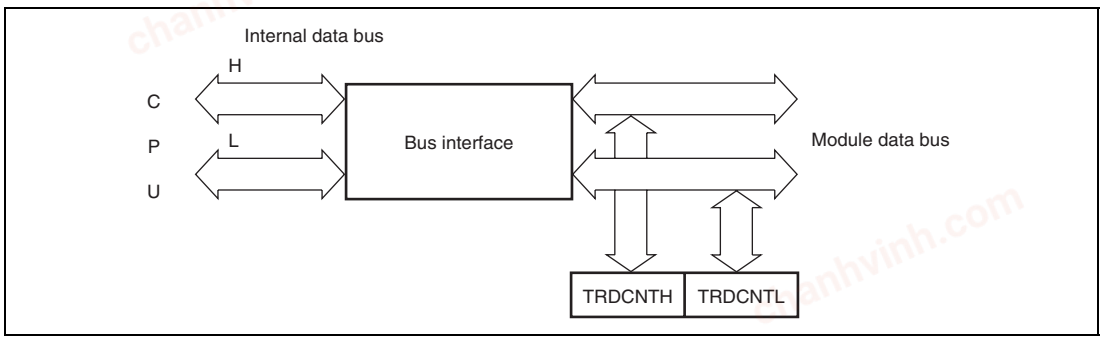



**Figure 16.5 Accessing Operation of 16-Bit Register (between CPU and TRDCNT (16 bits))**

#### (2) 8-Bit Register

Registers other than TRDCNT and GR are 8-bit registers that are connected internally with the CPU in an 8-bit width. Figure 16.6 shows an example of accessing the 8-bit registers.

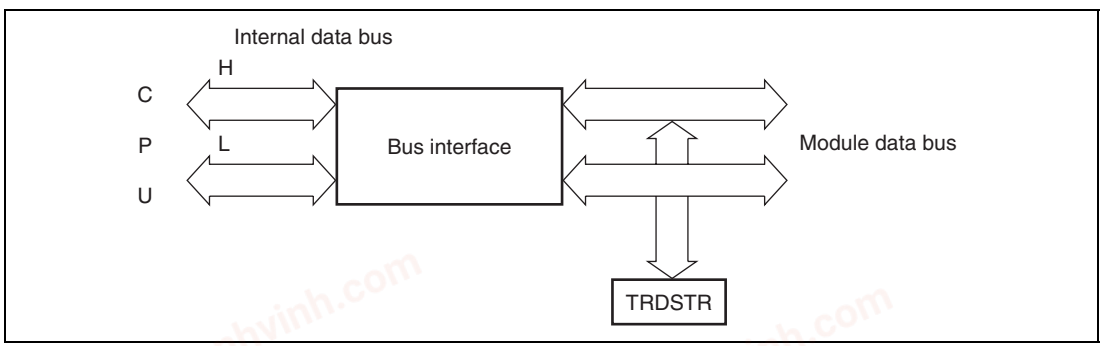



**Figure 16.6 Accessing Operation of 8-Bit Register (between CPU and TRDSTR (8 bits))**

chanhvinh.com



## 16.3 Operation

Timer RD has the following operating modes.

- Timer mode operation
  Enables output compare and input capture functions by setting the IOA2 to IOA0 and IOB2 to IOB0 bits in TRDIORA and the IOC3 to IOC0 and IOD3 to IOD0 bits in TRDIORC
- PWM mode operation
  Enables PWM mode operation by setting TRDPMR
- PWM3 mode operation
  Enables PWM3 mode operation by setting the PWM3 bit in TRDFCR
- Reset synchronous PWM mode operation
  Enables reset synchronous PWM mode operation by setting the CMD1 and CMD0 bits in TRDFCR
- Complementary PWM mode operation
  Enables complementary PWM mode operation by setting the CMD1 and CMD0 bits in TRDFCR

The following tables show the operating modes of the FTIOA0 to FTIOD0 and FTIOA1 to FTIOD1 pins set by the appropriate bits in the registers mentioned above. Set 1 to the PMR bits corresponding to the pins allocated by the PMC.

chanhvinh.com



- FTIOA0 pin

| Register Name | TRDOER1 | | TRDFCR | | TRDIORA | |
|---|---|---|---|---|---|---|
| **Bit Name** | **EA0** | **STCLK** | **CMD1, CMD0** | **PWM3** | **IOA2 to IOA0** | **Function** |
| Setting values | 0 | 0 | 00 | 0 | XXX | PWM3 mode waveform output |
| | 0 | 0 | 00 | 1 | 001, 01X | Timer mode waveform output (output compare function) |
| | X | 0 | 00 | 1 | 1XX | Timer mode (input capture function) |
| | X | 0 | 00 | 1 | 000 | General input port (when the corresponding pin PCR = 0) |
| | X | 1 | XX | X | 0XX | External clock input |
| | Other than above | | | | | Setting prohibited |

chanhvinh.com



- FTIOB0 pin

| Register Name | TRDOER1 | TRDFCR | | TRDPMR | TRDIORA | |
|---|---|---|---|---|---|---|
| **Bit Name** | **EB0** | **CMD1, CMD0** | **PWM3** | **PWMB0** | **IOB2 to IOB0** | **Function** |
| Setting values | 0 | 10, 11 | X | X | XXX | Complementary PWM mode waveform output |
| | 0 | 01 | X | X | XXX | Reset synchronous PWM mode waveform output |
| | 0 | 00 | 0 | X | XXX | PWM3 mode waveform output |
| | 0 | 00 | 1 | 1 | XXX | PWM mode waveform output |
| | 0 | 00 | 1 | 0 | 001, 01X | Timer mode waveform output (output compare function) |
| | X | 00 | 1 | 0 | 1XX | Timer mode (input capture function) |
| | X | 00 | 1 | 0 | 000 | General input port (when the corresponding pin PCR = 0) |
| | Other than above | | | | | Setting prohibited |

chanhvinh.com



- FTIOC0 pin

| Register Name | TRDOER1 | TRDFCR | | TRDPMR | TRDIORC | |
|---|---|---|---|---|---|---|
| Bit Name | EC0 | CMD1, CMD0 | PWM3 | PWMC0 | IOC2 to IOC0 | Function |
| Setting values | 0 | 10, 11 | X | X | XXX | Complementary PWM mode waveform output |
| | 0 | 01 | X | X | XXX | Reset synchronous PWM mode waveform output |
| | 0 | 00 | 1 | 1 | XXX | PWM mode waveform out |
| | 0 | 00 | 1 | 0 | 001, 01X | Timer mode waveform output (output compare function) |
| | X | 00 | 1 | 0 | 1XX | Timer mode (input capture function) |
| | X | 00 | 1 | 0 | 000 | General input port (when the corresponding pin PCR = 0) |
| | Other than above | | | | | Setting prohibited |

chanhvinh.com



- FTIOD0 pin

| Register Name | TRDOER1 | TRDFCR | | TRDPMR | TRDIORC | |
|---|---|---|---|---|---|---|
| **Bit Name** | **ED0** | **CMD1, CMD0** | **PWM3** | **PWMD0** | **IOD2 to IOD0** | **Function** |
| Setting values | 0 | 10, 11 | X | X | XXX | Complementary PWM mode waveform output |
| | 0 | 01 | X | X | XXX | Reset synchronous PWM mode waveform output |
| | 0 | 00 | 1 | 1 | XXX | PWM mode waveform out |
| | 0 | 00 | 1 | 0 | 001, 01X | Timer mode waveform output (output compare function) |
| | X | 00 | 1 | 0 | 1XX | Timer mode (input capture function) |
| | X | 00 | 1 | 0 | 000 | General input port (when the corresponding pin PCR = 0) |
| | Other than above | | | | | Setting prohibited |

chanhvinh.com



- FTIOA1 pin

| Register Name | TRDOER1 | TRDFCR | | TRDIORA | |
|---|---|---|---|---|---|
| Bit Name | EA1 | CMD1, CMD0 | PWM3 | IOA2 to IOA0 | Function |
| Setting values | 0 | 10, 11 | X | XXX | Complementary PWM mode waveform output |
| | 0 | 01 | X | XXX | Reset synchronous PWM mode waveform output |
| | 0 | 00 | 1 | 001, 01X | Timer mode waveform output (output compare function) |
| | X | 00 | 1 | 1XX | Timer mode (input capture function) |
| | X | 00 | 1 | 000 | General input port (when the corresponding pin PCR = 0) |
| | Other than above | | | | Setting prohibited |

chanhvinh.com



- FTIOB1 pin

| Register Name | TRDOER1 | TRDFCR | | TRDPMR | TRDIORA | |
|---|---|---|---|---|---|---|
| **Bit Name** | **EB1** | **CMD1, CMD0** | **PWM3** | **PWMB1** | **IOB2 to IOB0** | **Function** |
| Setting values | 0 | 10, 11 | X | X | XXX | Complementary PWM mode waveform output |
| | 0 | 01 | X | X | XXX | Reset synchronous PWM mode waveform output |
| | 0 | 00 | 1 | 1 | XXX | PWM mode waveform out |
| | 0 | 00 | 1 | 0 | 001, 01X | Timer mode waveform output (output compare function) |
| | X | 00 | 1 | 0 | 1XX | Timer mode (input capture function) |
| | X | 00 | 1 | 0 | 000 | General input port (when the corresponding pin PCR = 0) |
| | Other than above | | | | | Setting prohibited |

chanhvinh.com



- FTIOC1 pin

| Register Name | TRDOER1 | TRDFCR | | TRDPMR | TRDIORC | |
|---|---|---|---|---|---|---|
| **Bit Name** | **EC1** | **CMD1, CMD0** | **PWM3** | **PWMC1** | **IOC2 to IOC0** | **Function** |
| Setting values | 0 | 10, 11 | X | X | XXX | Complementary PWM mode waveform output |
| | 0 | 01 | X | X | XXX | Reset synchronous PWM mode waveform output |
| | 0 | 00 | 1 | 1 | XXX | PWM mode waveform out |
| | 0 | 00 | 1 | 0 | 001, 01X | Timer mode waveform output (output compare function) |
| | X | 00 | 1 | 0 | 1XX | Timer mode (input capture function) |
| | X | 00 | 1 | 0 | 000 | General input port (when the corresponding pin PCR = 0) |
| | Other than above | | | | | Setting prohibited |

chanhvinh.com



- FTIOD1 pin

| Register Name | TRDOER1 | TRDFCR | | TRDPMR | TRDIORC | |
|---|---|---|---|---|---|---|
| **Bit Name** | **ED1** | **CMD1, CMD0** | **PWM3** | **PWMD1** | **IOD2 to IOD0** | **Function** |
| Setting values | 0 | 10, 11 | X | X | XXX | Complementary PWM mode waveform output |
| | 0 | 01 | X | X | XXX | Reset synchronous PWM mode waveform output |
| | 0 | 00 | 1 | 1 | XXX | PWM mode waveform out |
| | 0 | 00 | 1 | 0 | 001, 01X | Timer mode waveform output (output compare function) |
| | X | 00 | 1 | 0 | 1XX | Timer mode (input capture function) |
| | X | 00 | 1 | 0 | 000 | General input port (when the corresponding pin PCR = 0) |
| | Other than above | | | | | Setting prohibited |

chanhvinh.com



### 16.3.1 Counter Operation

When one of bits STR0 and STR1 in TRDSTR is set to 1, the TRDCNT counter for the corresponding channel begins counting. TRDCNT can operate as a free-running counter, periodic counter, for example. Figure 16.7 shows an example of the counter operation setting procedure.

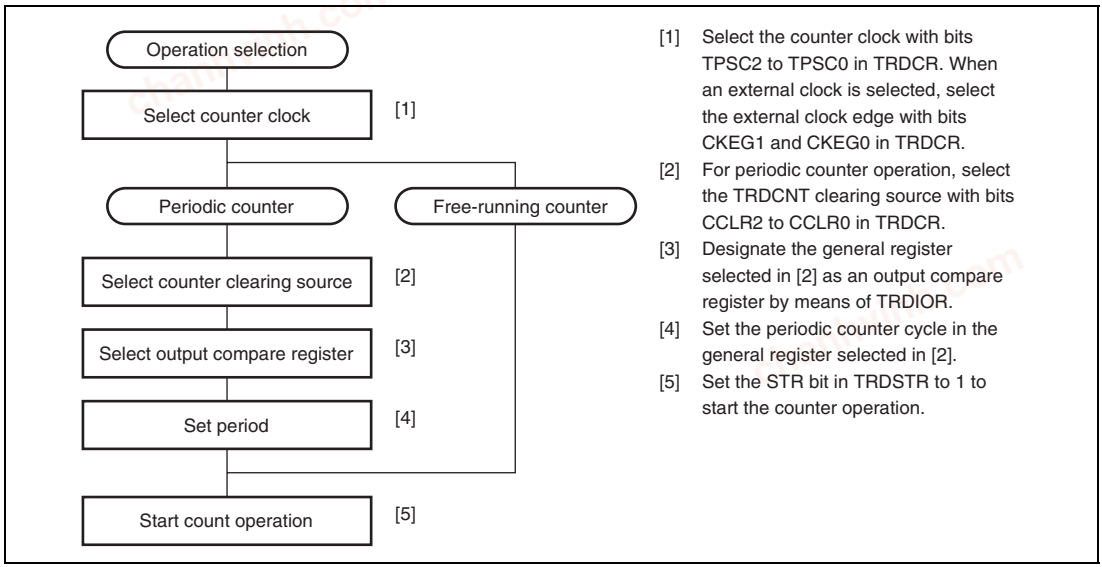



**Figure 16.7 Example of Counter Operation Setting Procedure**

chanhvinh.com



### (1) Free-Running Count Operation and Periodic Count Operation

Immediately after a reset, the TRDCNT counters for channels 0 and 1 are all designated as free-running counters. When the relevant bit in TRDSTR is set to 1, the corresponding TRDCNT counter starts an increment operation as a free-running counter. When TRDCNT overflows, the OVF flag in TRDSR is set to 1. If the value of the OVIE bit in the corresponding TRDIER is 1 at this point, timer RD requests an interrupt. After overflow, TRDCNT starts an increment operation again from H'0000.

Figure 16.8 illustrates free-running counter operation.

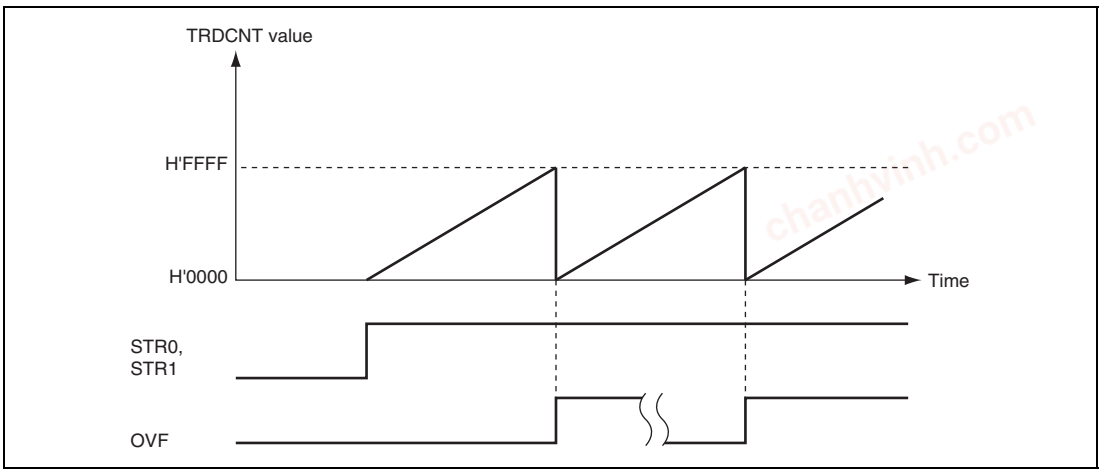



**Figure 16.8 Free-Running Counter Operation**

When compare match is selected as the TRDCNT clearing source, the TRDCNT counter for the relevant channel performs periodic count operation. The GR registers for setting the period are designated as output compare registers, and counter clearing by compare match is selected by means of bits CCLR1 and CCLR0 in TRDCR. After the settings have been made, TRDCNT starts an increment operation as a periodic counter when the corresponding bit in TRDSTR is set to 1. When the count value matches the value in GR, the IMFA, IMFB, IMFC, or IMFD flag in TRDSR is set to 1 and TRDCNT is cleared to H'0000. If the value of the corresponding IMIEA, IMIEB, IMIEC, or IMIED bit in TRDIER is 1 at this point, timer RD requests an interrupt. After a compare match, TRDCNT starts an increment operation again from H'0000.

chanhvinh.com



Figure 16.9 illustrates periodic counter operation.

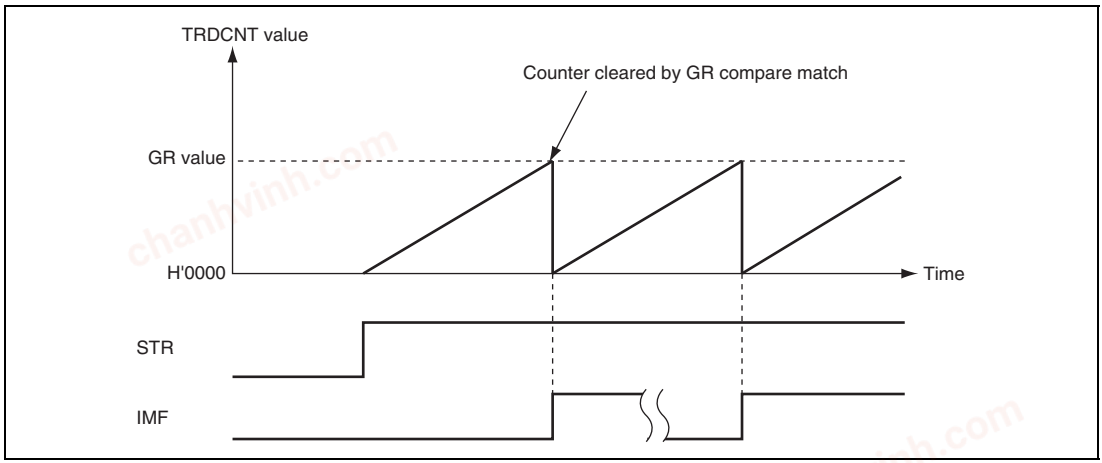



**Figure 16.9 Periodic Counter Operation**

### (2) TRDCNT Count Timing

- Internal clock operation

  A system clock (ϕ), or four types of clocks (ϕ/2, ϕ/4, ϕ/8, or ϕ/32) that are generated by dividing the system clock can be selected by bits TPSC2 to TPSC0 in TRDCR.

  Figure 16.10 illustrates this timing.

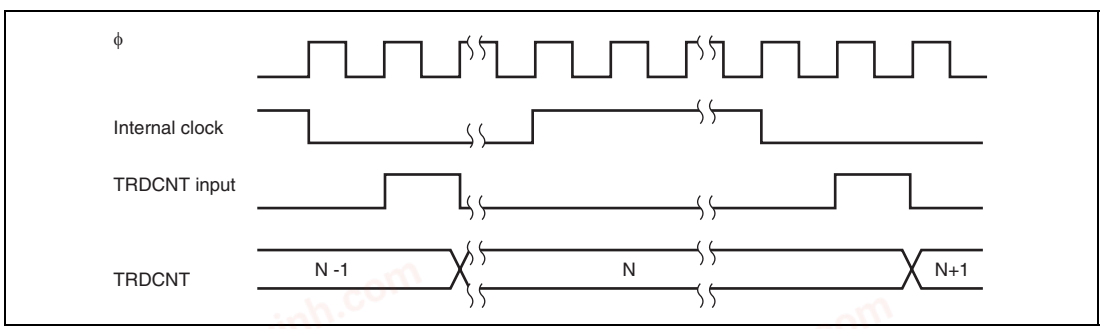



**Figure 16.10 Count Timing in Internal Clock Operation**

chanhvinh.com



- External clock operation

  An external clock input pin (TCLK) can be selected by bits TPSC2 to TPSC0 in TRDCR, and a detection edge can be selected by bits CKEG1 and CKEG0. To detect an external clock, the rising edge, falling edge, or both edges can be selected.

  Figure 16.11 illustrates the detection timing of the rising and falling edges.

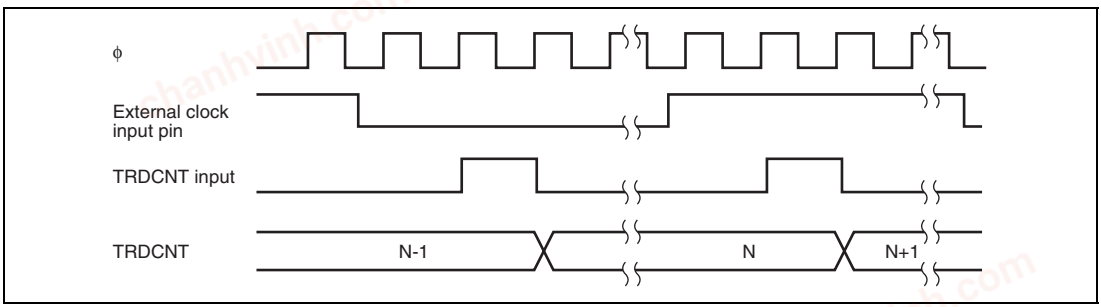



**Figure 16.11 Count Timing in External Clock Operation (Both Edges Detected)**

### 16.3.2 Waveform Output by Compare Match

Timer RD can perform 0, 1, or toggle output from the corresponding FTIOA, FTIOB, FTIOC, or FTIOD output pin using compare match A, B, C, or D.

Figure 16.12 shows an example of the setting procedure for waveform output by compare match.

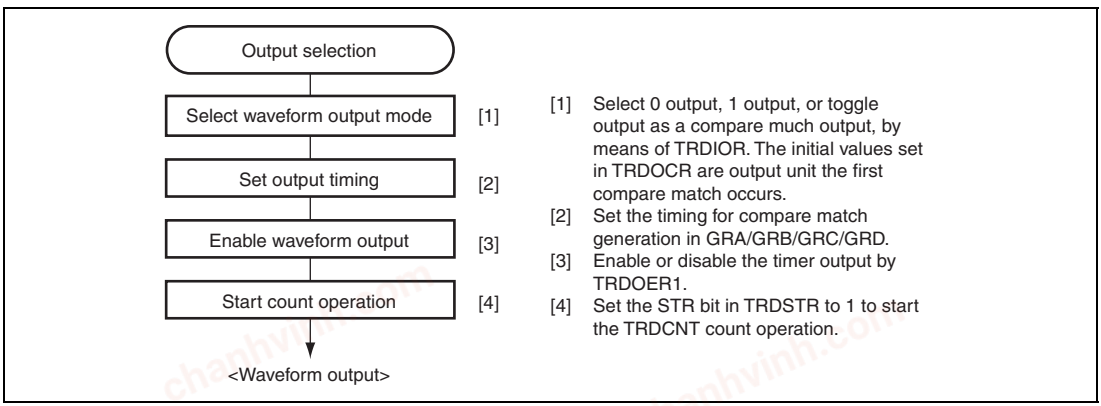



**Figure 16.12 Example of Setting Procedure for Waveform Output by Compare Match**

chanhvinh.com



### (1) Examples of Waveform Output Operation

Figure 16.13 shows an example of 0 output/1 output.

In this example, TRDCNT has been designated as a free-running counter, and settings have been made such that 0 is output by compare match A, and 1 is output by compare match B. When the set level and the pin level coincide, the pin level does not change.

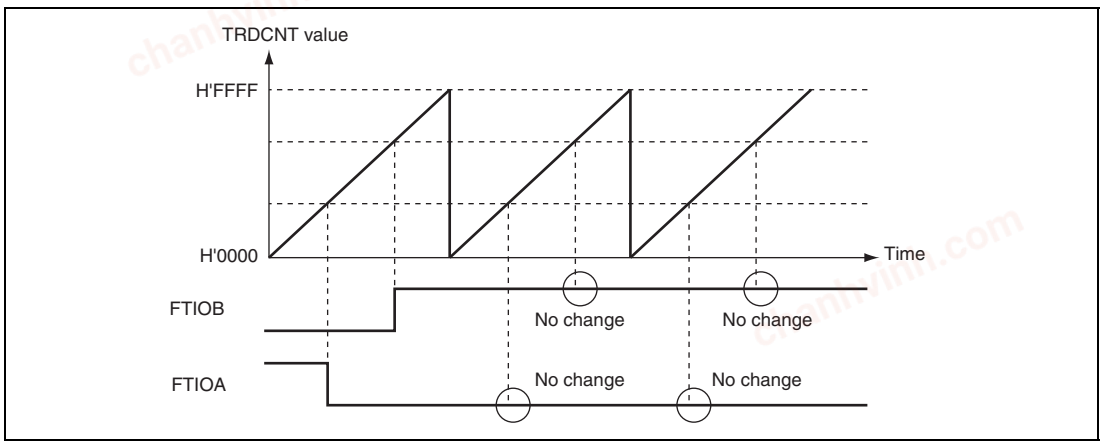



**Figure 16.13 Example of 0 Output/1 Output Operation**

Figure 16.14 shows an example of toggle output.

In this example, TRDCNT 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.

chanhvinh.com



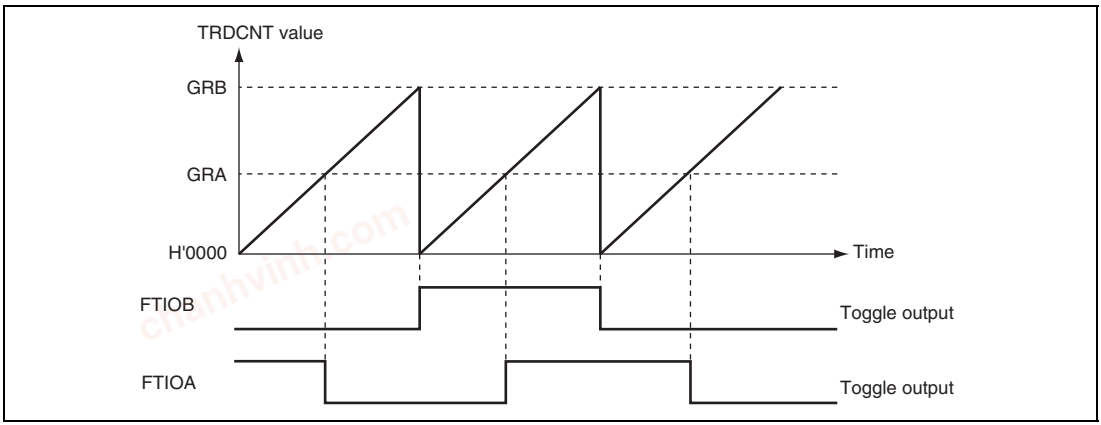



**Figure 16.14 Example of Toggle Output Operation**

### (2) Output Compare Timing

The compare match signal is generated in the last state in which TRDCNT and GR match (when TRDCNT changes from the matching value to the next value). When the compare match signal is generated, the output value selected in TRDIOR is output at the compare match output pin (FTIOA, FTIOB, FTIOC, or FTIOD). When TRDCNT matches GR, the compare match signal is generated only after the next TRDCNT input clock pulse is input.

Figure 16.15 shows an example of the output compare timing.

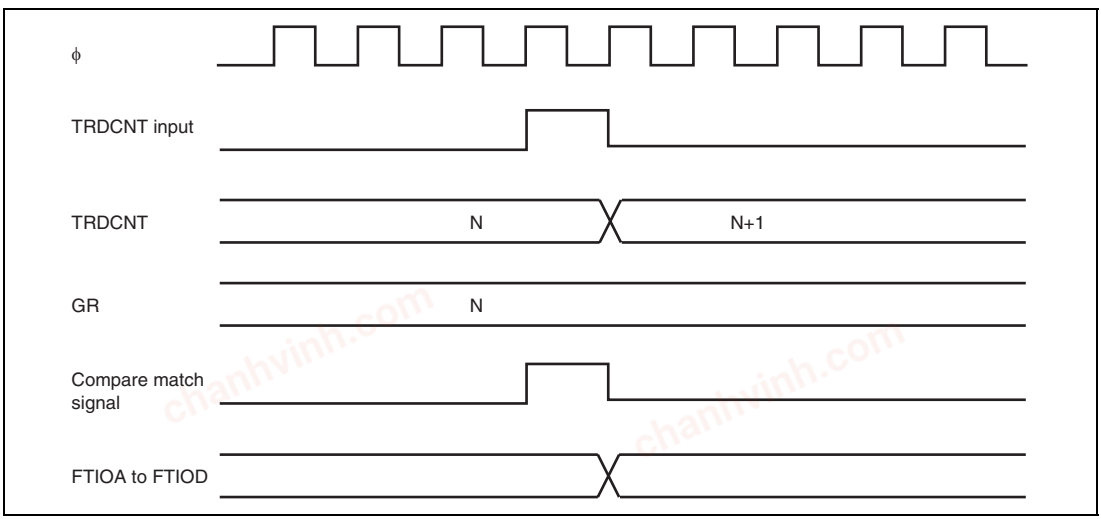



**Figure 16.15 Output Compare Timing**

chanhvinh.com



### 16.3.3 Input Capture Function

The TRDCNT value can be transferred to GR on detection of the input edge of the input capture/output compare pin (FTIOA, FTIOB, FTIOC, or FTIOD). Rising edge, falling edge, or both edges can be selected as the detected edge. When the input capture function is used, the pulse width or period can be measured.

Figure 16.16 shows an example of the input capture operation setting procedure.

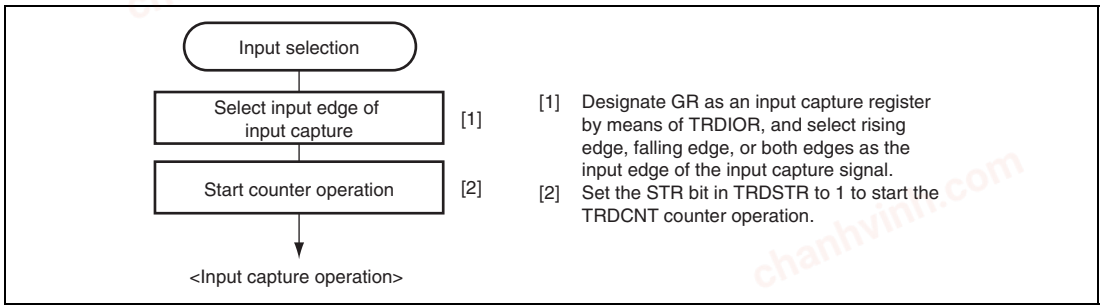



**Figure 16.16 Example of Input Capture Operation Setting Procedure**

chanhvinh.com



### (1) Example of Input Capture Operation

Figure 16.17 shows an example of input capture operation.

In this example, both rising and falling edges have been selected as the FTIOA pin input capture input edge, the falling edge has been selected as the FTIOB pin input capture input edge, and counter clearing by GRB input capture has been designated for TRDCNT.

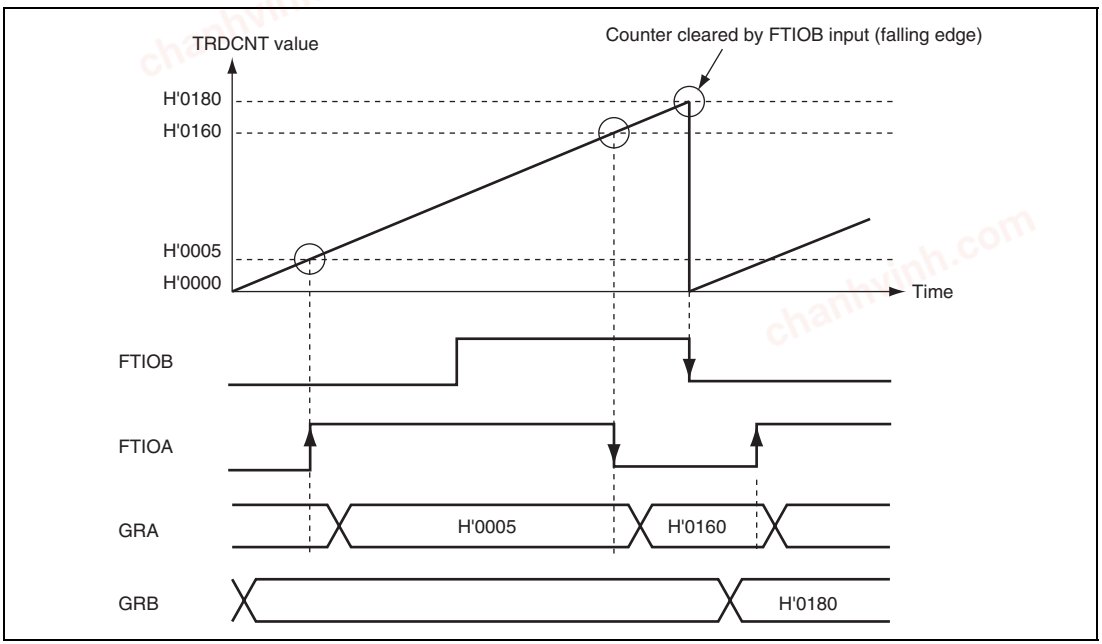



**Figure 16.17 Example of Input Capture Operation**

chanhvinh.com



### (2) Input Capture Signal Timing

Input capture on the rising edge, falling edge, or both edges can be selected through settings in TRDIOR. Figure 16.18 shows the timing when the rising edge is selected.

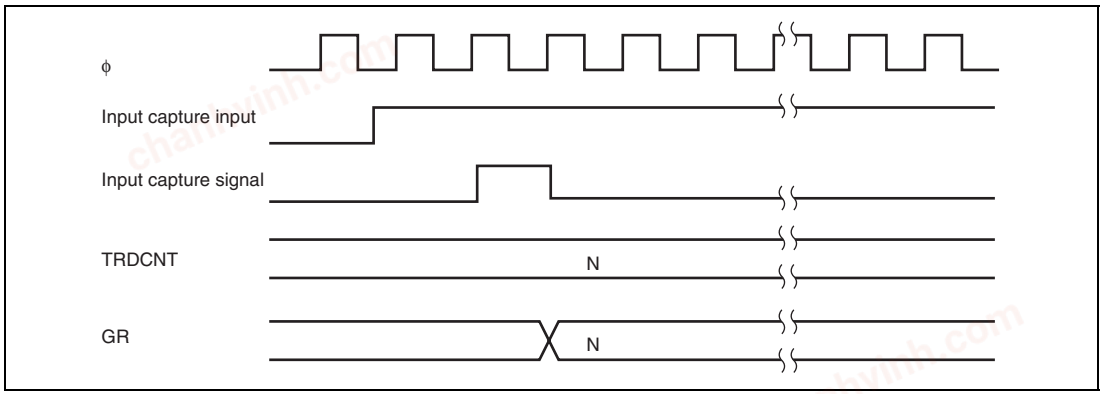



**Figure 16.18 Input Capture Signal Timing**

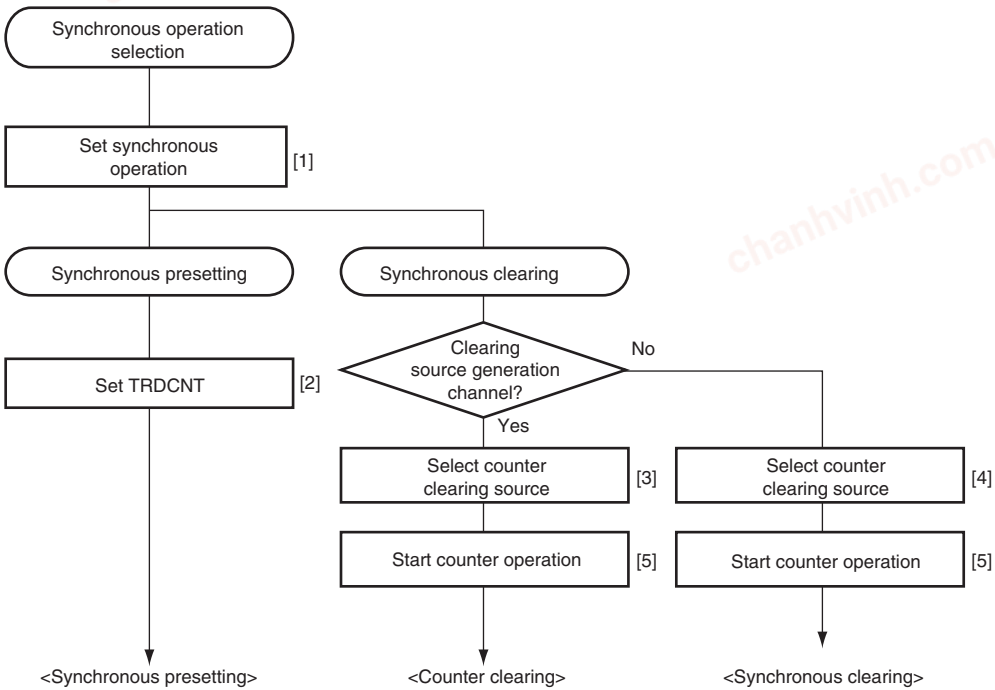
chanhvinh.com



### 16.3.4 Synchronous Operation

In synchronous operation, the values in a number of TRDCNT counters can be rewritten simultaneously (synchronous presetting). Also, a number of TRDCNT counters can be cleared simultaneously by making the appropriate setting in TRDCR (synchronous clearing). Synchronous operation enables GR to be increased with respect to a single time base.

Figure 16.19 shows an example of the synchronous operation setting procedure.




[1] Set the SYNC bits in TRDMDR to 1.
[2] When a value is written to either of the TRDCNT counters, the same value is simultaneously written to the other TRDCNT counter.
[3] Set bits CCLR1 and CCLR0 in TRDCR to specify counter clearing by compare match/input capture.
[4] Set bits CCLR1 and CCLR0 in TRDCR to designate synchronous clearing for the counter clearing source.
[5] Set the STR bit in TRDSTR to 1 to start the count operation.

**Figure 16.19 Example of Synchronous Operation Setting Procedure**

chanhvinh.com



Figure 16.20 shows an example of synchronous operation. In this example, synchronous operation has been selected, FTIOB0 and FTIOB1 have been designated for PWM mode, GRA_0 compare match has been set as the channel 0 counter clearing source, and synchronous clearing has been set for the channel 1 counter clearing source. The same input clock has been set for the channel 0 and channel 1 counter input clocks. Two-phase PWM waveforms are output from pins FTIOB0 and FTIOB1. At this time, synchronous presetting and synchronous operation by GRA_0 compare match are performed by TRDCNT counters.

For details on PWM mode, see section 16.3.5, PWM Mode.

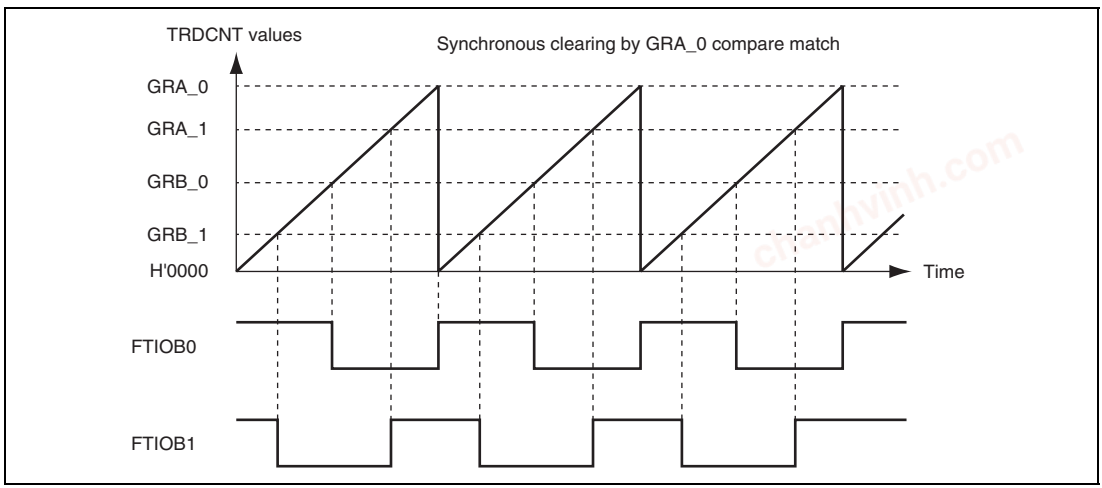



**Figure 16.20 Example of Synchronous Operation**

### 16.3.5 PWM Mode

In PWM mode, PWM waveforms are output from the FTIOB, FTIOC, and FTIOD output pins with GRA as a cycle register and GRB, GRC, and GRD as duty registers. The initial output level of the corresponding pin depends on the setting values of TRDOCR and POCR. Table 16.4 shows an example of the initial output level of the FTIOB0 pin.

The output level is determined by the POLB to POLD bits corresponding to POCR. When POLB is 0, the FTIOB output pin is set to 0 by compare match B and set to 1 by compare match A. When POLB is 1, the FTIOB output pin is set to 1 by compare match B and cleared to 0 by compare match A. In PWM mode, maximum 6-phase PWM outputs are possible.

chanhvinh.com



Figure 16.21 shows an example of the PWM mode setting procedure.

**Table 16.4 Initial Output Level of FTIOB0 Pin**

| TOB0 | POLB | Initial Output Level |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

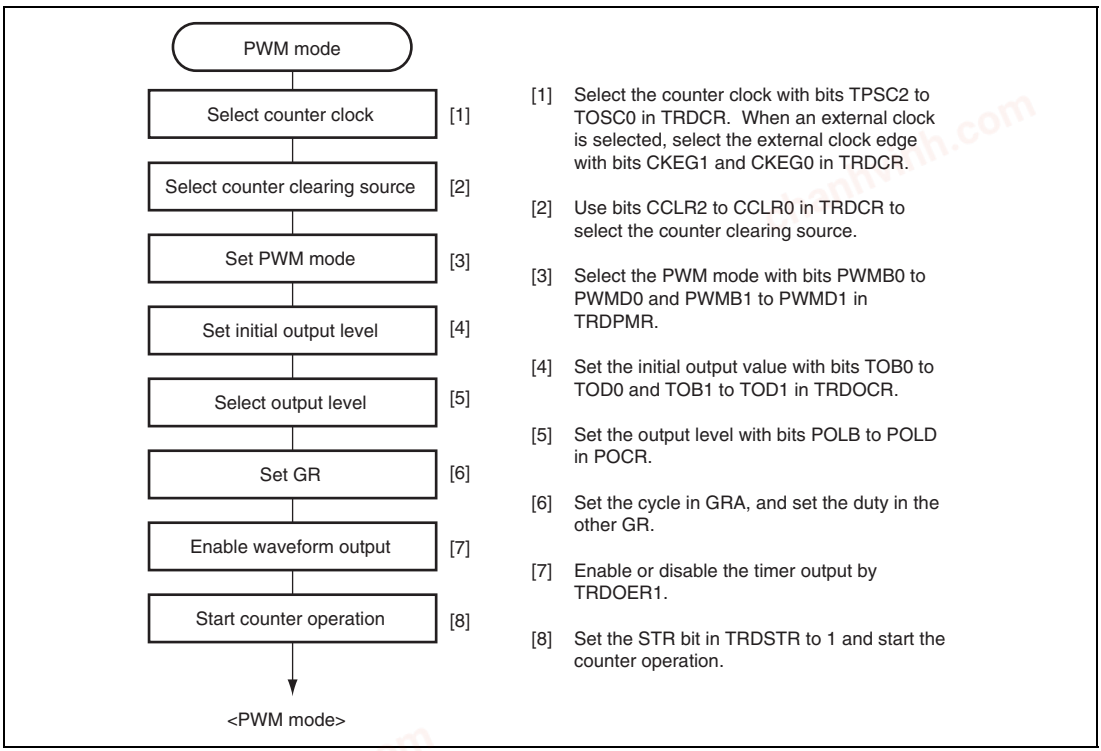



**Figure 16.21 Example of PWM Mode Setting Procedure**

chanhvinh.com



Figure 16.22 shows an example of operation in PWM mode. The output signals go to 1 and TRDCNT is reset at compare match A, and the output signals go to 0 at compare match B, C, and D (TOB, TOC, and TOD = 0, POLB, POLC, and POLD = 0).

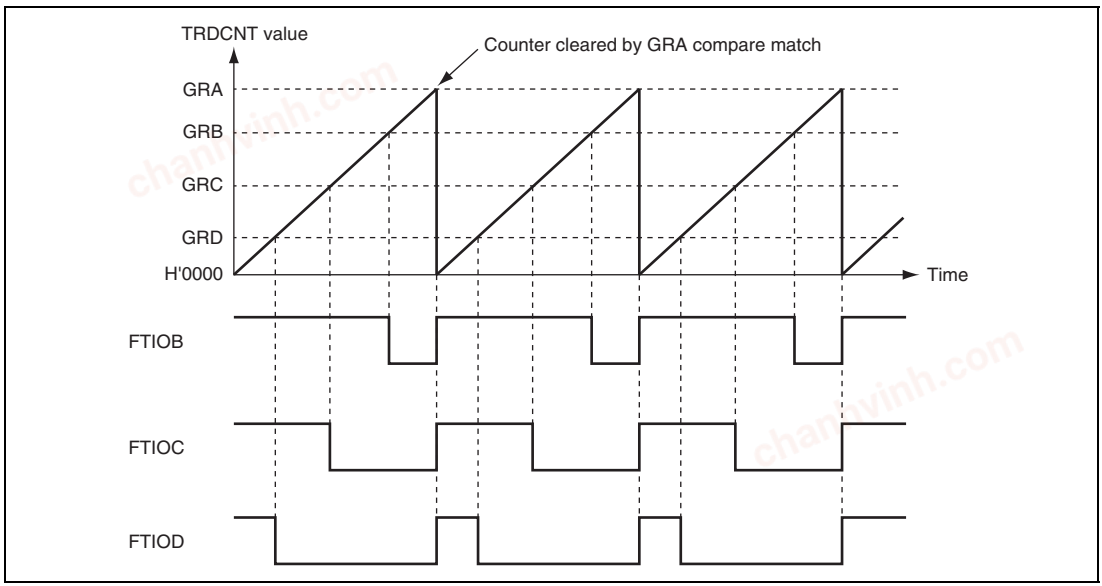



**Figure 16.22 Example of PWM Mode Operation (1)**

chanhvinh.com



Figure 16.23 shows another example of operation in PWM mode. The output signals go to 0 and TRDCNT is reset at compare match A, and the output signals go to 1 at compare match B, C, and D (TOB, TOC, and TOD = 0, POLB, POLC, and POLD = 1).

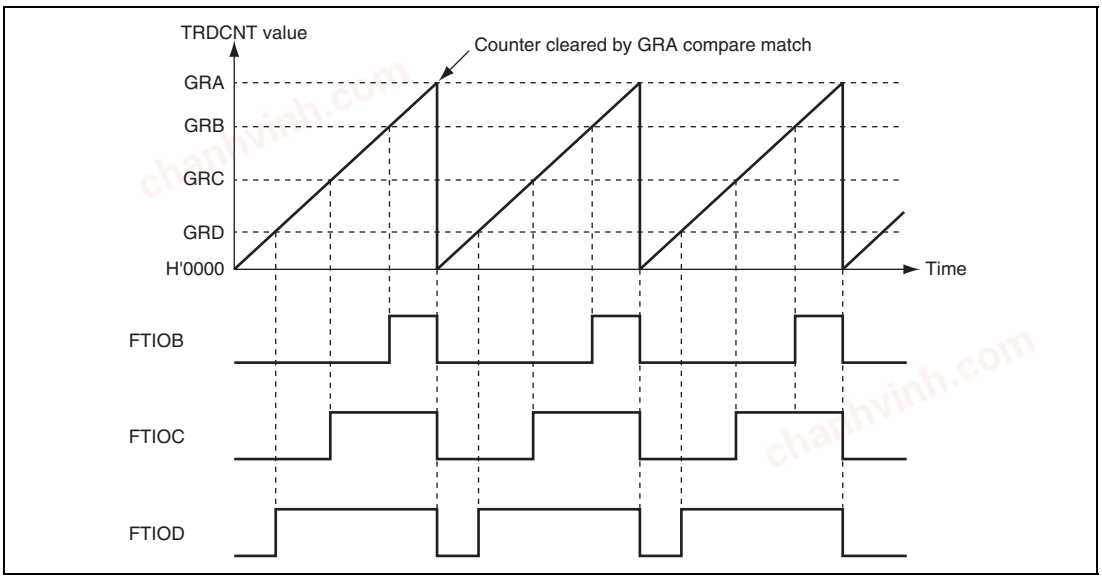



**Figure 16.23 Example of PWM Mode Operation (2)**

chanhvinh.com



Figures 16.24 (when TOB, TOC, and TOD = 0, POLB, POLC, and POLD = 0) and 16.25 (when TOB, TOC, and TOD = 0, POLB, POLC, and POLD = 1) show examples of the output of PWM waveforms with duty cycles of 0% and 100% in PWM mode.

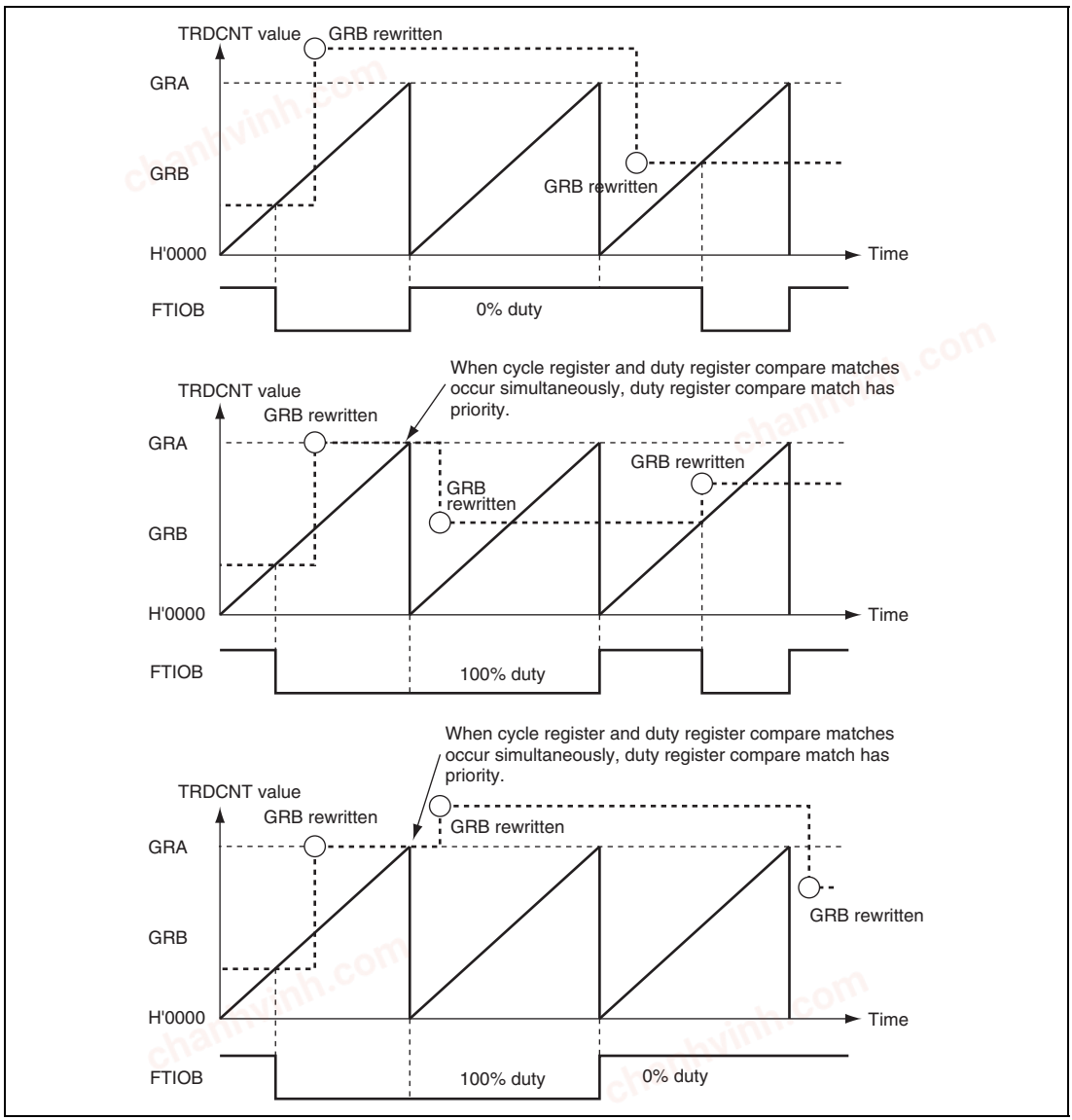



**Figure 16.24 Example of PWM Mode Operation (3)**

chanhvinh.com



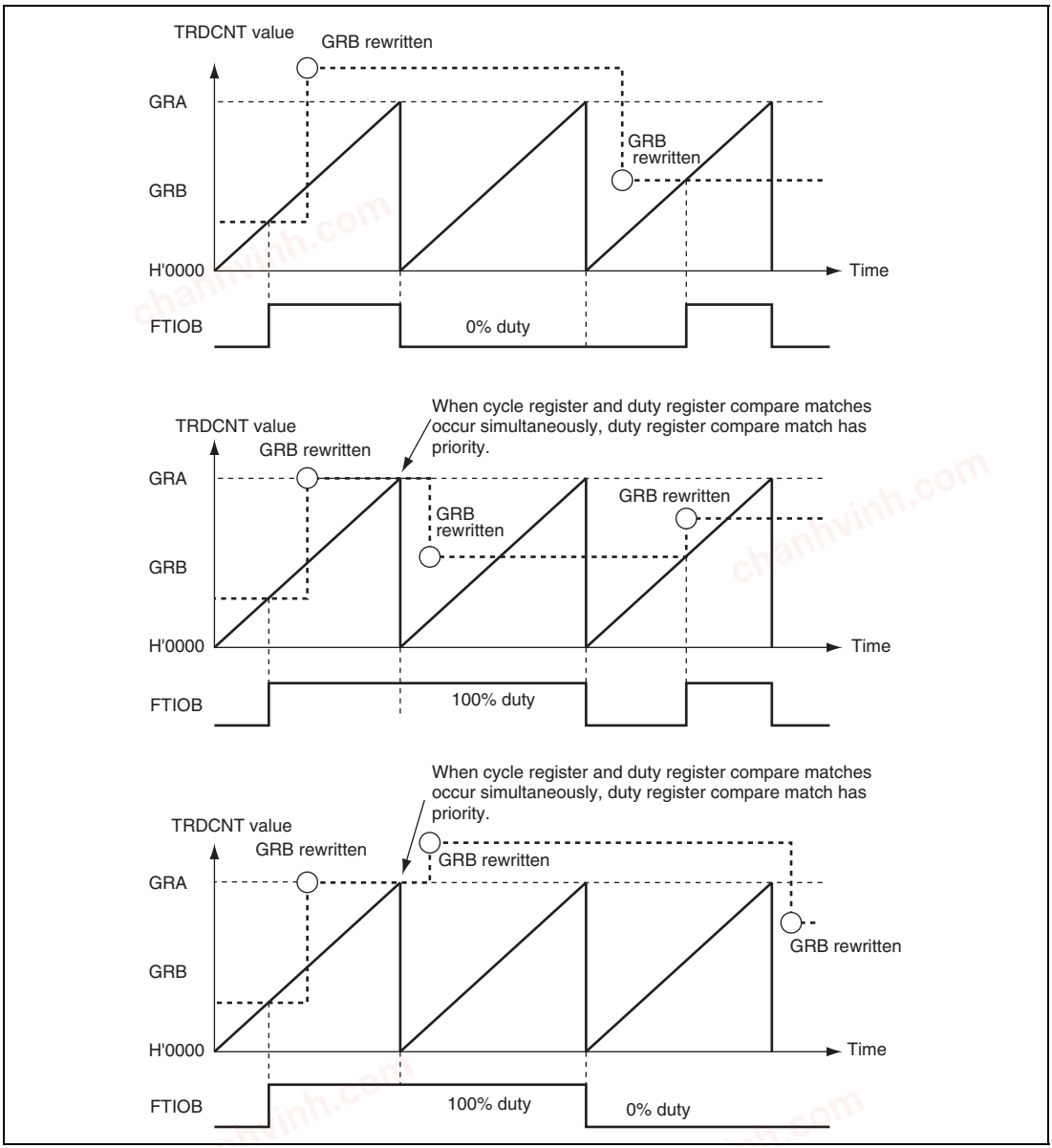



**Figure 16.25 Example of PWM Mode Operation (4)**

chanhvinh.com



### 16.3.6 Reset Synchronous PWM Mode

Three normal- and counter-phase PWM waveforms are output by combining channels 0 and 1 that one of changing points of waveforms will be common.

In reset synchronous PWM mode, the FTIOB0 to FTIOD0 and FTIOA1 to FTIOD1 pins become PWM-output pins automatically. TRDCNT_0 performs an increment operation. Tables 16.5 and 16.6 show the PWM-output pins used and the register settings, respectively.

Figure 16.29 shows the example of reset synchronous PWM mode setting procedure.

**Table 16.5 Output Pins in Reset Synchronous PWM Mode**

| Channel | Pin Name | Input/Output | Pin Function |
|---|---|---|---|
| 0 | FTIOC0 | Output | Toggle output in synchronous with PWM cycle |
| 0 | FTIOB0 | Output | PWM output 1 |
| 0 | FTIOD0 | Output | PWM output 1 (counter-phase waveform of PWM output 1) |
| 1 | FTIOA1 | Output | PWM output 2 |
| 1 | FTIOC1 | Output | PWM output 2 (counter-phase waveform of PWM output 2) |
| 1 | FTIOB1 | Output | PWM output 3 |
| 1 | FTIOD1 | Output | PWM output 3 (counter-phase waveform of PWM output 3) |

**Table 16.6 Register Settings in Reset Synchronous PWM Mode**

| Register | Description |
|---|---|
| TRDCNT_0 | Initial setting of H'0000 |
| TRDCNT_1 | Not used (independently operates) |
| GRA_0 | Sets counter cycle of TRDCNT_0 |
| GRB_0 | Set a changing point of the PWM waveform output from pins FTIOB0 and FTIOD0. |
| GRA_1 | Set a changing point of the PWM waveform output from pins FTIOA1 and FTIOC1. |
| GRB_1 | Set a changing point of the PWM waveform output from pins FTIOB1 and FTIOD1. |

chanhvinh.com



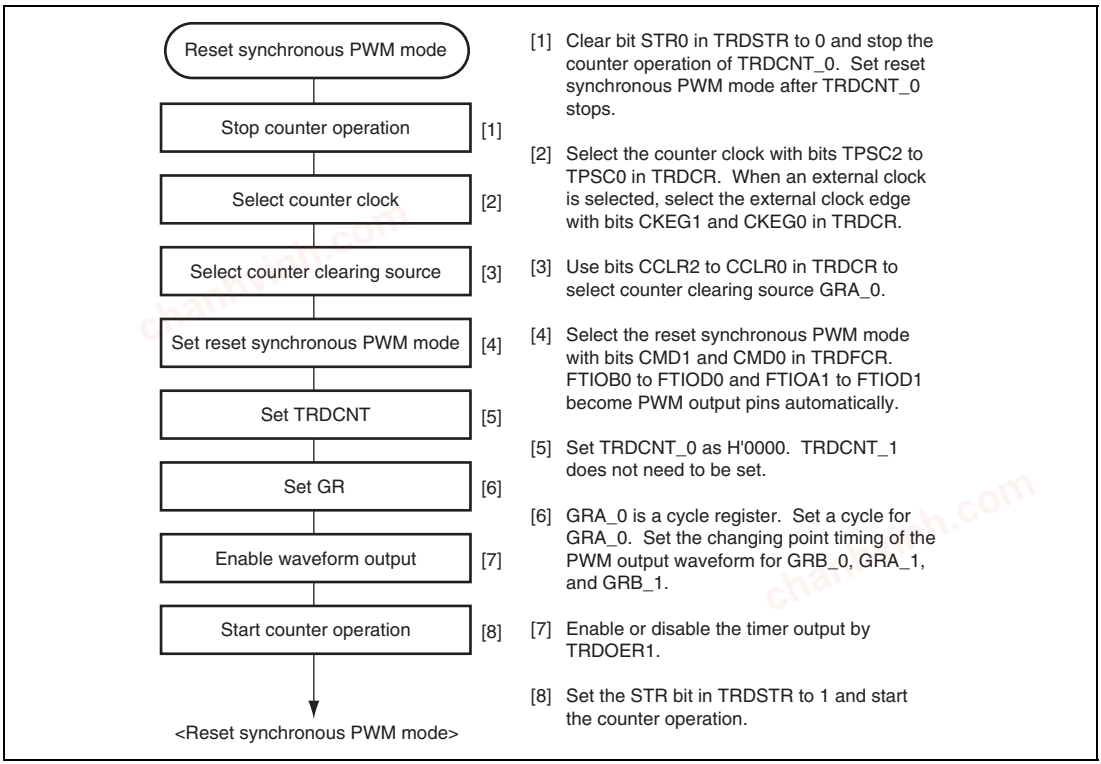



**Figure 16.26 Example of Reset Synchronous PWM Mode Setting Procedure**

chanhvinh.com



Figures 16.27 and 16.28 show examples of operation in reset synchronous PWM mode.

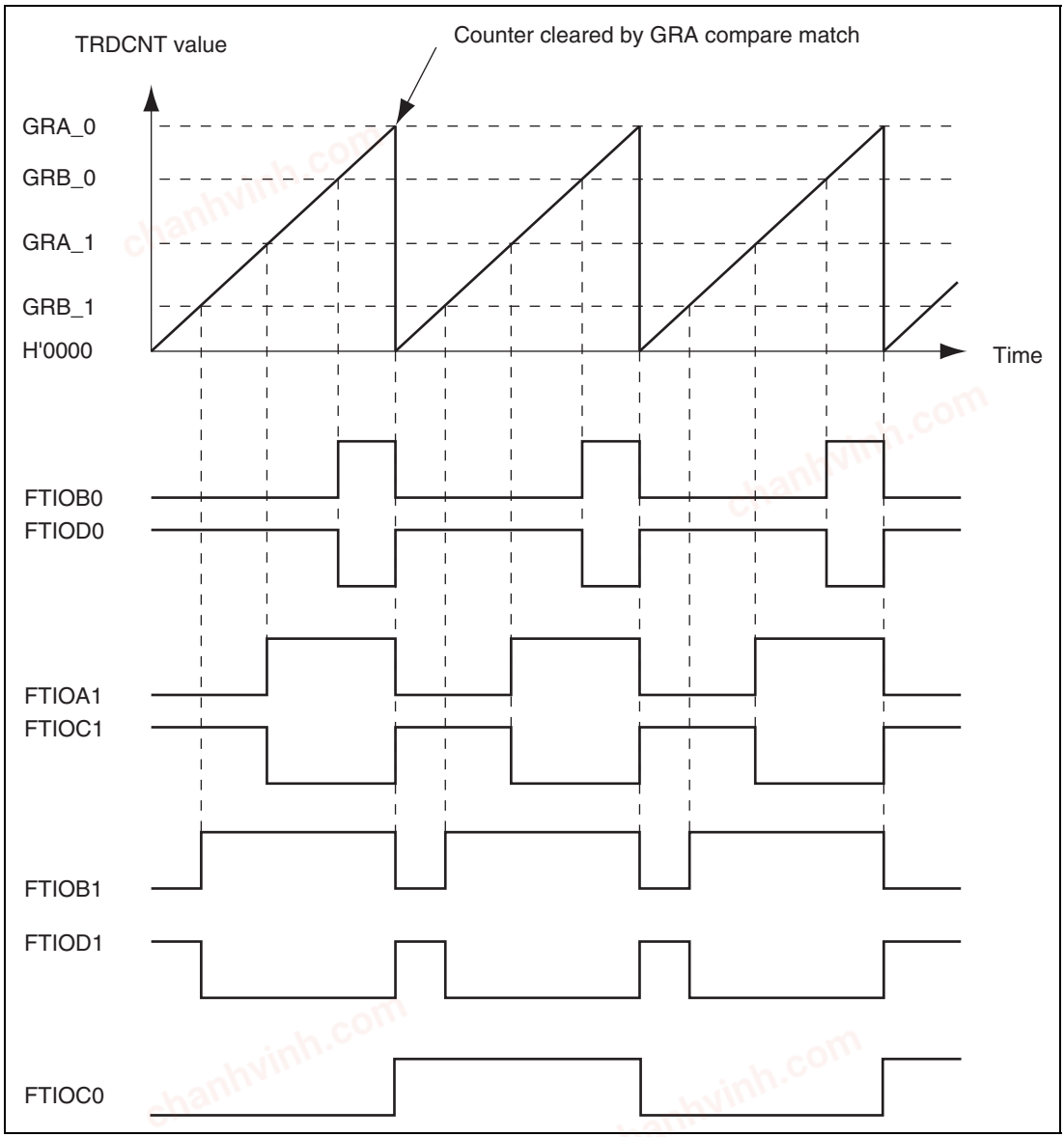



**Figure 16.27 Example of Reset Synchronous PWM Mode Operation (OLS0 = OLS1 = 1)**

chanhvinh.com



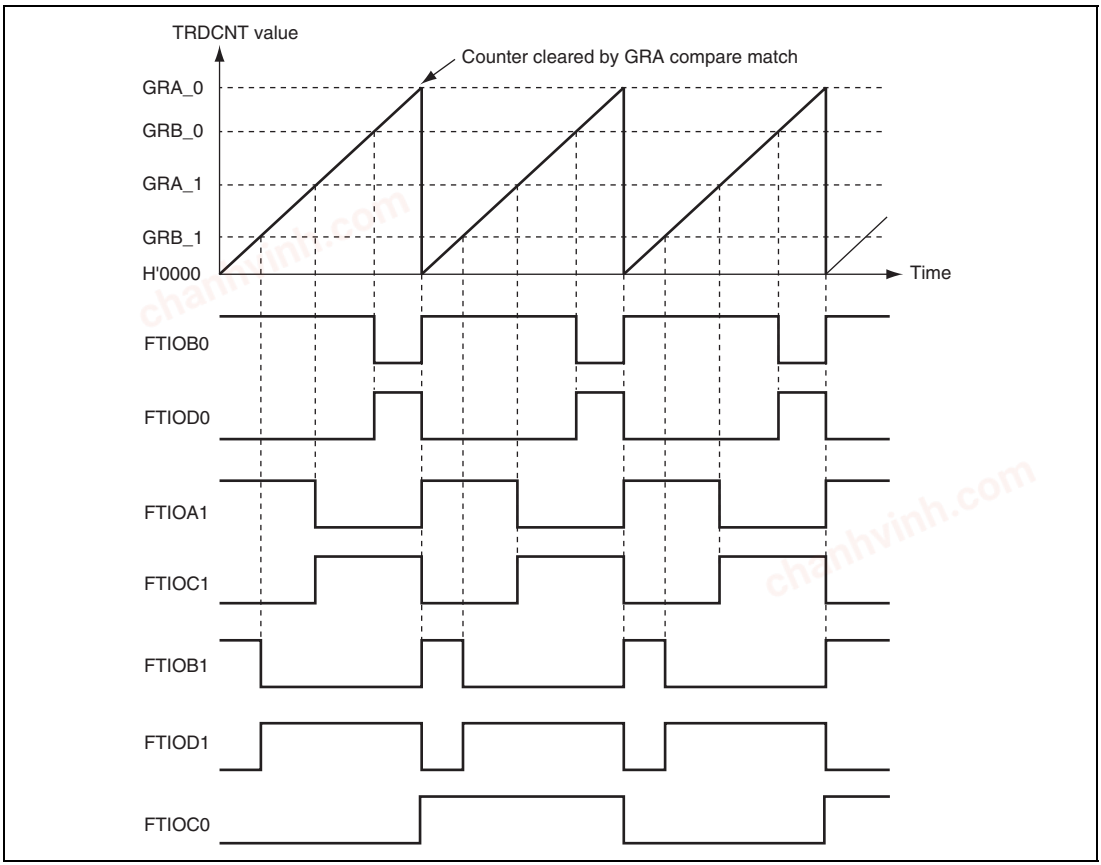



**Figure 16.28 Example of Reset Synchronous PWM Mode Operation (OLS0 = OLS1 = 0)**

In reset synchronous PWM mode, TRDCNT_0 and TRDCNT_1 perform increment and independent operations, respectively. However, GRA_1 and GRB_1 are separated from TRDCNT_1. When a compare match occurs between TRDCNT_0 and GRA_0, a counter is cleared and an increment operation is restarted from H'0000.

The PWM pin outputs 0 or 1 whenever a compare match between GRB_0, GRA_1, GRB_1 and TRDCNT_0 or counter clearing occur.

For details on operations when reset synchronous PWM mode and buffer operation are simultaneously set, see section 16.3.9, Buffer Operation.

chanhvinh.com



### 16.3.7 Complementary PWM Mode

Three PWM waveforms for non-overlapped normal and counter phases are output by combining channels 0 and 1.

In complementary PWM mode, the FTIOB0 to FTIOD0 and FTIOA1 to FTIOD1 pins become PWM-output pins automatically. TRDCNT_0 and TRDCNT_1 perform an increment or decrement operation. Tables 16.7 and 16.8 show the output pins and register settings in complementary PWM mode, respectively.

Figure 16.29 shows the example of complementary PWM mode setting procedure.

**Table 16.7 Output Pins in Complementary PWM Mode**

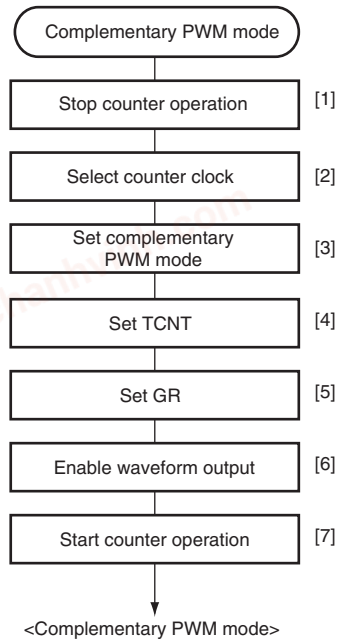
| Channel | Pin Name | Input/Output | Pin Function |
|---|---|---|---|
| 0 | FTIOC0 | Output | Toggle output in synchronous with PWM cycle |
| 0 | FTIOB0 | Output | PWM output 1 |
| 0 | FTIOD0 | Output | PWM output 1 (counter-phase waveform non-overlapped with PWM output 1) |
| 1 | FTIOA1 | Output | PWM output 2 |
| 1 | FTIOC1 | Output | PWM output 2 (counter-phase waveform non-overlapped with PWM output 2) |
| 1 | FTIOB1 | Output | PWM output 3 |
| 1 | FTIOD1 | Output | PWM output 3 (counter-phase waveform non-overlapped with PWM output 3) |

**Table 16.8 Register Settings in Complementary PWM Mode**

| Register | Description |
|---|---|
| TRDCNT_0 | Initial setting of non-overlapped periods (non-overlapped periods are differences with TRDCNT_1) |
| TRDCNT_1 | Initial setting of H'0000 |
| GRA_0 | Sets (upper limit value – 1) of TRDCNT_0 |
| GRB_0 | Set a changing point of the PWM waveform output from pins FTIOB0 and FTIOD0. |
| GRA_1 | Set a changing point of the PWM waveform output from pins FTIOA1 and FTIOC1. |
| GRB_1 | Set a changing point of the PWM waveform output from pins FTIOB1 and FTIOD1. |

chanhvinh.com






[1] Clear bits STR0 and STR1 in TRDSTR to 0, and stop the counter operation of TRDCNT_0. Stop TRDCNT_0 and TRDCNT_1 and set complementary PWM mode.

[2] Use bits TPSC2 to TPSC0 in TRDCR to select the same counter clock for channels 0 and 1. When an external clock is selected, select the edge of the external clock by bits CKEG1 and CKEG0 in TRDCR. Set bits CCLR2 to CCLR0 in TRDCR so that the counter is not cleared.

[3] Use bits CMD1 and CMD0 in TRDFCR to set complementary PWM mode. FTIOB0 to FTIOD0 and FTIOA1 to FTIOD1 automatically become PWM output pins.

[4] TRDCNT_1 must be H'0000. Set a non-overlapped period to TRDCNT_0.

[5] GRA_0 is a cycle register. Set the cycle to GRA_0. Set the timing to change the PWM output waveform to GRB_0, GRA_1, and GRB_1. The settings must be set so that a compare match occurs on TRDCNT_0 and TRDCNT_1. $T \leq X$ (X: Initial value in GRB_0, GRA_1, or GRB_1).

[6] Use TRDOER1 to enable or disable the timer output.

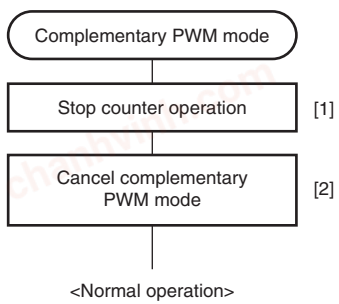
[7] Set the STR0 and STR1 bits in TRDSTR to 1 to start the count operation.

Note: To modify the settings for the complementary PWM mode, clear settings other than those for the mode. After that, repeat setting from step [1].

**Figure 16.29 Example of Complementary PWM Mode Setting Procedure**

### (1) Canceling Procedure of Complementary PWM Mode

Figure 16.30 shows the complementary PWM mode canceling procedure.




[1] Clear bit CMD1 in TRDFCR to 0, and set channels 0 and 1 to normal operation.

[2] After setting channels 0 and 1 to normal operation, clear bits STR0 and STR1 in TRDSTR to 0 and stop TRDCNT_0 and TRDCNT_1.

**Figure 16.30 Canceling Procedure of Complementary PWM Mode**

chanhvinh.com



### (2) Examples of Complementary PWM Mode Operation

Figure 16.31 shows an example of complementary PWM mode operation. In complementary PWM mode, TRDCNT_0 and TRDCNT_1 perform an increment or decrement operation. When TRDCNT_0 and GRA_0 are compared and their contents match, the counter is decremented. And when TRDCNT_1 underflows, the counter is incremented. In GRA_0, GRA_1, and GRB_1, compare match is carried out in the order of TRDCNT_0 → TRDCNT_1 → TRDCNT_1 → TRDCNT_0 and PWM waveform is output, during one cycle of an up/down counter. In this mode, the initial setting will be TRDCNT_0 > TRDCNT_1.

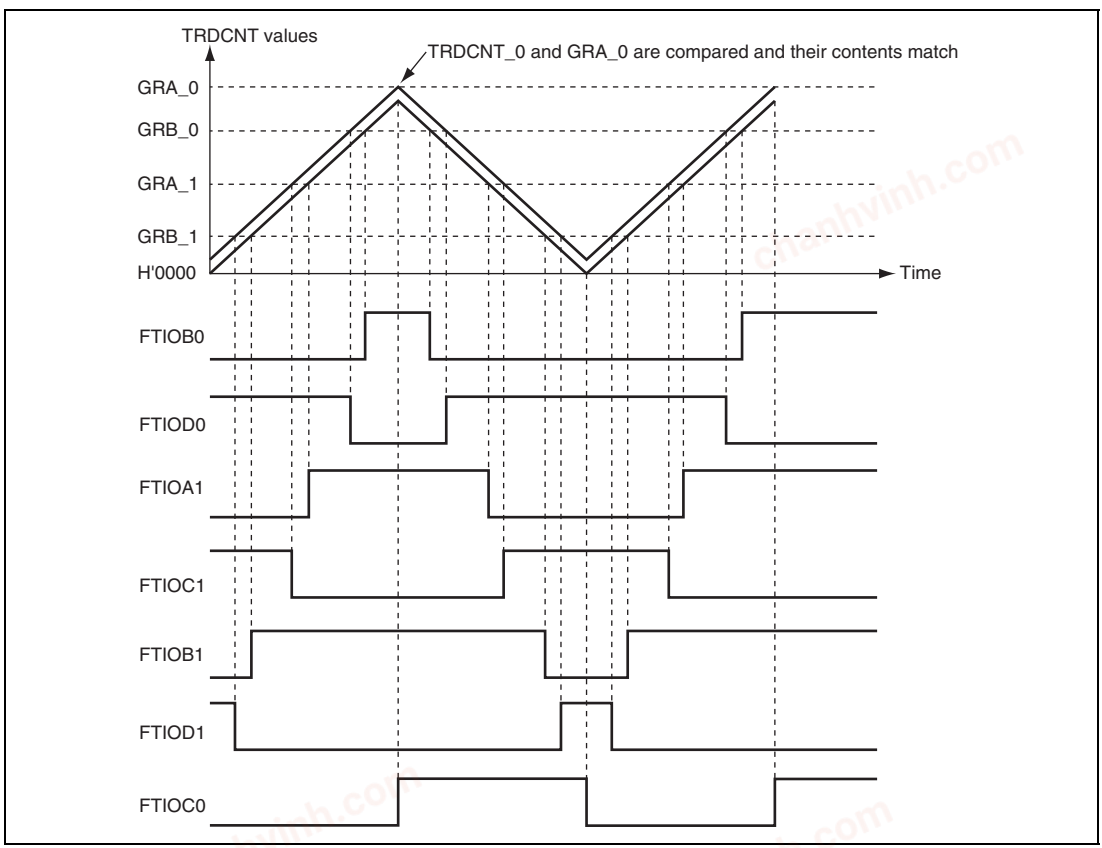



**Figure 16.31 Example of Complementary PWM Mode Operation (1)**

chanhvinh.com



Figure 16.32 shows an example of PWM waveform output with 0% duty and 100% duty in complementary PWM mode (for one phase).

In this figure, GRB_0 is set to a value equal to or greater than GRA_0 and H'0000. The waveform with a duty cycle of 0% and 100% can be output. When buffer operation is used together, the duty cycles can easily be changed, including the above settings, during operation. For details on buffer operation, see section 16.3.9, Buffer Operation.

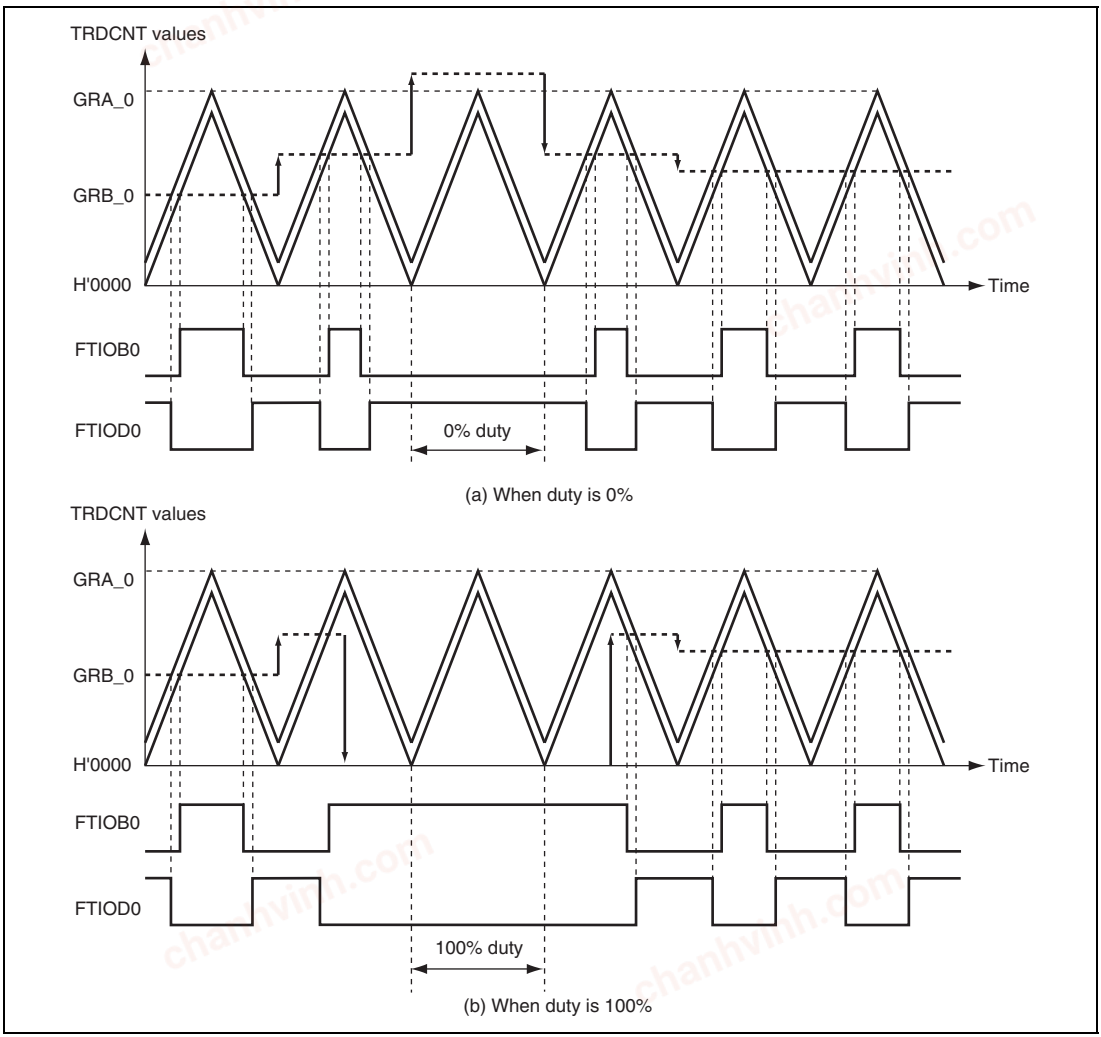



**Figure 16.32 Example of Complementary PWM Mode Operation (2)**

chanhvinh.com



In complementary PWM mode, when the counter switches from up-counter to down-counter or vice versa, TRDCNT_0 and TRDCNT_1 overshoots or undershoots, respectively. In this case, the conditions to set the IMFA flag in channel 0 and the UDF flag in channel 1 differ from usual settings. Also, the transfer conditions in buffer operation differ from usual settings. Such timings are shown in figures 16.33 and 16.34.

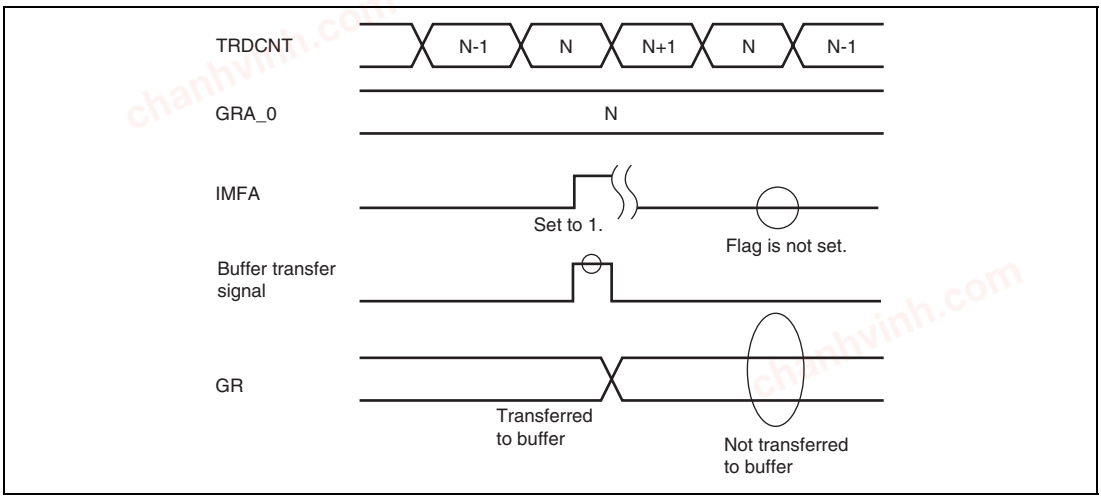



**Figure 16.33 Timing of Overshooting**

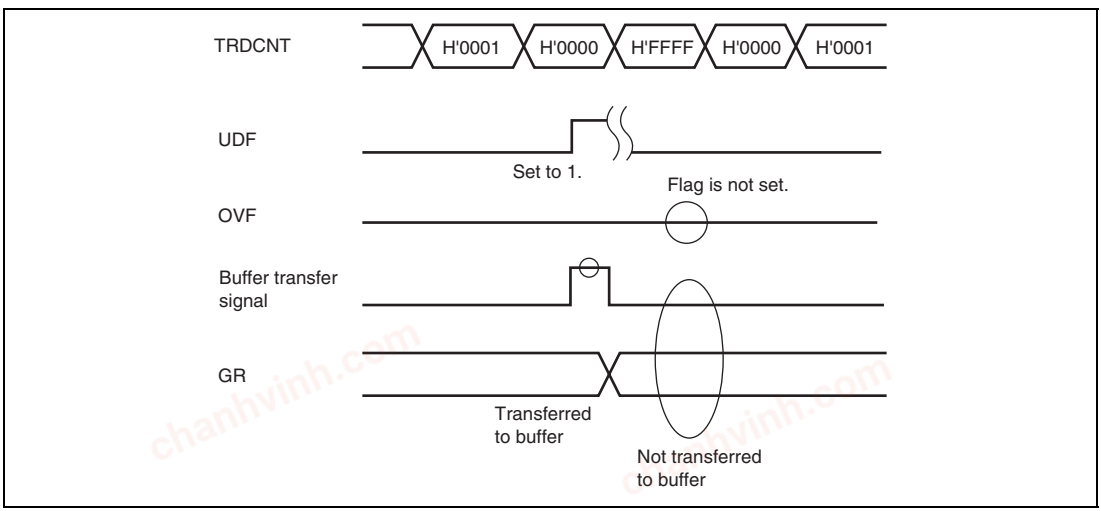



**Figure 16.34 Timing of Undershooting**

chanhvinh.com



When the counter is incremented or decremented, the IMFA flag of channel 0 is set to 1, and when the register is underflowed, the UDF flag of channel 0 is set to 1. After buffer operation has been designated for GR, the value in the buffer registers is transferred to GR when the counter is incremented by compare match A0 or when TRDCNT_1 is underflowed. In complementary PWM mode, the OVF flag is not set to 1 at the timing that the counter value changes from H'FFFF to H'0000 as shown in figure 16.34.

### (3) Setting GR Value in Complementary PWM Mode

To set the general register (GR) or modify GR during operation in complementary PWM mode, see the following notes.

1. Initial value
   - H'0000 to T – 1 (T: Initial value of TRDCNT_0) must not be set for the initial value.
   - GRA_0 – (T – 1) or more must not be set for the initial value.
   - When using buffer operation, the same values must be set in the buffer registers and corresponding general registers.
2. Modifying the setting value
   - Use the buffer operation to change the GR value. If the GR value is changed by writing to it directly, the intended waveform may not be output.
   - Do not change settings of GRA_0 during operation.

chanhvinh.com



### 16.3.8 PWM3 Mode Operation

In PWM3 mode, single-phase PWM waveforms can be output using TRDCNT_0. The waveform does not overlap its counter-phase waveform.

When the PWM3 mode is selected, the FTIOA0 and FTIOB0 pins are automatically set to output pins for the PWM function using TRDCNT_0 regardless of the TRDPMR value. The waveform is output on a GRA_0, GRA_1, GRB_0, or GRB_1 compare match according to bits TOA0 and TOB0 in TRDOCR regardless of the TRDIORA and TRDIORC settings.

- When TOA0 = 0, 1 is output on a compare match of GRA_1 and 0 is output on a compare match of GRA_0 on the FTIOA0 pin.
- When TOA0 = 1, 0 is output on a compare match of GRA_1 and 1 is output on a compare match of GRA_0 on the FTIOA0 pin.
- When TOB0 = 0, 1 is output on a compare match of GRB_1 and 0 is output on a compare match of GRB_0 on the FTIOB0 pin.
- When TOB0 = 1, 0 is output on a compare match of GRB_1 and 1 is output on a compare match of GRB_0 on the FTIOB0 pin.

Table 16.9 lists the correspondence between pin functions and GR registers, figure 16.35 shows a block diagram in PWM3 mode, and figure 16.36 shows a flowchart of setting in PWM3 mode.

When the buffer operation is used, set TRDMDR. The timer input/output pins, which are not used in PWM3 mode, can be used as general port pins. When the buffer operation is not set, since GRC or GRD is not used, a compare match interrupt can be generated when GRC or GRD matches with TRDCNT_1.

chanhvinh.com



**Table 16.9 Pin Configuration in PWM3 Mode and GR Registers**

| Channel | Pin Name | Input/Output | Compare Match Register | Buffer Register |
|---|---|---|---|---|
| 0 | FTIOA0 | Output | GRA_0 | GRC_0 |
| | | | GRA_1 | GRC_1 |
| | FTIOB0 | | GRB_0 | GRD_0 |
| | | | GRB_1 | GRD_1 |
| | FTIOC0 | I/O | General I/O port | General I/O port |
| | FTIOD0 | | | |
| 1 | FTIOA1 | | | |
| | FTIOB1 | | | |
| | FTIOC1 | | | |
| | FTIOD1 | | | |

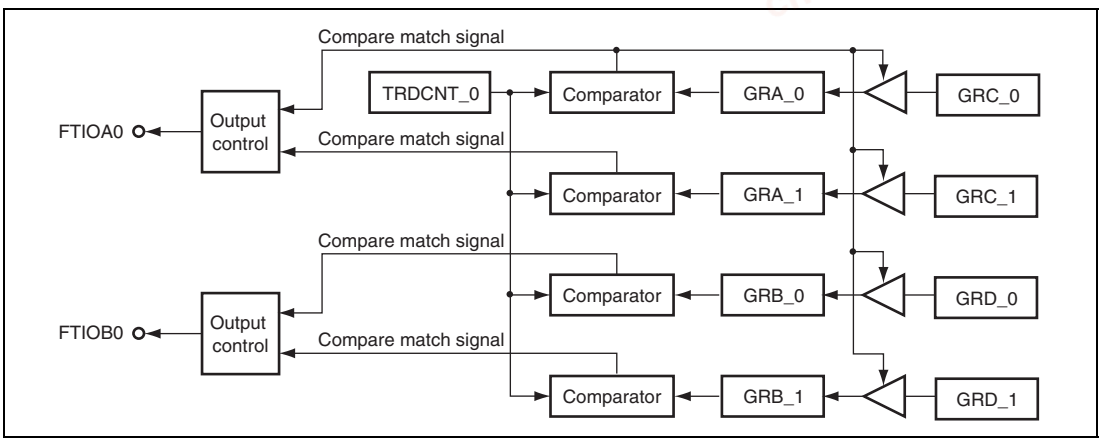
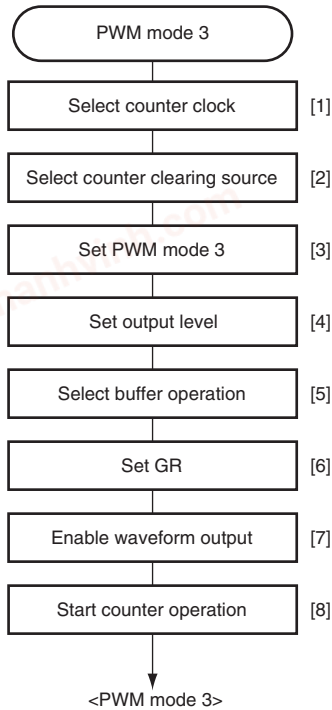



**Figure 16.35 Block Diagram in PWM3 Mode**

chanhvinh.com






[1] Select the counter clock with bits TPSC2 to TPSC0 in TRDCR. When an external clock is selected, select the external clock edge with bits CKEG1 and CKEG0 in TRDCR.

[2] Use bits CCLR2 to CCLR0 in TRDCR to select counter clearing source GRA_0.

[3] Select PWM mode 3 with bit PWM3 in TRDFCR.

[4] Set output levels with bits TOB0 and TOA0 in TRDOCR.

[5] Set the GR buffer operation with bits BFC0, BFC1, BFD0, and BFD1 in TRDMDR.

[6] Set a cycle in GRA. Set the duty cycle in other GR registers.

[7] Enable or disable the timer output by TRDOER1.

[8] Set the STR bit in TRDSTR to 1 and start the counter operation.

**Figure 16.36 Flowchart of Setting in PWM3 Mode**

chanhvinh.com



Figure 16.37 is an example when non-overlapped pulses are output on pins FTIOA0 and FTIOB0. In this example, TRDCNT_0 functions as a periodic counter which is cleared on compare match A0 (bits CCLR2 to CCLR0 in TRDCR_0 are set to B'001), and PWM3 mode is selected (bit PWM3 in TRDFCR is cleared to 0). The cycle of the pulse is arbitrary.

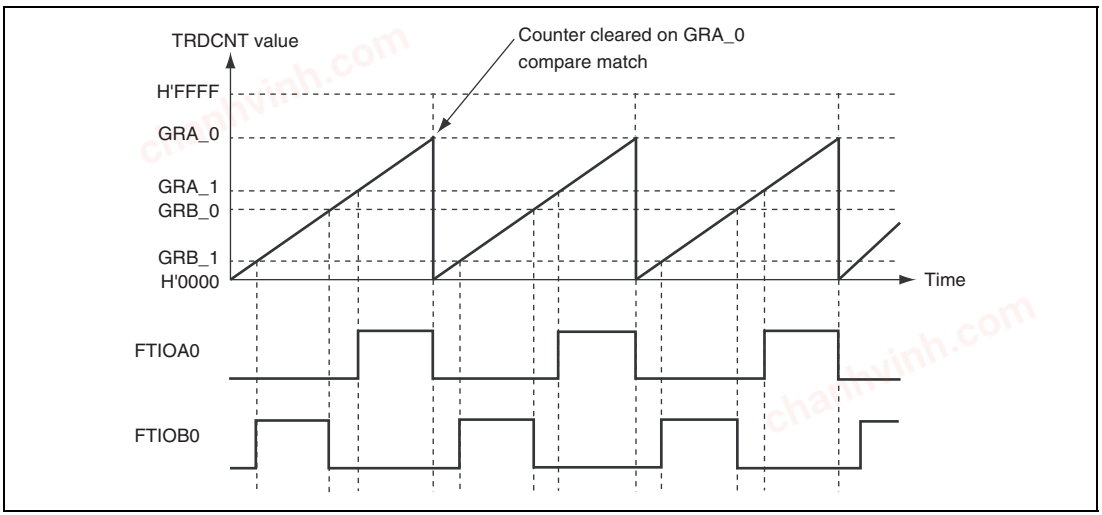



**Figure 16.37 Example of Non-Overlap Pulses**

chanhvinh.com



Figures 16.38 and 16.39 show examples of stopping operation of the counter in PWM3 mode, when the CCLR2 to CCLR0 bits in TRDCR are set to clear TRDCNT_0 on GRA_0 compare match. For details on PWM3 mode, see section 16.3.8, PWM3 Mode Operation.

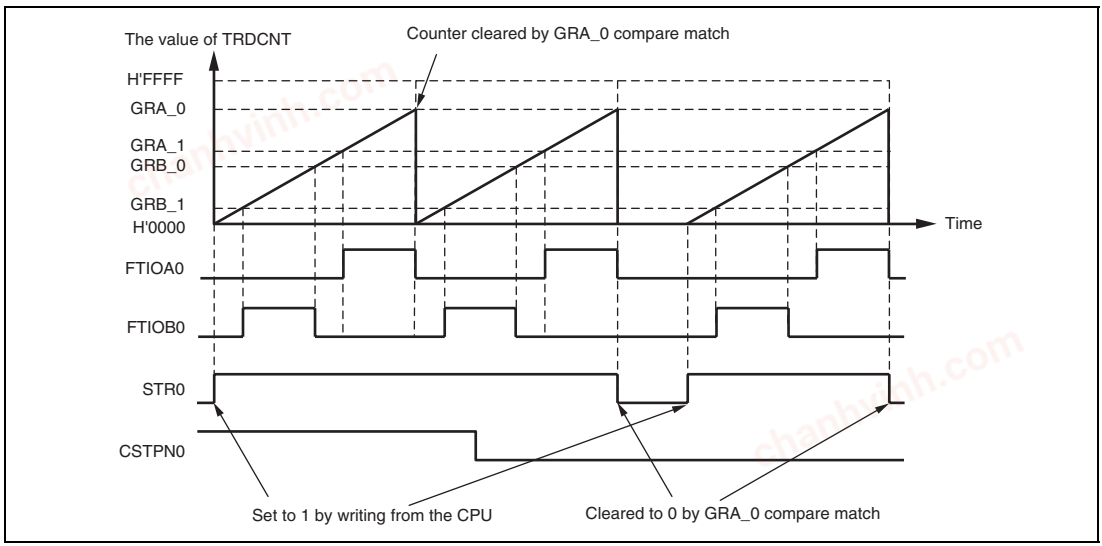



**Figure 16.38 Example (1) of Stopping Operation of the Counter (in PWM3 Mode)**

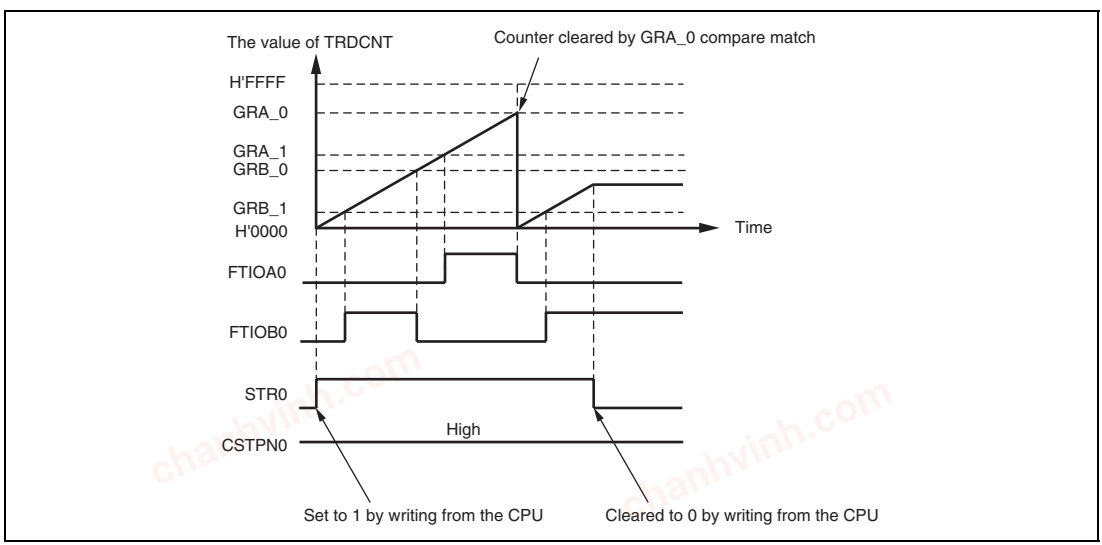



**Figure 16.39 Example (2) of Stopping Operation of the Counter (in PWM3 Mode)**

chanhvinh.com



Figure 16.40 shows an example of starting and stopping operations of counters in PWM3 mode, when TRDCN1_0 is set to be cleared and stopped on GRA_0 compare match (CCLR2 to CCLR0 = 001, CSTPN0 = 0) and TRDCNT_1 is used as a free-running counter. When TRDCNT_1 starts counting by setting the STR1 bit to 1 after TRDCNT_0 has started counting by setting the STR0 bit to 1, set 0 in the STR0 bit and 1 in the STR1 bit by using a MOV instruction. If the bit manipulation instruction is used to set 1 in the STR1 bit, there is a possibility that the STR0 bit is set to 1 after the counting has stopped on GRA_0 compare match, and that TRDCNT_0 starts counting again.

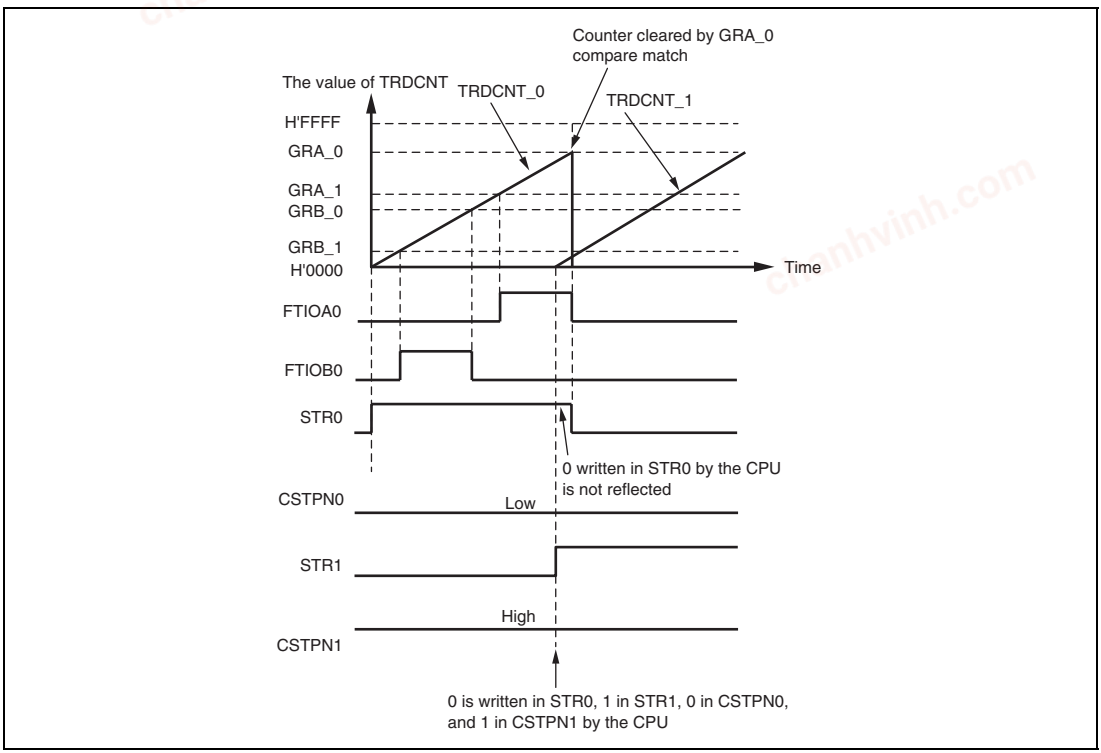



**Figure 16.40 Example of Starting and Stopping Operations of Counters (in PWM3 Mode)**

chanhvinh.com



### 16.3.9 Buffer Operation

Buffer operation differs depending on whether GR has been designated for an input capture register or an output compare register, or in reset synchronous PWM mode or complementary PWM mode.

Table 16.10 shows the register combinations used in buffer operation.

**Table 16.10 Register Combinations in Buffer Operation**

| General Register (GR) | Buffer Register |
|---|---|
| GRA | GRC |
| GRB | GRD |

#### (1) When GR is an Output Compare Register

When a compare match occurs, the value in the buffer register of the corresponding channel is transferred to the general register.

This operation is illustrated in figure 16.41.

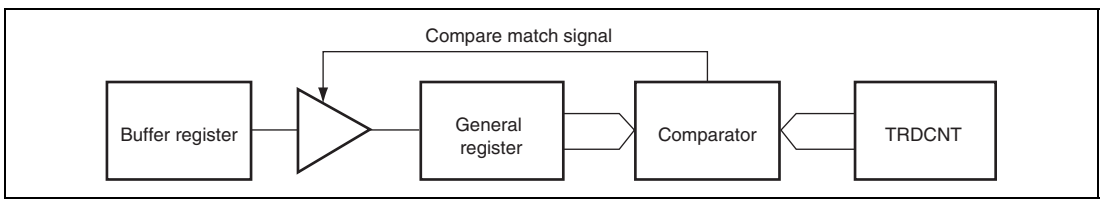



**Figure 16.41 Compare Match Buffer Operation**

chanhvinh.com



### (2) When GR is an Input Capture Register

When an input capture occurs, the value in TRDCNT is transferred to GR and the value previously stored in the general register is transferred to the buffer register.

This operation is illustrated in figure 16.42.

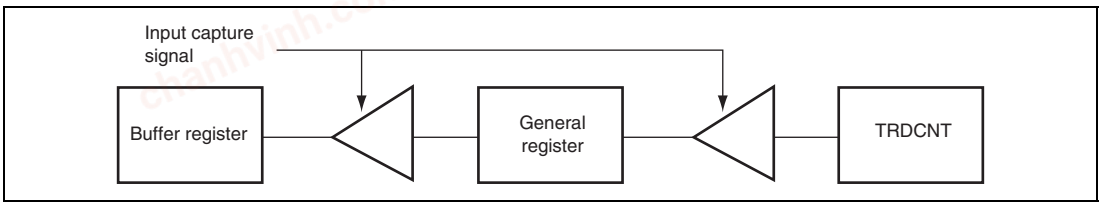



**Figure 16.42 Input Capture Buffer Operation**

### (3) PWM3 Mode

When compare match A0 occurs, the value of the buffer register is transferred to GR.

### (4) Complementary PWM Mode

When the counter switches from counting up to counting down or vice versa, the value of the buffer register is transferred to GR. Here, the value of the buffer register is transferred to GR in the following timing:

- When TRDCNT_0 and GRA_0 are compared and their contents match
- When TRDCNT_1 underflows

### (5) Reset Synchronous PWM Mode

When compare match A0 occurs, the value in the buffer register is transferred to GR.

chanhvinh.com



### (6) Example of Buffer Operation Setting Procedure

Figure 16.43 shows an example of the buffer operation setting procedure.

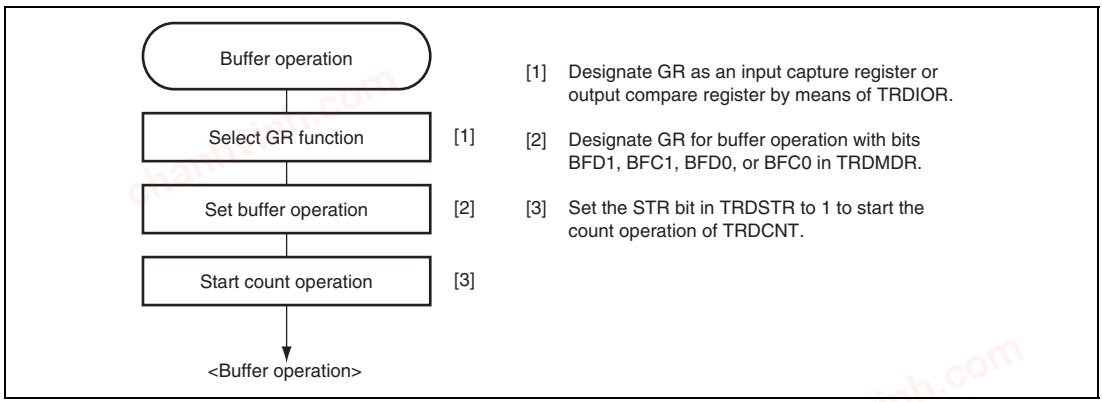



**Figure 16.43 Example of Buffer Operation Setting Procedure**

### (7) Examples of Buffer Operation

Figure 16.44 shows an operation example in which GRA has been designated as an output compare register, and buffer operation has been designated for GRA and GRC.

This is an example of TRDCNT operating as a periodic counter cleared by compare match B.

Pins FTIOA and FTIOB are set for toggle output by compare match A and B.

As buffer operation has been set, when compare match A occurs, the FTIOA pin performs toggle outputs and the value in buffer register is simultaneously transferred to the general register. This operation is repeated each time that compare match A occurs.

chanhvinh.com



The timing to transfer data is shown in figure 16.45.

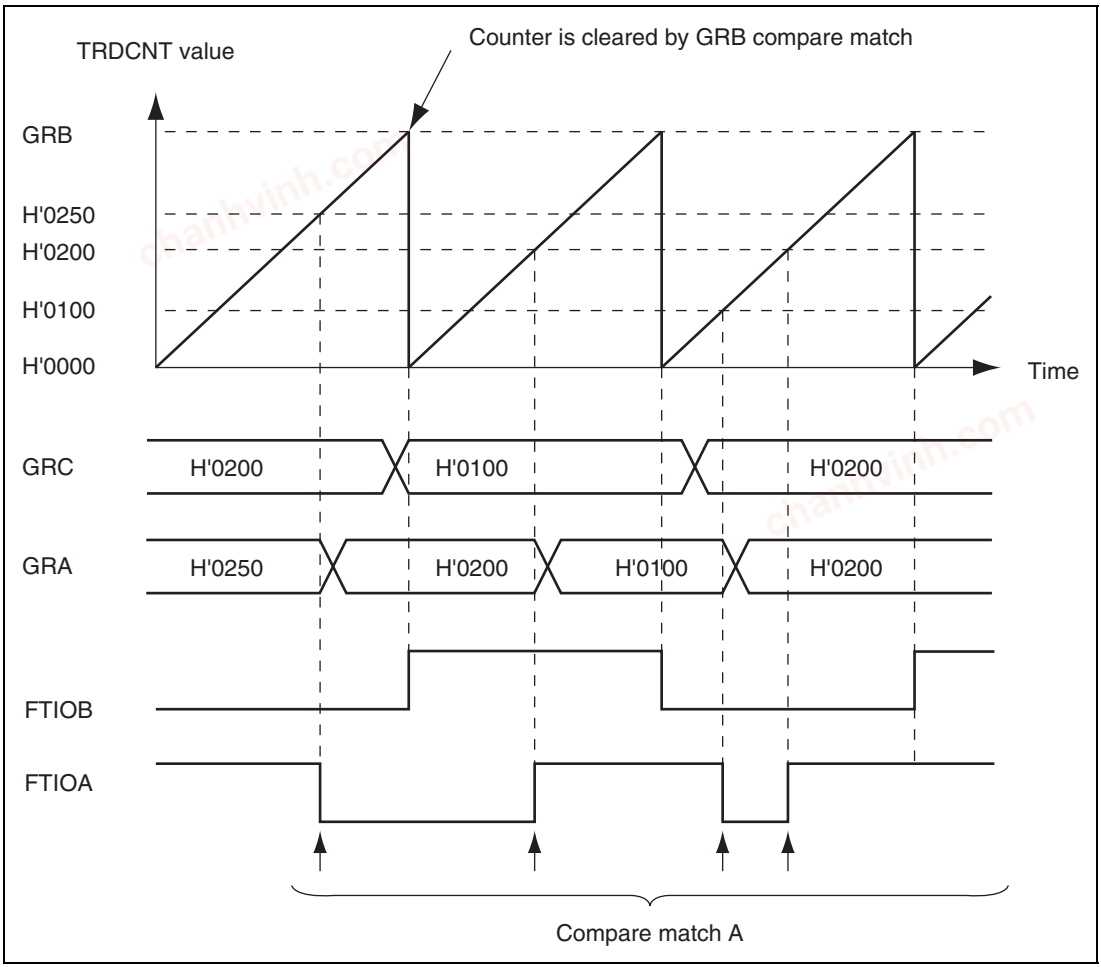



**Figure 16.44 Example of Buffer Operation (1)**
**(Buffer Operation for Output Compare Register)**

chanhvinh.com



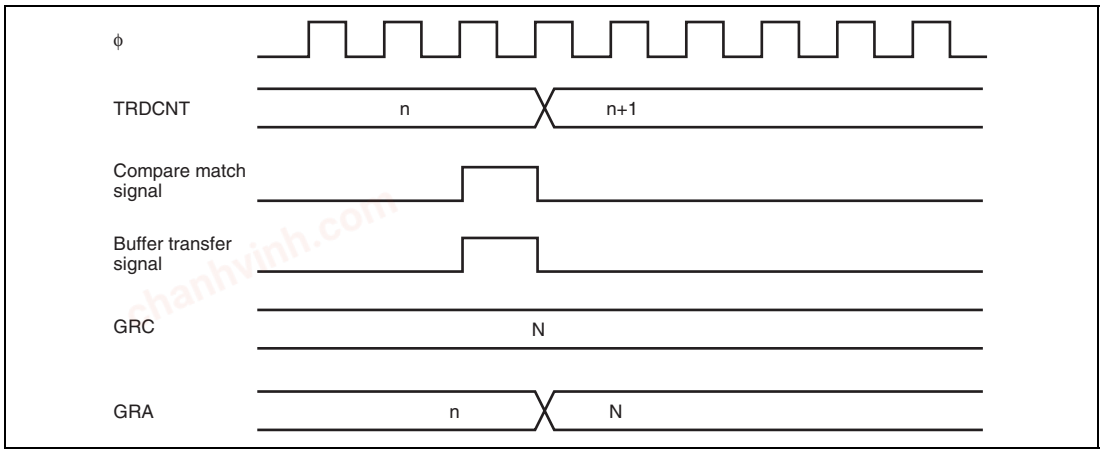



**Figure 16.45 Example of Compare Match Timing for Buffer Operation**

Figure 16.46 shows an operation example in which GRA has been designated as an input capture register, and buffer operation has been designated for GRA and GRC.

Counter clearing by input capture B has been set for TRDCNT, and falling edges have been selected as the FTIOB pin input capture input edge. And both rising and falling edges have been selected as the FTIOA pin input capture input edge.

As buffer operation has been set, when the TRDCNT value is stored in GRA upon the occurrence of input capture A, the value previously stored in GRA is simultaneously transferred to GRC. The transfer timing is shown in figure 16.47.

chanhvinh.com



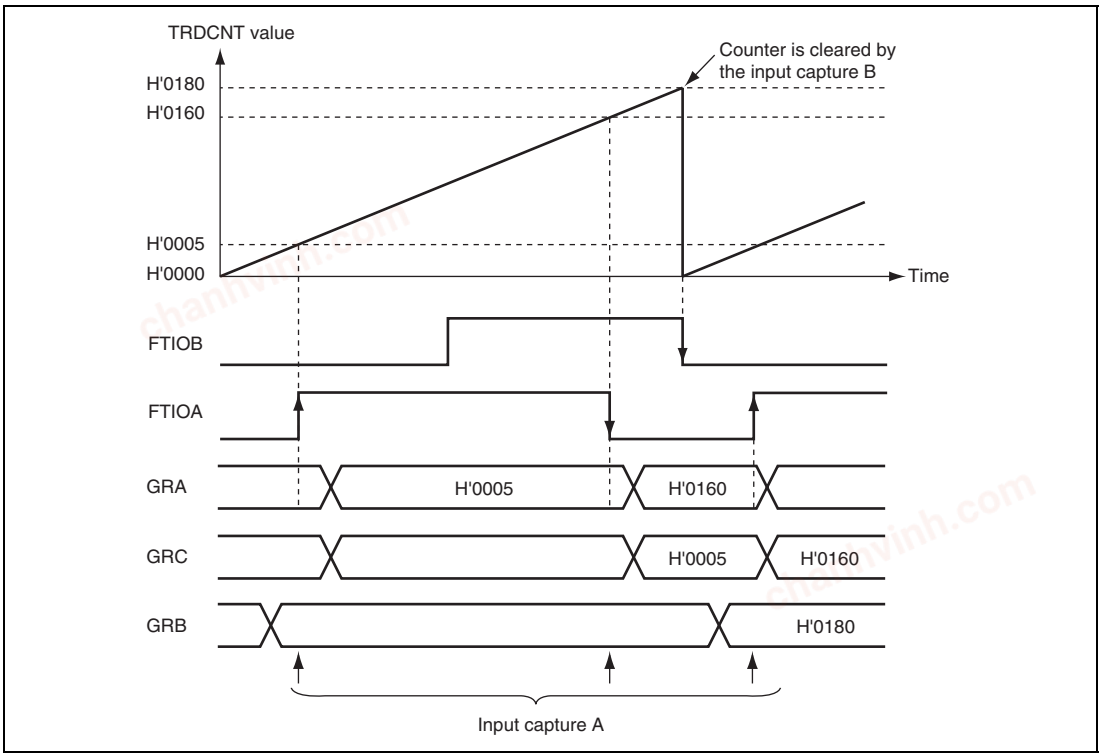



**Figure 16.46 Example of Buffer Operation (2)**
**(Buffer Operation for Input Capture Register)**

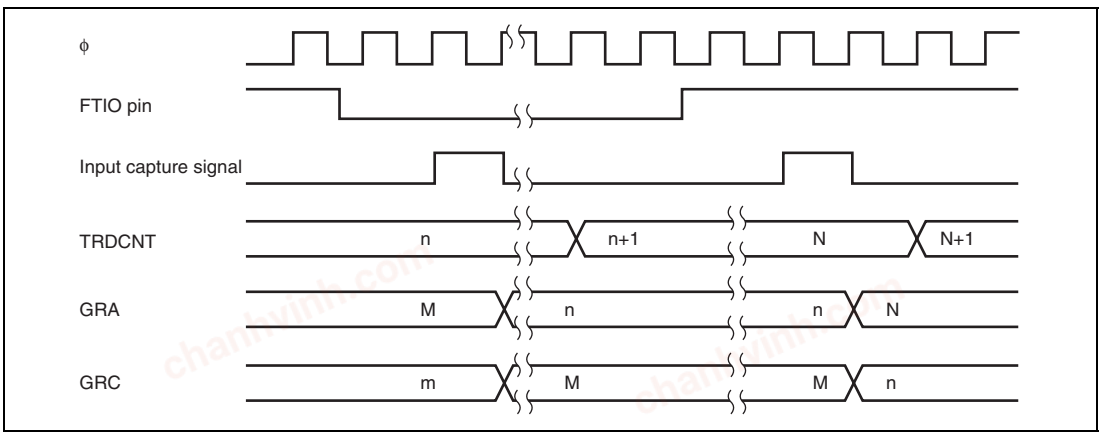



**Figure 16.47 Input Capture Timing of Buffer Operation**

chanhvinh.com



Figures 16.48 and 16.49 show the operation examples when buffer operation has been designated for GRB_0 and GRD_0 in complementary PWM mode. These are examples when a PWM waveform of 0% duty is created by using the buffer operation and performing GRD_0 ≥ GRA_0. Data is transferred from GRD_0 to GRB_0 according to the settings of CMD0 and CMD1 when TRDCNT_0 and GRA_0 are compared and their contents match or when TRDCNT_1 underflows. However, when GRD_0 ≥ GRA_0, data is transferred from GRD_0 to GRB_0 when TRDCNT_1 underflows regardless of the setting of CMD0 and CMD1. When GRD_0 = H'0000, data is transferred from GRD_0 to GRB_0 when TRDCNT_0 and GRA_0 are compared and their contents match regardless of the settings of CMD0 and CMD1.

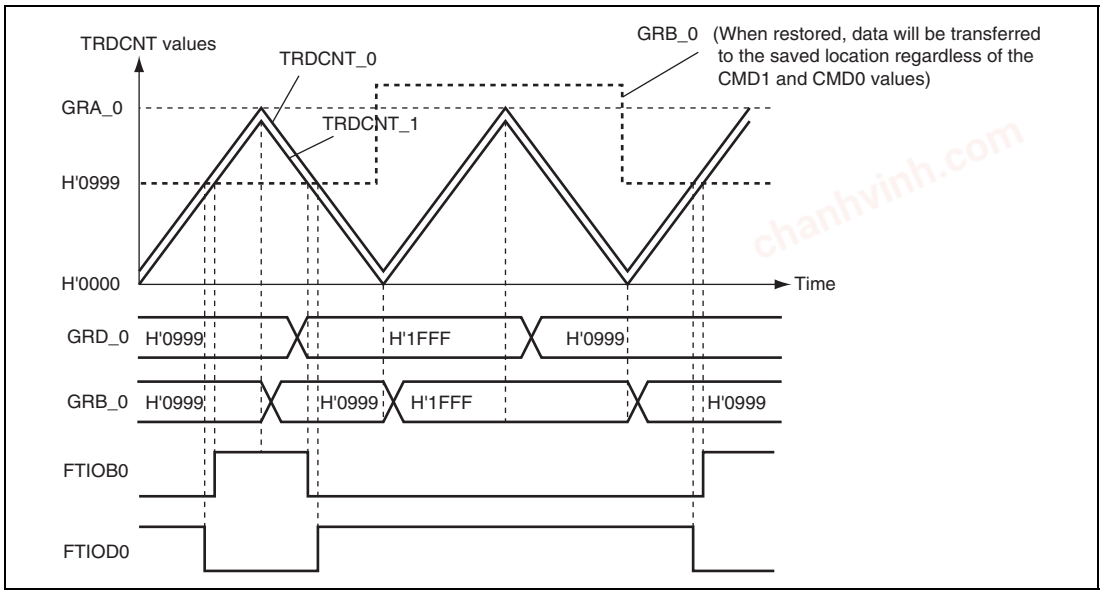



**Figure 16.48 Buffer Operation (3)**
**(Buffer Operation in Complementary PWM Mode CMD1 = CMD0 = 1)**

chanhvinh.com



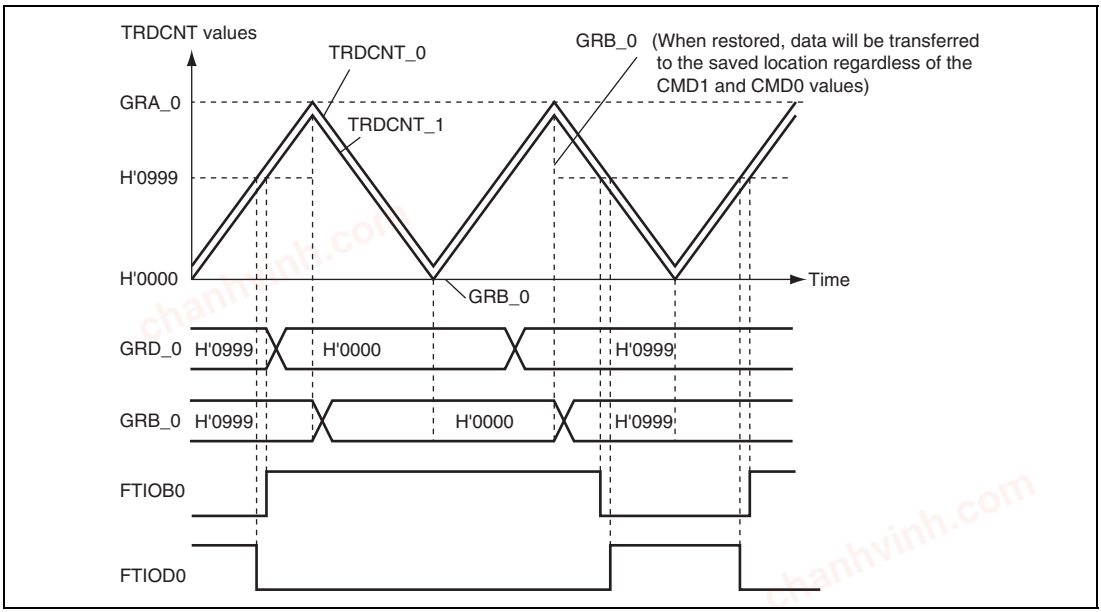



**Figure 16.49 Buffer Operation (4)**
**(Buffer Operation in Complementary PWM Mode CMD1 =1, CMD0 = 0)**

chanhvinh.com



### 16.3.10 Timer RD Output Timing

The outputs of channels 0 and 1 can be disabled or inverted by the settings of TRDOER1 and TRDOCR and the external level.

#### (1) Output Disable/Enable Timing of Timer RD by TRDOER1

Setting the master enable bit in TRDOER1 to 1 disables the output of timer RD. By setting the PCR and PDR of the corresponding I/O port beforehand, any value can be output. Figure 16.50 shows the timing to enable or disable the output of timer RD by TRDOER1.

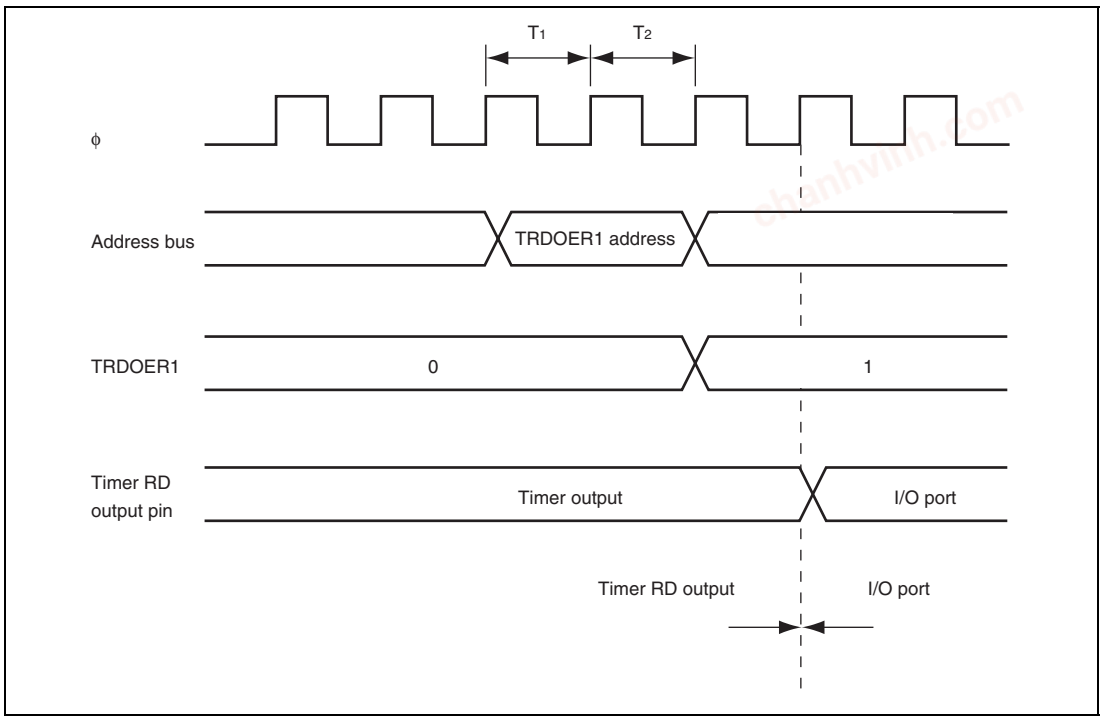



**Figure 16.50 Example of Output Disable Timing of Timer RD by Writing to TRDOER1**

chanhvinh.com



### (2) Output Disable Timing of Timer RD by External Trigger

When $\overline{\text{TRDOI}}$ is selected for inputs and low level is input to $\overline{\text{TRDOI}}$, the master enable bit in TRDOER1 is set to 1 and the output of timer RD will be disabled.

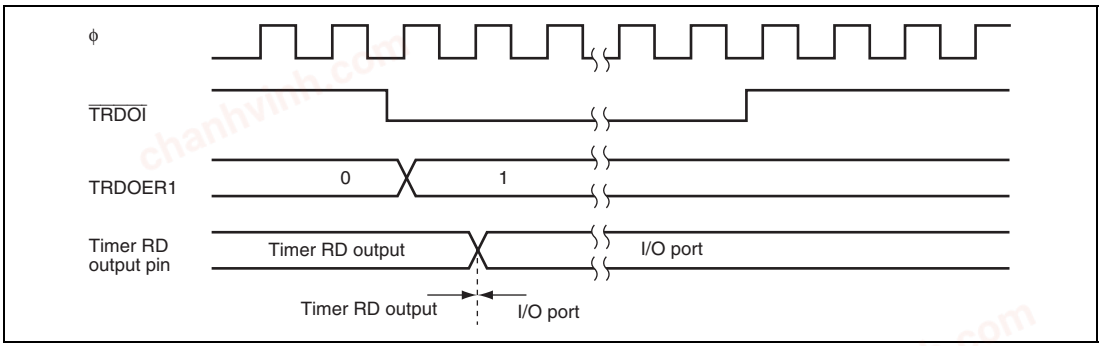



**Figure 16.51 Example of Output Disable Timing of Timer RD by External Trigger**

### (3) Output Inverse Timing by TRDFCR

The output level can be inverted by inverting the OLS1 and OLS0 bits in TRDFCR in reset synchronous PWM mode or complementary PWM mode. Figure 16.52 shows the timing.

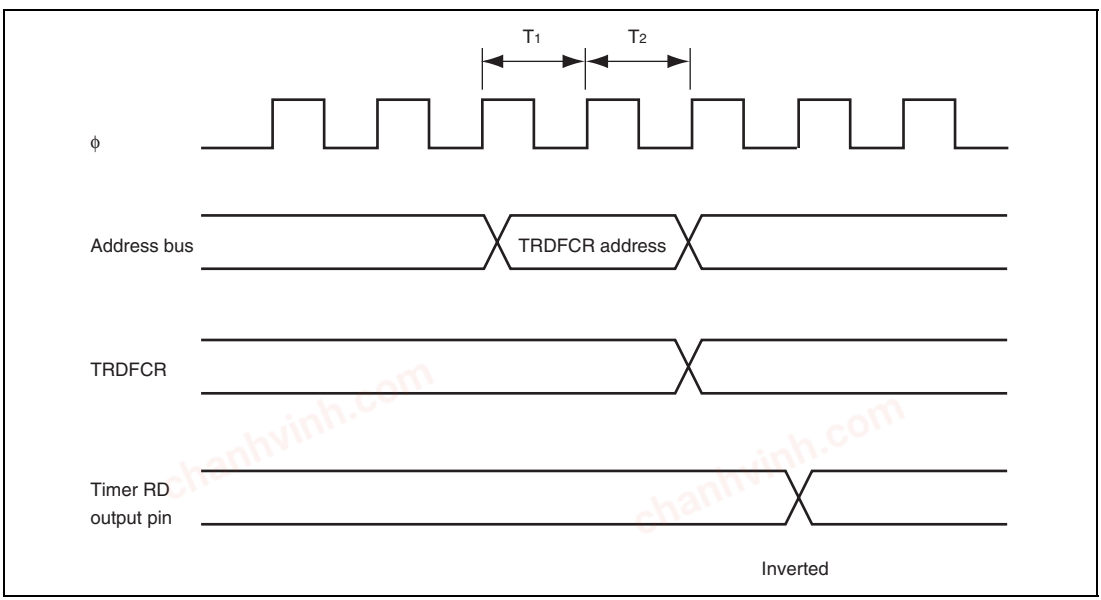



**Figure 16.52 Example of Output Inverse Timing of Timer RD by Writing to TRDFCR**

chanhvinh.com



### (4) Output Inverse Timing by POCR

The output level can be inverted by inverting the POLD, POLC, and POLB bits in POCR in PWM mode. Figure 16.53 shows the timing.

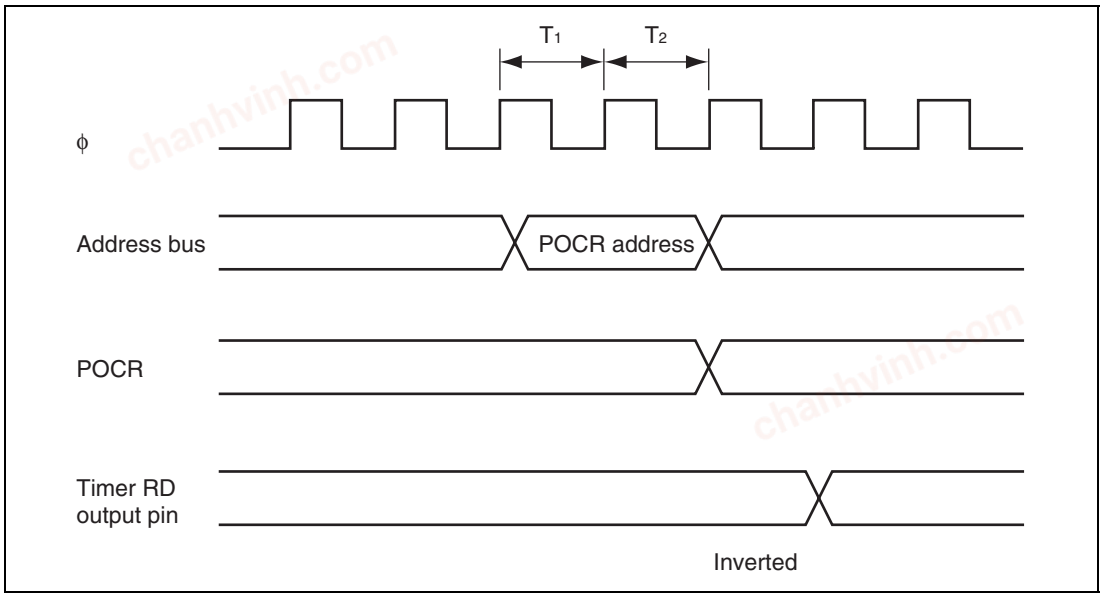



**Figure 16.53 Example of Output Inverse Timing of Timer RD by Writing to POCR**

chanhvinh.com



### 16.3.11 Digital Filtering Function for Input Capture Inputs

Input signals on the FTIOA to FTIOD pins can be input via the digital filters. The digital filter includes three latches connected in series and a match detector circuit. The latches operate on the sampling clock specified by bits DFCK1 and DFCK0 in TRDDF and stores an input signal on the FTIOA to FTIOD pins. When outputs of the three latches match, the match detector circuit outputs the signal level of the input. Otherwise, the output remains unchanged. That is, when a pulse width is equal to or greater than three sampling clock cycles, the pulse is input as a signal. When a pulse width is less than three sampling clock cycles, the pulse is considered as a noise to be removed.

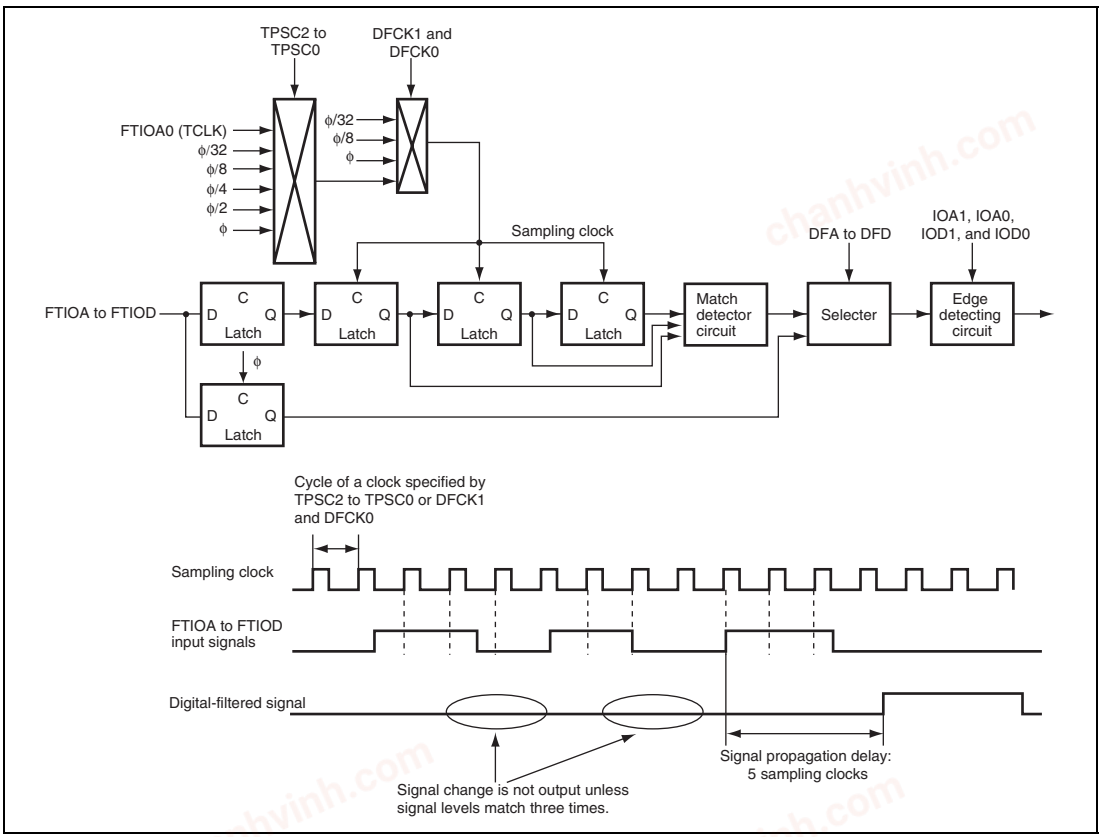



**Figure 16.54 Block Diagram of Digital Filter**

chanhvinh.com



### 16.3.12 Function of Changing Output Pins for GR

With the settings of bits IOC3 and IOD3 in TRDIORC, pins for outputs of compare match signals for GRC and GRD can be changed from the FTIOC and FTIOD pins to the FTIOA and FTIOB pins. This means that the compare match A signal ORed with the compare match C signal can be output on the FTIOA pin. The compare match B ORed with the compare match D signal can be output on the FTIOB pin. Figure 16.55 is a block diagram of this function. The setting for channel 0 is independent of that for channel 1.

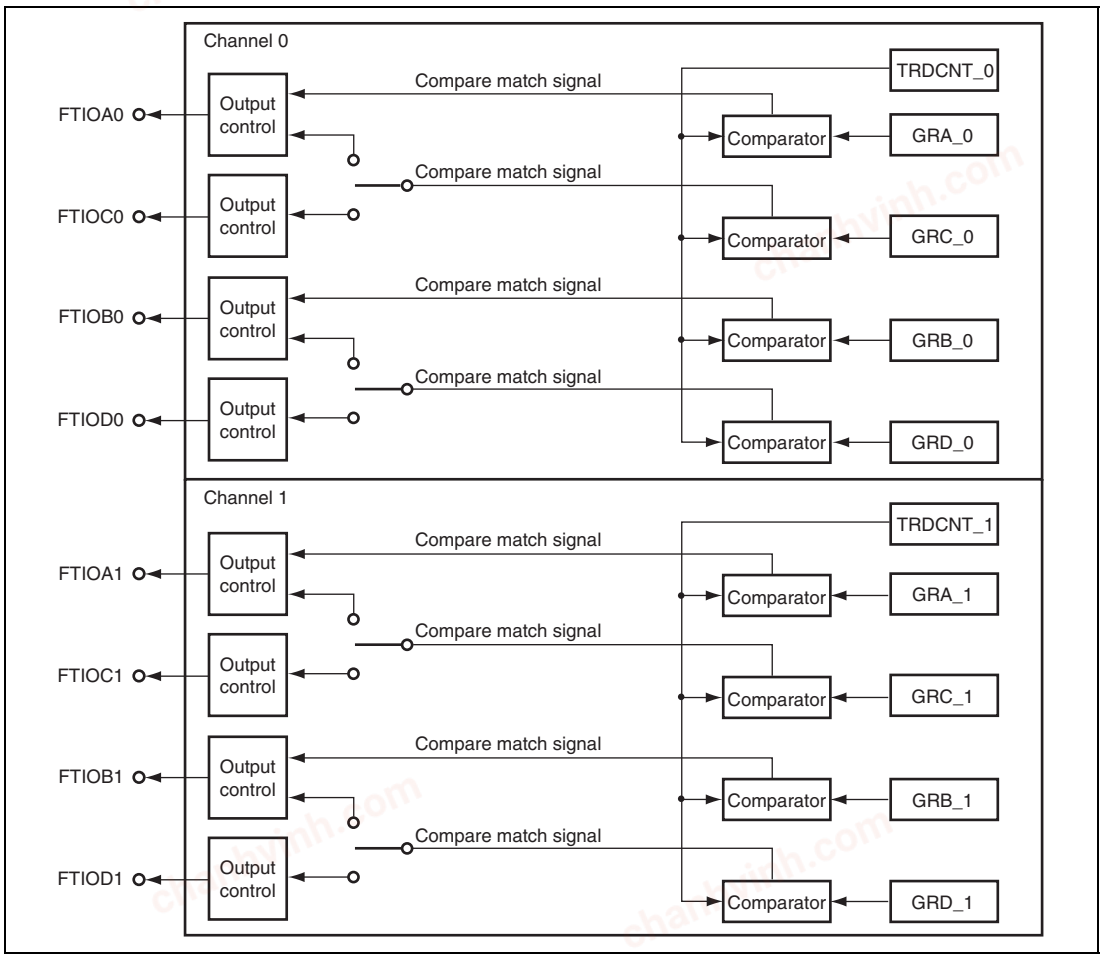



**Figure 16.55 Block Diagram of Output Pins for GR**

chanhvinh.com



Figure 16.56 is an example when non-overlapped pulses are output on pins FTIOA0 and FTIOB0. In this example, TRDCNT_0 functions as a periodic counter which is cleared on compare match A0 (bits CCLR2 to CCLR0 in TRDCR_0 are set to B'001), an output signal is toggled on compare match A (bits IOA2 to IOA0 in TRDIORA_1 are set to B'011), the output signal on the FTIOA pin is toggled on compare match C (GRC_0) (bits IOC3 to IOC0 in TRDIORC_1 are set to B'0X11), an output signal is toggled on compare match B (GRB_0) (bits IOB2 to IOB0 in TRDIORA_1 are set to B'011), and the output signal on the FTIOB pin is toggled on compare match D (GRD_0) (bits IOD3 to IOD0 in TRDIORC_1 are set to B'0X11). The cycle of the pulse is arbitrary.

Similarly, figure 16.57 is an example when non-overlapped pulses are output using TRDCNT_1.

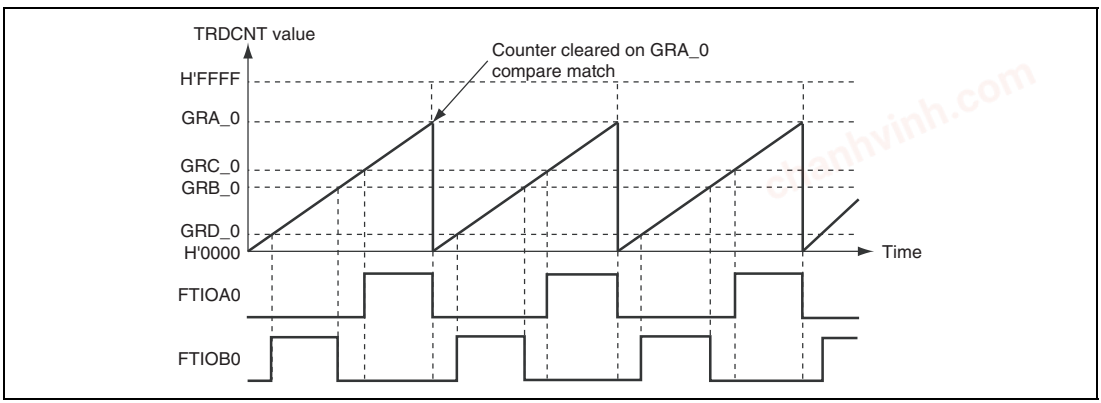



**Figure 16.56 Example of Non-Overlapped Pulses Output on Pins FTIOA0 and FTIOB0 (TRDCNT_0 Used)**

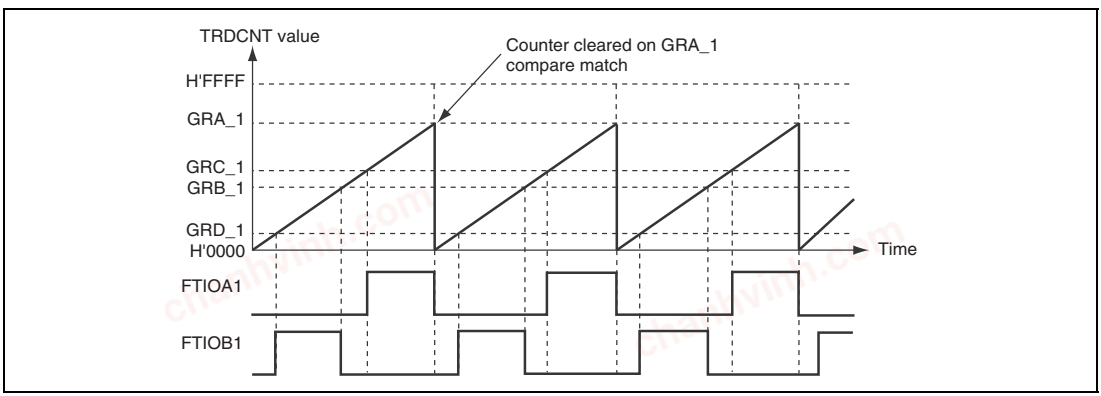



**Figure 16.57 Example of Non-Overlapped Pulses Output on Pins FTIOA1 and FTIOB1 (TRDCNT_1 Used)**

chanhvinh.com



### 16.3.13 A/D Conversion Start Trigger Setting Function

Timer RD can generate the A/D conversion start trigger signal by setting the timer RD A/D conversion start trigger control register (TRDADCR) or bits ADEG and ADTRG in the timer RD function control register (TRDFCR).

Figures 16.58 and 16.59 show examples of the A/D conversion trigger signal generation in complementary PWM mode.

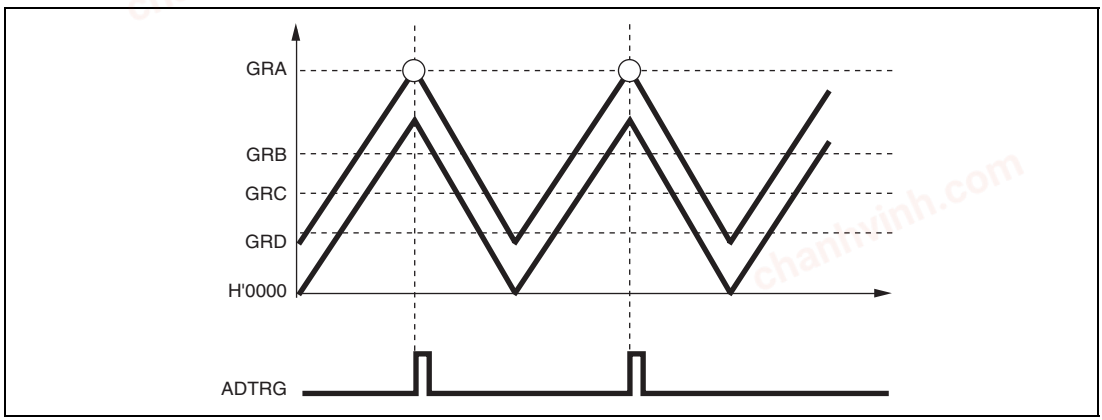



**Figure 16.58 Example of A/D Conversion Trigger Signal Generation in Complementary PWM Mode**
**(Trigger Asserted When TRDCNT_0 Matches GRA_0: ADEG = 0, ADTRG = 1)**

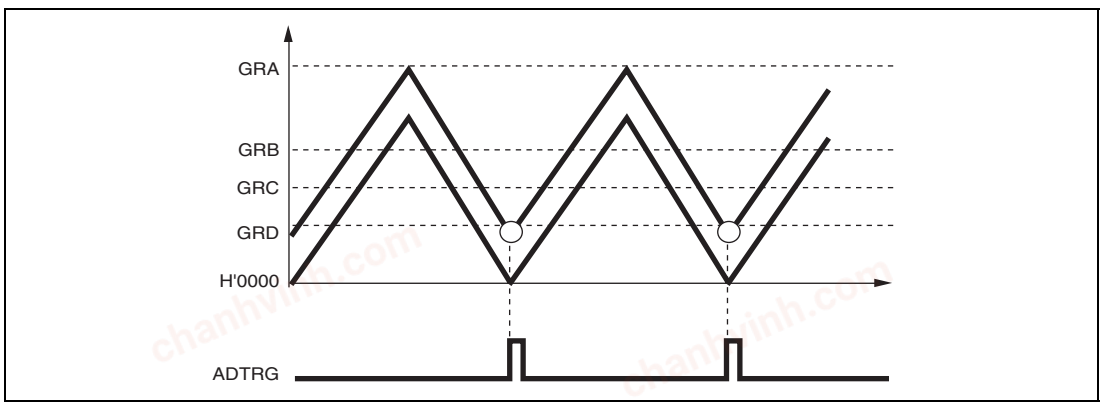



**Figure 16.59 Example of A/D Conversion Trigger Signal Generation in Complementary PWM Mode**
**(Trigger Asserted When TRDCNT_1 Underflows: ADEG = 1, ADTRG = 1)**

chanhvinh.com



Figure 16.60 shows an example where the A/D conversion start trigger signal is generated by compare match. In this case, the TRDADCR register must be set.

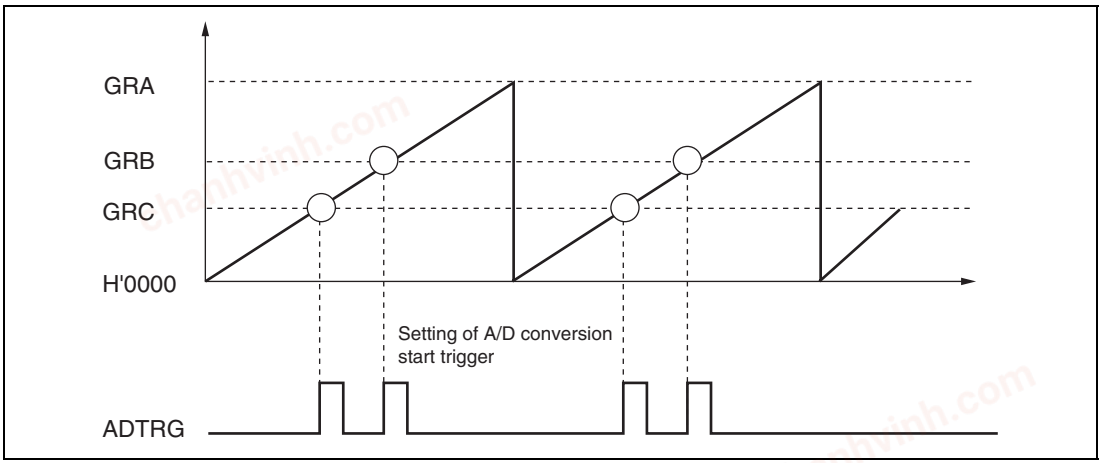



**Figure 16.60 Example of A/D Conversion Trigger Signal Generation by Compare Match**

Figure 16.61 shows the timing for generating the A/D conversion start trigger by compare match.

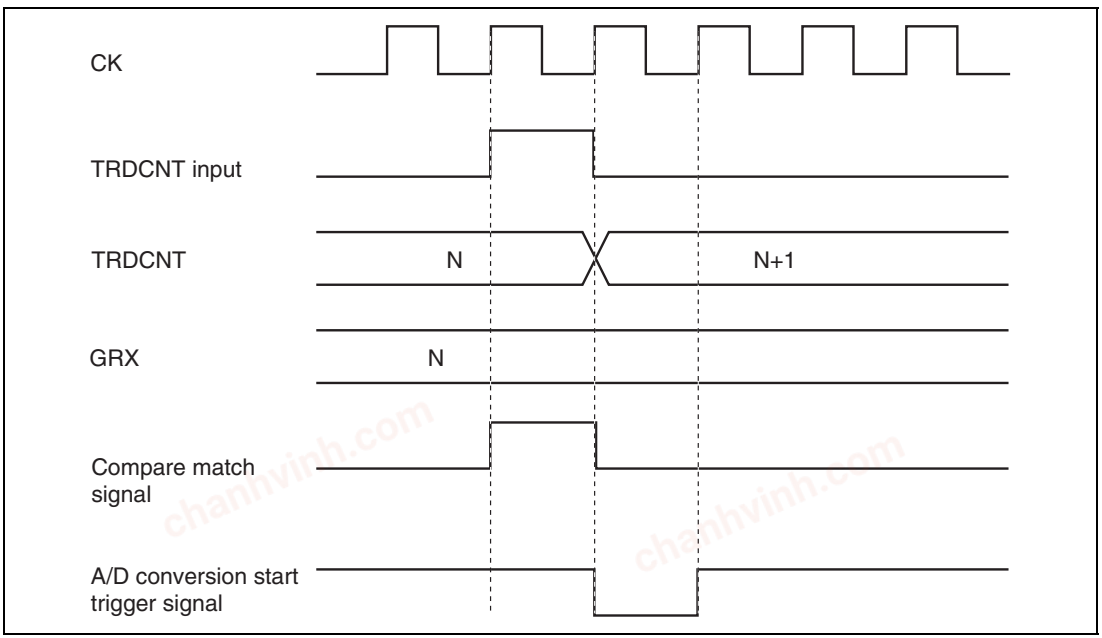



**Figure 16.61 Timing of A/D Conversion Start Trigger Generation**

chanhvinh.com



### 16.3.14 Operation by Event Clear

Using the event link controller (ELC), timer RD unit 0 can be made to operate in the following ways in relation to events occurring in other modules. Each channel 0 and 1 can be specified independently.

#### (1) Staring Counter Operation

The start of counting operations by timer RD can be selected by ELOPA and ELOPB of the ELC. When the event specified by ELSR3 and ELSR4 occur, the STR[1:0] bits in TRDSTR are set to 1, which stars counting by timer RD. However, if the specified event occurs when the STR bit has already been set to 1, the event is not effective.

#### (2) Counting Event

The counting of events by timer RD can be selected by ELOPA and ELOPB of the ELC. When the event specified in ELSR3 and ELSR4 occurs, event counter operation proceeds with that event as the source to drive counting, regardless of the setting of the TPSC[2:0] bits in TRDCR and the STR1 and STR0 bits in TRDSTR. When the value of the counter is read, the value read out is the actual number of input events.

#### (3) Input Capture

Input capture operation of timer RD can be selected by ELOPA and ELOPB of the ELC. When the event specified in ELSR3 and ELSR4 occurs, GRD captures the value of TRDCNT. When input capture operation initiated by an event link is in use, set the IOD[3:0] bits = b'1101 in TRDIORC of timer RD, set the STR bit in TRDSTR to 1, and then start the counter. Since input on the FTIOD pin becomes valid at the same time, fix the input to the FTIOD pin or take other measures such as not allocating the FTIOD pin to the port in the PMC, etc.

chanhvinh.com



## 16.4 Interrupt Sources

There are three kinds of timer RD interrupt sources; input capture/compare match, overflow, and underflow. An interrupt is requested when the corresponding interrupt request flag is set to 1 while the corresponding interrupt enable bit is set to 1.

### 16.4.1 Status Flag Set Timing

#### (1) IMF Flag Set Timing

The IMF flag is set to 1 by the compare match signal that is generated when the GR matches with the TRDCNT. The compare match signal is generated at the last state of matching (timing to update the counter value when the GR and TRDCNT match). Therefore, when the TRDCNT and GR matches, the compare match signal will not be generated until the TRDCNT input clock is generated. Figure 16.62 shows the timing to set the IMF flag.

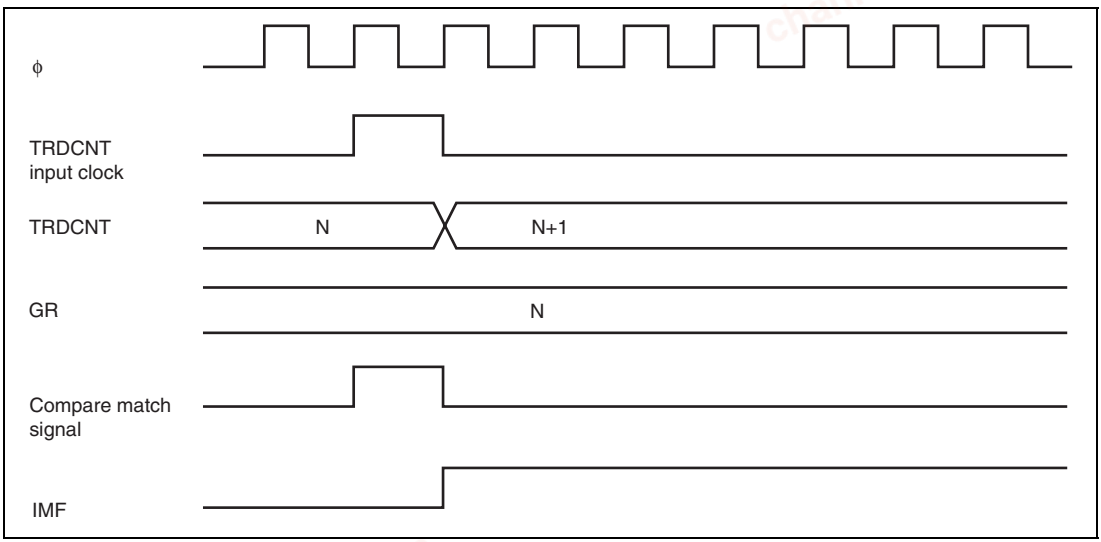



**Figure 16.62 IMF Flag Set Timing when Compare Match Occurs**

chanhvinh.com



### (2) IMF Flag Set Timing at Input Capture

When an input capture signal is generated, the IMF flag is set to 1 and the value of TRDCNT is simultaneously transferred to corresponding GR. Figure 16.63 shows the timing.

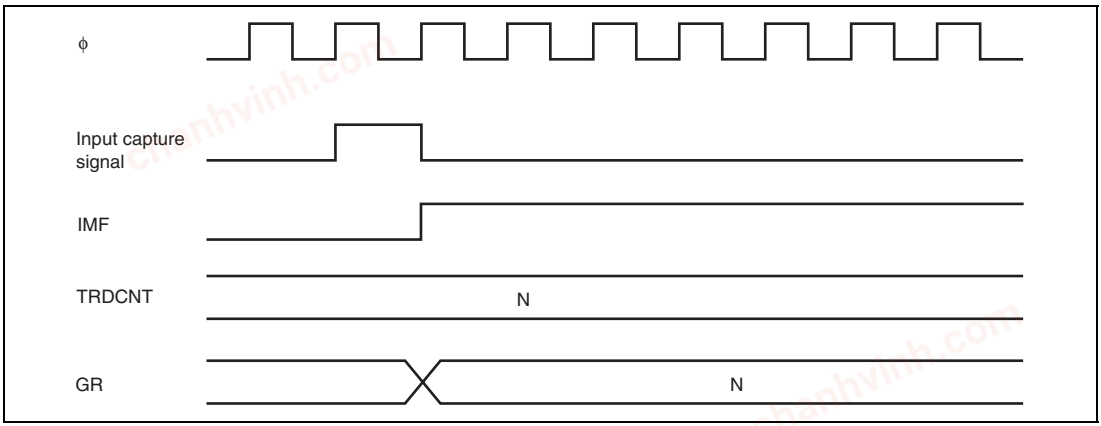



**Figure 16.63 IMF Flag Set Timing at Input Capture**

### (3) Overflow Flag (OVF) Set Timing

The overflow flag is set to 1 when the TRDCNT overflows. Figure 16.64 shows the timing.

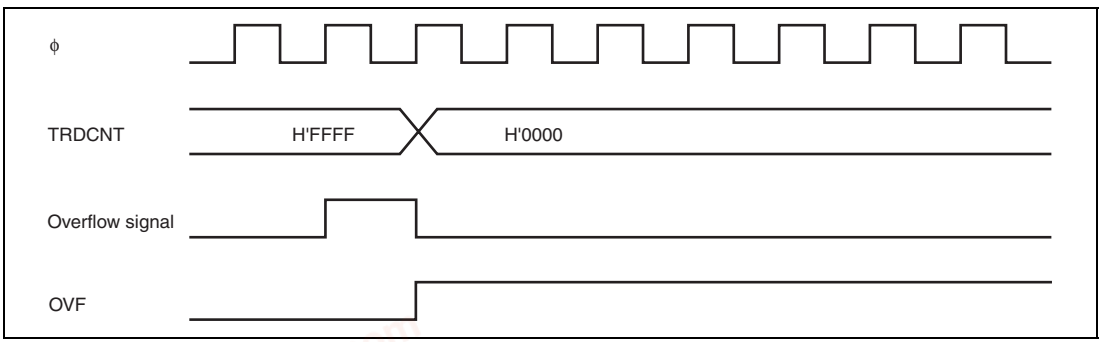



**Figure 16.64 OVF Flag Set Timing**

chanhvinh.com



### 16.4.2 Status Flag Clearing Timing

The status flag can be cleared by writing 0 after reading 1 from the CPU. Figure 16.65 shows the timing in this case.

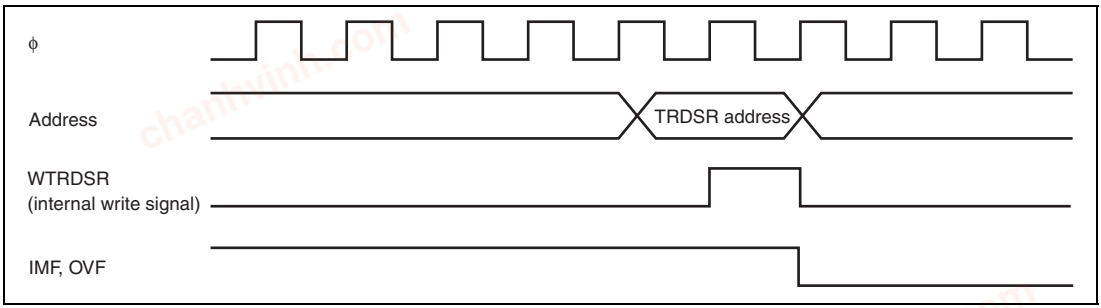



**Figure 16.65 Status Flag Clearing Timing**

## 16.5 Usage Notes

### (1) Input Pulse Width of Input Clock Signal and Input Capture Signal

When the digital filtering function for input is not in use, the pulse width of the input clock signal and the input capture signal must be at least three system clock (ϕ) cycles when the TPSC2 to TPSC0 bits in TRDCR = B'0XX or B'10X; shorter pulses will not be detected correctly.

chanhvinh.com



### (2) Conflict between TRDCNT Write and Clear Operations

If a counter clear signal is generated in the $T_2$ state of a TRDCNT write cycle, TRDCNT clearing has priority and the TRDCNT write is not performed. Figure 16.66 shows the timing in this case.

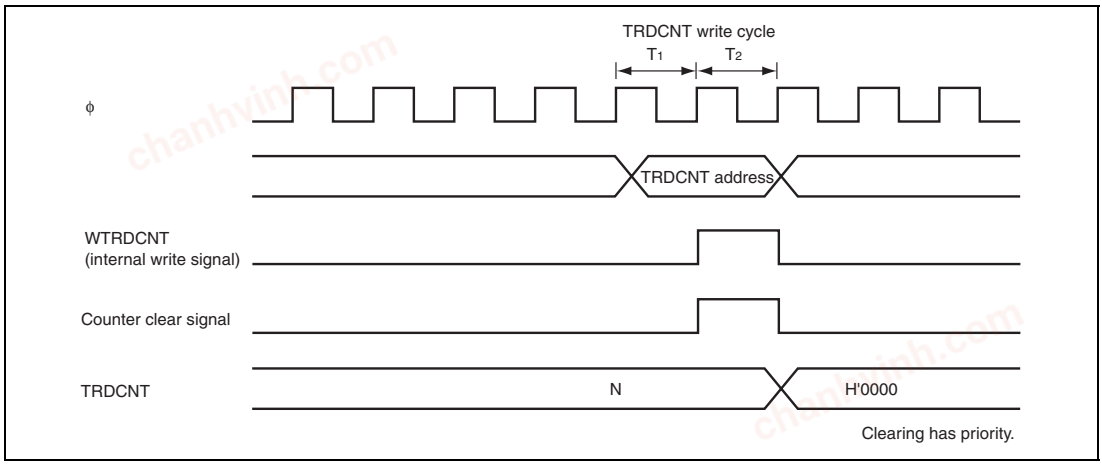



**Figure 16.66 Conflict between TRDCNT Write and Clear Operations**

### (3) Conflict between TRDCNT Write and Increment Operations

If TRDCNT is incremented in the $T_2$ state of a TRDCNT write cycle, writing has priority. Figure 16.67 shows the timing in this case.

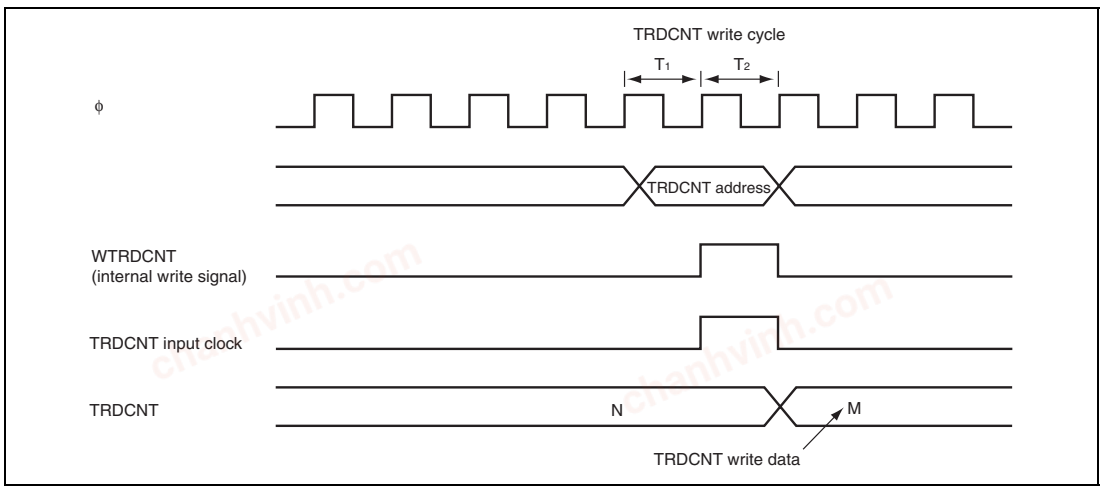



**Figure 16.67 Conflict between TRDCNT Write and Increment Operations**

chanhvinh.com



### (4) Conflict between GR Write and Compare Match

If a compare match occurs in the $T_2$ state of a GR write cycle, GR write has priority and the compare match signal is disabled. Figure 16.68 shows the timing in this case.

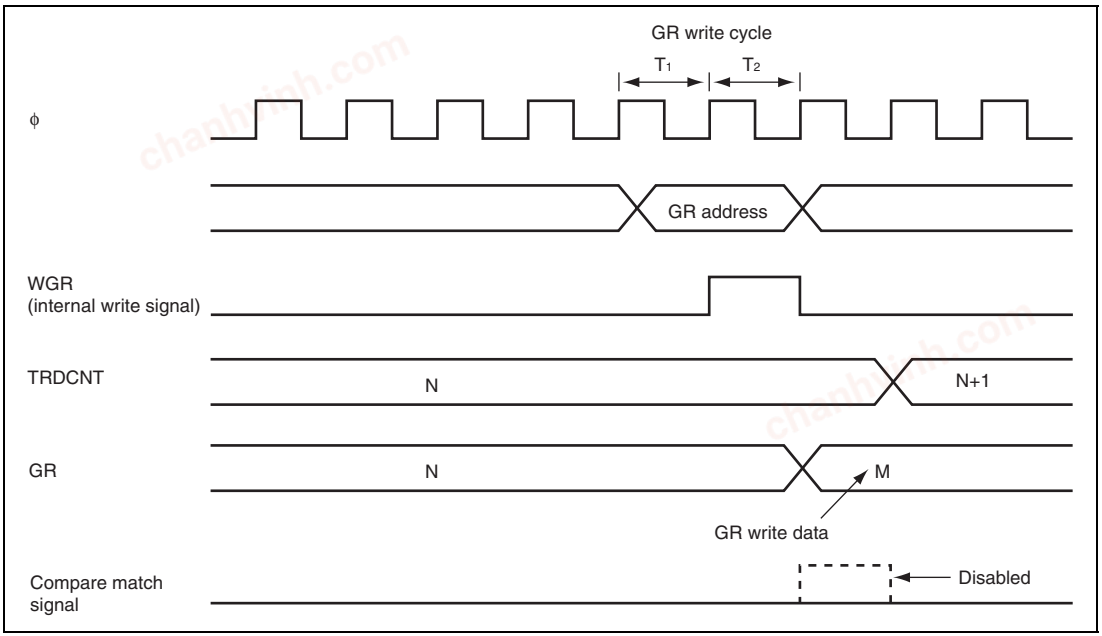



**Figure 16.68 Conflict between GR Write and Compare Match**

chanhvinh.com



### (5) Conflict between TRDCNT Write and Overflow/Underflow

If overflow/underflow occurs in the $T_2$ state of a TRDCNT write cycle, TRDCNT write has priority without an increment operation. At this time, the OVF flag is set to 1. Figure 16.69 shows the timing in this case.

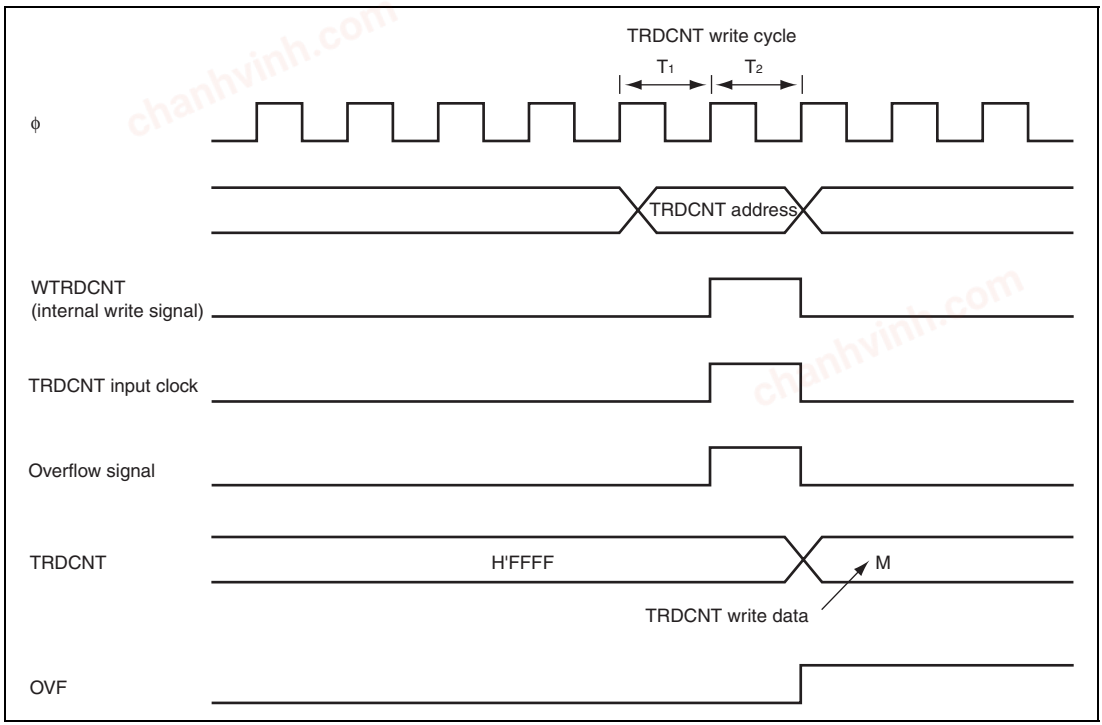



**Figure 16.69 Conflict between TRDCNT Write and Overflow**

chanhvinh.com



### (6) Conflict between GR Read and Input Capture

If an input capture signal is generated in the $T_2$ state of a GR read cycle, the data that is read will be transferred before input capture transfer. Figure 16.70 shows the timing in this case.

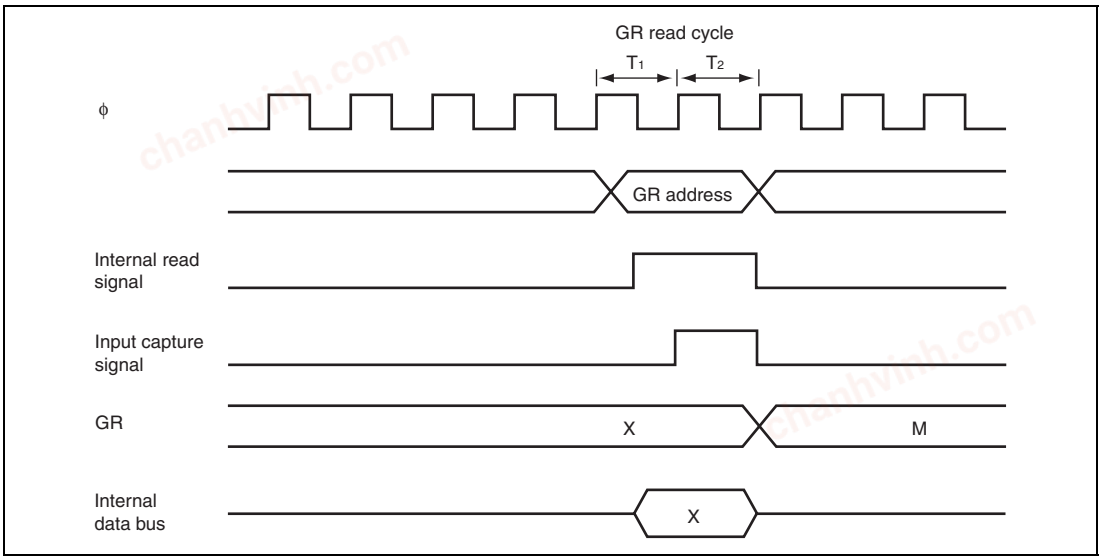



**Figure 16.70 Conflict between GR Read and Input Capture**

chanhvinh.com



### (7) Conflict between Count Clearing and Increment Operations by Input Capture

If an input capture and increment signals are simultaneously generated, count clearing by the input capture operation has priority without an increment operation. The TRDCNT contents before clearing counter are transferred to GR. Figure 16.71 shows the timing in this case.

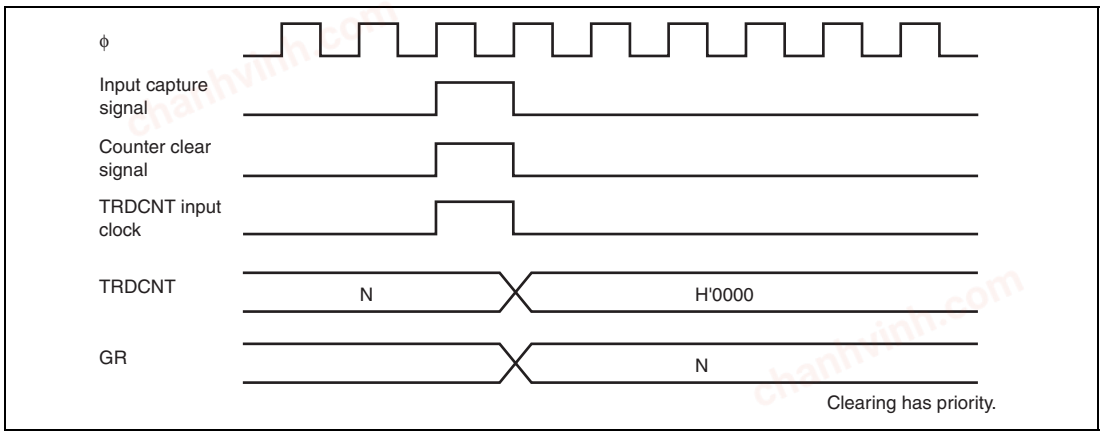



**Figure 16.71 Conflict between Count Clearing and Increment Operations by Input Capture**

chanhvinh.com



### (8) Conflict between GR Write and Input Capture

If an input capture signal is generated in the $T_2$ state of a GR write cycle, the input capture operation has priority and the write to GR is not performed. Figure 16.72 shows the timing in this case.

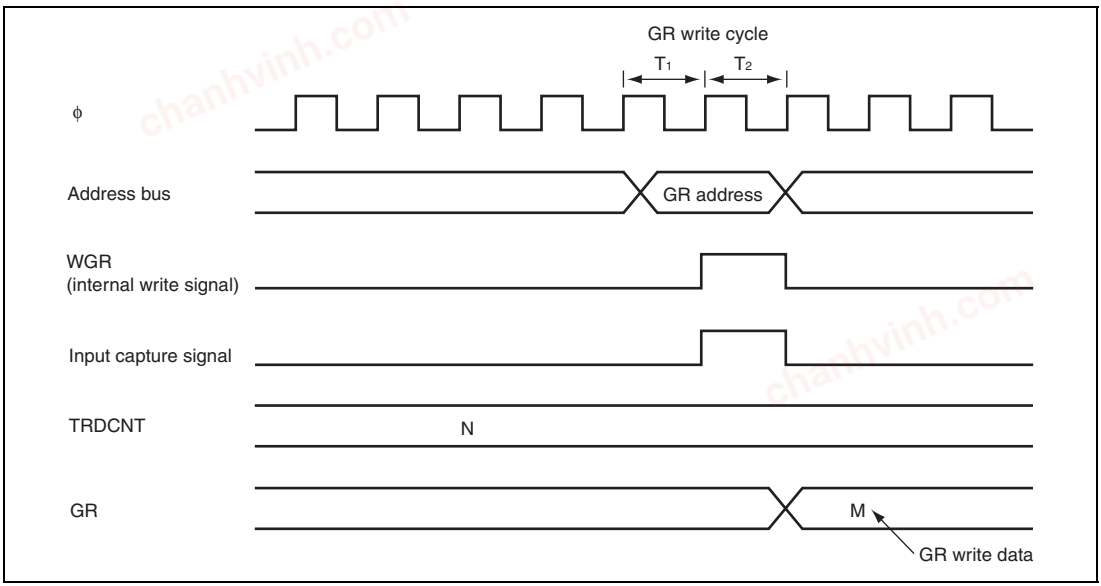



**Figure 16.72 Conflict between GR Write and Input Capture**

chanhvinh.com



### (9) Notes on Setting Reset Synchronous PWM Mode/Complementary PWM Mode

When bits CMD1 and CMD0 in TRDFCR are set, note the following:

- Write bits CMD1 and CMD0 while TRDCNT_1 and TRDCNT_0 are halted.
- Changing the settings of reset synchronous PWM mode to complementary PWM mode or vice versa is disabled. Set reset synchronous PWM mode or complementary PWM mode after the normal operation (bits CMD1 and CMD0 are cleared to 0) has been set.

### (10) Note on Writing to the TOA0 to TOD0 Bits and the TOA1 to TOD1 Bits in TRDOCR

The TOA0 to TOD0 bits and the TOA1 to TOD1 bits in TRDOCR decide the value of the FTIO pin, which is output until the first compare match occurs. Once a compare match occurs and this compare match changes the values of FTIOA0 to FTIOD0 and FTIOA1 to FTIOD1 output, the values of the FTIOA0 to FTIOD0 and FTIOA1 to FTIOD1 pin output and the values read from the TOA0 to TOD0 and TOA1 to TOD1 bits may differ. Moreover, when the writing to TRDOCR and the generation of the compare match A0 to D0 and A1 to D1 occur at the same timing, the writing to TRDOCR has the priority. Thus, output change due to the compare match is not reflected to the FTIOA0 to FTIOD0 and FTIOA1 to FTIOD1 pins. Therefore, when bit manipulation instruction is used to write to TRDOCR, the values of the FTIOA0 to FTIOD0 and FTIOA1 to FTIOD1 pin output may result in an unexpected result. When TRDOCR is to be written to while compare match is operating, stop the counter once before accessing to TRDOCR, read the port 6 state to reflect the values of FTIOA0 to FTIOD0 and FTIOA1 to FTIOD1 output, to TOA0 to TOD0 and TOA1 to TOD1, and then restart the counter. Figure 16.73 shows an example when the compare match and the bit manipulation instruction to TRDOCR occur at the same timing.

chanhvinh.com



TRDOCR has been set to H'06. Compare match B0 and compare match C0 are used. The FTIOB0 pin is in the 1 output state, and is set to the toggle output or the 0 output by compare match B0.
When BCLR#2, @TRDOCR is executed to clear the TOC0 bit (the FTIOC0 signal is low) and compare match B0 occurs at the same timing as shown below, the H'02 writing to TRDOCR has priority and compare match B0 does not drive the FTIOB0 signal low; the FTIOB0 signal remains high.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TRDOCR | TOD1 | TOC1 | TOB1 | TOA1 | TOD0 | TOC0 | TOB0 | TOA0 |
| Set value | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |

BCLR#2, @TRDOCR
(1) TRDOCR read operation: Read H'06
(2) Modify operation: Modify H'06 to H'02
(3) Write operation to TRDOCR: Write H'02

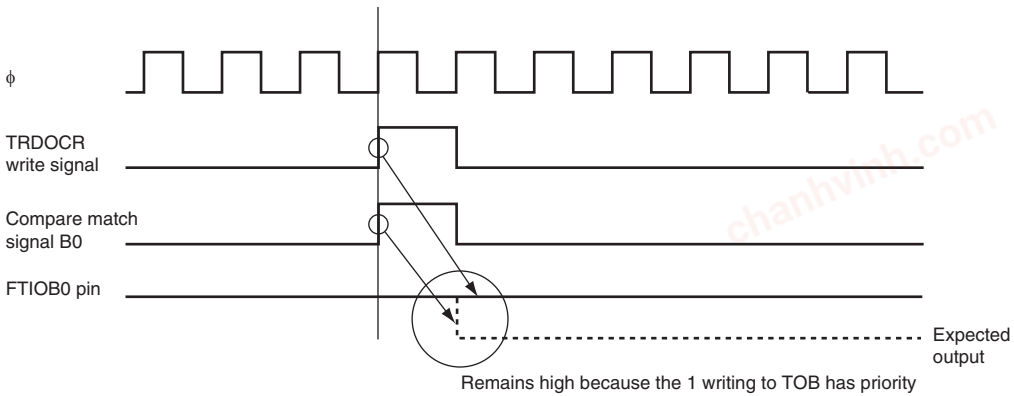



**Figure 16.73 When Compare Match and Bit Manipulation Instruction to TRDOCR Occur at the Same Timing**

chanhvinh.com



# Section 17 Timer RE

Timer RE is a timer that provides a realtime clock function to count time ranging from a second to a week and a compare-match function. Figure 17.1 shows a block diagram of the timer RE.

## 17.1 Features

- Realtime clock mode
  - Counts seconds, minutes, hours, and day-of-week
  - Start/stop function
  - Reset function
  - Readable/writable counter of seconds, minutes, hours, and day-of-week with BCD codes
  - Periodic (seconds, minutes, hours, days, and weeks) interrupts
- Output-compare mode
  - 8-bit counter with a compare-match function
  - Selection of clock source
  - Compare-match interrupt

chanhvinh.com



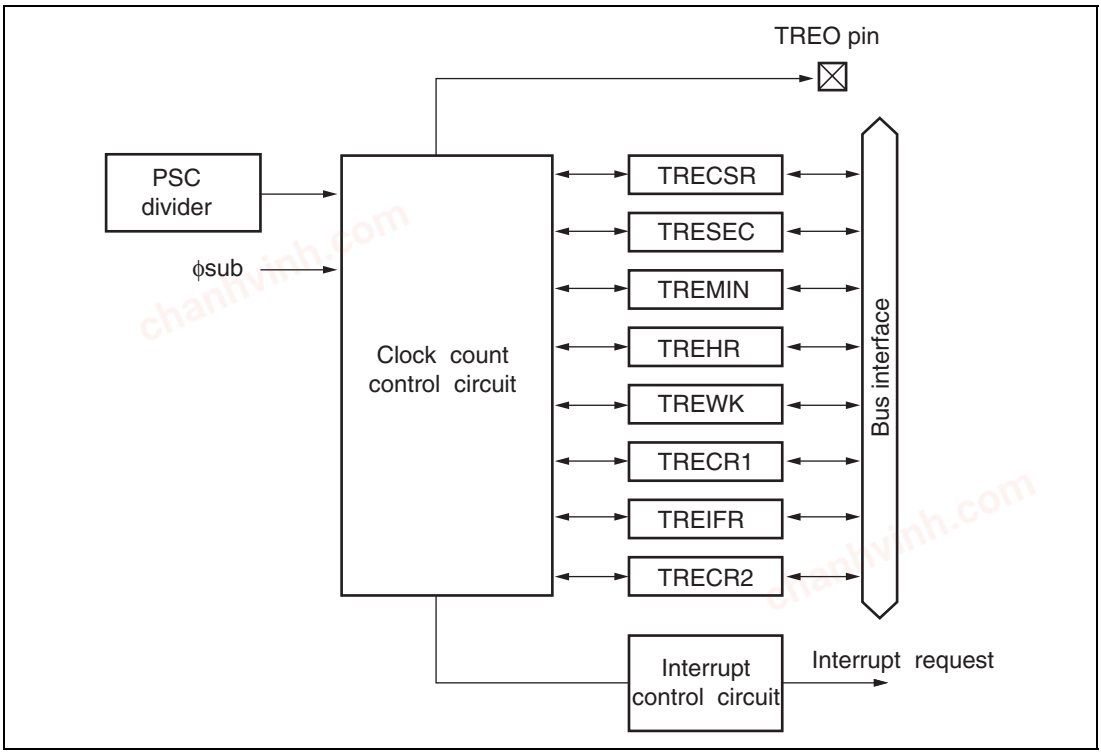



**Figure 17.1 Block Diagram of Timer RE**

Table 17.1 shows the timer RE input/output pin.

**Table 17.1 Pin Configuration**

| Pin Name | I/O | Function |
|---|---|---|
| TREO | Output | Clock or compare-match output |

chanhvinh.com



## 17.2 Register Descriptions

The timer RE has the following registers.

- Timer RE second data register/counter data register (TRESEC)
- Timer RE minute data register/compare data register (TREMIN)
- Timer RE hour data register (TREHR)
- Timer RE day-of-week data register (TREWK)
- Timer RE control register 1 (TRECR1)
- Timer RE control register 2 (TRECR2)
- Timer RE clock source select register (TRECSR)
- Timer RE interrupt flag register (TREIFR)

chanhvinh.com
chanhvinh.com
chanhvinh.com

chanhvinh.com



### 17.2.1 Timer RE Second Data Register/Counter Data Register (TRESEC)

Address: H'FFFFA8

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | BSY | SC12 | SC11 | SC10 | SC03 | SC02 | SC01 | SC00 |
| Value after reset: | — | — | — | — | — | — | — | — |

- Realtime clock mode

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | BSY | Timer RE busy | This bit is set to 1 when the timer RE is updating (calculating) the values of second, minute, hour, and day-of-week data registers. When this bit is 0, the values of second, minute, hour, and day-of-week data registers must be adopted. | R |
| 6 | SC12 | Counting ten's position of seconds | Counts on 0 to 5 for 60-second counting. | R/W |
| 5 | SC11 | | | R/W |
| 4 | SC10 | | | R/W |
| 3 | SC03 | 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. | R/W |
| 2 | SC02 | | | R/W |
| 1 | SC01 | | | R/W |
| 0 | SC00 | | | R/W |

- Output-compare mode

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | BSY | — | Used as an 8-bit register for reading the counter data. | R |
| 6 | SC12 | | The counter value is retained when counting is stopped. | R/W |
| 5 | SC11 | | This register is initialized to H'00 with a compare-match. | R/W |
| 4 | SC10 | | | R/W |
| 3 | SC03 | | | R/W |
| 2 | SC02 | | | R/W |
| 1 | SC01 | | | R/W |
| 0 | SC00 | | | R/W |

TRESEC counts the BCD-coded second value in realtime clock mode. TRESEC is incremented from decimal 00 to 59. TRESEC is used as an 8-bit register for reading the counter data in output-compare mode.

chanhvinh.com



### 17.2.2 Timer RE Minute Data Register/Compare Data Register (TREMIN)

Address: H'FFFFA9

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | BSY | MN12 | MN11 | MN10 | MN03 | MN02 | MN01 | MN00 |
| Value after reset: | — | — | — | — | — | — | — | — |

- Realtime clock mode

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | BSY | Timer RE busy | This bit is set to 1 when the timer RE is updating (calculating) the values of second, minute, hour, and day-of-week data registers. When this bit is 0, the values of second, minute, hour, and day-of-week data registers must be adopted. | R |
| 6 | MN12 | Counting ten's position of minutes | Counts on 0 to 5 for 60-minute counting. | R/W |
| 5 | MN11 | | | R/W |
| 4 | MN10 | | | R/W |
| 3 | MN03 | Counting one's position of minutes | Counts on 0 to 9 once per minute. When a carry is generated, 1 is added to the ten's position. | R/W |
| 2 | MN02 | | | R/W |
| 1 | MN01 | | | R/W |
| 0 | MN00 | | | R/W |

- Output-compare mode

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | BSY | — | Used as an 8-bit register for storing the compare data. The setting range is H'01 to H'FF.<br>This register can be written to only when counting is stopped (when TSTART and TCSTF in TRECR1 are 0). | R |
| 6 | MN12 | | | R/W |
| 5 | MN11 | | | R/W |
| 4 | MN10 | | | R/W |
| 3 | MN03 | | | R/W |
| 2 | MN02 | | | R/W |
| 1 | MN01 | | | R/W |
| 0 | MN00 | | | R/W |

TREMIN counts the BCD-coded minute value on the carry generated once per minute by the TRESEC counting in realtime clock mode. TREMIN is incremented from decimal 00 to 59. TREMIN is used as an 8-bit register for storing the compare data in output-compare mode.

chanhvinh.com



### 17.2.3 Timer RE Hour Data Register (TREHR)

Address: H'FFFFAA

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | BSY | — | HR11 | HR10 | HR03 | HR02 | HR01 | HR00 |
| Value after reset: | — | 0 | — | — | — | — | — | — |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | BSY | Timer RE busy | This bit is set to 1 when the timer RE is updating (calculating) the values of second, minute, hour, and day-of-week data registers. When this bit is 0, the values of second, minute, hour, and day-of-week data registers must be adopted. | R |
| 6 | — | Reserved | This bit is read as 0. The write value should be 0. | — |
| 5 | HR11 | Counting ten's position of hours | Counts on 0 to 2 for ten's position of hours | R/W |
| 4 | HR10 | | | R/W |
| 3 | HR03 | 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. | R/W |
| 2 | HR02 | | | R/W |
| 1 | HR01 | | | R/W |
| 0 | HR00 | | | R/W |

TREHR is used in realtime clock mode and counts the BCD-coded hour value on the carry generated once per hour by TREMIN. TREHR is incremented either from decimal 00 to 11 or 00 to 23 by the selection of the 12/24 bit in TRECR1. This register is not used in output-compare mode.

chanhvinh.com



### 17.2.4 Timer RE Day-of-Week Data Register (TREWK)

Address: H'FFFFAB

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | BSY | — | — | — | — | WK[2:0] | | |
| Value after reset: | — | 0 | 0 | 0 | 0 | — | — | — |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | BSY | Timer RE busy | This bit is set to 1 when the timer RE is updating (calculating) the values of second, minute, hour, and day-of-week data registers. When this bit is 0, the values of second, minute, hour, and day-of-week data registers must be adopted. | R |
| 6 to 3 | — | Reserved | These bits are read as 0. The write value should be 0. | — |
| 2 to 0 | WK[2:0] | Day-of-week counting | 000: Sunday<br>001: Monday<br>010: Tuesday<br>011: Wednesday<br>100: Thursday<br>101: Friday<br>110: Saturday<br>111: Setting prohibited | R/W |

TREWK is used in realtime clock mode and counts the BCD-coded day-of-week value on the carry generated once per day by TREHR. Bits WK[2:0] indicate the day of the week with a binary code, ranging from decimal 0 to 6. This register is not used in output-compare mode.

chanhvinh.com



### 17.2.5 Timer RE Control Register 1 (TRECR1)

Address: H'FFFFAC

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TSTART | H12_H24 | PM | TRERST | INT | TOENA | TCSTF | — |
| Value after reset: | — | — | — | 0 | 0 | 0 | — | 0 |

- Realtime clock mode

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | TSTART | Counter operation start | 0: Stops timer counter operation<br>1: Starts timer counter operation | R/W |
| 6 | H12_H24*[1] | Operating mode | 0: The timer RE operates in 12-hour mode. TREHR counts on 0 to 11.<br>1: The timer RE operates in 24-hour mode. TREHR counts on 0 to 23. | R/W |
| 5 | PM*[1] | a.m./p.m. | 0: Indicates a.m. when the timer RE is in the 12-hour mode.<br>1: Indicates p.m. when the timer RE is in the 12-hour mode. | R/W |
| 4 | TRESET | Reset | 0: Normal operation<br>1: Resets all the registers and control circuits, except TRECSR and the TOENA and TRESET bits in this register. Clear this bit to 0 after having been set to 1. | R/W |
| 3 | INT*[1] | Interrupt generation timing | 0: Generates a second, minute, hour, or day-of-week periodic interrupt during timer RE busy period.<br>1: Generates a second, minute, hour, or day-of-week periodic interrupt immediately after completing timer RE busy period.*[2] | R/W |
| 2 | TOENA | TREO pin output enable | 0: Disables timer RE divided clock output.<br>1: Enables timer RE divided clock output. | R/W |
| 1 | TCSTF | Operation status flag | 0: Indicates that timer RE operation has been stopped.<br>1: Indicates that timer RE operation is in progress. | R |
| 0 | — | Reserved | This bit is read as 0. The write value should be 0. | — |

Notes: 1. Bits H12_H24, PM, and INT should be set when the timer RE operation is stopped.
2. This bit should be set to 1 in realtime clock mode and cleared to 0 in output compare mode.

chanhvinh.com



TRECR1 controls start/stop and reset of the counter. For the definition of time expression, see figure 17.2.

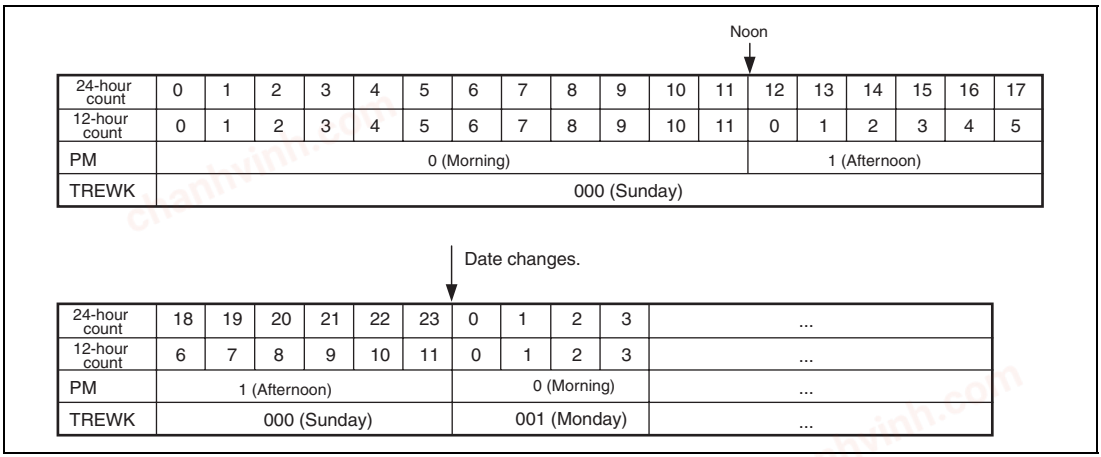



**Figure 17.2 Definition of Time Expression**

chanhvinh.com



- Output-compare mode

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | TSTART | Counter operation start | 0: Stops timer counter operation.<br>1: Starts timer counter operation. | R/W |
| 6 | H12_H24 | Operating mode | 0 should be written to this bit in output-compare mode. | R/W |
| 5 | PM | a.m./p.m. | 0 should be written to this bit in output-compare mode. | R/W |
| 4 | TRESET | Reset | 0: Normal operation<br>1: Resets all the registers and control circuits, except TRECSR and the TOENA and TRESET bits in this register. Clear this bit to 0 after having been set to 1. | R/W |
| 3 | INT | Interrupt generation timing | 0 should be written to this bit in output-compare mode. | R/W |
| 2 | TOENA | TREO pin output enable | 0: Disables timer RE divided clock output.<br>1: Enables timer RE divided clock output. | R/W |
| 1 | TCSTF | Operation status flag | 0: Indicates that timer RE operation has been stopped.<br>1: Indicates that timer RE operation is in progress. | R |
| 0 | — | Reserved | This bit is read as 0. The write value should be 0. | — |

Note: After writing 1 to TSTART, the timer RE should not be accessed before reading 1 from TCSTF, with the exception of reading TCSTF. Similarly, after writing 0 to TSTART, the timer RE should not be accessed before reading 0 from TCSTF, with the exception of reading TCSTF.

chanhvinh.com



### 17.2.6 Timer RE Control Register 2 (TRECR2)

Address: H'FFFFAD

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | COMIE | WKIE | DYIE | HRIE | MNIE | SEIE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7, 6 | — | Reserved | These bits are read as 0. The write value should be 0. | — |
| 5 | COMIE | Compare-match interrupt enable | 0: Disables a compare-match interrupt<br>1: Enables a compare-match interrupt<br>This bit should be 0 in realtime clock mode. | R/W |
| 4 | WKIE | Week periodic interrupt enable | 0: Disables a week periodic interrupt<br>1: Enables a week periodic interrupt<br>This bit should be 0 in output-compare mode. | R/W |
| 3 | DYIE | Day periodic interrupt enable | 0: Disables a day periodic interrupt<br>1: Enables a day periodic interrupt<br>This bit should be 0 in output-compare mode. | R/W |
| 2 | HRIE | Hour periodic interrupt enable | 0: Disables an hour periodic interrupt<br>1: Enables an hour periodic interrupt<br>This bit should be 0 in output-compare mode. | R/W |
| 1 | MNIE | Minute periodic interrupt enable | 0: Disables a minute periodic interrupt<br>1: Enables a minute periodic interrupt<br>This bit should be 0 in output-compare mode. | R/W |
| 0 | SEIE | Second periodic interrupt enable | 0: Disables a second periodic interrupt<br>1: Enables a second periodic interrupt<br>This bit should be 0 in output-compare mode. | R/W |

Notes: 1. When using interrupts, this register should be set last after other registers are set.
2. The COMIE bit should be set when counting operation is stopped.
3. Bits WKIE, DYIE, HRIE, MNIE, and SEIE should be set when timer RE operation is stopped.

TRECR2 controls timer RE periodic interrupts of weeks, days, hours, minutes, and seconds in realtime clock mode. Enabling interrupts of weeks, days, hours, minutes, and seconds sets the interrupt request flag to 1 in the timer RE interrupt flag register (TREIFR) when an interrupt occurs. It also controls a compare-match interrupt when output-compare mode is used.

chanhvinh.com



### 17.2.7 Timer RE Interrupt Flag Register (TREIFR)

Address: H'FFFFAE

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | COMF | WKF | DYF | HRF | MNF | SECF |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7, 6 | — | Reserved | These bits are read as 0. The write value should be 0. | — |
| 5 | COMF | Compare-match interrupt request flag | [Setting condition]<br>• When the counter value matches the value set in TREMIN in output-compare mode.<br>[Clearing condition]<br>• When 1 is read from the bit and then 0 is written to the bit. | R/W |
| 4 | WKF | Week periodic interrupt request flag | [Setting condition]<br>• When bits WK[2:0] in TREWK reach B'000 in realtime clock mode.<br>[Clearing condition]<br>• When 1 is read from the bit and then 0 is written to the bit.<br>When the DTC is activated with a week periodic interrupt and the DISEL bit in the MRB register of the DTC is 1. | R/W |
| 3 | DYF | Day periodic interrupt request flag | [Setting condition]<br>• Each time TREWK is updated in realtime clock mode. (Occurs every day)<br>[Clearing conditions]<br>• When 1 is read from the bit and then 0 is written to the bit.<br>• When the DTC is activated with a day periodic interrupt and the DISEL bit in the MRB register of the DTC is 1. | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 2 | HRF | Hour periodic interrupt request flag | [Setting condition]<br>• Each time TREHR is updated in realtime clock mode. (Occurs every hour)<br>[Clearing conditions]<br>• When 1 is read from the bit and then 0 is written to the bit.<br>• When the DTC is activated with an hour periodic interrupt and the DISEL bit in the MRB register of the DTC is 1. | R/W |
| 1 | MNF | Minute periodic interrupt request flag | [Setting condition]<br>• Each time TREMIN is updated in realtime clock mode. (Occurs every minute)<br>[Clearing conditions]<br>• When 1 is read from the bit and then 0 is written to the bit.<br>• When the DTC is activated with a minute periodic interrupt and the DISEL bit in the MRB register of the DTC is 1. | R/W |
| 0 | SECF | Second periodic interrupt request flag | [Setting condition]<br>• Each time TRESEC is updated in realtime clock mode. (Occurs every second)<br>[Clearing conditions]<br>• When 1 is read from the bit and then 0 is written to the bit.<br>• When the DTC is activated with a second periodic interrupt and the DISEL bit in the MRB register of the DTC is 1. | R/W |

chanhvinh.com



### 17.2.8 Timer RE Clock Source Select Register (TRECSR)

Address: H'FFFFAF

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | RCS[6:4] | | | RCS3 | RCS2 | RCS[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is read as 0. The write value should be 0. | — |
| 6 to 4 | RCS[6:4]*2 | Clock output select | 000: ϕ/2<br>001: ϕ/4<br>010: ϕ/8<br>011: Compare-match output (Only valid in output-compare mode)<br>100: ϕ/sub (32.768 kHz)<br>101: 1 Hz (Only valid in realtime clock mode)<br>11x: Setting prohibited | R/W |
| 3 | RCS3 | Mode select | 0: Output-compare mode<br>1: Realtime clock mode | R/W |
| 2 | RCS2 | 4-bit counter select | (Only valid in output-compare mode)<br>0: Does not use the 4-bit counter.<br>1: Uses the 4-bit counter. | R/W |
| 1, 0 | RCS[1:0] *1*3 | Clock source select | 00: ϕ/2<br>01: ϕ/4<br>10: ϕ/8<br>11: ϕ/sub | R/W |

[Legend]

X: Don't care

Notes: 1. RCS[1:0] should be set when realtime clock mode is used or when counter operation is stopped.

2. RCS[6:4] should be set when the TOENA bit in TRECR1 is 0.

3. In output compare mode, when the CPU is in a ϕloco clock mode, do not select the ϕsub clock as the clock source for the timer.

chanhvinh.com



TRECSR selects clock output, operating mode, and clock source.

- RCS6 to RCS4 (clock output select)
  Selects a clock output from the TREO pin when the TOENA bit in TRECR1 is set to 1.
- RCS1 and RCS0 (clock source select)
  Selects a clock source for output-compare mode. For realtime clock mode, the subclock ϕsub (32.768 kHz) is selected regardless of the setting of these bits.

## 17.3 Operation of Realtime Clock Mode

### 17.3.1 Initial Settings of Registers after Power-On

The timer RE registers that contain second, minute, hour, and day-of-week data are not initialized by a reset by the $\overline{\text{RES}}$ pin, LVD, or watchdog timer. Therefore, all registers must be set to their initial values after power-on. Once the register settings are made, the timer RE provides an accurate time as long as power is supplied regardless of the $\overline{\text{RES}}$ pin, LVD, or watchdog timer reset.

### 17.3.2 Initial Setting Procedure

Figure 17.3 shows the procedure for the initial setting of the timer RE to be used in realtime clock mode. To set the timer RE again, also follow this procedure.

chanhvinh.com



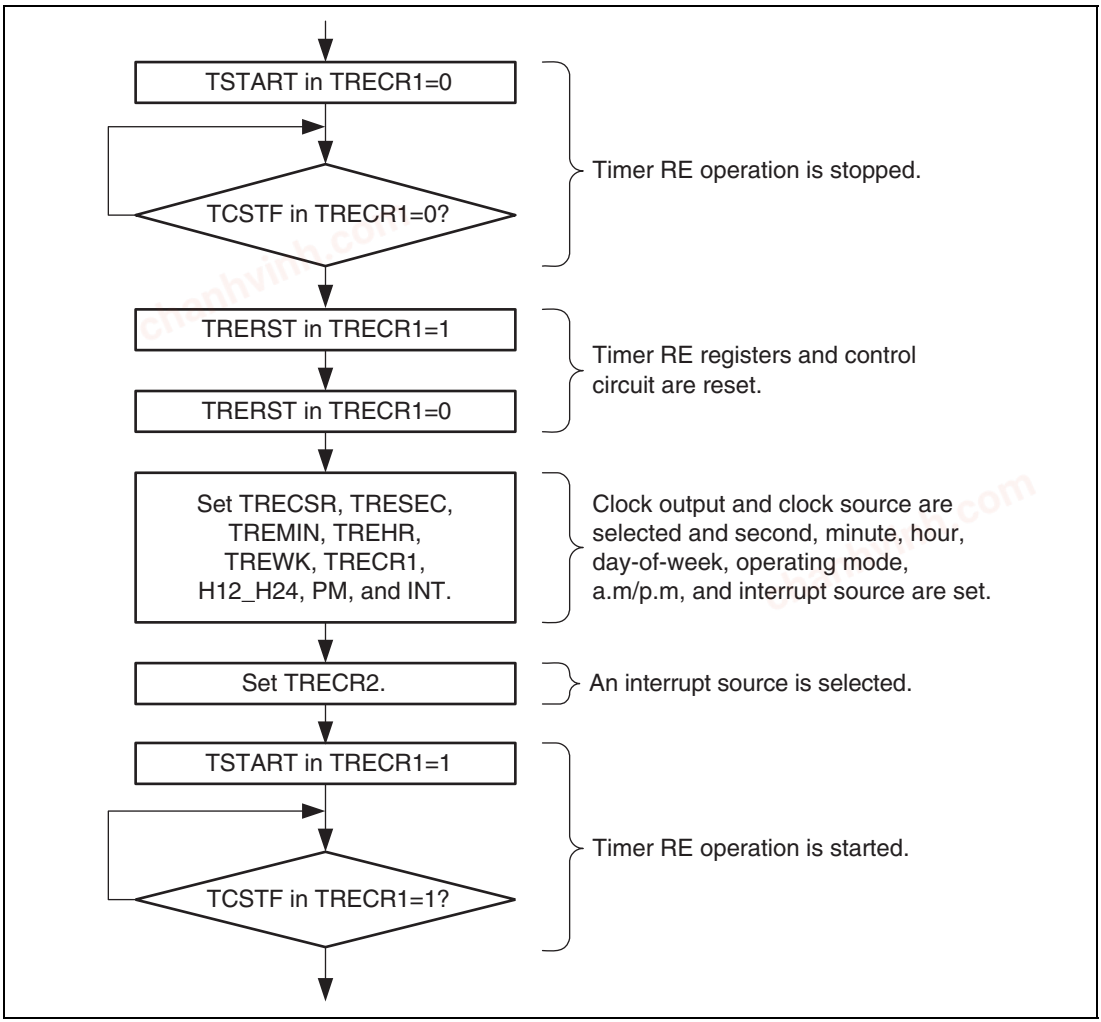



**Figure 17.3 Initial Setting Procedure**

chanhvinh.com



### 17.3.3 Data Reading Procedure in Realtime Clock Mode

When the seconds, minutes, hours, or day-of-week datum is updated while time data is being read, the data obtained may not be correct, and so the time data must be read again. Figure 17.4 shows an example in which correct data is not obtained. In this example, since only TRESEC is read after data update, about 1-minute inconsistency occurs.

The following three methods can be used to avoid reading in this timing:

1. Check the setting of the BSY bit, and when the BSY bit changes from 1 to 0, read from the second, minute, hour, and day-of-week registers. When about 62.5 ms is passed after the BSY bit is set to 1, the registers are updated, and the BSY bit is cleared to 0.
2. Making use of interrupts, read from the second, minute, hour, and day-of week registers after the SECF flag in TREIFR is set to 1 and the BSY bit is confirmed to be 0.
3. Read from the second, minute, hour, and day-of week registers twice in a row, and if there is no change in the read data, the read data is used.

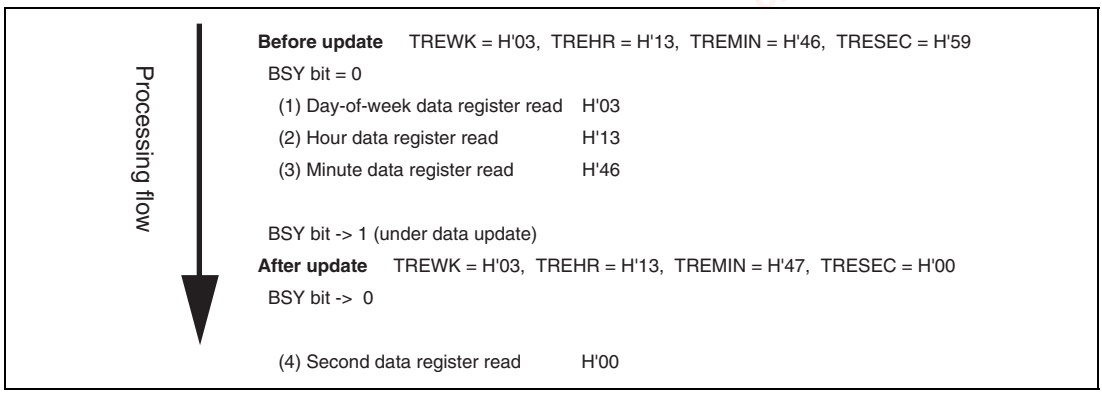



**Figure 17.4 Example: Reading of Inaccurate Time Data**

chanhvinh.com



### 17.3.4 Operation in Realtime Clock Mode

Figure 17.5 shows an example of realtime clock mode operation.

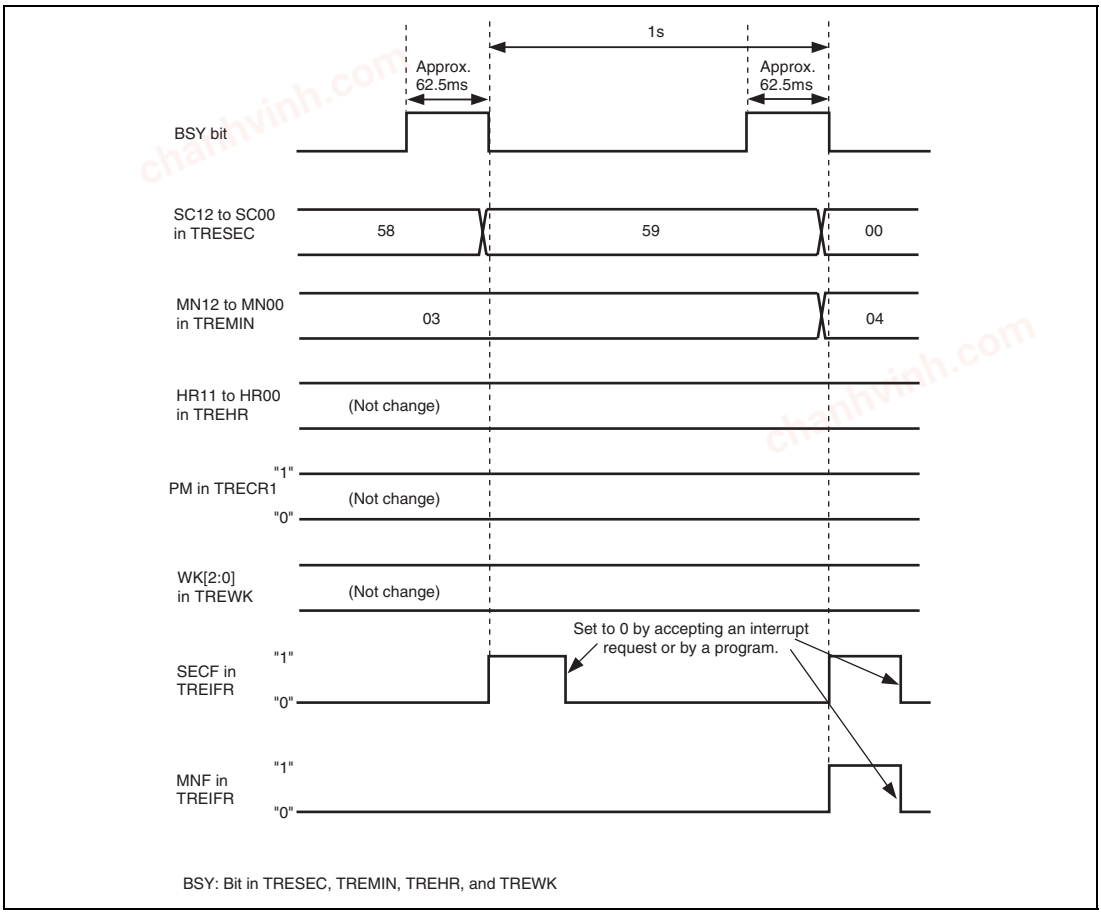



**Figure 17.5 Example of Realtime Clock Mode Operation**

chanhvinh.com



## 17.4 Operation of Output Compare Mode

Writing 0 to the RCS3 bit in TRECSR sets the timer RE in output compare mode and causes it to operate as a counter provided with an 8-bit compare match function. Four count sources can be selected. When used in output compare mode, the timer RE should be initialized. When timer RE is initialized, first select the output-compare mode through the RCS3 bit in TRECSR and then perform the initial setting procedure shown in figure 17.3.

The count source selected by the RCS1 and RCS0 bits is divided into two and counted with an 8-bit counter. Setting 1 to the RCS2 bit in TRECSR causes the count source divided into two to be counted with a 4-bit counter, and the 8-bit counter counts overflows of the 4-bit counter.

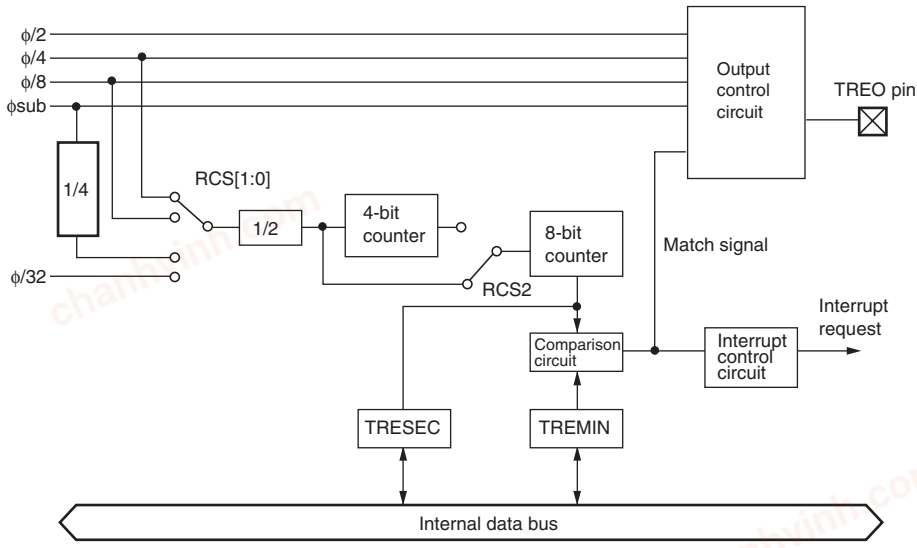
TREMIN sets a compare value. By reading TRESEC, it is possible to read values from the 8-bit counter. In this mode, TREHR or TREWK is not used. Setting bits RCS6 to RCS4 in TRECSR to B'011 and setting the TOENA bit in TRECR1 to 1 produces toggle output from the TREO pin each time the value of the 8-bit counter matches the value of TREMIN (initial value: low output).

Also, by setting the COMIE bit in TRECR2 to 1, it is possible to generate a compare match interrupt request. The counter, using the TSTART bit in TRECR1, controls the start/stop of counter operation.

Figure 17.6 shows a block diagram of output compare mode; figure 17.7 shows an operation example.

chanhvinh.com






[Legend]
TRESEC: Timer RE second data register/counter data register
TREMIN: Timer RE minute data register/compare data register
RCS2 to RCS0: Bits 2 to 0 in TRECSR

**Figure 17.6 Block Diagram of Output Compare Mode**

chanhvinh.com



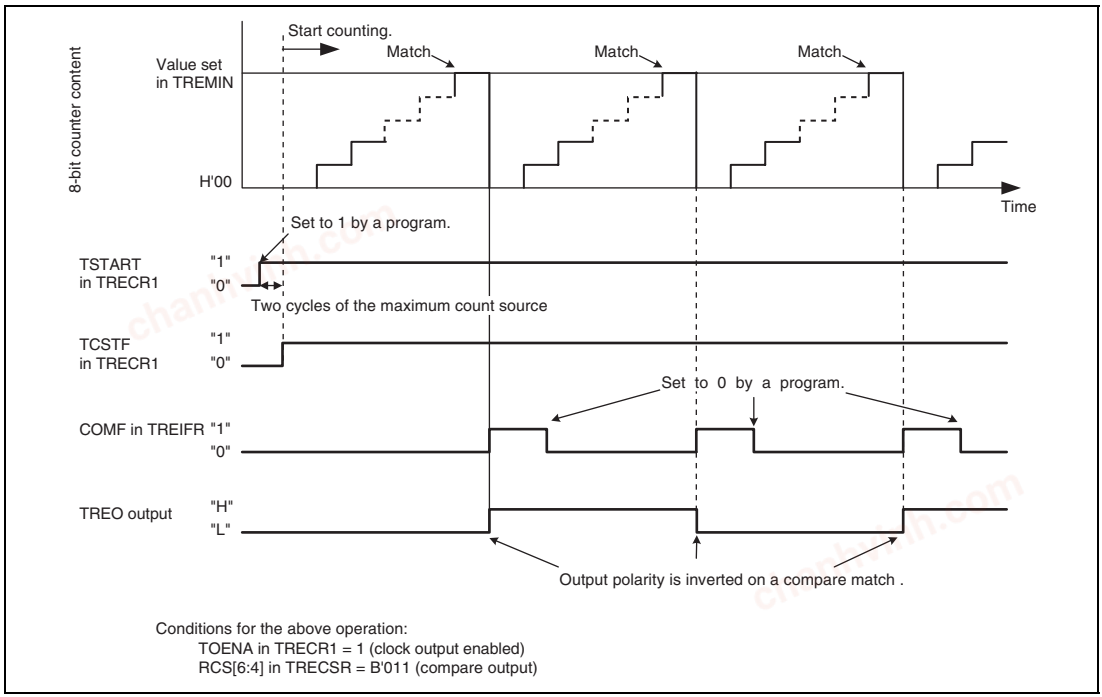



**Figure 17.7 Example of Output Compare Mode Operation**

chanhvinh.com



## 17.5 Interrupt Sources

There are six kinds of timer RE interrupts: week interrupts, day interrupts, hour interrupts, minute interrupts, and second interrupts in realtime clock mode, and compare-match interrupts in output compare mode. Table 17.2 shows the interrupt sources.

When using an interrupt, initiate the timer RE last after other registers are set. Independent vector addresses are allocated to each timer RE interrupt source.

**Table 17.2 Interrupt Sources**

| Interrupt Name | Interrupt Source | Interrupt Enable Bit |
|---|---|---|
| Compare-match interrupt | Occurs when the count value matches the compare data. | COMIE |
| Week periodic interrupt | Occurs every week when the day-of-week data register value becomes 0. | WKIE |
| Day periodic interrupt | Occurs every day when the day-of-week data register value is incremented. | DYIE |
| Hour periodic interrupt | Occurs every hour when the hour date register value is incremented. | HRIE |
| Minute periodic interrupt | Occurs every minute when the minute data register value is incremented. | MNIE |
| Second periodic interrupt | Occurs every second when the second data register value is incremented. | SEIE |

chanhvinh.com



## 17.6 Usage Notes

### (1) Starting and Stopping Counting Process

The timer RE includes a TSTART bit that directs the start or stop of the counting process, and a TCSTF bit that indicates that the counting process has started or stopped.

Setting the TSTART bit to 1 causes the timer RE to start counting and assigns 1 to the TCSTF bit. From the time the TSTART bit is set to 1 and to the time the TCSTF bit turns 1, a maximum of 2 cycles of count sources are required. During this time period, the timer RE related registers*, with the exception of the TCSTF bit, should not be accessed.

Similarly, clearing the TSTART bit to 0 causes the timer RE to stop counting, and assigns 0 to the TCSTF bit. From the time the TSTART bit is set to 0 and to the time the TCSTF bit turns 0, the timer RE related registers*, with the exception of the TCSTF bit, should not be accessed.

Note: Timer RE related registers: TRESEC, TREMIN, TREHR, TREWK, TRECR1, TRECR2, and TRECSR

### (2) Register Settings of Timer RE

The following registers and bits should be written when the timer RE is stopped.

The condition "timer RE stopped" refers to the condition in which both the TSTART and TCSTF bits in TRECR1 are 0. Set TRECR2 at the end of setting the above registers and bits (before the timer RE counting process is started).

- Registers TRESEC, TREMIN, TREHR, TREWK, and TRECR2
- Bits H12_H24 bit, PM, and INT in TRECR1
- Bits RCS0 to RCS3 in TRECSR

### (3) Sampling Circuit for Noise Canceler in ϕ Subclock Signal

When selecting the ϕsub as the clock source for the timer RE, always enable the sampling circuit with the SUBNC[1:0] bits in SYSCCR. For details of the SUBNC[1:0] bits, see section 5.2.2, System Clock Control Register (SYSCCR).

### (4) Restrictions on Clock Selection in Output Compare Mode

In output compare mode, do not select the ϕsub clock as the clock source for the timer if the CPU is in ϕloco mode.

chanhvinh.com



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

chanhvinh.com



# Section 18 Timer RG

Timer RG is a 16-bit timer with output compare and input capture functions. Timer RG can count using a number of internal or external clocks and output pulses with a desired duty cycle using the compare match function between the timer counter and two general registers. Timer RG is also able to decode the phase difference between two external clocks and increment. Timer RG therefore provides an ideal solution for many systems with a requirement to decide position based on a rotary encoder or tachometer as well as a wide range of other applications.

## 18.1 Features

- Selection of six counter clock sources
  Internal clocks: ϕ, ϕ/2, ϕ/4, ϕ/8, and ϕ/32
  External clocks: TCLKA, TCLKB
- Timer mode
  Waveform output by compare match (Selection of 0 output, 1 output, or toggle output)
  Input capture function (Rising edge, falling edge, or both edges)
- PWM mode
  Generates pulses with a desired period and duty cycle.
- Phase counting mode
  Detects phase difference between two external clock inputs and increments/decrements the TCNT.
- Fast access via internal 16-bit bus
  Performs high-speed accesses to the timer counter and general registers using the 16-bit bus interface.
- Four interrupt sources
  TRGCNT overflow, TRGCNT underflow, compare match, and input capture

chanhvinh.com



**Table 18.1 Functions of Timer RG**

| Item | | Counter | Input/Output Pins: TGIOA | Input/Output Pins: TGIOB |
|---|---|---|---|---|
| Counter clock | | Internal clocks: φ, φ/2, φ/4, φ/8, and φ/32<br>External clock: TCLKA, TCLKB | | |
| General registers (multiplexed registers with output compare/input capture) | | — | GRA | GRB |
| Buffer register | | — | BRA | BRB |
| Counter clearing function | | — | Compare match/ input capture | Compare match/ input capture |
| Initial output value setting function | | — | — | — |
| Buffer operation | | — | Yes | Yes |
| Compare match output | 0 output | — | Yes | Yes |
| | 1 output | — | Yes | Yes |
| | Toggle output | — | Yes | Yes |
| Input capture function | | — | Yes | Yes |
| PWM mode | | — | Yes | Yes |
| Phase counting mode | | — | Yes | Yes |
| Interrupt sources | | Overflow<br>Underflow | Compare match/ input capture | Compare match/ input capture |

chanhvinh.com



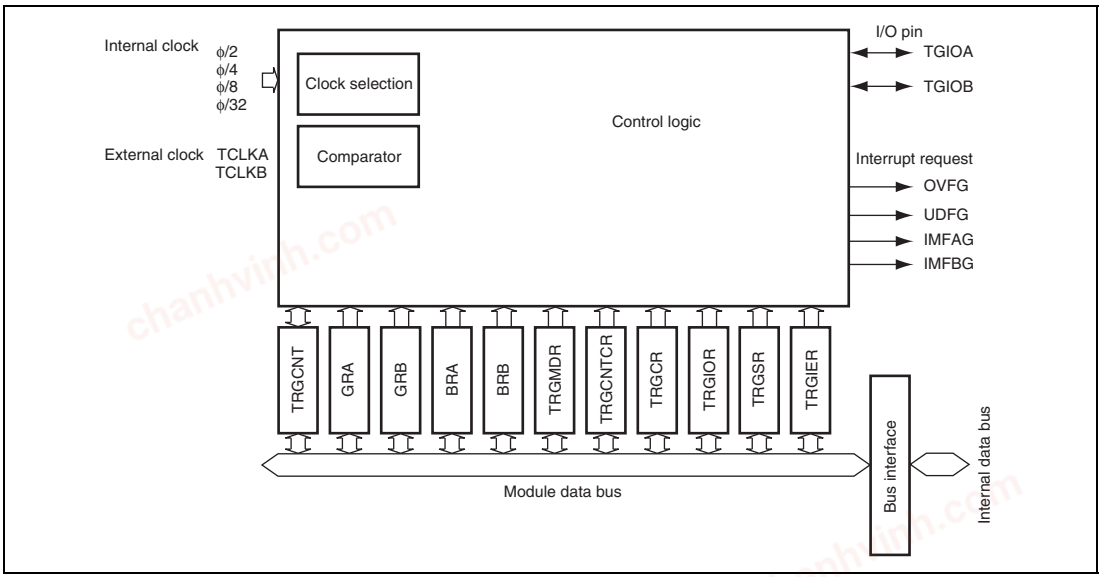



**Figure 18.1 Timer RG Block Diagram**

Table 18.2 summarizes the timer RG pins.

**Table 18.2 Pin Configuration**

| Pin Name | I/O | Function |
|---|---|---|
| TCLKA | Input | External clock A input pin (Phase A input pin in phase counting mode) |
| TCLKB | Input | External clock B input pin (Phase B input pin in phase counting mode) |
| TGIOA | I/O | GRA output compare output pin/<br>GRA input capture input pin/<br>PWM output pin in PWM mode |
| TGIOB | I/O | GRB output compare output pin/<br>GRB input capture input pin |

chanhvinh.com



## 18.2 Register Descriptions

Timer RG has the following registers.

- Timer RG mode register (TRGMDR)
- Timer RG counter control register (TRGCNTCR)
- Timer RG control register (TRGCR)
- Timer RG I/O control register (TRGIOR)
- Timer RG status register (TRGSR)
- Timer RG interrupt enable register (TRGIER)
- Timer RG counter (TRGCNT)
- General register A (GRA)
- General register B (GRB)
- GRA buffer register (BRA)
- GRB buffer register (BRB)

chanhvinh.com
chanhvinh.com
chanhvinh.com

chanhvinh.com



### 18.2.1 Timer RG Mode Register (TRGMDR)

Address: H'FF0646

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | STR | — | DFCK[1:0] | | DFB | DFA | MDF | PWM |
| Value after reset: | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | STR | Counter start | 0: TRGCNT stops counting.<br>1: TRGCNT performs counting. | R/W |
| 6 | — | Reserved | This bit is read as 1. The write value should be 1. | — |
| 5, 4 | DFCK[1:0] | Digital filter clock select | 00: φ/32 (initial value)<br>01: φ/8<br>10: φ<br>11: φ/32Clock specified by bits CKS2 to CKS0 in TRGCR | R/W |
| 3 | DFB | TGIOB pin digital filter function select | 0: Disables the digital filter for the TGIOB pin.<br>1: Enables the digital filter for the TGIOB pin. | R/W |
| 2 | DFA | TGIOB pin digital filter function select | 0: Disables the digital filter for the TGIOA pin.<br>1: Enables the digital filter for the TGIOA pin. | R/W |
| 1 | MDF | Phase counting mode select | 0: Increments the counter.*1<br>1: Phase counting mode | R/W |
| 0 | PWM | PWM mode select | 0: Usual mode*2<br>1: PWM mode | R/W |

Notes: 1. Select counting up in PWM mode.
2. Select normal mode here when the MDF bit is set for phase counting mode.

- STR bit (Counter start)

  Clearing this bit to 0 stops counting by TRGCNT. Counting by TRGCNT proceeds while this bit is set to 1.

  This bit is set to 1 if the specified event occurs when operation of timer RG has been selected in ELOPC of the event link controller.

chanhvinh.com



- MDF bit (Phase counting mode select)

  When this bit is 0, the counter counts the clock pulses specified with the TPSC2 to TPSC0 bits in TRGCR. When this bit is 1, the counter counts the phases produced by TCLKA and TCLKB as specified in TRGCNTCR.

### 18.2.2 Timer RG Counter Control Register (TRGCNTCR)

Address: H'FF0647

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | CNTEN7 | CNTEN6 | CNTEN5 | CNTEN4 | CNTEN3 | CNTEN2 | CNTEN1 | CNTEN0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | CNTEN7 | Count enable bit 7 | 0: Not affected by the TCLKB rising edge when TCLKA is low.<br>1: Incremented at the TCLKB rising edge when TCLKA is low. | R/W |
| 6 | CNTEN6 | Count enable bit 6 | 0: Not affected by the TCLKA rising edge when TCLKB is high.<br>1: Incremented at the TCLKA rising edge when TCLKB is high. | R/W |
| 5 | CNTEN5 | Count enable bit 5 | 0: Not affected by the TCLKB falling edge when TCLKA is high.<br>1: Incremented at the TCLKB falling edge when TCLKA is high. | R/W |
| 4 | CNTEN4 | Count enable bit 4 | 0: Not affected by the TCLKA falling edge when TCLKB is low.<br>1: Incremented at the TCLKA falling edge when TCLKB is low. | R/W |
| 3 | CNTEN3 | Count enable bit 3 | 0: Not affected by the TCLKA falling edge when TCLKB is high.<br>1: Incremented at the TCLKA falling edge when TCLKB is high. | R/W |
| 2 | CNTEN2 | Count enable bit 2 | 0: Not affected by the TCLKB falling edge when TCLKA is low.<br>1: Incremented at the TCLKB falling edge when TCLKA is low. | R/W |
| 1 | CNTEN1 | Count enable bit 1 | 0: Not affected by the TCLKA rising edge when TCLKB is low.<br>1: Incremented at the TCLKA rising edge when TCLKB is low. | R/W |
| 0 | CNTEN0 | Count enable bit 0 | 0: Not affected by the TCLKB rising edge when TCLKA is high.<br>1: Incremented at the TCLKB rising edge when TCLKA is high. | R/W |

chanhvinh.com



### 18.2.3 Timer RG Control Register (TRGCR)

Address: H'FF0648

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | CCLR[1:0] | | CKEG[1:0] | | TPSC[2:0] | | |
| Value after reset: | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is read as 1. The write value should be 1. | — |
| 6, 5 | CCLR[1:0] | Counter clear source select | 00: Disables clearing TRGCNT.<br>01: Clears TRGCNT with a GRA compare match/input capture.<br>1X: Clears TRGCNT with a GRB compare match/input capture. | R/W |
| 4, 3 | CKEG[1:0] | External clock detection edge select | 00: Incremented at the rising edges.<br>01: Incremented at the falling edges.<br>1x: Incremented at the rising and falling edges. | R/W |
| 2 to 0 | TPSC[2:0] | TRGCNT count clock select | 000: TRGCNT counts the internal clock φ<br>001: TRGCNT counts the internal clock φ/2<br>010: TRGCNT counts the internal clock φ/4<br>011: TRGCNT counts the internal clock φ/8<br>100: TRGCNT counts the internal clock φ/32<br>101: TRGCNT counts the TCLKA pin input<br>110: Setting prohibited<br>111: TRGCNT counts the TCLKB pin input | R/W |

[Legend]
X: Don't care.

- CKEG[1:0] bits (external clock detection edge select)
  Selects an edge of the external clock to be detected. When phase counting mode is used, the phase counting operation is performed regardless of the CKEG[1:0] setting.

- TPSC[2:0] bits (TRGCNT count clock select)
  The settings are invalid in phase counting mode.

chanhvinh.com



### 18.2.4 Timer RG I/O Control Register (TRGIOR)

Address: H'FF0649

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | BUFB | IOB2 | IOB[1:0] | | BUFA | IOA2 | IOA[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | BUFB | BRB function select | 0: BRB does not function as the GRB buffer register.<br>1: BRB functions as the GRB buffer register. | R/W |
| 6 | IOB2 | GRB function select | 0: GRB is used as a compare match register.<br>1: GRB is used as an input capture register. | R/W |
| 5, 4 | IOB[1:0] | GRB I/O function select | When IOB2 = 0,<br>00: Disables pin output at a compare match.<br>01: Outputs 0 to the TGIOB pin at a GRB compare match.<br>10: Outputs 1 to the TGIOB pin at a GRB compare match.<br>11: Toggles the output to the TGIOB pin at a GRB compare match.<br>When IOB2 = 1,<br>00: Input capture to GRB at the rising edge of the TGIOB pin.<br>01: Input capture to GRB at the falling edge of the TGIOB pin.<br>1X: Input capture to GRB at the rising and falling edges of the TGIOB pin. | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 3 | BUFA | BRA function select | 0: BRA does not function as the GRA buffer register.<br>1: BRA functions as the GRA buffer. | R/W |
| 2 | IOA2 | GRA function select | 0: GRA is used as a compare match register.<br>1: GRA is used as an input capture register. | R/W |
| 1, 0 | IOA[1:0] | GRA I/O function select | When IOA2 = 0,<br>00: Disables pin output at a compare match.<br>01: Outputs 0 to the TGIOA pin at a GRA compare match.<br>10: Outputs 1 to the TGIOA pin at a GRA compare match.<br>11: Toggles the output to the TGIOA pin at a GRA compare match.<br>When IOA2 = 1,<br>00: Input capture to GRA at the rising edge of the TGIOA pin.<br>01: Input capture to GRA at the falling edge of the TGIOA pin.<br>1X: Input capture to GRA at the rising and falling edges of the TGIOA pin. | R/W |

[Legend]

X: Don't care.

chanhvinh.com



### 18.2.5 Timer RG Status Register (TRGSR)

Address: H'FF064A

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | DIRF | OVF | UDF | IMFB | IMFA |
| Value after reset: | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 to 5 | — | Reserved | These bits are read as 1. The write value should be 1. | — |
| 4 | DIRF | Count direction flag | 0: TRGCNT is decremented.<br>1: TRGCNT is incremented. | R |
| 3 | OVF | Overflow flag | [Setting condition]<br>• When TRGCNT overflows from H'FFFF to H'0000<br>[Clearing condition]<br>• When OVF is read when OVF = 1, then 0 is written to. | R/W |
| 2 | UDF | Underflow flag | [Setting condition]<br>• When TRGCNT underflows from H'0000 to H'FFFF<br>[Clearing condition]<br>• When UDF is read when UDF = 1, then 0 is written to.<br>UDF is valid when phase counting mode is used (MDF in TRGMDR is 1). | R/W |
| 1 | IMFB | Input capture/ compare match flag B | [Setting conditions]<br>• TRGCNT = GRB when GRB functions as an output compare register<br>• The TRGCNT value is transferred to GRB by an input capture signal when GRB functions as an input capture register<br>[Clearing condition]<br>• When the DTC is activated by a IMFB interrupt, and the DISEL bit in MRB of the DTC is 0.<br>• When IMFB is read when IMFB = 1, then 0 is written to. | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 0 | IMFA | Input capture/ compare match flag A | [Setting conditions]<br>• TRGCNT = GRA when GRA functions as an output compare register<br>• The TRGCNT value is transferred to GRA by an input capture signal when GRA functions as an input capture register<br>[Clearing condition]<br>• When the DTC is activated by an IMFA interrupt, and the DISEL bit in MRB of the DTC is 0.<br>• When IMFA is read when IMFA = 1, then 0 is written to. | R/W |

### 18.2.6 Timer RG Interrupt Enable Register (TRGIER)

TRGIER is a register that controls interrupt requests of timer RG.

Address: H'FF064B

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | OVIE | UDIE | IMIEB | IMIEA |
| Value after reset: | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 to 4 | — | Reserved | These bits are read as 1. The write value should always be 1. | — |
| 3 | OVIE | Overflow interrupt enable | 0: Interrupt by the OVF flag is disabled.<br>1: Interrupt by the OVF flag is enabled. | R/W |
| 2 | UDIE | Underflow interrupt enable | 0: Interrupt by the UDF flag is disabled.<br>1: Interrupt by the UDF flag is enabled. | R/W |
| 1 | IMIEB | Input capture/ compare match B enable | 0: Interrupt by the IMFB flag is disabled.<br>1: Interrupt by the IMFB flag is enabled. | R/W |
| 0 | IMIEA | Input capture/ compare match A enable | 0: Interrupt by the IMFA flag is disabled.<br>1: Interrupt by the IMFA flag is enabled. | R/W |

chanhvinh.com



### 18.2.7 Timer RG Counter (TRGCNT)

Address: H'FF0640

| Bit: | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

TRGCNT is a 16-bit readable/writable register that performs count operation with an input clock. The input clock is selected by bits TPSC2 to TPSC0 in TRGCR.

TRGCNT is incremented or decremented in phase counting mode and is only incremented in other modes.

TRGCNT can be cleared to H'0000 by a compare match with the relevant GRA or GRB or by an input capture to GRA or GRB (counter clearing function).

When TRGCNT overflows (changes from H'FFFF to H'0000), the OVF flag in TRGSR is set to 1. When TRGCNT underflows (changes from H'0000 to H'FFFF), the UDF flag in TRGSR is set to 1.

TRGCNT must always be read from or written to in units of 16 bits; 8-bit accesses are not allowed. TRGCNT is initialized to H'0000 by a reset.

chanhvinh.com



### 18.2.8 General Registers A and B (GRA, GRB), GRA and GRB Buffer Registers (BRA, BRB)

GRA

Address: H'FF0642

| Bit: | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

GRB

Address: H'FF0644

| Bit: | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

BRA

Address: H'FF064C

| Bit: | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

BRB

Address: H'FF064E

| Bit: | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Each of GRA and GRB is a 16-bit readable/writable register that can function as either an output-compare register or an input-capture register. The function is selected with TRGIOR.

When a general register is used as an output-compare register, its value is constantly compared with the TRGCNT value. When the two values match (a compare match), the corresponding flag (the IMFA or IMFB bit) in TRGSR is set to 1. A compare match output can be selected in TRGIOR.

When a general register is used as an input-capture register, an external input-capture signal is detected and the current TRGCNT value is stored in the general register. The corresponding flag (the IMFA or IMFB bit) in TRGSR is set to 1. The edge of the input-capture signal is selected in TRGIOR. The setting of TRGIOR is ignored in PWM mode.

chanhvinh.com



BRA and BRB can be used as buffer registers of GRA and GRB, respectively, by setting BUFA and BUFB in TRGIOR.

For example, when GRA is set as an output-compare register and BRA is set as the buffer register for GRA, the value in TRGCNT is sent to GRA whenever compare match A is generated.

When GRA is set as an input-capture register and BRA is set as the buffer register for GRA, the value in TRGCNT is transferred to GRA and the value in GRA is transferred to the buffer register BRA whenever an input capture is generated.

General registers and buffer registers must be written or read in 16-bit units. General registers are set as output compare registers and initialized to H'FFFF by a reset.

chanhvinh.com



## 18.3 Operation

Timer RG has the following operating modes.

- Timer mode (the waveform output function by a compare match, and the input-capture function)
- PWM mode
- Phase counting mode

The TGIOA and TGIOB pins indicate the functions by each register setting.

- TGIOA pin

| Register Name | PMR | PCR | TRGMDR | TRGIOR | |
|---|---|---|---|---|---|
| **Bit Name** | **PMR** | **PCR** | **PWM** | **IOA2 to IOA0** | **Function** |
| Setting values | 1 | X | 1 | XXX | PWM mode waveform output |
| | | X | 0 | 001, 01X | Timer mode waveform output (output compare function) |
| | | 0 | 0 | 1XX | Timer mode (input capture function) |
| | 0 | 1 | X | XXX | General output port |
| | | 0 | X | XXX | General input port |
| | Other than the above | | | | Setting prohibited |

[Legend]
X: Don't care.

chanhvinh.com



- TGIOB pin

| Register Name | PMR | PCR | TRGMDR | TRGIOR | |
|---|---|---|---|---|---|
| **Bit Name** | **PMR** | **PCR** | **PWM** | **IOB2 to IOB0** | **Function** |
| Setting values | 1 | X | 0 | 001, 01X | Timer mode waveform output (output compare function) |
| | | 0 | 0 | 1XX | Timer mode (input capture function) |
| | 0 | 1 | X | XXX | General output port |
| | | 0 | X | XXX | General input port |
| | Other than the above | | | | Setting prohibited |

[Legend]
X: Don't care.

### 18.3.1 Timer Mode

TRGCNT performs up-counting, and is also capable of free-running operation, periodic counting, and external event counting.

Each of GRA and GRB can be used as an input capture register or output compare register.

**(1) Waveform Output by Compare Match:**

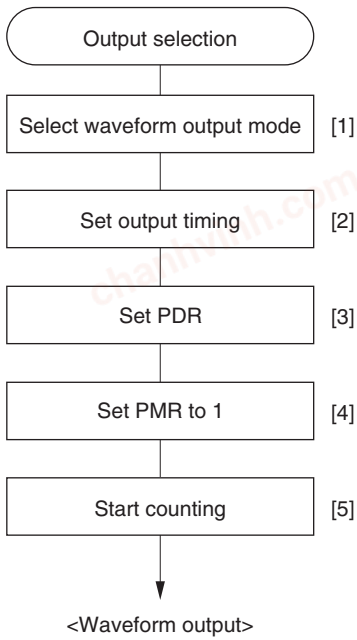
The TPU can perform 0, 1, or toggle output from the corresponding output pin using a compare match.

**(a) Example of setting procedure for waveform output by compare match**

Figure 18.2 shows an example of the setting procedure for waveform output by a compare match.

chanhvinh.com






[1] Select 0 output, 1 output, or toggle output for compare match output by means of TRGIOR.

[2] Set the timing for compare match generation in GRA/GRB.

[3] Write the same value output initially from the compare match output pin (TGIOA or TGIOB) to the PDR bit of the corresponding pin specified by the PMC.

[4] When the PMR bit of the corresponding pin specified by the PMC is set to 1, the port functions as the compare match output pin.

[5] Set the STR bit in TRGMDR to 1 to start the TRGCNT count operation.

**Figure 18.2 Example of Setting Procedure for Waveform Output by Compare Match**

**Table 18.3 Initial Output Values until the First Compare Match Occurs**

| Pin | 0 is Output by Compare Match | 1 is Output by Compare Match | Output is Toggled by Compare Match |
|---|---|---|---|
| TGIOA | 1 | 0 | 0* |
| TGIOB | 1 | 0 | 0* |

Note: * When the initial toggled output immediately after release from the reset state is selected. In case where switching was from another output, the output value is that which preceded the switch.

chanhvinh.com



### (b) Examples of waveform output operation

Figure 18.3 shows an example of 0 output/1 output.

In this example, TRGCNT has been designated as a free-running counter, and settings have been made so that 1 is output by compare match A, and 0 is output by compare match B. When the set level and the pin level match, the pin level does not change.

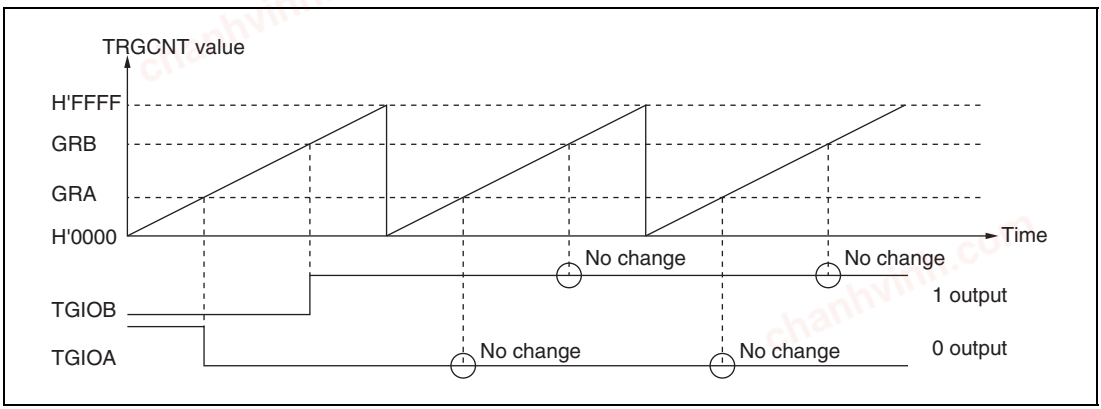



**Figure 18.3 Example of 0 Output/1 Output Operation**

Figure 18.4 shows an example of toggle output. In this example TRGCNT has been designated as a periodic counter (with counter clearing performed by compare match B), and settings have been made so that output is toggled by both compare match A and compare match B.

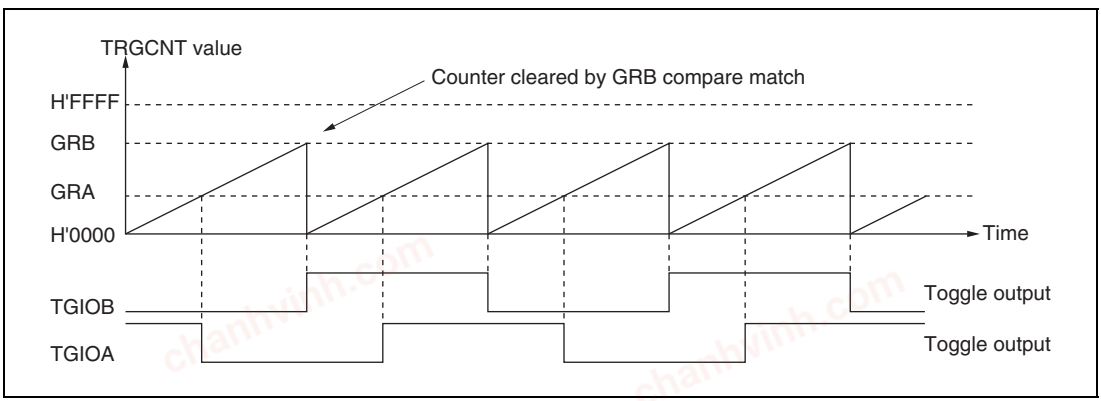



**Figure 18.4 Example of Toggle Output Operation**

chanhvinh.com



### (c) Output compare output timing

A compare match signal is generated in the final state in which TRGCNT and GR match (the point at which the count value matched by TRGCNT is updated). When a compare match signal is generated, the output value set in TRGIOR is output at the output compare output pin (TGIOA, TGIOB). After a match between TRGCNT and GR, the compare match signal is not generated until the TRGCNT input clock is generated.

Figure 18.5 shows output compare output timing.

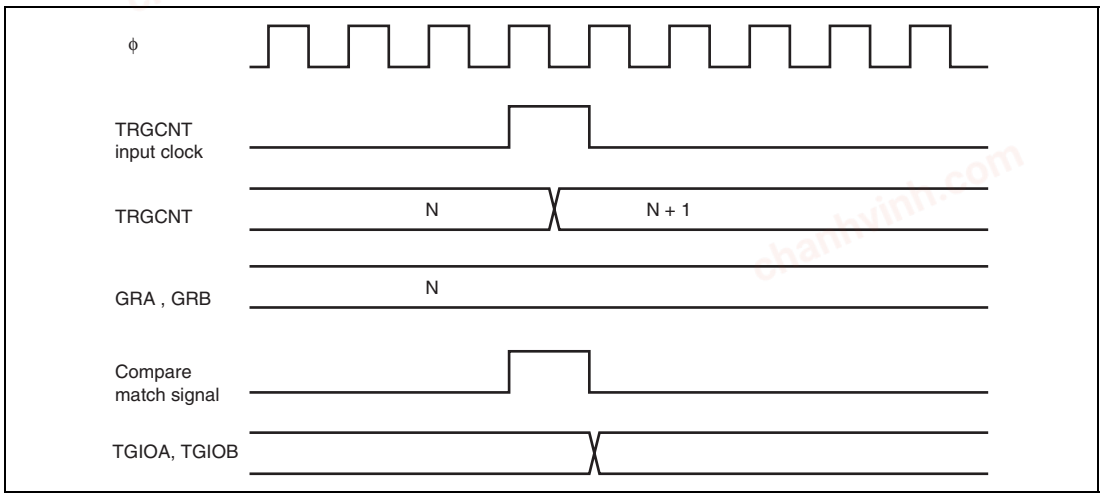



**Figure 18.5 Output Compare Output Timing**

chanhvinh.com

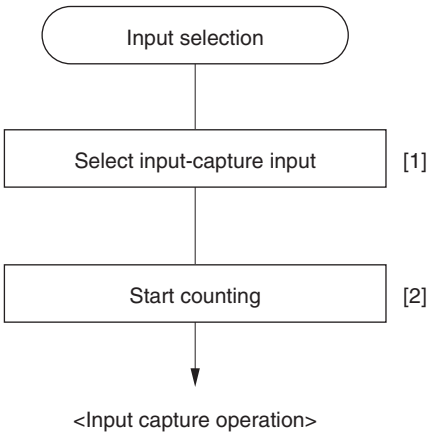


### (2) Input Capture Function

The TRGCNT value can be transferred to GR on detection of the input-capture/output-compare pin (TGIOA, TGIOB) input edge. Rising edge, falling edge, or both edges can be selected as the detection edge. The pulse width and cycle period can be measured using the input capture function.

#### (a) Example of setting procedure for input capture operation

Figure 18.6 shows an example of the setting procedure for input capture operation.




[1] Designate GR as an input capture register by means of TRGIOR, and select the input capture signal input edge (rising edge, falling edge, or both edges).
[2] Set the STR bit in TRGMDR to 1 to start the TRGCNT count operation.

**Figure 18.6 Example of Setting Procedure for Input Capture Operation**

chanhvinh.com



### (b) Example of input capture operation

Figure 18.7 shows an example of input capture operation.

In this example both rising and falling edges have been selected as the TGIOA pin input capture input edge, falling edge has been selected as the TGIOB pin input capture input edge, and counter clearing by GRB input capture has been designated for TRGCNT.

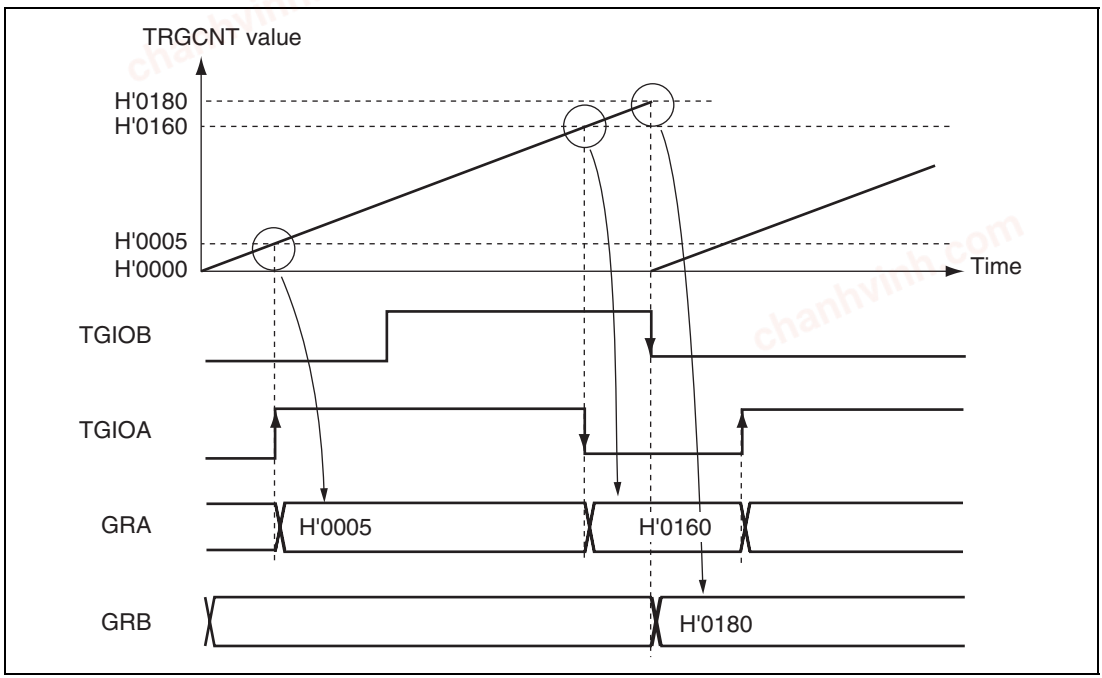



**Figure 18.7 Example of Input Capture Operation**

### (c) Input capture signal timing

Rising edge, falling edge, or both edges can be selected as the detection edge for input capture with TRGIOR.

Figure 18.8 shows input capture signal timing when the falling edge has been selected.

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.

chanhvinh.com



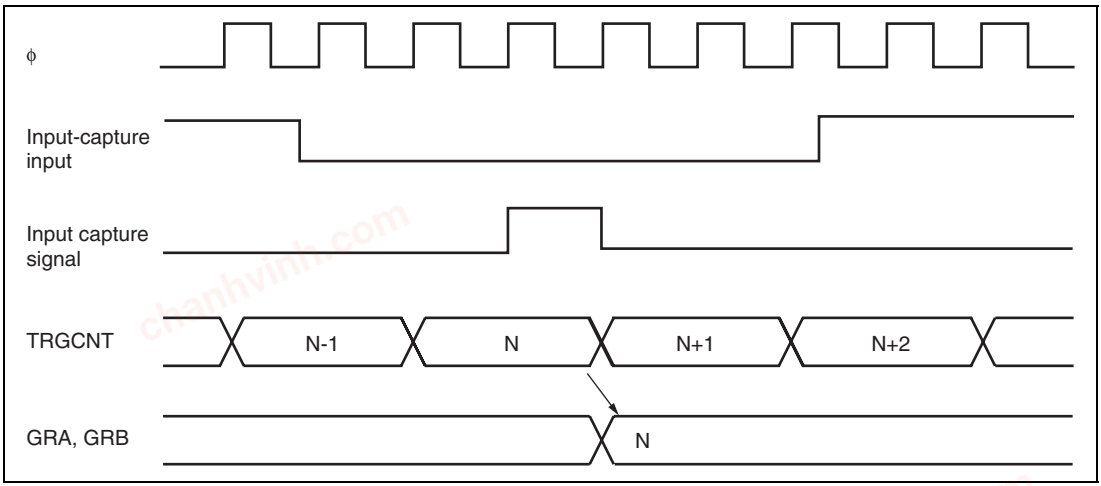



**Figure 18.8 Input Capture Input Signal Timing**

### 18.3.2 PWM Mode

In PWM mode, the PWM waveform is output from the TGIOA output pin by using GRA and GRB as a pair. When an output pin is set for PWM mode, the TRGIOR output setting is ignored. The high level output timing for PWM waveform is set in GRA and the low level output timing in GRB.

Designating GRA or GRB compare match as the TRGCNT counter clearing source enables outputting a PWM waveform in the range of 0% to 100% duty cycle from the TGIOA pin.

The correspondence between PWM output pins and registers is shown in table 18.4. When the same value is set in GRA and GRB, the output value does not change even if a compare match occurs.

**Table 18.4 PWM Output Pins and Registers**

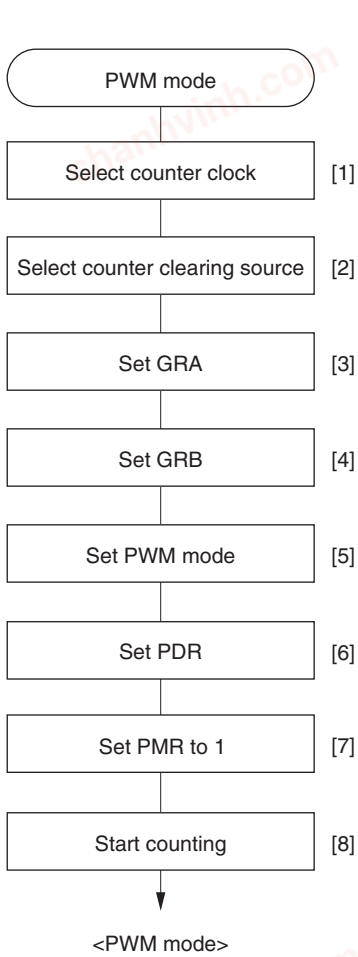
| Output Pin | Output 1 | Output 0 |
|---|---|---|
| TGIOA | GRA | GRB |
| TGIOB | Functions as general I/O port | |

chanhvinh.com



### (1) Example of PWM Mode Setting Procedure

Figure 18.9 shows an example of the PWM mode setting procedure.




[1] Select the counter clock with bits TPSC2 to TPSC0 in TRGCR. When an external clock is selected, select the external clock edge with bits CKEG1 and CKEG0 in TRGCR.

[2] Use bits CCLR1 and CCLR0 in TRGCR to select the counter clearing source.

[3] Set the 1-output timing for the output PWM waveform with GRA.

[4] Set the 0-output timing for the output PWM waveform with GRB.

[5] Select the PWM mode with the PWM bit in TRGMDR. When PWM mode is set, GRA and GRB are used as output compare registers for setting 1-output/0-output for PWM output waveform, regardless of the TRGIOR setting.

[6] Write the same value output initially from the PWM output pin (TGIOA) to the PDR bit of the corresponding pin specified by the PMC.

[7] When the PMR bit of the corresponding pin specified by the PMC is set to 1, the TGIOA pin functions as the PWM output pin.

[8] Set the STR bit in TRGMDR to 1 to start the count operation.

**Figure 18.9 Example of PWM Mode Setting Procedure**

chanhvinh.com



### (2) Examples of PWM Mode Operation

Figure 18.10 shows examples of PWM mode operation.

When PWM mode is set, the TGIOA pin is automatically set as an output pin. The TGIOA pin outputs 1 on a GRA compare match and outputs 0 on a GRB compare match. The TGIOB pin always functions as an I/O pin for the relevant port.

In the examples shown in the figure, GRA and GRB compare matches are set as the TRGCNT clearing source. The initial value of TGIOA differs according to the counter clearing source. The correspondence between counter clearing sources and initial values is shown in table 18.5.

**Table 18.5 Correspondence between Counter Clearing Sources and TGIOA Initial Values**

| Counter Clearing Source | TGIOA Initial Value |
|---|---|
| GRA compare match | 1 |
| GRB compare match | 0 |

chanhvinh.com



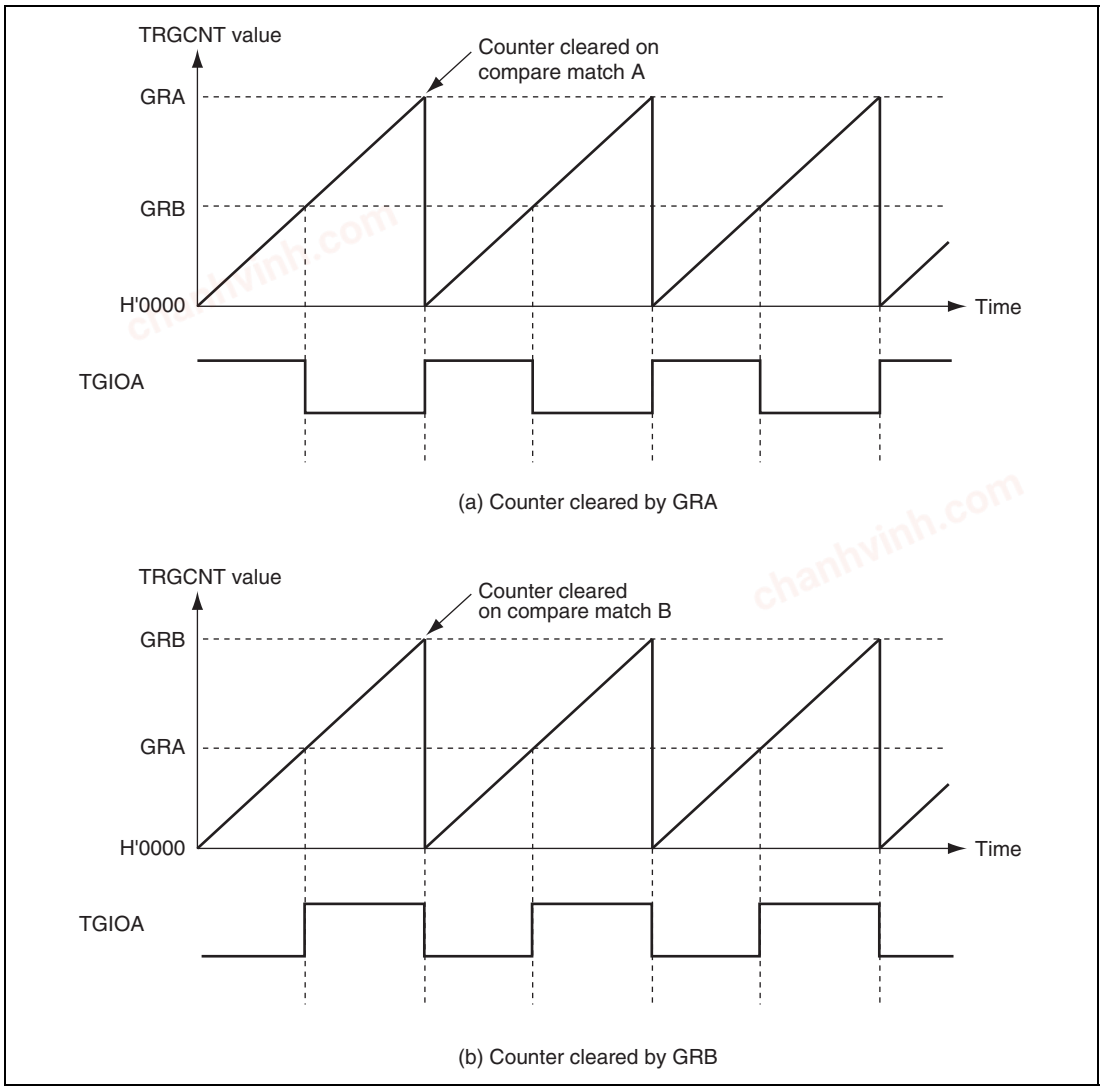



**Figure 18.10 Example of PWM Mode Operation (1)**

Figure 18.11 shows examples of PWM waveform output with 0% duty cycle and 100% duty cycle in PWM mode. When GRB compare match is set as the counter clearing source and the set value in GRA is greater than the value in GRB, the duty cycle of the PWM waveform is 0%. When GRA compare match is set as the counter clearing source and the set value in GRB is greater than the value in GRA, the duty cycle is 100%.

chanhvinh.com



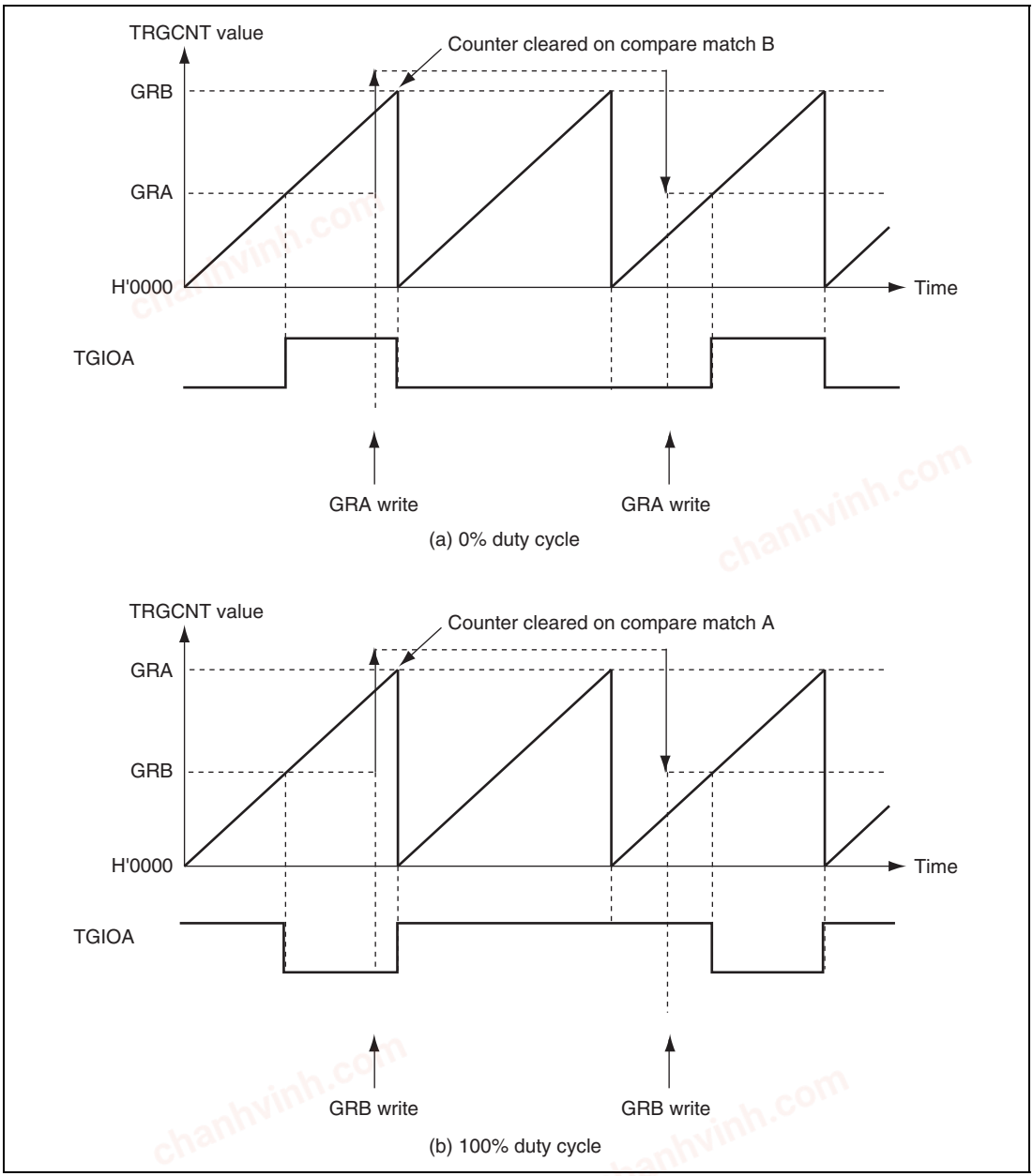



**Figure 18.11 Example of PWM Mode Operation (2)**

chanhvinh.com



### 18.3.3 Phase Counting Mode

In phase counting mode, the phase difference between two external clock inputs (TCLKA and TCLKB pins) is detected and TRGCNT is incremented/decremented accordingly.

When phase counting mode is set, the TCLKA and TCLK pins function as external clock input pins and TRGCNT operates as an up/down-counter regardless of the setting of bits TPSC2 to TPSC0 and bits CKEG1 and CKEG0 in TRGCR.

#### (1) Example of Phase Counting Mode Setting Procedure

Figure 18.12 shows an example of the phase counting mode setting procedure.

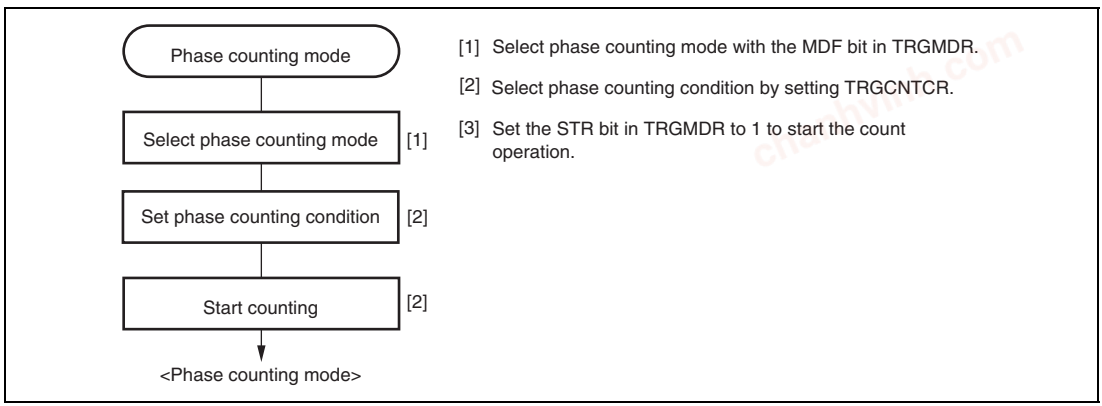



**Figure 18.12 Example of Phase Counting Mode Setting Procedure**

chanhvinh.com



### (2) Examples of Phase Counting Mode Operation

Figures 18.13 to 18.16 show examples of phase counting mode operation, and tables 18.6 to 18.9 summarize the TRGCNT increment/decrement conditions.

**Table 18.6 Increment/Decrement Conditions in Phase Counting Mode Operation Example 1 (TRGCNTCR = H'FF)**

| TRGCNTCR | Set Value | TCLKA | TCLKB | Operation |
|---|---|---|---|---|
| CNTEN7 | 1 | Low level | ↑ | Increment |
| CNTEN6 | 1 | ↑ | High level | |
| CNTEN5 | 1 | High level | ↓ | |
| CNTEN4 | 1 | ↓ | Low level | |
| CNTEN3 | 1 | ↓ | High level | Decrement |
| CNTEN2 | 1 | Low level | ↓ | |
| CNTEN1 | 1 | ↑ | Low level | |
| CNTEN0 | 1 | High level | ↑ | |

[Legend]
↑ : Rising edge
↓ : Falling edge

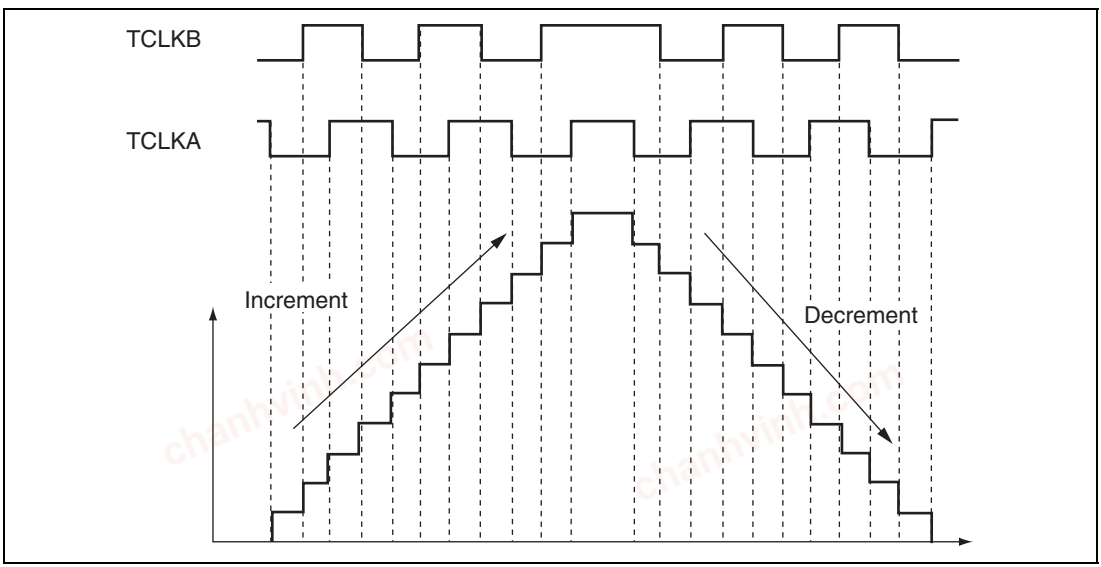



**Figure 18.13 Phase Counting Mode Operation Example 1 (TRGCNTCR = H'FF)**

chanhvinh.com



**Table 18.7 Increment/Decrement Conditions in Phase Counting Mode Operation Example 2 (TRGCNTCR = H'24)**

| TRGCNTCR | Set Value | TCLKA | TCLKB | Operation |
|---|---|---|---|---|
| CNTEN7 | 0 | Low level | ↑ | Don't care |
| CNTEN6 | 0 | ↑ | High level | |
| CNTEN5 | 1 | High level | ↓ | Increment |
| CNTEN4 | 0 | ↓ | Low level | Don't care |
| CNTEN3 | 0 | ↓ | High level | |
| CNTEN2 | 1 | Low level | ↓ | Decrement |
| CNTEN1 | 0 | ↑ | Low level | Don't care |
| CNTEN0 | 0 | High level | ↑ | |

[Legend]
↑ : Rising edge
↓ : Falling edge

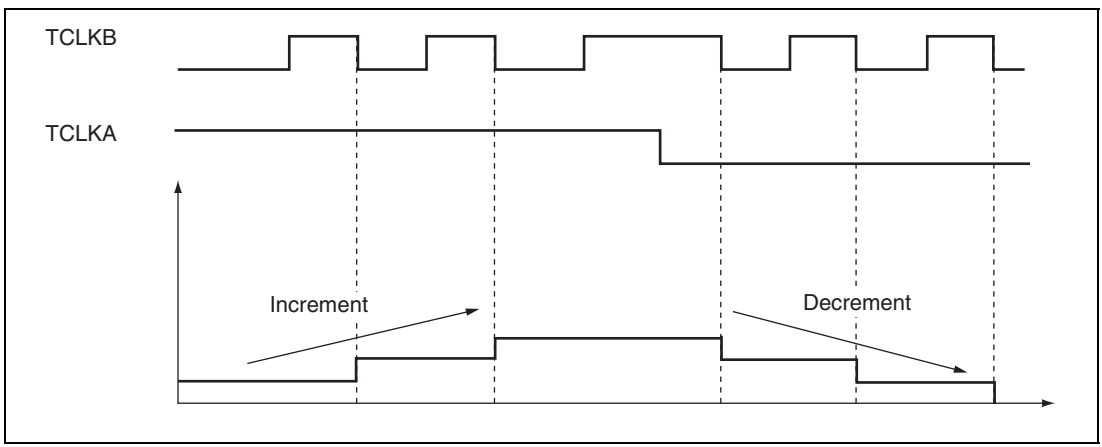



**Figure 18.14 Phase Counting Mode Operation Example 2 (TRGCNTCR = H'24)**

chanhvinh.com



**Table 18.8 Increment/Decrement Conditions in Phase Counting Mode Operation Example 3 (TRGCNTCR = H'28)**

| TRGCNTCR | Set Value | TCLKA | TCLKB | Operation |
|---|---|---|---|---|
| CNTEN7 | 0 | Low level | ↑ | Don't care |
| CNTEN6 | 0 | ↑ | High level | |
| CNTEN5 | 1 | High level | ↓ | Increment |
| CNTEN4 | 0 | ↓ | Low level | Don't care |
| CNTEN3 | 1 | ↓ | High level | Decrement |
| CNTEN2 | 0 | Low level | ↓ | Don't care |
| CNTEN1 | 0 | ↑ | Low level | |
| CNTEN0 | 0 | High level | ↑ | |

[Legend]
↑ : Rising edge
↓ : Falling edge

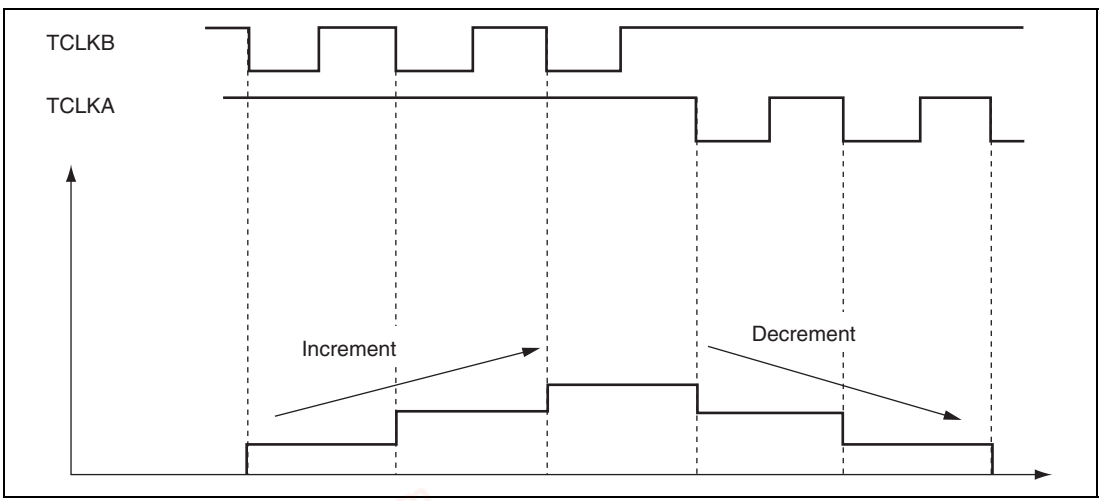



**Figure 18.15 Phase Counting Mode Operation Example 3 (TRGCNTCR = H'28)**

chanhvinh.com



**Table 18.9 Increment/Decrement Conditions in Phase Counting Mode Operation Example 4 (TRGCNTCR = H'5A)**

| TRGCNTCR | Set Value | TCLKA | TCLKB | Operation |
|---|---|---|---|---|
| CNTEN7 | 0 | Low level | ↑ | Don't care |
| CNTEN6 | 1 | ↑ | High level | Increment |
| CNTEN5 | 0 | High level | ↓ | Don't care |
| CNTEN4 | 1 | ↓ | Low level | Increment |
| CNTEN3 | 1 | ↓ | High level | Decrement |
| CNTEN2 | 0 | Low level | ↓ | Don't care |
| CNTEN1 | 1 | ↑ | Low level | Decrement |
| CNTEN0 | 0 | High level | ↑ | Don't care |

[Legend]
↑ : Rising edge
↓ : Falling edge

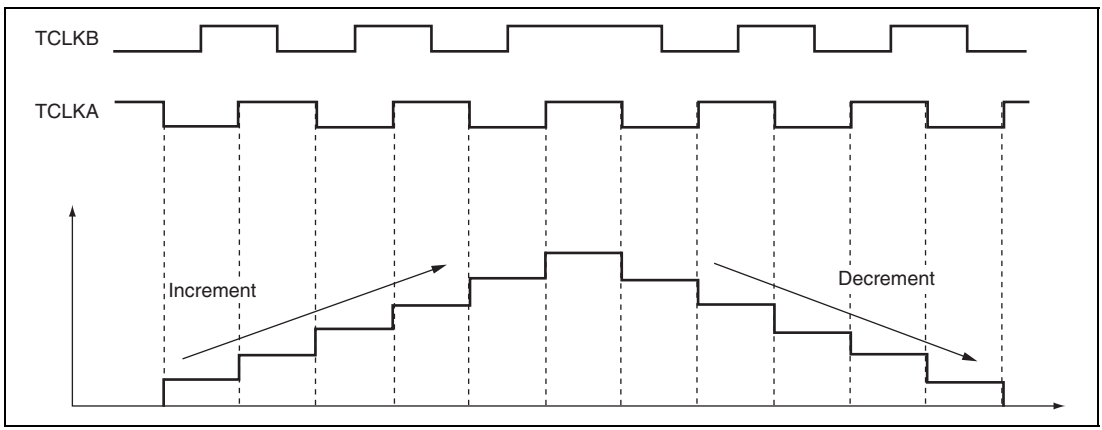



**Figure 18.16 Phase Counting Mode Operation Example 4 (TRGCNTCR = H'5A)**

chanhvinh.com



### (3) Note on Phase Counting Mode

In phase counting mode, the phase difference and overlap between TCLKA and TCLKB must be at least 1.5 × ϕ cycle of the system clock when bits TPSC2 to TPSC0 in TRGCR = B'0XX or B'100, and the pulse width must be at least 3 × ϕ cycle. Figure 18.17 shows the input clock conditions in phase counting mode.

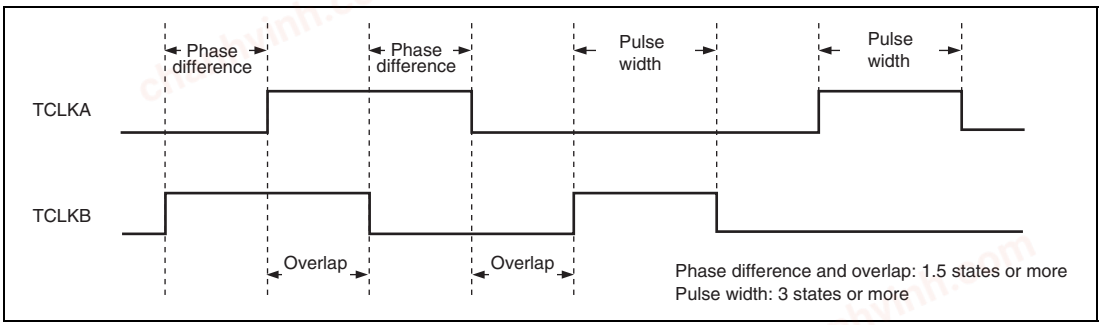



**Figure 18.17 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode**

Note: When CNTEN7 to CNTEN0 in TRGCNTCR are cleared, the counting is not performed even if an increment/decrement condition matches.

### 18.3.4 Buffer Operation

Buffer operation differs according to whether GR has been designated as an output compare register or an input capture register.

Table 18.10 shows the register combinations used in buffer operation.

**Table 18.10 Register Combinations in Buffer Operation**

| General Register | Buffer Register |
|---|---|
| GRA | BRA |
| GRB | BRB |

chanhvinh.com



### (1) When GR is an output compare register

When a compare match occurs, the value in the buffer register for the corresponding channel is transferred to the general register. This operation is illustrated in figure 18.18.

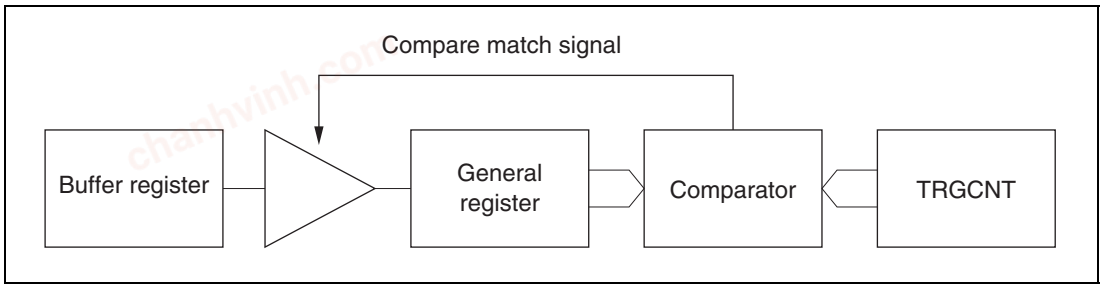



**Figure 18.18 Compare Match Buffer Operation**

### (2) When GR is an input capture register

When input capture occurs, the value in TRGCNT is transferred to GR and the value previously held in the general register is transferred to the buffer register. This operation is illustrated in figure 18.19.

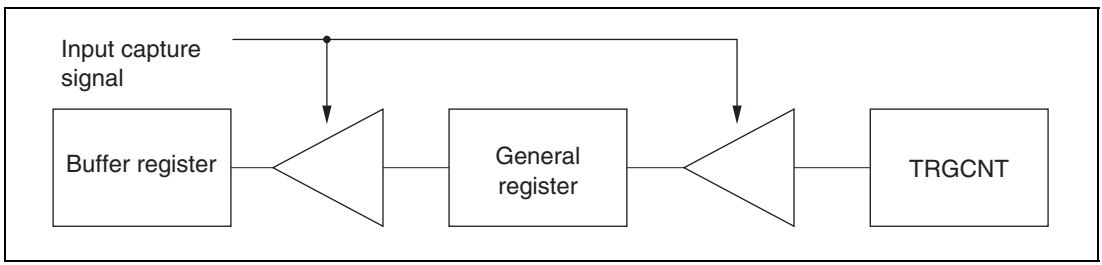



**Figure 18.19 Input Capture Buffer Operation**

chanhvinh.com



Figures 18.20 and 18.21 show the timings in buffer operation.

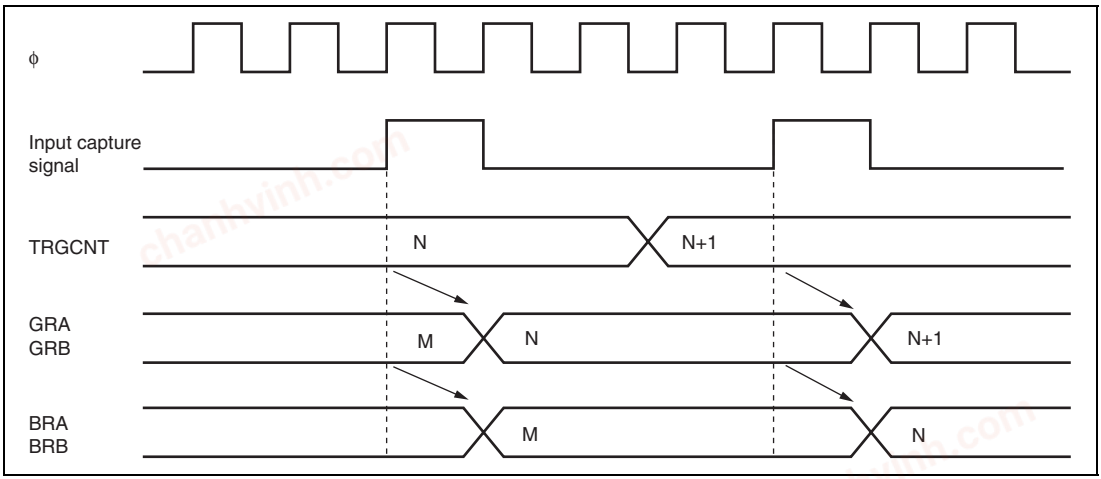



**Figure 18.20 Buffer Operation Timing (Compare Match)**

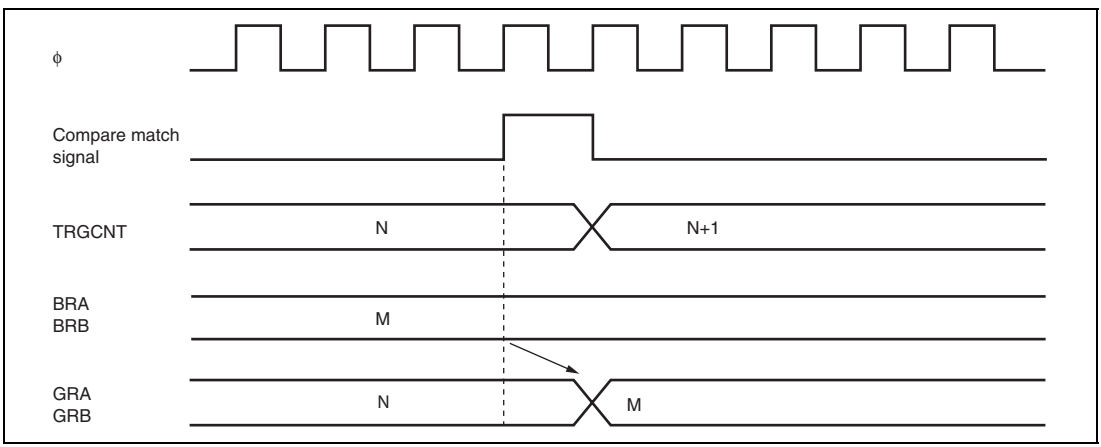



**Figure 18.21 Buffer Operation Timing (Input Capture)**

chanhvinh.com



### 18.3.5 Operation through an Event Link

Using the event link controller (ELC), timer RG can be made to operate in the following ways in relation to events occurring in other modules.

#### (1) Staring Counter Operation

The start of counting operations by timer RG can be selected by ELOPC of the ELC. When the event specified by ELSR8 occur, the STR bit in TRGMDR is set to 1, which starts counting by timer RG. However, if the specified event occurs when the STR bit has already been set to 1, the event is not effective.

#### (2) Counting Event

The counting of events by timer RG can be selected by ELOPC of the ELC. When the event specified in ELSR8 occurs, event counter operation proceeds with that event as the source to drive counting, regardless of the setting of TPSC[2:0] bits in TRGCR and the STR bit in TRGMDR. When the value of the counter is read, the value read out is the actual number of input events.

#### (3) Input Capture

Input capture operation of timer RG can be selected by ELOPC of the ELC. When the event specified in ELSR8 occurs, GRB captures the value of TRGCNT. When input capture operation initiated by an event link is in use, set IOB[2:0] = b'101 in the TRGIOR register of timer RG, set the STR bit in TRGMDR, and then start the counter. Since input on the TGIOB pin becomes valid at the same time, fix the input to the TGIOB pin or take other measures such as not allocating the TGIOB pin to the port in the PMC, etc.

chanhvinh.com



### 18.3.6 Digital Filtering Function for Input Capture Inputs

Input signals on the TGIOA and TGIOB pins can be input via the digital filters. The digital filter includes three latches connected in series and a matching detecting circuit. The input signals on the TGIOA and TGIOB pins are operated on the sampling clock specified by the DFCK1 and DFCK0 bits in TRGMDR. When outputs of the three latches match, the matching detecting circuit outputs the signal level of the input. Otherwise, the output remains unchanged. That is, when a pulse width is equal to or greater than three sampling clock cycles, the pulse is input as a signal. When a pulse width is less than three sampling clock cycles, the pulse is considered as a noise to be removed.

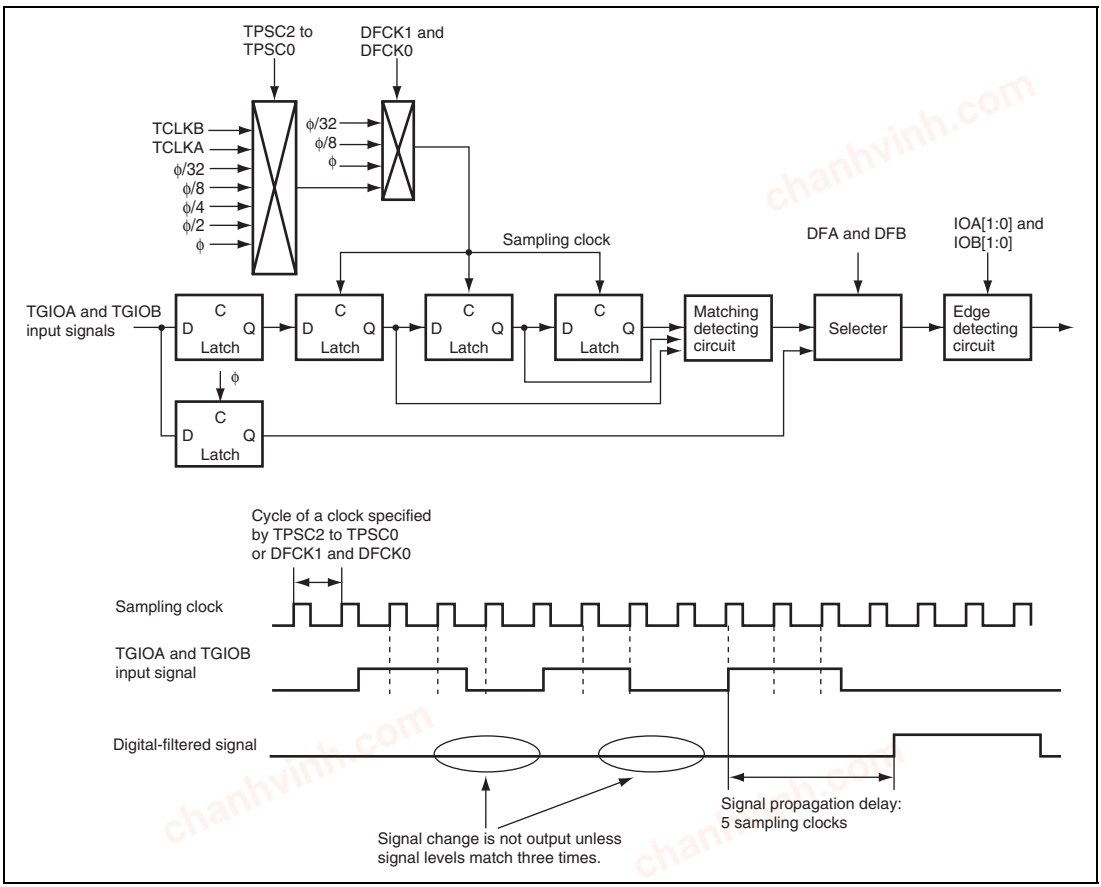



**Figure 18.22 Block Diagram of Digital Filter**

chanhvinh.com



# Section 19 Watchdog Timer (WDT)

The watchdog timer (WDT) is an 8-bit timer that can generate an internal reset signal for this LSI if a system crash prevents the CPU from writing to the timer counter, thus allowing it to overflow.

The block diagram of the watchdog timer is shown in figure 19.1.

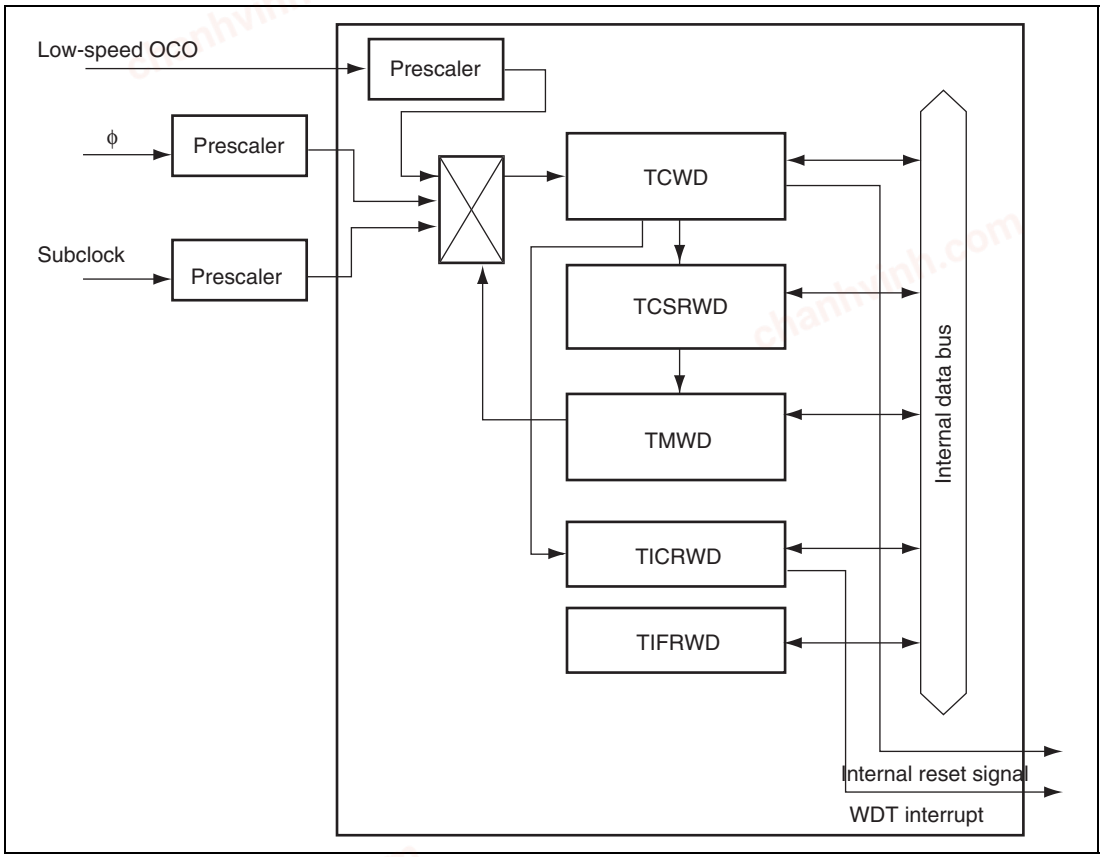



**Figure 19.1 Block Diagram of Watchdog Timer**

chanhvinh.com



## 19.1 Features

- Selectable from fifteen clock sources
  - Eight clocks generated by dividing ϕ: ϕ/64, ϕ/128, ϕ/256, ϕ/512, ϕ/1024, ϕ/2048, ϕ/4096, and ϕ/8192
  - Five clocks generated by dividing low-speed OCO clock: ϕloco/8, ϕloco/32, ϕloco/128, ϕloco/512, and ϕloco/1024
  - Two clocks generated by dividing subclock: ϕsub/4 and ϕsub/256

  When the low-speed OCO clock or subclock is selected, the WDT operates as the watchdog timer in any operating mode.
- Reset signal generated on counter overflow

  An overflow period of 1 to 256 times the selected clock can be set.
- The watchdog timer is enabled in the initial state.

  The watchdog timer starts operating after a reset is released.
- Periodic timer function

  The timer counter can also be used as a periodic timer. Interrupts can be generated with a specific count value.

chanhvinh.com



## 19.2 Register Descriptions

The watchdog timer has the following registers.

- Timer control/status register WD (TCSRWD)
- Timer counter WD (TCWD)
- Timer mode register WD (TMWD)
- Timer interrupt control register WD (TICRWD)
- Timer interrupt flag register WD (TIFRWD)

### 19.2.1 Timer Control/Status Register WD (TCSRWD)

Address: H'FFFF9A

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | B6WI | TCWE | B4WI | TCSRWE | TMWLOCK | TMWI | — | — |
| Value after reset: | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | B6WI | Bit 6 write inhibit | 0: Writing to the TCWE bit (bit 6 in this register) is enabled.<br>1: Writing to the TCWE bit (bit 6 in this register) is disabled.<br>This bit is always read as 1. | R/W |
| 6 | TCWE | Timer counter WD write enable | 0: Writing to the TCWD register is disabled.<br>1: Writing to the TCWD register is enabled.<br>Before writing data to this bit, the B6WI bit must be cleared to 0. | R/W |
| 5 | B4WI | Bit 4 write inhibit | 0: Writing to the TCSRWE bit (bit 4) is enabled.<br>1: Writing to the TCSRWE bit (bit 4) is disabled.<br>This bit is always read as 1. | R/W |
| 4 | TCSRWE | Timer control/status register WD write enable | 0: Writing to TMWLOCK and TMWI (bits 3 and 2 in this register) is disabled.<br>1: 0: Writing to TMWLOCK and TMWI (bits 3 and 2 in this register) is enabled.<br>Before writing data to this bit, the B4WI bit must be cleared to 0. | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 3 | TMWLOCK | Timer mode register WD lockdown | The TMWD register is write-protected when this bit is 1. Once this bit is set to 1, this bit can be cleared only by a reset.<br>0: Writing to the TMWD register is enabled.<br>1: Writing to the TMWD register is disabled.<br>[Setting condition]<br>• When 1 is written to this bit<br>[Clearing condition]<br>• Resetting | R/W |
| 2 | TMWI | Timer mode register WD write inhibit | 0: Writing to the TMWD register is enabled.<br>1: Writing to the TMWD register is disabled.<br>[Setting conditions]<br>• This bit is automatically set to 1 after TMWD is written to.<br>• When 1 is written to this bit.<br>[Clearing condition]<br>• When 0 is written to TMWI while TCSRWE is 1 | R/W |
| 1, 0 | — | Reserved | These bits are read as 1. The write value should always be 1. | — |

Note: TCSRWD must be rewritten by using the MOV instruction. The bit manipulation instruction cannot be used to change the setting value.

### 19.2.2 Timer Counter WD (TCWD)

Address: H'FFFF98

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

TCWD is an 8-bit readable/writable up-counter. When TCWD overflows from H'FF to H'00, the internal reset signal is generated. TCWD is initialized to H'00. TCWD can also be used as a periodic timer. It issues an interrupt request to the CPU when the upper two bits in TCWD are B'01, B'10, or B'11 according to the TICRWD setting.

chanhvinh.com



### 19.2.3 Timer Mode Register WD (TMWD)

Address: H'FFFF99

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | CKS[3:0] | | | |
| Value after reset: | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 to 4 | — | Reserved | These bits are read as 1. The write value should always be 1. | — |
| 3 to 0 | CKS[3:0] | Clock select | 0000: Internal clock: counts on φloco/8 (initial value) | R/W |
| | | | 0001: Internal clock: counts on φloco/32 | |
| | | | 0010: Internal clock: counts on φloco/128 | |
| | | | 0011: Internal clock: counts on φloco/512 | |
| | | | 0100: Internal clock: counts on φloco/1024 | |
| | | | 0101: Internal clock: counts on sub/4 | |
| | | | 0110: Internal clock: counts on φsub/256 | |
| | | | 0111: Clock input prohibited. | |
| | | | 1000: Internal clock: counts on φ/64 | |
| | | | 1001: Internal clock: counts on φ/128 | |
| | | | 1010: Internal clock: counts on φ/256 | |
| | | | 1011: Internal clock: counts on φ/512 | |
| | | | 1100: Internal clock: counts on φ/1024 | |
| | | | 1101: Internal clock: counts on φ/2048 | |
| | | | 1110: Internal clock: counts on φ/4096 | |
| | | | 1111: Internal clock: counts on φ8192 | |

Note: Only write values to this register while the bus master operation clock φs is not being frequency-divided (the value of the PHIS[2:0] bits in LPCR3 is B'000).

- CKS[3:0] bits (clock select)

  The method by which this register is written differs from other registers. The register must be written by using the MOV instruction twice in succession. First, write the data to be loaded to TMWD in a first operation, then write a bit reversal value of the data (b3 to b0) to be loaded in a second operation. When correct operation is executed, CKS[3:0] bits are rewritten after the second write. If the first data and the second reversal data do not match, all bits are not modified. Set CKS[3:0] bits to B'0111 (clock input prohibited) to stop WDT operation.

chanhvinh.com



### 19.2.4 Timer Interrupt Control Register WD (TICRWD)

Address: H'FFFF9B

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | INTSEL[1:0] | | IWIE | — | — | — | — | — |
| Value after reset: | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7, 6 | INTSEL[1:0] | WDT periodic interrupt condition select | 00: Setting prohibited<br>01: An interrupt is generated when the upper two bits in TCWD is B'01.<br>10: An interrupt is generated when the upper two bits in TCWD is B'10.<br>11: An interrupt is generated when the upper two bits in TCWD is B'11. (Initial value) | R/W |
| 5 | IWIE | WDT periodic interrupt enable | 0: Periodic interrupt request is disabled.<br>1: Periodic interrupt request is enabled. | R/W |
| 4 to 0 | — | Reserved | These bits are read as 1. The write value should always be 1. | — |

chanhvinh.com



### 19.2.5 Timer Interrupt Flag Register WD (TIFRWD)

Address: H'FFFF9C

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | IWF | — | — | — | — | — | — | — |
| Value after reset: | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | IWF | WDT periodic interrupt request flag | 0: No periodic interrupt request<br>1: Periodic interrupt request is generated.<br>[Setting condition]<br>• When the upper two bits in the timer counter WD agree with the value set by the INTSEL[1:0] bits in TICRWD.<br>[Clearing condition]<br>• When 0 is written to IWF after reading IWF = 1. | R/W |
| 6 to 0 | — | Reserved | These bits are read as 1. The write value should always be 1. | — |

chanhvinh.com



## 19.3 Operation

### 19.3.1 Watchdog Timer Overflow Reset

The watchdog timer is provided with an 8-bit counter. After a reset is released, TCWD starts counting up. When the TCWD count value overflows H'FF, an internal reset signal is generated. Since TCWD is a writable counter, it starts counting from the value set in TCWD. An overflow period in the range of 1 to 256 input clock cycles can therefore be set, according to the TCWD set value.

When the watchdog timer is not used, write 0 simultaneously to TMWLOCK and TMWI in TCSRWD while the TCSRWE bit is 1 and set CKS[3:0] in TMWD to B'0111 (clock input prohibited).

Figure 19.2 shows an example of watchdog timer operation.

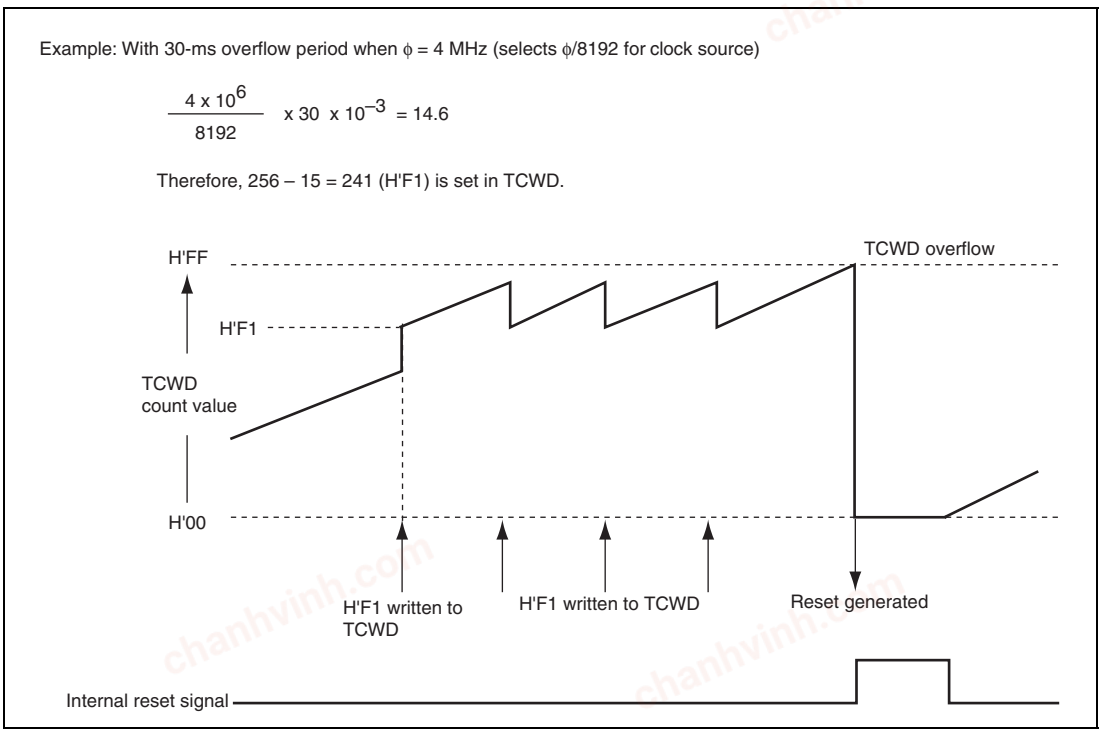



**Figure 19.2 Watchdog Timer Operation Example**

chanhvinh.com



### 19.3.2 Watchdog Timer Setting Flow

The watchdog timer should be set using the procedure shown in figure 19.3.

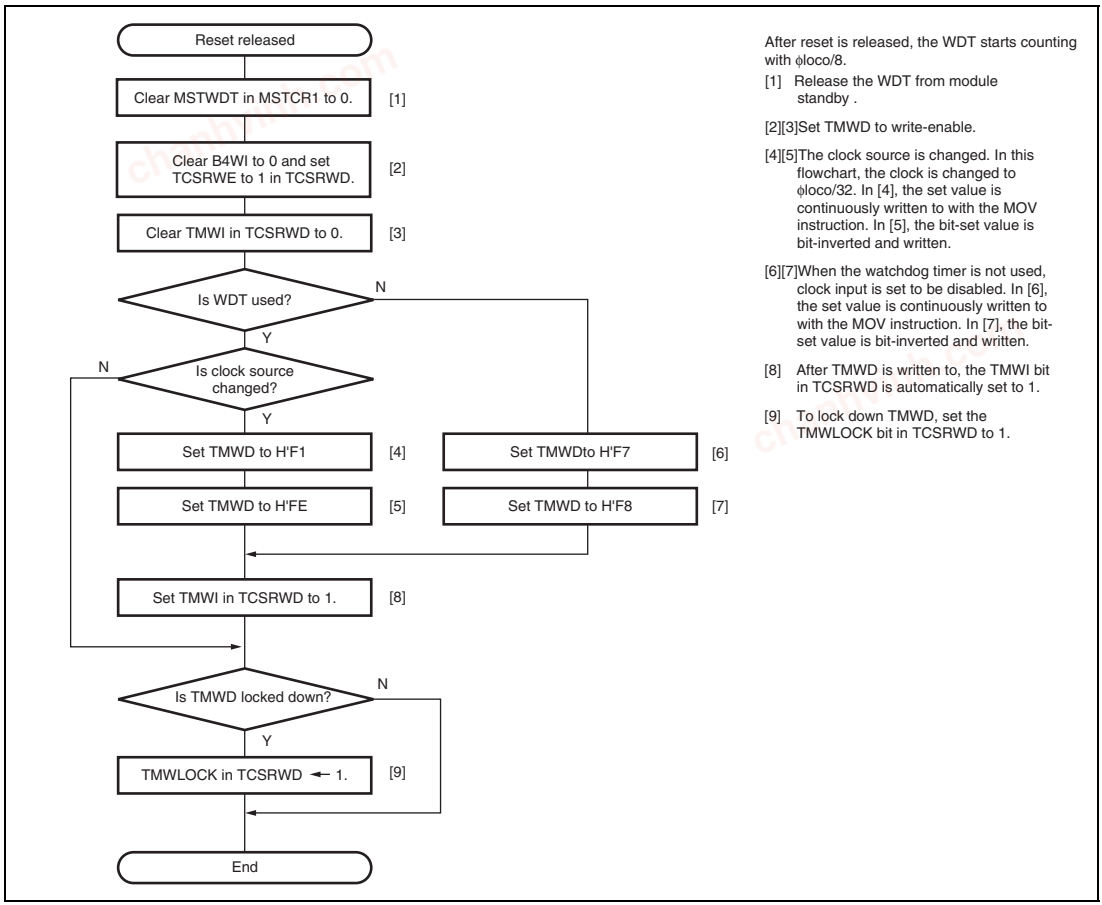



**Figure 19.3 Watchdog Timer Setting Flow**

chanhvinh.com



### 19.3.3 Watchdog Timer Periodic Interrupt

When the INTSEL[1:0] bits in TICRWD are set and the timer WD counter reaches the set value, the IWF bit in TIFRWD is set to 1. At this time, if the IWIE bit in TICRWD is 1, an interrupt request is generated. Figure 19.4 shows the interrupt generation timing when INTSEL is B'01.

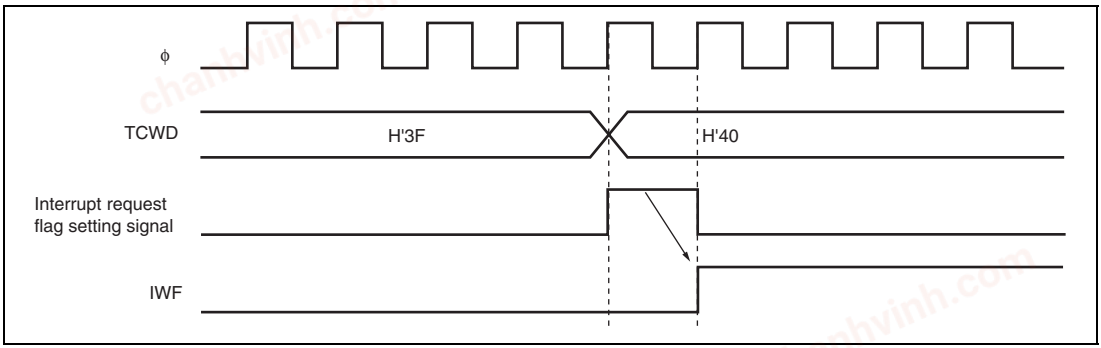



**Figure 19.4 Periodic Interrupt Generation Timing (INTSEL = B'01)**

chanhvinh.com



## 19.4 Usage Notes

### 19.4.1 Notes on System Design

While the watchdog timer is a useful function that restores the LSI to normal condition if the system runs erratically for some reason, the watchdog timer may fail to be reset properly in situations such as the perpetuation of an endless loop in a specific programming routine in which a counter setting operation is executed. Also, there is a possibility of the watchdog timer not being reset properly despite an erratic system condition if an interrupt is enabled and a counter value is set within the interrupt processing.

These notes should be taken into consideration in the system design phases.

### 19.4.2 Notes on Stopping the Watchdog Timer or Switching the Count Clock

The MSTWDT bit in MSTCR1 is set to 1 after release from a reset, but the watchdog timer will operate since ϕloco/8 is selected as the counter clock. (and, since the WDT is in module standby mode, access to the registers is disabled). To stop the watchdog timer or switch the count clock, proceed after releasing the WDT from module standby by clearing the MSTWDT bit in MSTCR1 to 0.

chanhvinh.com



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

chanhvinh.com



# Section 20 Serial Communication Interface 3 (SCI3, IrDA)

This LSI includes a serial communication interface 3 (SCI3), which has three independent channels. The SCI3 can handle both asynchronous and clocked synchronous serial communication. In asynchronous mode, serial data communication can be carried out using standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or an Asynchronous Communication Interface Adapter (ACIA). A function is also provided for serial communication between processors (multiprocessor communication function).

SCI3_2 provides IrDA (Infrared Data Association) communication waveform transmission/reception according IrDA standard version 1.0.

Table 20.1 shows the SCI3 channel configuration and figure 20.1 shows a block diagram of the SCI3. Since pin functions are identical for each of the three channels (SCI3, SCI3_2, and SCI3_3), separate explanations are not given in this section.

## 20.1 Features

- Choice of asynchronous or clocked synchronous serial communication mode
- Full-duplex communication capability

  The transmitter and receiver are mutually independent, enabling transmission and reception to be executed simultaneously.

  Double-buffering is used in both the transmitter and the receiver, enabling continuous transmission and continuous reception.
- On-chip baud rate generator allows any bit rate to be selected
- External clock or on-chip baud rate generator can be selected as a transfer clock source.
- Six interrupt sources

  Transmit-end, transmit-data-empty, receive-data-full, overrun error, framing error, and parity error. The DTC can be activated by the transmit-data-empty interrupt and receive-data-full interrupt sources.

Asynchronous mode

- Data length: 7 or 8 bits
- Stop bit length: 1 or 2 bits
- Parity: Even, odd, or none
- Receive error detection: Parity, overrun, and framing errors
- Break detection: Break can be detected by reading the RXD pin level directly in the case of a framing error

chanhvinh.com



Clocked synchronous mode

- Data length: 8 bits
- Receive error detection: Overrun errors

**Table 20.1 Channel Configuration**

| Channel | Abbreviation | Pin | Register | Register Address | Noise Canceler |
|---|---|---|---|---|---|
| Channel 1 | SCI3*[1] | SCK3<br>RXD<br>TXD | SMR | H'FF0550 | Available |
| | | | BRR | H'FF0551 | |
| | | | SCR3 | H'FF0552 | |
| | | | TDR | H'FF0553 | |
| | | | SSR | H'FF0554 | |
| | | | RDR | H'FF0555 | |
| | | | RSR | — | |
| | | | TSR | — | |
| | | | SPMR | H'FF0556 | |
| Channel 2 | SCI3_2*[2] | SCK3_2<br>RXD_2/IrRxD<br>TXD_2/IrTxD | SMR_2 | H'FF0558 | Available |
| | | | BRR_2 | H'FF0559 | |
| | | | SCR3_2 | H'FF055A | |
| | | | TDR_2 | H'FF055B | |
| | | | SSR_2 | H'FF055C | |
| | | | RDR_2 | H'FF055D | |
| | | | RSR_2 | — | |
| | | | TSR_2 | — | |
| | | | SPMR_2 | H'FF055E | |
| | | | IrCR | H'FF05DE | |

chanhvinh.com



| Channel | Abbreviation | Pin | Register | Register Address | Noise Canceler |
|---|---|---|---|---|---|
| Channel 3 | SCI3_3 | SCK3_3<br>RXD_3<br>TXD_3 | SMR_3 | H'FF0560 | Available |
| | | | BRR_3 | H'FF0561 | |
| | | | SCR3_3 | H'FF0562 | |
| | | | TDR_3 | H'FF0563 | |
| | | | SSR_3 | H'FF0564 | |
| | | | RDR_3 | H'FF0565 | |
| | | | RSR_3 | — | |
| | | | TSR_3 | — | |
| | | | SPMR_3 | H'FF0566 | |

Notes: 1. Channel 1 of the SCI3 is used in on-board programming mode by boot mode.
2. SCI3_2 provides IrDA (Infrared Data Association) communication waveform transmission/reception according IrDA standard version 1.0.

chanhvinh.com



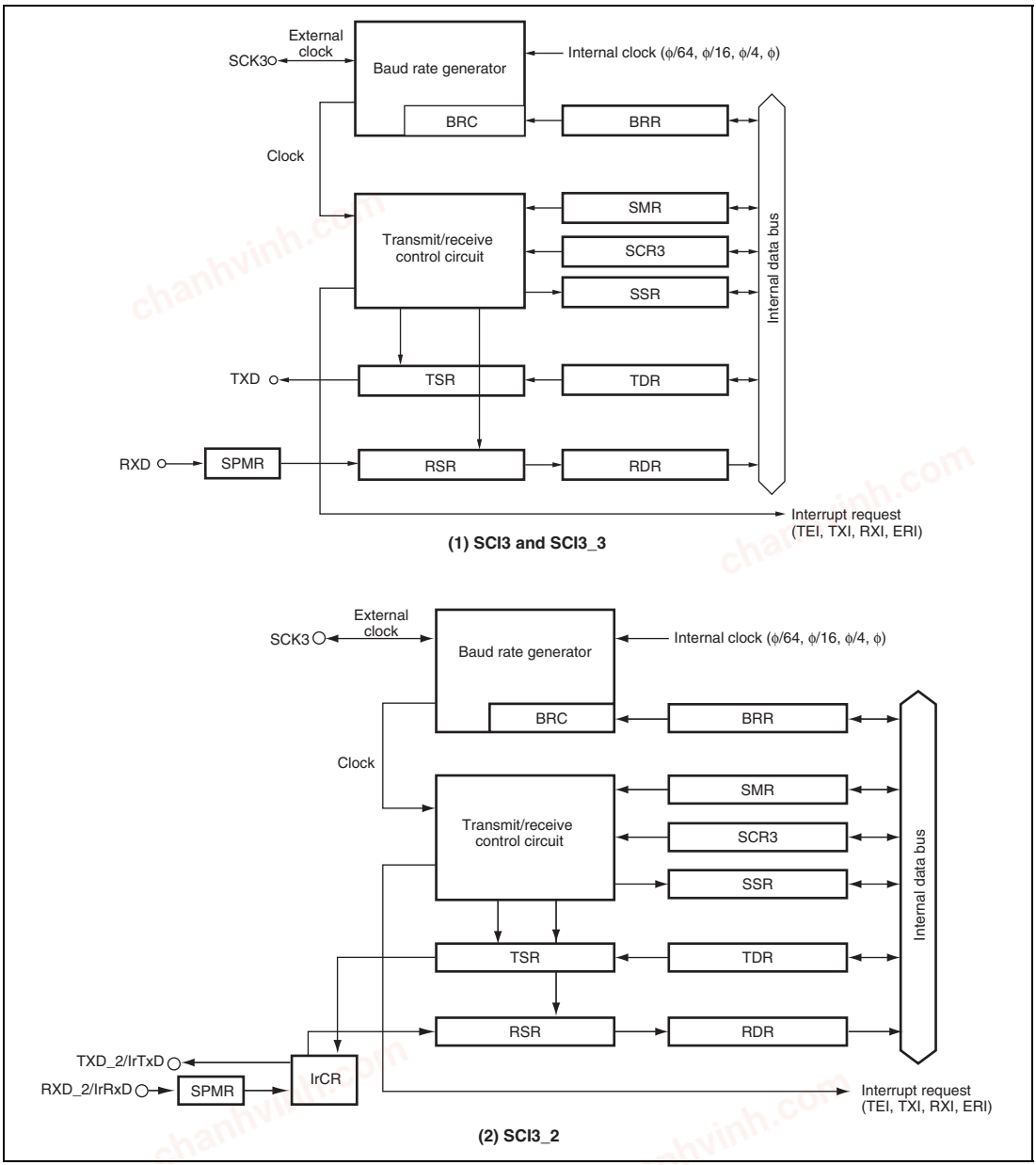



**Figure 20.1 Block Diagram of SCI3**

chanhvinh.com



Table 20.2 shows the SCI3 pin configuration.

**Table 20.2 Pin Configuration**

| Channel | Pin Name | I/O | Function |
|---|---|---|---|
| 1 | SCK3 | I/O | Clock input/output for channel 1 |
| | RXD | Input | Receive data input for channel 1 |
| | TXD | Output | Transmit data output for channel 1 |
| 2 | SCK3_2 | I/O | Clock input/output for channel 2 |
| | RXD_2/IrRxD | Input | Receive data input for channel 2/IrDA receive data input |
| | TXD_2/IrTxD | Output | Transmit data output for channel 2/IrDA transmit data output |
| 3 | SCK3_3 | I/O | Clock input/output for channel 3 |
| | RXD_3 | Input | Receive data input for channel 3 |
| | TXD_3 | Output | Transmit data output for channel 3 |

chanhvinh.com



## 20.2 Register Descriptions

The SCI3 has the following registers.

Channel 1
- Receive shift register (RSR)
- Receive data register (RDR)
- Transmit shift register (TSR)
- Transmit data register (TDR)
- Serial mode register (SMR)
- Serial control register (SCR3)
- Serial status register (SSR)
- Bit rate register (BRR)
- Sampling mode register (SPMR)

Channel 2
- Receive shift register_2 (RSR_2)
- Receive data register_2 (RDR_2)
- Transmit shift register_2 (TSR_2)
- Transmit data register_2 (TDR_2)
- Serial mode register_2 (SMR_2)
- Serial control register_2 (SCR3_2)
- Serial status register_2 (SSR_2)
- Bit rate register_2 (BRR_2)
- Sampling mode register_2 (SPMR_2)
- IrDA control register_2 (IrCR_2)

Channel 3
- Receive shift register_3 (RSR_3)
- Receive data register_3 (RDR_3)
- Transmit shift register_3 (TSR_3)
- Transmit data register_3 (TDR_3)
- Serial mode register_3 (SMR_3)
- Serial control register_3 (SCR3_3)
- Serial status register_3 (SSR_3)
- Bit rate register_3 (BRR_3)
- Sampling mode register_3 (SPMR_3)

chanhvinh.com



### 20.2.1 Receive Shift Register (RSR)

Address: —

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | — | — | — | — | — | — | — | — |

RSR is a shift register that is used to receive serial data input from the RXD pin and convert it into parallel data. When one frame of data has been received, it is transferred to RDR automatically. RSR cannot be directly accessed by the CPU.

### 20.2.2 Receive Data Register (RDR)

Address: H'FF0555, H'FF055D, H'FF0565

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

RDR is an 8-bit register that stores received data. When the SCI3 has received one frame of data, it transfers the received data from RSR to RDR, where it is stored. After this, RSR is receive-enabled. As RSR and RDR function as a double buffer in this way, continuous receive operations are possible. After confirming that the RDRF bit in SSR is set to 1, read RDR only once. RDR cannot be written to by the CPU. RDR is initialized to H'00.

### 20.2.3 Transmit Shift Register (TSR)

Address: —

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | — | — | — | — | — | — | — | — |

TSR is a shift register that transmits serial data. To perform serial data transmission, the SCI3 first transfers transmit data from TDR to TSR automatically, then sends the data that starts from the LSB to the TXD pin. TSR cannot be directly accessed by the CPU.

chanhvinh.com



### 20.2.4 Transmit Data Register (TDR)

Address: H'FF0553, H'FF055B, H'FF0563

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

TDR is an 8-bit register that stores data for transmission. When the SCI3 detects that TSR is empty, it transfers the transmit data written in TDR to TSR and starts transmission. The double-buffered structure of TDR and TSR enables continuous transmission. If the next transmit data has already been written to TDR during transmission of one-frame data, the SCI3 transfers the written data to TSR to continue transmission. To achieve reliable serial transmission, write transmit data to TDR only once after confirming that the TDRE bit in SSR is set to 1. TDR is initialized to H'FF.

### 20.2.5 Serial Mode Register (SMR)

Address: H'FF0550, H'FF0558, H'FF0560

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | COM | CHR | PE | PM | STOP | MP | CKS[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | COM | Communication mode | 0: Asynchronous mode<br>1: Clocked synchronous mode | R/W |
| 6 | CHR | Character length | (Enabled only in asynchronous mode)<br>0: Selects 8 bits as the data length.<br>1: Selects 7 bits as the data length. | R/W |
| 5 | PE | Parity enable | (Enabled only in asynchronous mode)<br>0: Parity bit addition and parity check are disabled.<br>1: The parity bit is added in transmission and the parity bit is checked in reception. | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 4 | PM | Parity mode | (Enabled only when the PE bit is 1 in asynchronous mode)<br>0: Selects even parity.<br>1: Selects odd parity. | R/W |
| 3 | STOP | Stop bit length | (Enabled only in asynchronous mode)<br>0: 1 stop bit<br>1: 2 stop bits | R/W |
| 2 | MP | Multiprocessor mode | 0: The multiprocessor communication function is disabled.<br>1: The multiprocessor communication function is enabled*2 | R/W |
| 1, 0 | CKS[1:0] | Clock select 0 and 1 | 00: $\phi$ clock (n = 0)<br>01: $\phi$/4 clock (n = 1)<br>10: $\phi$/14 clock (n = 2)<br>11: $\phi$/64 clock (n = 3) | R/W |

Notes: 1. The SMR value is retained when (module) standby mode is entered.
2. In clocked synchronous mode, clear this bit to 0.

- STOP bit (stop bit length)

  Selects the stop bit length in transmission. For reception, only the first stop bit is checked, regardless of the value in the bit. If the second stop bit is 0, it is treated as the start bit of the next transmit character.

- MP bit (multiprocessor mode)

  When this bit is set to 1, the multiprocessor communication function is enabled. The PE bit and PM bit settings are invalid in multiprocessor mode.

- CKS[1:0] bits (clock select 1, 0)

  These bits select the clock source for the baud rate generator.

  For the relationship between the bit rate register setting and the baud rate, see section 20.2.8, Bit Rate Register (BRR). n is the decimal representation of the value of n in BRR (see section 20.2.8, Bit Rate Register (BRR)).

chanhvinh.com



### 20.2.6 Serial Control Register 3 (SCR3)

Address: H'FF0552, H'FF055A, H'FF0562

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TIE | RIE | TE | RE | MPIE | TEIE | CKE[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | TIE | Transmit interrupt enable | 0: The TXI interrupt request is disabled.<br>1: The TXI interrupt request is enabled. | R/W |
| 6 | RIE | Receive interrupt enable | 0: RXI and ERI interrupt requests are disabled.<br>1: RXI and ERI interrupt requests are enabled. | R/W |
| 5 | TE | Transmit enable | 0: Transmission is disabled.<br>1: Transmission is enabled. | R/W |
| 4 | RE | Receive enable | 0: Reception is disabled.<br>1: Reception is enabled. | R/W |
| 3 | MPIE | Multiprocessor interrupt enable | (Enabled only when the MP bit in SMR is 1 in asynchronous mode)<br>When this bit is set to 1, receive data in which the multiprocessor bit is 0 is skipped, and setting of the RDRF, FER, and OER status flags in SSR is disabled. On receiving data in which the multiprocessor bit is 1, this bit is automatically cleared and normal reception is resumed. For details, see section 20.5, Multiprocessor Communication Function. | R/W |
| 2 | TEIE | Transmit end interrupt enable | 0: The TEI interrupt request is disabled.<br>1: The TEI interrupt request is enabled. | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 1, 0 | CKE[1:0] | Clock enable 0 and 1 | Selects the clock source.<br>Asynchronous mode:<br>00: On-chip baud rate generator<br>01: On-chip baud rate generator<br>Outputs a clock of the same frequency as the bit rate from the SCK3 pin.<br>10: External clock<br>A clock with a frequency 16 times the bit rate should be input from the SCK3 pin.<br>11:Reserved<br>Clocked synchronous mode:<br>00: On-chip clock (The SCK3 pin functions as clock output.)<br>01: Reserved<br>10: External clock (The SCK3 pin functions as clock input.)<br>11: Reserved | R/W |

Notes: 1. The TE and RE bits are reset and the other bits are retained when (module) standby mode is entered.
2. For details on interrupt requests, see section 20.8, Interrupt Requests.

chanhvinh.com



### 20.2.7 Serial Status Register (SSR)

Address: H'FF0554, H'FF055C, H'FF0564

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TDRE | RDRF | OER | FER | PER | TEND | MPBR | MPBT |
| Value after reset: | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | TDRE | Transmit data register empty flag | [Setting conditions]<br>• When the TE bit in SCR3 is 0<br>• When data is transferred from TDR to TSR<br>[Clearing conditions]<br>• When the CPU writes 0 after reading TDRE = 1.<br>• When the CPU writes transmit data to TDR.<br>• When the DTC transfers data to TDR with a TXI interrupt request and the DTC settings satisfy the flag clearing conditions. * | R/W |
| 6 | RDRF | Receive data register full flag | [Setting condition]<br>• When reception ends normally and receive data is transferred from RSR to RDR<br>[Clearing conditions]<br>• When the CPU writes 0 after reading RDRF = 1.<br>• When the CPU reads data from RDR.<br>• When the DTC transfers data from RDR with an RXI interrupt request and the DTC settings satisfy the flag clearing conditions. * | R/W |
| 5 | OER | Overrun error flag | [Setting condition]<br>• When an overrun error occurs in reception<br>[Clearing condition]<br>• When the CPU writes 0 after reading OER = 1. | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 4 | FER | Framing error flag | [Setting condition]<br>• When a framing error occurs in reception<br>[Clearing condition]<br>• When the CPU writes 0 after reading FER = 1. | R/W |
| 3 | PER | Parity error flag | [Setting condition]<br>• When a parity error is detected during reception<br>[Clearing condition]<br>• When the CPU writes 0 after reading PER = 1. | R/W |
| 2 | TEND | Transmit end flag | [Setting conditions]<br>• When the TE bit in SCR3 is 0<br>• When TDRE = 1 at transmission of the last bit of a transmit character<br>[Clearing conditions]<br>• When 0 is written to TDRE after reading TDRE = 1<br>• When the transmit data is written to TDR | R/W |
| 1 | MPBR | Multiprocessor bit receive | Stores the multiprocessor bit in the receive character data. When the RE bit in SCR3 is cleared to 0, its state is retained. | R/W |
| 0 | MPBT | Multiprocessor bit transfer | Specifies the multiprocessor bit value to be added to the transmit character data. | R/W |

Notes: * The DTC clears the peripheral module flags when all of the following three conditions are satisfied:
1. The DISEL bit is 0.
2. The value in the transfer counter (count register CRA in normal and repeat modes or count register CRB in block mode) is not 0.
3. A chain transfer is not used.

chanhvinh.com



### 20.2.8 Bit Rate Register (BRR)

Address: H'FF0551, H'FF0559, H'FF0561

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

BRR is an 8-bit register that adjusts the bit rate. The initial value of BRR is H'FF. Table 20.3 shows the relationship between the N setting in BRR and the n setting in bits CKS1 and CKS0 SMR in asynchronous mode. Table 20.4 shows the maximum bit rate for each frequency in asynchronous mode. The values shown in both tables 20.3 and 20.4 are values in active (high-speed) mode. Table 20.5 shows of the relationship between the N setting in BRR and the n setting in bits CKS1 and CKS0 of SMR in clocked synchronous mode. The values shown in table 20.5 are values in active (high-speed) mode. The N setting in BRR and error for other operating frequencies and bit rates can be obtained by the following formulas:

Note: The BRR value is retained in (module) standby mode.

**[Asynchronous Mode]**

$$N = \frac{\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

$$\text{Error (\%)} = \left\{ \frac{\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

**[Clocked Synchronous Mode]**

$$N = \frac{\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

[Legend]
B: Bit rate (bit/s)
N: BRR setting for baud rate generator ($0 \le N \le 255$)
$\phi$: Operating frequency (MHz)
n: CSK1 and CSK0 settings in SMR ($0 \le n \le 3$)

chanhvinh.com



**Table 20.3 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode)**

| | Operating Frequency ϕ (MHz) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 4 | | | 4.9152 | | | 5 | | | 6 | | |
| Bit Rate (bit/s) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) |
| 110 | 2 | 70 | 0.03 | 2 | 86 | 0.31 | 2 | 88 | -0.25 | 2 | 106 | -0.44 |
| 150 | 1 | 207 | 0.16 | 1 | 255 | 0.00 | 2 | 64 | 0.16 | 2 | 77 | 0.16 |
| 300 | 1 | 103 | 0.16 | 1 | 127 | 0.00 | 1 | 129 | 0.16 | 1 | 155 | 0.16 |
| 600 | 0 | 207 | 0.16 | 0 | 255 | 0.00 | 1 | 64 | 0.16 | 1 | 77 | 0.16 |
| 1200 | 0 | 103 | 0.16 | 0 | 127 | 0.00 | 0 | 129 | 0.16 | 0 | 155 | 0.16 |
| 2400 | 0 | 51 | 0.16 | 0 | 63 | 0.00 | 0 | 64 | 0.16 | 0 | 77 | 0.16 |
| 4800 | 0 | 25 | 0.16 | 0 | 31 | 0.00 | 0 | 32 | -1.36 | 0 | 38 | 0.16 |
| 9600 | 0 | 12 | 0.16 | 0 | 15 | 0.00 | 0 | 15 | 1.73 | 0 | 19 | -2.34 |
| 19200 | 0 | 6 | -6.99 | 0 | 7 | 0.00 | 0 | 7 | 1.73 | 0 | 9 | -2.34 |
| 31250 | 0 | 3 | 0.00 | 0 | 4 | -1.70 | 0 | 4 | 0.00 | 0 | 5 | 0.00 |
| 38400 | 0 | 2 | 8.51 | 0 | 3 | 0.00 | 0 | 3 | 1.73 | 0 | 4 | -2.34 |

| | Operating Frequency ϕ (MHz) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 6.144 | | | 7.3728 | | | 8 | | | 9.8304 | | |
| Bit Rate (bit/s) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) |
| 110 | 2 | 108 | 0.08 | 2 | 130 | -0.07 | 2 | 141 | 0.03 | 2 | 174 | -0.26 |
| 150 | 2 | 79 | 0.00 | 2 | 95 | 0.00 | 2 | 103 | 0.16 | 2 | 127 | 0.00 |
| 300 | 1 | 159 | 0.00 | 1 | 191 | 0.00 | 1 | 207 | 0.16 | 1 | 255 | 0.00 |
| 600 | 1 | 79 | 0.00 | 1 | 95 | 0.00 | 1 | 103 | 0.16 | 1 | 127 | 0.00 |
| 1200 | 0 | 159 | 0.00 | 0 | 191 | 0.00 | 0 | 207 | 0.16 | 0 | 255 | 0.00 |
| 2400 | 0 | 79 | 0.00 | 0 | 95 | 0.00 | 0 | 103 | 0.16 | 0 | 127 | 0.00 |
| 4800 | 0 | 39 | 0.00 | 0 | 47 | 0.00 | 0 | 51 | 0.16 | 0 | 63 | 0.00 |
| 9600 | 0 | 19 | 0.00 | 0 | 23 | 0.00 | 0 | 25 | 0.16 | 0 | 31 | 0.00 |
| 19200 | 0 | 9 | 0.00 | 0 | 11 | 0.00 | 0 | 12 | 0.16 | 0 | 15 | 0.00 |
| 31250 | 0 | 5 | 2.40 | 0 | 6 | 5.33 | 0 | 7 | 0.00 | 0 | 9 | -1.70 |
| 38400 | 0 | 4 | 0.00 | 0 | 5 | 0.00 | 0 | 6 | -6.99 | 0 | 7 | 0.00 |

chanhvinh.com



| | Operating Frequency ϕ (MHz) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 10 | | | 12 | | | 12.888 | | | 14 | | |
| Bit Rate (bit/s) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) |
| 110 | 2 | 177 | -0.25 | 2 | 212 | 0.03 | 2 | 217 | 0.08 | 2 | 248 | -0.17 |
| 150 | 2 | 129 | 0.16 | 2 | 155 | 0.16 | 2 | 159 | 0.00 | 2 | 181 | 0.16 |
| 300 | 2 | 64 | 0.16 | 2 | 77 | 0.16 | 2 | 79 | 0.00 | 2 | 90 | 0.16 |
| 600 | 1 | 129 | 0.16 | 1 | 155 | 0.16 | 1 | 159 | 0.00 | 1 | 181 | 0.16 |
| 1200 | 1 | 64 | 0.16 | 1 | 77 | 0.16 | 1 | 79 | 0.00 | 1 | 90 | 0.16 |
| 2400 | 0 | 129 | 0.16 | 0 | 155 | 0.16 | 0 | 159 | 0.00 | 0 | 181 | 0.16 |
| 4800 | 0 | 64 | 0.16 | 0 | 77 | 0.16 | 0 | 79 | 0.00 | 0 | 90 | 0.16 |
| 9600 | 0 | 32 | -1.36 | 0 | 38 | 0.16 | 0 | 39 | 0.00 | 0 | 45 | -0.93 |
| 19200 | 0 | 15 | 1.73 | 0 | 19 | -2.34 | 0 | 19 | 0.00 | 0 | 22 | -0.93 |
| 31250 | 0 | 9 | 0.00 | 0 | 11 | 0.00 | 0 | 11 | 2.40 | 0 | 13 | 0.00 |
| 38400 | 0 | 7 | 1.73 | 0 | 9 | -2.34 | 0 | 9 | 0.00 | — | — | — |

| | Operating Frequency ϕ (MHz) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 14.7456 | | | 16 | | | 18 | | | 20 | | |
| Bit Rate (bit/s) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) |
| 110 | 3 | 64 | 0.70 | 3 | 70 | 0.03 | 3 | 79 | -0.12 | 3 | 88 | -0.25 |
| 150 | 2 | 191 | 0.00 | 2 | 207 | 0.16 | 2 | 233 | 0.16 | 3 | 64 | 0.16 |
| 300 | 2 | 95 | 0.00 | 2 | 103 | 0.16 | 2 | 114 | 0.16 | 2 | 129 | 0.16 |
| 600 | 1 | 191 | 0.00 | 1 | 207 | 0.16 | 1 | 233 | 0.16 | 2 | 64 | 0.16 |
| 1200 | 1 | 95 | 0.00 | 1 | 103 | 0.16 | 1 | 114 | 0.16 | 1 | 129 | 0.16 |
| 2400 | 0 | 191 | 0.00 | 0 | 207 | 0.16 | 0 | 233 | 0.16 | 1 | 64 | 0.16 |
| 4800 | 0 | 95 | 0.00 | 0 | 103 | 0.16 | 0 | 114 | 0.16 | 0 | 129 | 0.16 |
| 9600 | 0 | 47 | 0.00 | 0 | 51 | 0.16 | 0 | 58 | -0.96 | 0 | 64 | 0.16 |
| 19200 | 0 | 23 | 0.00 | 0 | 25 | 0.16 | 0 | 28 | 1.02 | 0 | 32 | -1.36 |
| 31250 | 0 | 14 | -1.70 | 0 | 15 | 0.00 | 0 | 17 | 0.00 | 0 | 19 | 0.00 |
| 38400 | 0 | 11 | 0.00 | 0 | 12 | 0.16 | 0 | 14 | -2.34 | 0 | 15 | 1.73 |

[Legend]

—: A setting is available but error occurs.

chanhvinh.com



**Table 20.4 Maximum Bit Rate for Each Frequency (Asynchronous Mode)**

| ϕ (MHz) | Maximum Bit Rate (bit/s) | n | N | ϕ (MHz) | Maximum Bit Rate (bit/s) | n | N |
|---|---|---|---|---|---|---|---|
| 4 | 125000 | 0 | 0 | 12 | 375000 | 0 | 0 |
| 4.9152 | 153600 | 0 | 0 | 12.288 | 384000 | 0 | 0 |
| 5 | 156250 | 0 | 0 | 14 | 437500 | 0 | 0 |
| 6 | 187500 | 0 | 0 | 14.7456 | 460800 | 0 | 0 |
| 6.144 | 192000 | 0 | 0 | 16 | 500000 | 0 | 0 |
| 7.3728 | 230400 | 0 | 0 | 17.2032 | 537600 | 0 | 0 |
| 8 | 250000 | 0 | 0 | 18 | 562500 | 0 | 0 |
| 9.8304 | 307200 | 0 | 0 | 20 | 625000 | 0 | 0 |
| 10 | 312500 | 0 | 0 | | | | |

chanhvinh.com



**Table 20.5 Examples of BRR Settings for Various Bit Rates (Clocked Synchronous Mode)**

| Bit Rate (bit/s) | Operating Frequency ϕ (MHz) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 4 | | 8 | | 10 | | 16 | | 18 | | 20 | |
| | n | N | n | N | n | N | n | N | n | N | n | N |
| 110 | | | | | | | | | | | | |
| 250 | 2 | 249 | 3 | 124 | — | — | 3 | 249 | | | | |
| 500 | 2 | 124 | 2 | 249 | — | — | 3 | 124 | — | — | — | — |
| 1k | 1 | 249 | 2 | 124 | — | — | 2 | 249 | — | — | — | — |
| 2.5k | 1 | 99 | 1 | 199 | 1 | 249 | 2 | 99 | — | — | 2 | 124 |
| 5k | 0 | 199 | 1 | 99 | 1 | 124 | 1 | 199 | 1 | 224 | 1 | 249 |
| 10k | 0 | 99 | 0 | 199 | 0 | 249 | 1 | 99 | — | — | 1 | 124 |
| 25k | 0 | 39 | 0 | 79 | 0 | 99 | 0 | 159 | 0 | 179 | 0 | 199 |
| 50k | 0 | 19 | 0 | 39 | 0 | 49 | 0 | 79 | 0 | 89 | 0 | 99 |
| 100k | 0 | 9 | 0 | 19 | 0 | 24 | 0 | 39 | 0 | 44 | 0 | 49 |
| 250k | 0 | 3 | 0 | 7 | 0 | 9 | 0 | 15 | 0 | 17 | 0 | 19 |
| 500k | 0 | 1 | 0 | 3 | 0 | 4 | 0 | 7 | 0 | 8 | 0 | 9 |
| 1M | 0 | 0* | 0 | 1 | — | — | 0 | 3 | 0 | 4 | 0 | 4 |
| 2M | | | 0 | 0* | — | — | 0 | 1 | — | — | — | — |
| 2.5M | | | | | 0 | 0* | — | — | — | — | 0 | 1 |
| 5M | | | | | | | | | | | 0 | 0* |

[Legend]

Blank: No setting is available.

—: A setting is available but error occurs.

*: Continuous transfer is not possible.

chanhvinh.com



### 20.2.9 Sampling Mode Register (SPMR)

Address: H'FF0556, H'FF055E, H'FF0566

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | NFEN | — | — |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 to 3 | — | Reserved | These bits are read as 1. The write value should be 1. | — |
| 2 | NFEN | Noise cancellation function select | 0: The noise cancellation function is invalid for the RXD pin input.<br>1: The noise cancellation function is valid for the RXD pin input (when the COM bit in SMR is 0). | R/W |
| 1, 0 | — | Reserved | These bits are read as 1. The write value should be 1. | — |

Note: The SPMR value is retained in (module) standby mode.

- NFEN bit (noise cancellation function select)

  Performs noise cancellation for the RXD pin input when the COM bit in SMR is 0 and NFEN bit is 1.

chanhvinh.com



### 20.2.10 IrDA Control Register (IrCR)

Address: H'FF05DE

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | IrE | IrCK[2:0] | | | IrTXINV | IrRXINV | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | IrE | IrDA enable | 0: The TXD_2/IrTXD and RXD_2/IrRXD pins function as the TXD_2 and RXD_2 pins.<br>1: The TXD_2/IrTXD and RXD_2/IrRXD pins function as the IrTXD and IrRXD pins. | R/W |
| 6 to 4 | IrCK[2:0] | IrDA clock select 2 to 0 | 000: Bit rate × 3/16<br>001: φ/2<br>010: φ/4<br>011: φ/8<br>100: φ/16<br>101: φ/32<br>110: φ/64<br>111: φ/128 | R/W |
| 3 | IrTXINV | IrTX data polarity inversion | 0: Transmit data is output from IrTXD as is.<br>1: Transmit data is inverted to be output from IrTXD. | R/W |
| 2 | IrRXINV | IrRX data polarity inversion | 0: IrRXD input is used for receive data as is.<br>1: IrRXD input is inverted to be used for receive data. | R/W |
| 1, 0 | — | Reserved | These bits are read as 1. The write value should be 1. | — |

Note: The IrCR value is retained in (module) standby mode.

chanhvinh.com



- IrE bit (IrDA enable)
  Selects the SCI3_2 I/O pin function between the usual serial function and IrDA function.

- IrCK[2:0] bit (IrDA clock select 2 to 0)
  Sets the high pulse width for IrTXD output pulse encoding when the IrDA function is

- IrTXINV bit (IrTX data polarity inversion)
  Sets to invert the logic level of the IrTXD output. When inversion is specified, the high pulse width set with IrCR[2:0] is handled as low pulse width.

- IrRXINV bit (IrRX data polarity inversion)
  Sets to invert the logic level of the IrRXD input. When inversion is specified, the high pulse width set with IrCR[2:0] is handled as low pulse width.

chanhvinh.com



## 20.3 Operation in Asynchronous Mode

Figure 20.2 shows the general format for asynchronous communication. One character (or frame) consists of a start bit (low level), followed by data (in LSB-first order), a parity bit (high or low level), and finally stop bits (high level). Inside the SCI3, the transmitter and receiver are independent units, enabling full-duplex. Both the transmitter and the receiver also have a double-buffered structure, so data can be read or written during transmission or reception, enabling continuous data transfer.

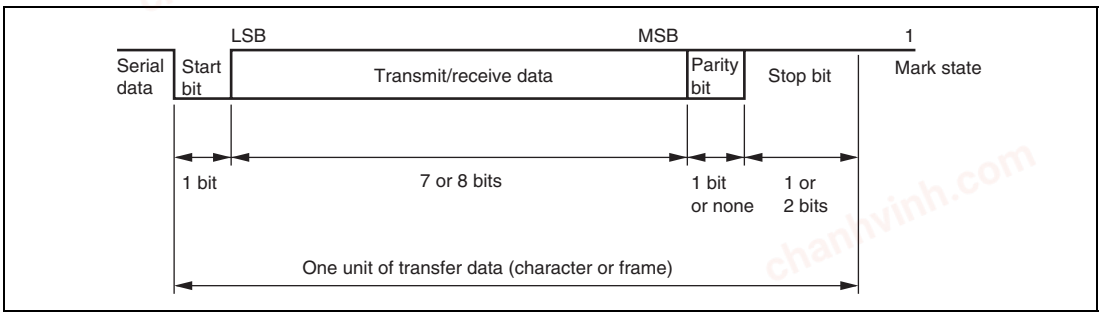



**Figure 20.2 Data Format in Asynchronous Communication**

### 20.3.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input at the SCK3 pin can be selected as the SCI3's transfer clock, according to the setting of the COM bit in SMR and the CKE0 and CKE1 bits in SCR3. When an external clock is input at the SCK3 pin, the clock frequency should be 16 times the bit rate used.

When the SCI3 is operated on an internal clock, the clock can be output from the SCK3 pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is in the middle of the transmit data, as shown in figure 20.3.

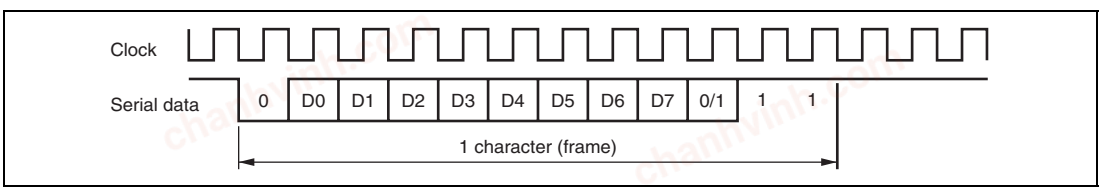


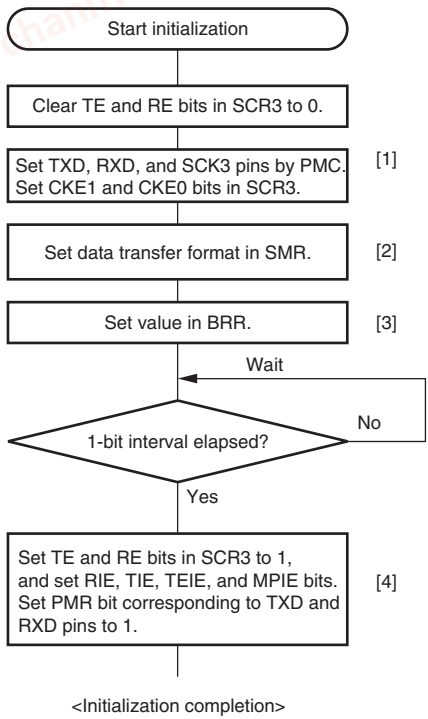

**Figure 20.3 Relationship between Output Clock and Transfer Data Phase (Asynchronous Mode) (Example with 8-Bit Data, Parity, Two Stop Bits)**

chanhvinh.com



### 20.3.2 SCI3 Initialization

Figure 20.4 shows a sample flowchart to initialize the SCI3. When the TE bit is cleared to 0, the TDRE flag is set to 1. Note that clearing the RE bit to 0 does not initialize the contents of the RDRF, PER, FER, and OER flags, or the contents of RDR. When the external clock is used in asynchronous mode, the clock must be supplied even during initialization.




[1] With the PMC, select which of the TXD, RXD, and SCK3 pins are to be used. Set the clock selection in SCR3. Be sure to clear the other bits in SCR3 to 0. When clock output is selected in asynchronous mode, after the CKE1 and CKE0 settings have been made, output of the clock signal begins immediately upon setting of the PMR bits that correspond to pins selected by SCK3. When clock output is selected with reception in clock-synchronous mode, and CKE1, CKE0, and RE are set to 1, output of the clock signal begins immediately upon setting of the PMR bits that correspond to pins selected by SCK3.

[2] Set the data transfer format in SMR.

[3] Write the value corresponding to the bit rate to BRR. Not necessary if an external clock is used.

[4] Wait at least one bit interval, then set the TE bit or RE bit in SCR3 to 1. For transmission, enable use of the TXD output pin by setting the PMR bit for the pin selected as TXD by the PMC to 1. For reception, enable use of the RXD input pin by setting the PMR bit for the pin selected as RXD by the PMC to 1.

Also set the RIE, TIE, TEIE, and MPIE bits, according to the required interrupts. In asynchronous mode, SCI3 is in the mark state (active) for transmission and in the space state (idle) while waiting for the start bit during reception. After the TE bit has been set to 1 in the case of transmission, transmission is enabled after the output of a frame with all bits 1.

**Figure 20.4 Sample Flowchart for Initializing SCI3**

chanhvinh.com



### 20.3.3 Data Transmission

Figure 20.5 shows an example of operation for transmission in asynchronous mode. In transmission, the SCI3 operates as described below.

1. The SCI3 monitors the TDRE flag in SSR. If the flag is cleared to 0, the SCI3 recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. The SCI3 sets the TDRE flag to 1 and starts transmission. If the TIE bit is set to 1 at this time, a TXI interrupt request is generated. Continuous transmission is possible because the TXI interrupt routine writes next transmit data to TDR before transmission of the current transmit data has been completed.
3. The SCI3 checks the TDRE flag at the timing for sending the stop bit.
4. If the TDRE flag is 0, the data is transferred from TDR to TSR, the stop bit is sent, and then transmission of the next frame is started.
5. If the TDRE flag is 1, the TEND flag in SSR is set to 1, the stop bit is sent, and then the "mark state" is entered, in which 1 is output. If the TEIE bit in SCR3 is set to 1 at this time, a TEI interrupt request is generated.
6. Figure 20.6 shows a sample flowchart for transmission in asynchronous mode.

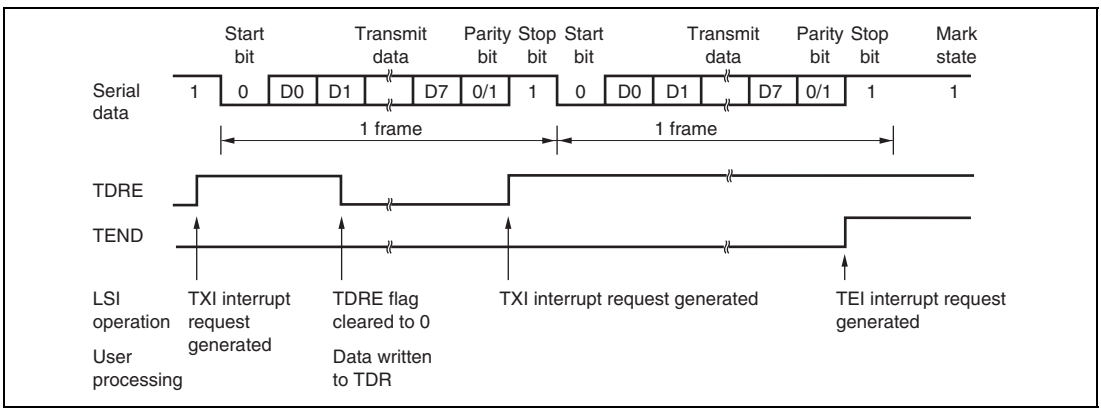
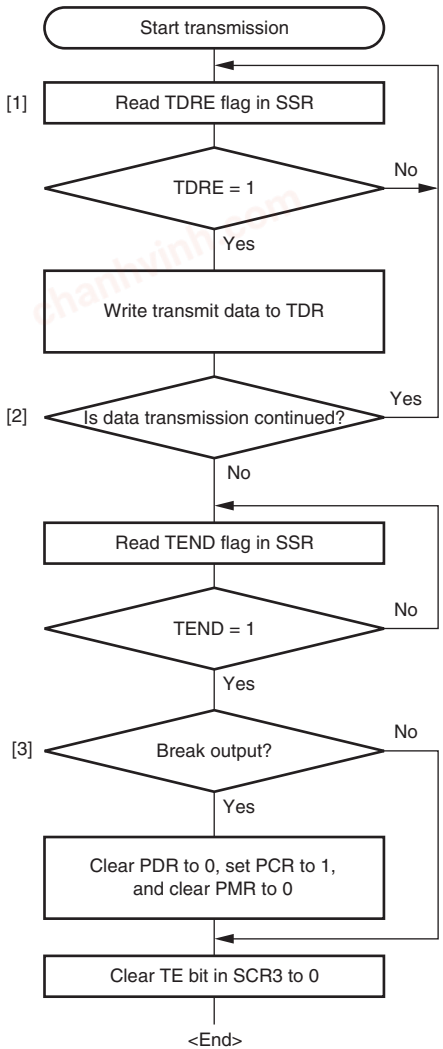



**Figure 20.5 Example of Transmission in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit)**

chanhvinh.com






[1] Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR. When data is written to TDR, the TDRE flag is automaticaly cleared to 0.

[2] To continue data transmission, read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR. When data is written to TDR, the TDRE flag is automaticaly cleared to 0.
If data is transferred to TDR by the DTC with a transmit data empty interrupt (TXI) request, the TDRE flag is automatically checked and cleared.

[3] To output a break at the end of data transmission, clear PMR corresponding to TxD to 0, after setting PCR to 1 and PDR to 0, then clear the TE bit in SCR3 to 0.

**Figure 20.6 Sample Flowchart for Transmitting Data (Asynchronous Mode)**

chanhvinh.com



### 20.3.4 Data Reception

Figure 20.7 shows an example of operation for reception in asynchronous mode. In reception, the SCI3 operates as described below.

1. The SCI3 monitors the communication line. If a start bit is detected, the SCI3 performs internal synchronization, receives receive data in RSR, and checks the parity bit and stop bit.
2. If an overrun error occurs (when reception of the next data is completed while the RDRF flag is still set to 1), the OER bit in SSR is set to 1. If the RIE bit in SCR3 is set to 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR.
3. If a parity error is detected, the PER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR3 is set to 1 at this time, an ERI interrupt request is generated.
4. If a framing error is detected (when the stop bit is 0), the FER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR3 is set to 1 at this time, an ERI interrupt request is generated.
5. If reception is completed successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR3 is set to 1 at this time, an RXI interrupt request is generated. Continuous reception is possible because the RXI interrupt routine reads the receive data transferred to RDR before reception of the next receive data has been completed.

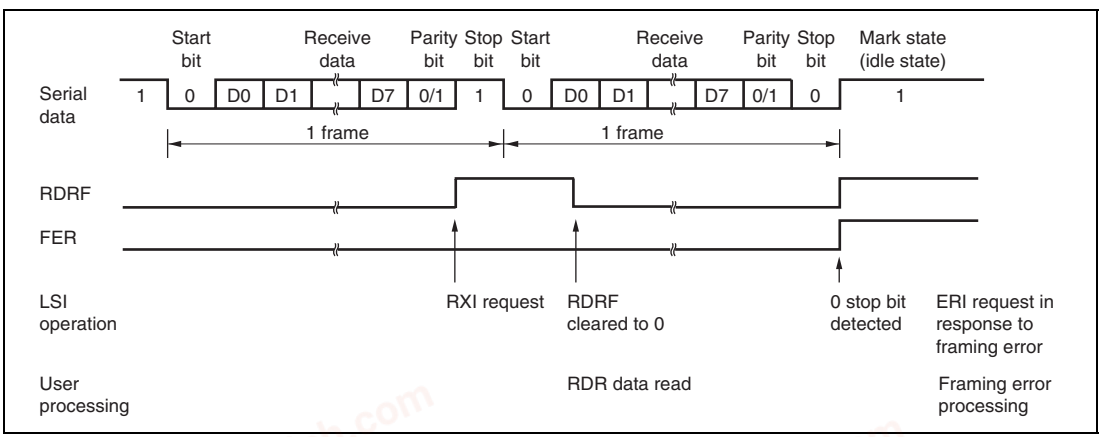



**Figure 20.7 Example of Reception in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit)**

chanhvinh.com



Table 20.6 shows the states of the SSR status flags and receive data handling when a receive error is detected. If a receive error is detected, the RDRF flag retains its state before receiving data. Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the OER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 20.8 shows a sample flowchart for data reception.

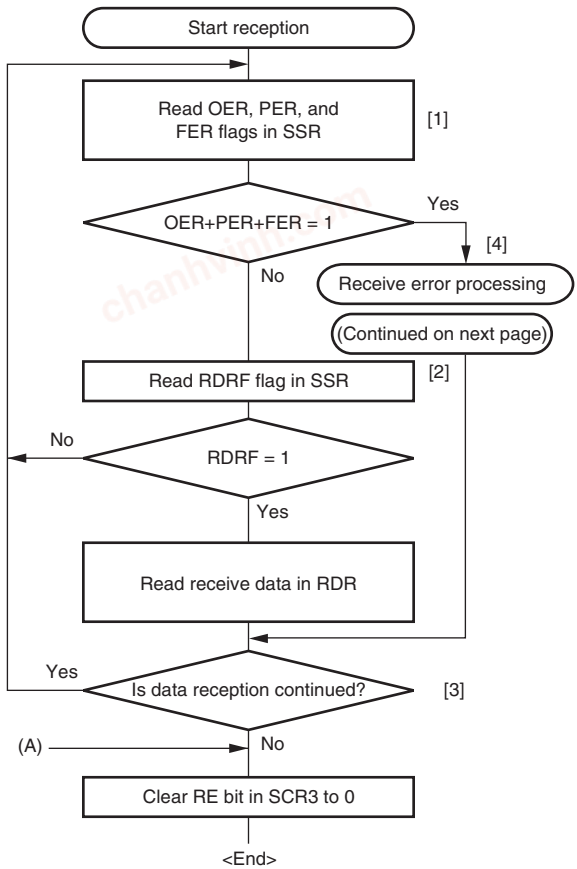
**Table 20.6 SSR Status Flags and Receive Data Handling**

| | SSR Status Flag | | | | |
|---|---|---|---|---|---|
| **RDRF*** | **OER** | **FER** | **PER** | **Receive Data** | **Receive Error Type** |
| 1 | 1 | 0 | 0 | Lost | Overrun error |
| 0 | 0 | 1 | 0 | Transferred to RDR | Framing error |
| 0 | 0 | 0 | 1 | Transferred to RDR | Parity error |
| 1 | 1 | 1 | 0 | Lost | Overrun error + framing error |
| 1 | 1 | 0 | 1 | Lost | Overrun error + parity error |
| 0 | 0 | 1 | 1 | Transferred to RDR | Framing error + parity error |
| 1 | 1 | 1 | 1 | Lost | Overrun error + framing error + parity error |

Note: * The RDRF flag retains the state it had before data reception.

chanhvinh.com






[1] Read the OER, PER, and FER flags in SSR to identify the error. If a receive error occurs, performs the appropriate error processing.

[2] Read SSR and check that RDRF = 1, then read the receive data in RDR. The RDRF flag is cleared automatically.

[3] To continue data reception, before the stop bit for the current frame is received, read the RDRF flag and read RDR. The RDRF flag is cleared automatically by te RDR read. If RDR data is transferred by the DTC which was activated by an RXI interrupt, the RDRF flag is cleared automatically.

[4] If a receive error occurs, read the OER, PER, and FER flags in SSR to identify the error. After performing the appropriate error processing, ensure that the OER, PER, and FER flags are all cleared to 0. Reception cannot be resumed if any of these flags are set to 1. In the case of a framing error, a break can be detected by reading the value of the RxD pin.

**Figure 20.8 Sample Flowchart for Data Reception (Asynchronous Mode) (1)**

chanhvinh.com



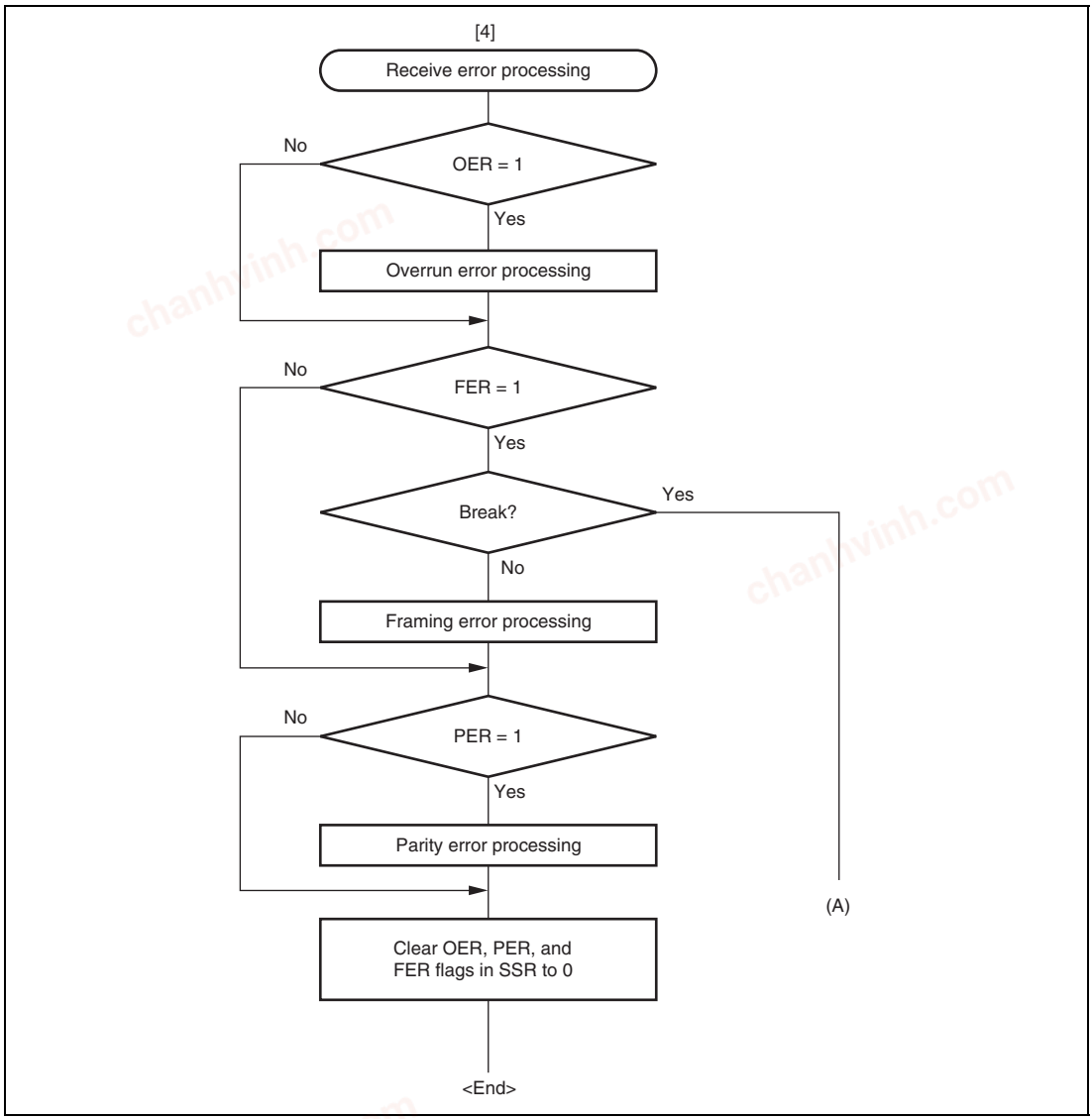



**Figure 20.8 Sample Flowchart for Data Reception (Asynchronous Mode) (2)**

chanhvinh.com



## 20.4 Operation in Clocked Synchronous Mode

Figure 20.9 shows the format for clocked synchronous communication. In clocked synchronous mode, data is transmitted or received synchronous with clock pulses. A single character in the transmit data consists of the 8-bit data starting from the LSB. In transmission, data is output from one falling edge of the synchronization clock to the next. In reception, data is received in synchronization with the rising edge of the synchronization clock. After 8-bit data is output, the transmission line holds the MSB state. In clocked synchronous mode, no parity or multiprocessor bit is added. Inside the SCI3, the transmitter and receiver are independent units, enabling full-duplex communication through the use of a common clock. Both the transmitter and the receiver also have a double-buffered structure, so data can be read or written during transmission or reception, enabling continuous data transfer.

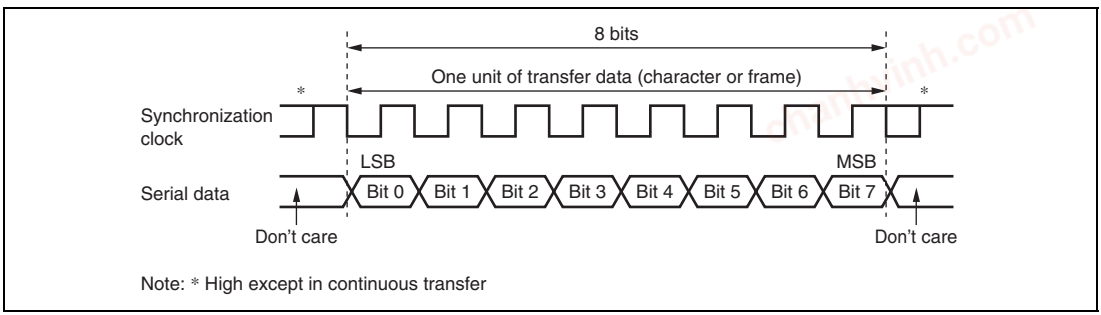



**Figure 20.9 Data Format in Clocked Synchronous Communication**

### 20.4.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCK3 pin can be selected, according to the setting of the COM bit in SMR and CKE0 and CKE1 bits in SCR3. When the SCI3 is operated on an internal clock, the synchronization clock is output from the SCK3 pin. Eight synchronization clock pulses are output in the transfer of one character, and when no transfer is performed the clock is fixed high.

### 20.4.2 SCI3 Initialization

Before transmitting and receiving data, the SCI3 should be initialized as described in a sample flowchart in figure 20.4.

chanhvinh.com



### 20.4.3 Data Transmission

Figure 20.10 shows an example of SCI3 operation for transmission in clocked synchronous mode. In transmission, the SCI3 operates as described below.

1. The SCI3 monitors the TDRE flag in SSR, and if the flag is 0, the SCI3 recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. The SCI3 sets the TDRE flag to 1 and starts transmission. If the TIE bit in SCR3 is set to 1 at this time, a transmit data empty interrupt (TXI) is generated.
3. The SCI3 outputs eight synchronization clock pulses when clock output mode has been specified. Data is output in synchronization with the input clock when use of an external clock has been specified. Serial data is transmitted sequentially from the LSB (bit 0), from the TXD pin.
4. The SCI3 checks the TDRE flag at the timing for sending the MSB (bit 7).
5. If the TDRE flag is cleared to 0, data is transferred from TDR to TSR, and transmission of the next frame is started.
6. If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, and the TDRE flag maintains the output state of the MSB. If the TEIE bit in SCR3 is set to 1 at this time, a TEI interrupt request is generated.
7. The SCK3 pin is fixed high at the end of transmission.

Figure 20.11 shows a sample flowchart for data transmission. Transmission will not start while a receive error flag (OER, FER, or PER) is set to 1. Make sure that the receive error flags are cleared to 0 before starting transmission.

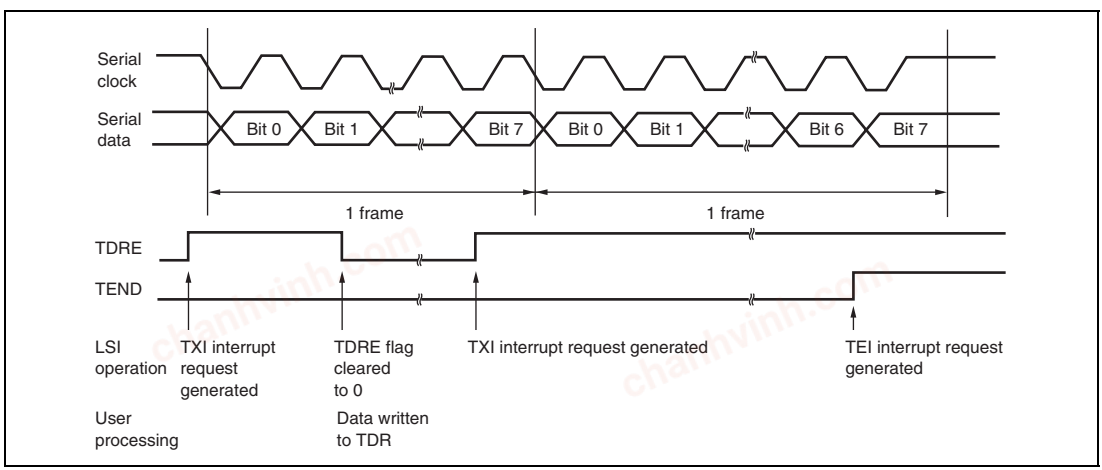
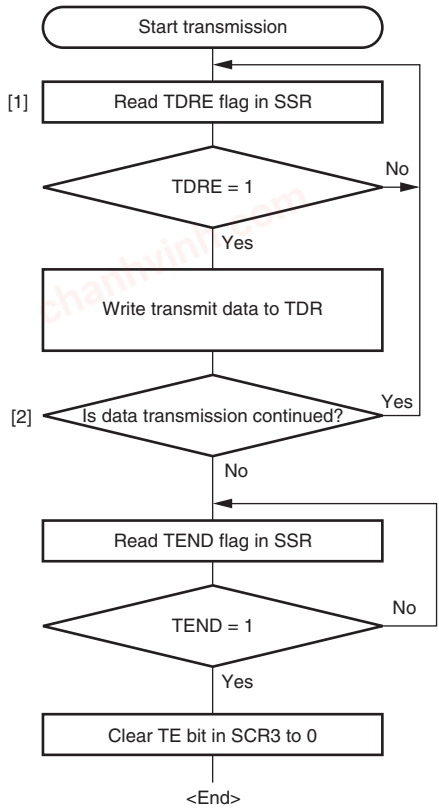



**Figure 20.10 Example of Transmission in Clocked Synchronous Mode**

chanhvinh.com






[1] Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR. When data is written to TDR, the TDRE flag is automatically cleared to 0 and clocks are output to start the data transmission. If data is transferred to TDR by the DTC with a transmit data empty interrupt (TXI) request, the TDRE flag is automatically checked and cleared.

[2] To continue data transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR. When data is written to TDR, the TDRE flag is automatically cleared to 0.

**Figure 20.11 Sample Flowchart for Data Transmission (Clocked Synchronous Mode)**

chanhvinh.com



### 20.4.4 Data Reception (Clocked Synchronous Mode)

Figure 20.12 shows an example of SCI3 operation for reception in clocked synchronous mode. In reception, the SCI3 operates as described below.

1. The SCI3 performs internal initialization synchronous with a synchronization clock input or output and starts receiving data.
2. The SCI3 stores the receive data in RSR.
3. If an overrun error occurs (when reception of the next data is completed while the RDRF flag in SSR is still set to 1), the OER bit in SSR is set to 1. If the RIE bit in SCR3 is set to 1 at this time, an ERI interrupt request is generated, receive data is not transferred to RDR, and the RDRF flag remains to be set to 1.
4. If reception is completed successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR3 is set to 1 at this time, an RXI interrupt request is generated.

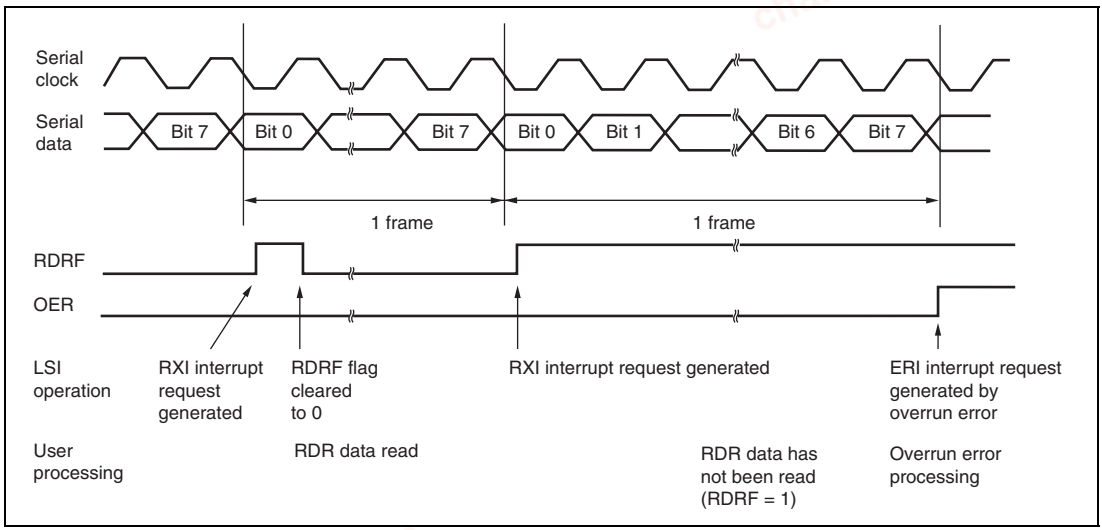
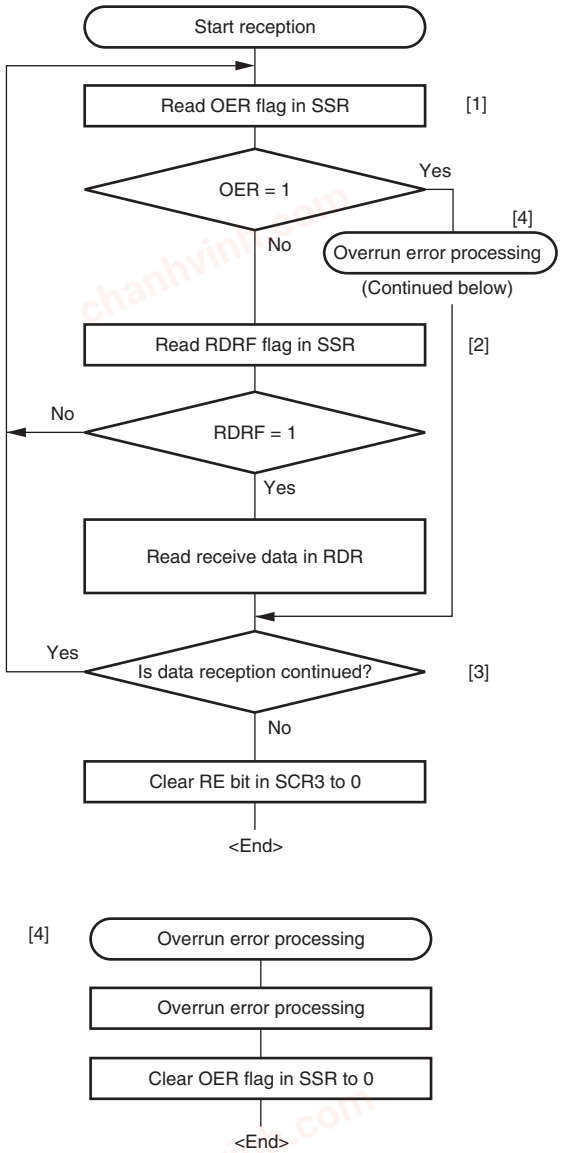



**Figure 20.12 Example of Reception in Clocked Synchronous Mode**

Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the OER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 20.13 shows a sample flowchart for data reception.

chanhvinh.com






[1] Read the OER flag in SSR to determine if there is an error. If an overrun error has occurred, execute overrun error processing.

[2] Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR. When data is read from RDR, the RDRF flag is automatically cleared to 0.
If RDR data is transferred by the DTC with a receive data full interrupt (RXI) request, the RDRF flag is cleared automatically.

[3] To continue data reception, before the MSB (bit 7) of the current frame is received, reading the RDRF flag and reading RDR should be finished. When data is read from RDR, the RDRF flag is automatically cleared to 0.

[4] If an overrun error occurs, read the OER flag in SSR, and after performing the appropriate error processing, clear the OER flag to 0. Reception cannot be resumed if the OER flag is set to 1.

**Figure 20.13 Sample Flowchart for Data Reception (Clocked Synchronous Mode)**

chanhvinh.com

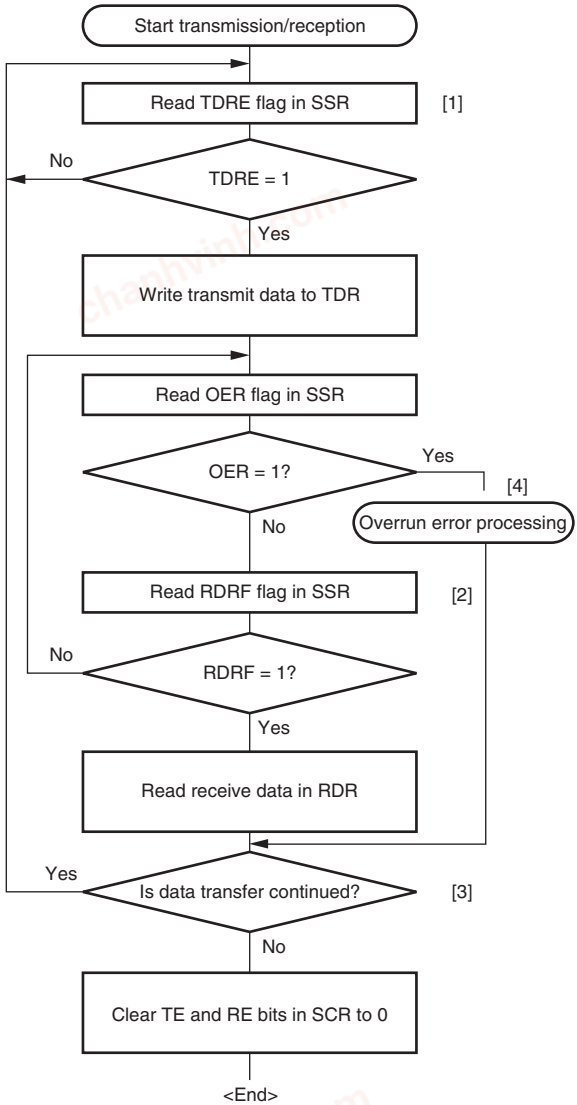


### 20.4.5 Simultaneous Data Transmission and Reception

Figure 20.14 shows a sample flowchart for simultaneous transmit and receive operations. The following procedure should be used for simultaneous data transmit and receive operations. To switch from transmit mode to simultaneous transmit and receive mode, after checking that the SCI3 has finished transmission and the TDRE and TEND flags are set to 1, clear TE to 0. Then simultaneously set TE and RE to 1 with a single instruction. To switch from receive mode to simultaneous transmit and receive mode, after checking that the SCI3 has finished reception, clear RE to 0. Then after checking that the RDRF and receive error flags (OER, FER, and PER) are cleared to 0, simultaneously set TE and RE to 1 with a single instruction.

chanhvinh.com






[1] Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR.
When data is written to TDR, the TDRE flag is automatically cleared to 0.

[2] Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR.
When data is read from RDR, the RDRF flag is automatically cleared to 0.

[3] To continue data transmission/ reception, before the MSB (bit 7) of the current frame is received, finish reading the RDRF flag, reading RDR. Also, before the MSB (bit 7) of the current frame is transmitted, read 1 from the TDRE flag to confirm that writing is possible. Then write data to TDR.
When data is written to TDR, the TDRE flag is automatically cleared to 0. When data is read from RDR, the RDRF flag is automatically cleared to 0.
If data is transferred to TDR by the DTC with a transmit data empty interrupt (TXI) request, the TDRE flag is automatically checked and cleared. If RDR data is transferred by the DTC with a receive data full iterrupt (RXI) request, the RDRF flag is automatically cleared.

[4] If an overrun error occurs, read the OER flag in SSR, and after performing the appropriate error processing, clear the OER flag to 0. Transmission/reception cannot be resumed if the OER flag is set to 1. For overrun error processing, see figure 20.13.

**Figure 20.14 Sample Flowchart of Simultaneous Transmit and Receive Operations (Clocked Synchronous Mode)**

chanhvinh.com



## 20.5 Multiprocessor Communication Function

Use of the multiprocessor communication function enables data transfer between a number of processors sharing communication lines by asynchronous serial communication using the multiprocessor format, in which a multiprocessor bit is added to the transfer data. When multiprocessor communication is performed, each receiving station is addressed by a unique ID code. The serial communication cycle consists of two component cycles; an ID transmission cycle that specifies the receiving station, and a data transmission cycle. The multiprocessor bit is used to differentiate between the ID transmission cycle and the data transmission cycle. If the multiprocessor bit is 1, the cycle is an ID transmission cycle; if the multiprocessor bit is 0, the cycle is a data transmission cycle. Figure 20.15 shows an example of inter-processor communication using the multiprocessor format. The transmitting station first sends the ID code of the receiving station with which it wants to perform serial communication as data with a 1 multiprocessor bit added. It then sends transmit data as data with a 0 multiprocessor bit added. When data with a 1 multiprocessor bit is received, the receiving station compares that data with its own ID. The station whose ID matches then receives the data sent next. Stations whose IDs do not match continue to skip data until data with a 1 multiprocessor bit is again received.

The SCI3 uses the MPIE bit in SCR3 to implement this function. When the MPIE bit is set to 1, transfer of receive data from RSR to RDR, error flag detection, and setting the SSR status flags, RDRF, FER, and OER, to 1, are inhibited until data with a 1 multiprocessor bit is received. On reception of a receive character with a 1 multiprocessor bit, the MPBR bit in SSR is set to 1 and the MPIE bit is automatically cleared, thus normal reception is resumed. If the RIE bit in SCR3 is set to 1 at this time, an RXI interrupt is generated.

When the multiprocessor format is selected, the parity bit setting is rendered invalid. All other bit settings are the same as those in normal asynchronous mode. The clock used for multiprocessor communication is the same as that in normal asynchronous mode.

chanhvinh.com



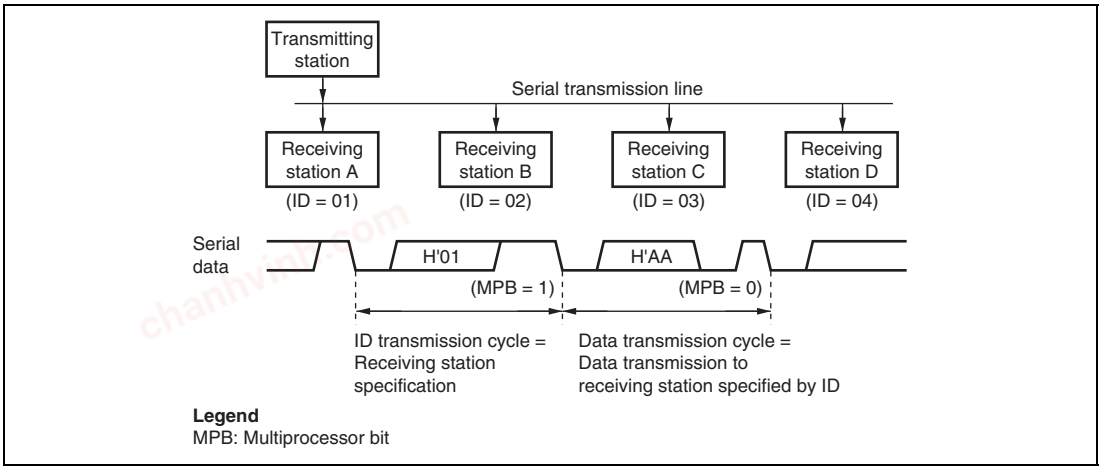


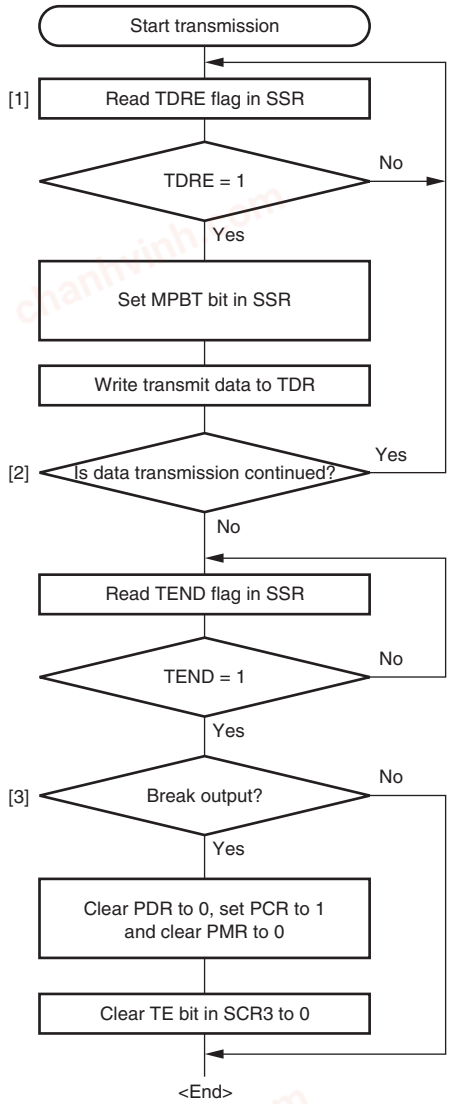

**Figure 20.15 Example of Inter-Processor Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)**

### 20.5.1 Multiprocessor Data Transmission

Figure 20.16 shows a sample flowchart for multiprocessor data transmission. For an ID transmission cycle, set the MPBT bit in SSR to 1 before transmission. For a data transmission cycle, clear the MPBT bit in SSR to 0 before transmission. All other SCI3 operations are the same as those in asynchronous mode.

chanhvinh.com






[1] Read SSR and check that the TDRE flag is set to 1, set the MPBT bit in SSR to 0 or 1, then write transmit data to TDR. When data is written to TDR, the TDRE flag is automatically cleared to 0.

[2] To continue data transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR. When data is written to TDR, the TDRE flag is automatically cleared to 0.
If data is transferred to TDR by the DTC with a transmit data empty interrupt (TXI) request, the TDRE flag is automatically checked and cleared.

[3] To output a break in serial transmission, set the port PCR to 1, clear PDR and PMR to 0, then clear the TE bit in SCR3 to 0.

**Figure 20.16 Sample Flowchart for Multiprocessor Data Transmission**

chanhvinh.com



### 20.5.2 Multiprocessor Data Reception

Figure 20.17 shows a sample flowchart for multiprocessor data reception. If the MPIE bit in SCR3 is set to 1, data is skipped until data with a 1 multiprocessor bit is sent. On receiving data with a 1 multiprocessor bit, the receive data is transferred to RDR. An RXI interrupt request is generated at this time. All other SCI3 operations are the same as those in asynchronous mode. Figure 20.18 shows an example of SCI3 operation for multiprocessor data reception.

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

chanhvinh.com



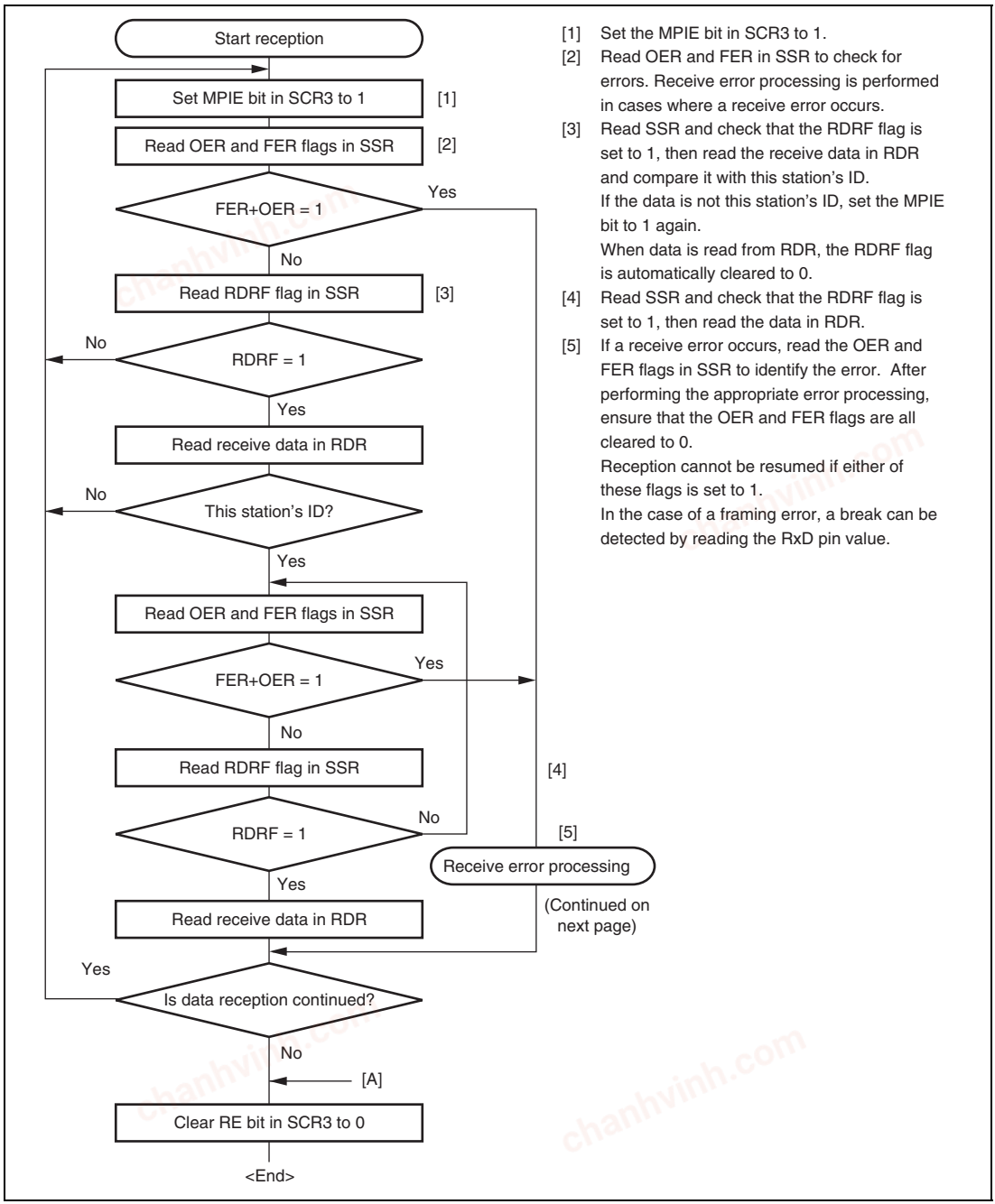



**Figure 20.17 Sample Flowchart for Multiprocessor Data Reception (1)**

chanhvinh.com



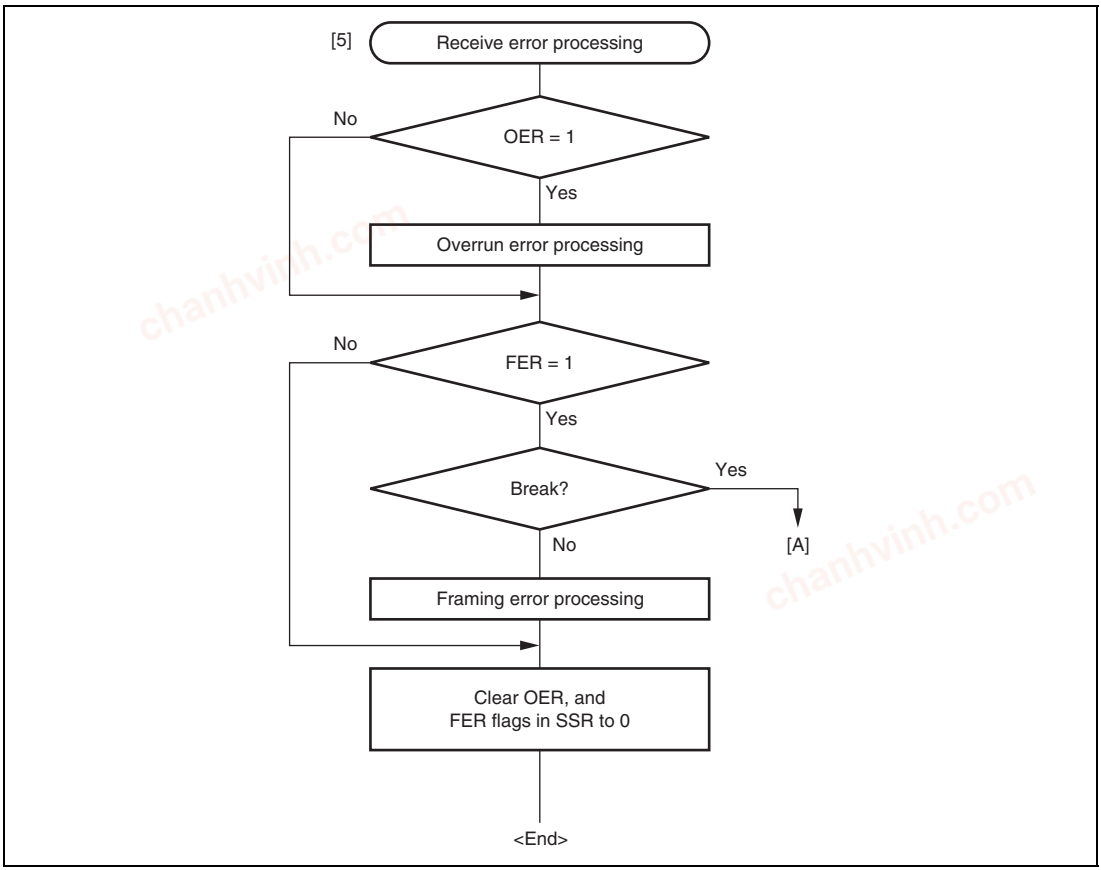



**Figure 20.17 Sample Flowchart for Multiprocessor Data Reception (2)**

chanhvinh.com



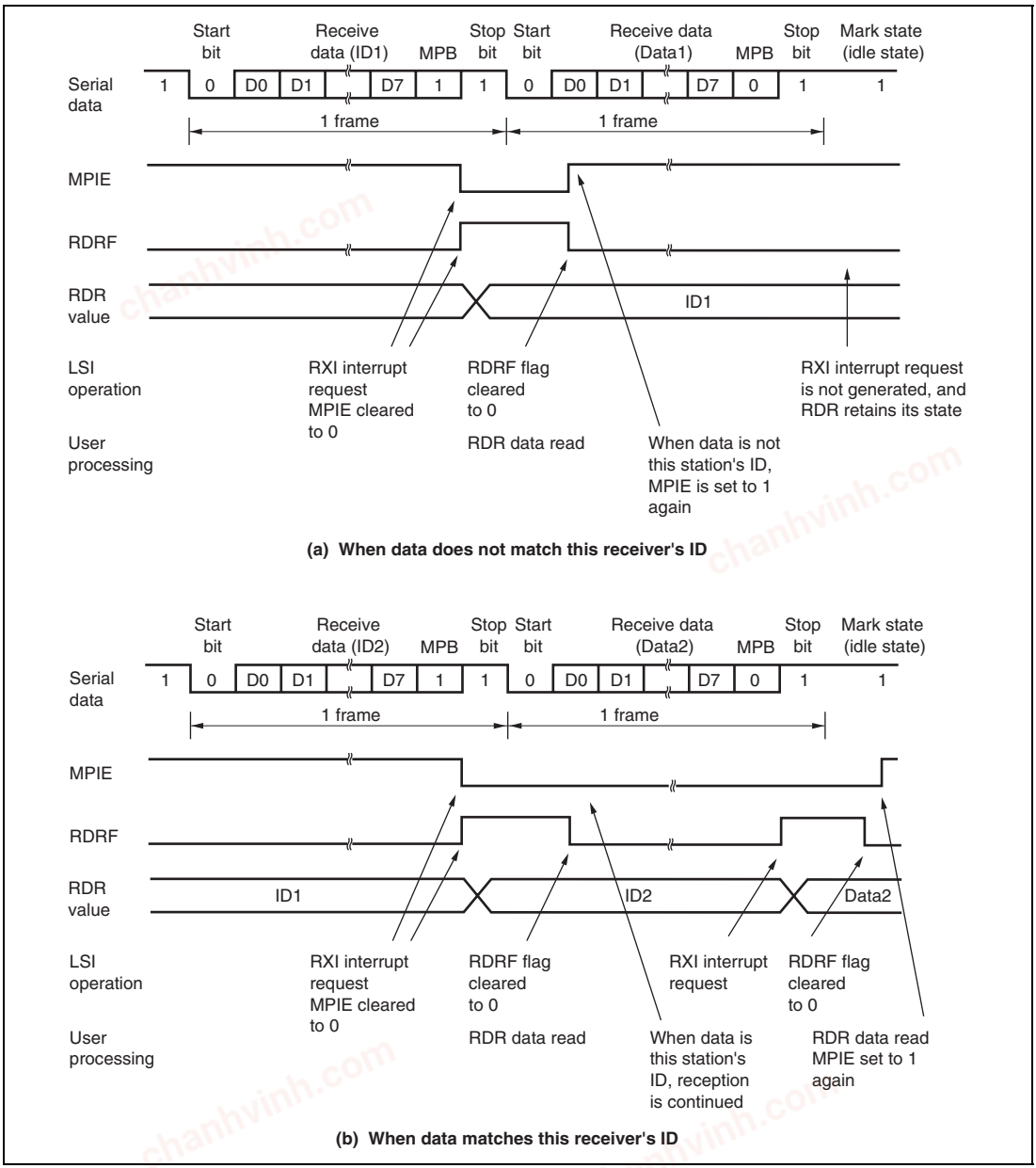



**Figure 20.18 Example of Reception Using Multiprocessor Format (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)**

chanhvinh.com



## 20.6 IrDA Operation

The SCI3_2 provides the IrDA function. If the IrDA function is enabled using the IrE bit in IrCR, the TXD_2 and RXD_2 pins in SCI3_2 are allowed to encode and decode the waveform based on the IrDA Specifications version 1.0 (function as the IrTxD and IrRxD pins)*. Connecting these pins to the infrared data transceiver achieves infrared data communications based on the system defined by the IrDA Specifications version 1.0.

In the system defined by the IrDA Specifications version 1.0, communication is started at a transfer rate of 9600 bps, which can be modified later as required. Since the IrDA interface provided by this LSI does not incorporate the capability of automatic modification of the transfer rate, the transfer rate must be modified through programming.

Figure 20.19 is the IrDA block diagram.

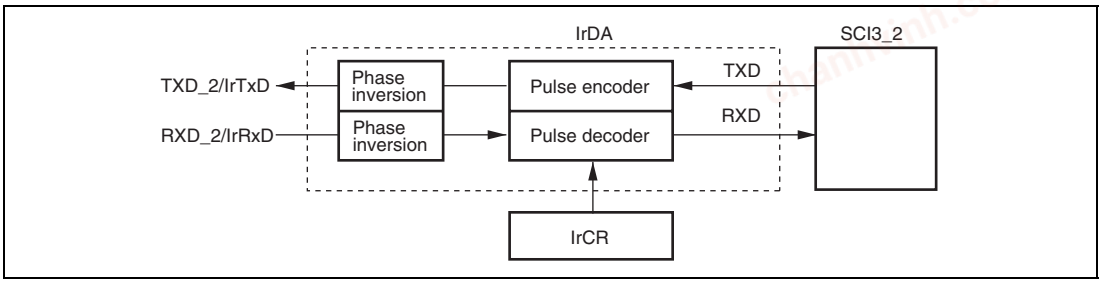



**Figure 20.19 IrDA Block Diagram**

IrDA operation should be set according to the following procedures.

(1) Set the corresponding pin in the PMCR register or PMR register.
(2) Set the IrCR register.
(3) Set the register related to SCI3_2.

chanhvinh.com



### 20.6.1 Transmission

During transmission, the output signals from the SCI3_2 (UART frames) are converted to IR frames using the IrDA interface (see figure 20.20). For serial data of level 0, a high-level pulse having a width of 3/16 of the bit rate (1-bit interval) is output (initial setting). The high-level pulse can be selected using the IrCK2 to IrCK0 bits in IrCR. The high-level pulse width is defined to be 1.41 µs at minimum and (3/16 + 2.5%) × bit rate or (3/16 × bit rate) +1.08 µs at maximum. For example, when the frequency of system clock ϕ is 20 MHz, a high-level pulse width of 1.6 µs can be specified because it is the smallest value in the range greater than 1.41 µs. For serial data of level 1, no pulses are output.

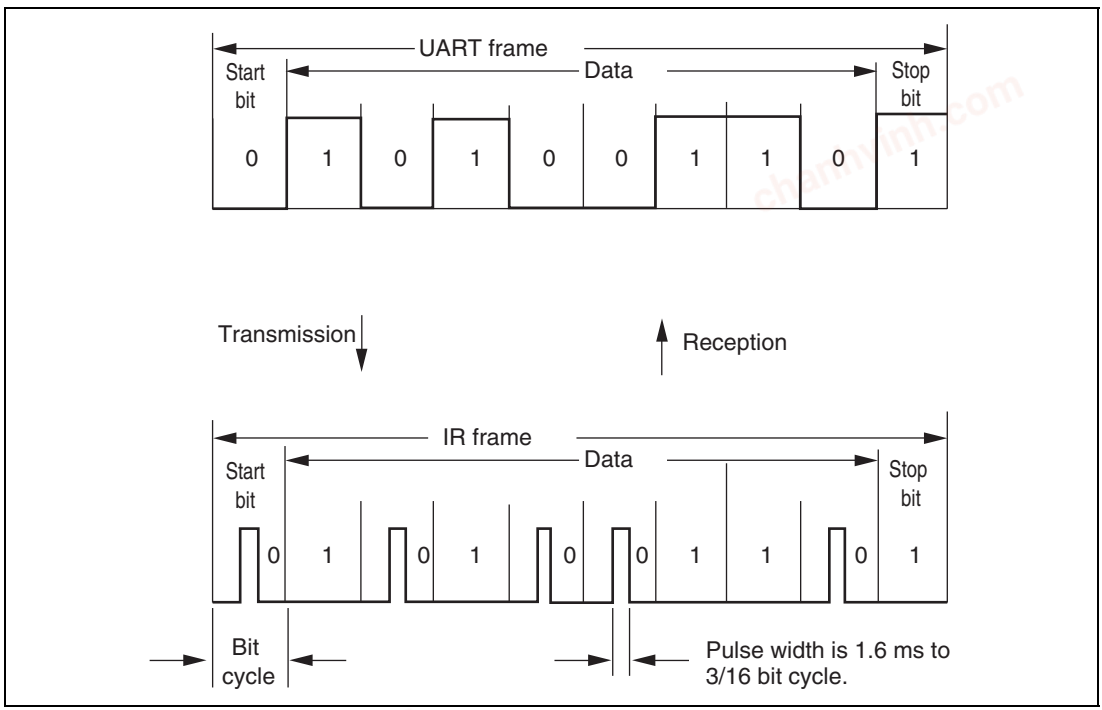



**Figure 20.20 IrDA Transmission and Reception**

### 20.6.2 Reception

During reception, IR frames are converted to UART frames using the IrDA interface before inputting to SCI3_2. 0 is output when the high level pulse is detected while 1 is output when no pulse is detected during one bit period. Note that a pulse shorter than the minimum pulse width of 1.41 µs is regarded as a 0 signal.

chanhvinh.com



### 20.6.3 High-Level Pulse Width Selection

Table 20.7 shows possible settings for bits IrCK2 to IrCK0 (minimum pulse width), and this LSI's operating frequencies and bit rates, for making the pulse width shorter than 3/16 times the bit rate in transmission.

**Table 20.7 Settings of Bits IrCK2 to IrCK0**

| Operating Frequency ϕ (MHz) | Bit Rate (bps) (Above)/Bit Period × 3/16 (Below) | | | | | |
|---|---|---|---|---|---|---|
| | 2400 | 9600 | 19200 | 38400 | 57600 | 115200 |
| | 78.13 | 19.53 | 9.77 | 4.88 | 3.26 | 1.63 |
| 4.9152 | 011 | 011 | 011 | 011 | 011 | 011 |
| 5 | 011 | 011 | 011 | 011 | 011 | 011 |
| 6 | 100 | 100 | 100 | 100 | 100 | 100 |
| 6.144 | 100 | 100 | 100 | 100 | 100 | 100 |
| 7.3728 | 100 | 100 | 100 | 100 | 100 | 100 |
| 8 | 100 | 100 | 100 | 100 | 100 | 100 |
| 9.3804 | 100 | 100 | 100 | 100 | 100 | 100 |
| 10 | 100 | 100 | 100 | 100 | 100 | 100 |
| 12 | 101 | 101 | 101 | 101 | 101 | 101 |
| 12.288 | 101 | 101 | 101 | 101 | 101 | 101 |
| 14 | 101 | 101 | 101 | 101 | 101 | 101 |
| 14.7456 | 101 | 101 | 101 | 101 | 101 | 101 |
| 16 | 101 | 101 | 101 | 101 | 101 | 101 |
| 16.9344 | 101 | 101 | 101 | 101 | 101 | 101 |
| 17.2032 | 101 | 101 | 101 | 101 | 101 | 101 |
| 18 | 101 | 101 | 101 | 101 | 101 | 101 |
| 19.6608 | 101 | 101 | 101 | 101 | 101 | 101 |
| 20 | 101 | 101 | 101 | 101 | 101 | 101 |

chanhvinh.com



## 20.7 Noise Canceler

Figure 20.21 shows a block diagram of the noise canceler circuit. When the noise canceler function is enabled, the RXD input signal is routed through the noise canceler before being provided internally. The noise canceler consists of three cascaded latches and a match detector. The RXD input signal is sampled at the basic clock frequency, 16 times the transfer rate, and when the outputs of three latches agree, the level is passed to the next circuit. If they do not agree, the previous value is held.

In other words, if the input level changes and the level remains the same for three or more clock cycles after the change, it is recognized as a signal. However, if the level remains the same for less than three clock cycles, it is recognized as a noise, not as a signal.

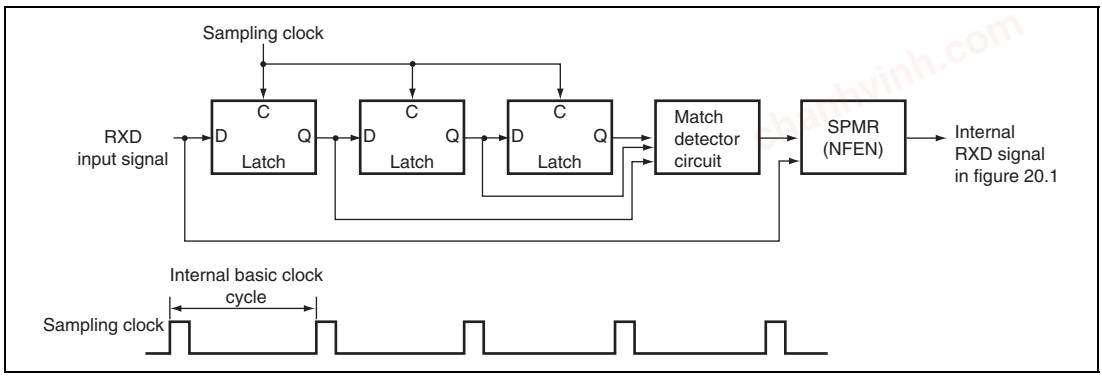



**Figure 20.21 Block Diagram of Noise Canceler**

chanhvinh.com



## 20.8 Interrupt Requests

The SCI3 creates the following six interrupt requests: transmission end, transmit data empty, receive data full, and receive errors (overrun error, framing error, and parity error). Table 20.8 shows the interrupt sources.

**Table 20.8 SCI3 Interrupt Requests**

| Interrupt Requests | Abbreviation | Interrupt Sources | DTC Activation |
|---|---|---|---|
| Receive Data Full | RXI | Setting RDRF in SSR | Possible |
| Transmit Data Empty | TXI | Setting TDRE in SSR | Possible |
| Transmission End | TEI | Setting TEND in SSR | Impossible |
| Receive Error | ERI | Setting OER, FER, and PER in SSR | Impossible |

When the TDRE flag in SSR is set to 1, a TXI interrupt request is generated. When the TEND flag in SSR is set to 1, a TEI interrupt request is generated. The DTC can be activated to perform data transfers with the TXI interrupt request. The TDRE flag is automatically cleared to 0 by the DTC data transfer.

When the RDRF flag in SSR is set to 1, a RXI interrupt request is generated. When any of the OER, PER and FER flags is set to 1, an ERI interrupt request is generated. The DTC can be activated to perform data transfers with the RXI interrupt request. The RDRF flag is automatically cleared to 0 by the DTC data transfer.

The TEI interrupt is generated if the TEND flag is set to 1 when the TEIE bit is 1. If the TEI and TXI interrupts are generated at the same time, the TXI interrupt is accepted first. Therefore, if the TDRE and TEND flags are to be simultaneously cleared in a TXI interrupt routine, branching to a TEI interrupt routine cannot be performed.

The initial value of the TDRE flag in SSR is 1. Thus, when the TIE bit in SCR3 is set to 1 before transferring the transmit data to TDR, a TXI interrupt request is generated even if the transmit data is not ready. The initial value of the TEND flag in SSR is 1. Thus, when the TEIE bit in SCR3 is set to 1 before transferring the transmit data to TDR, a TEI interrupt request is generated even if the transmit data has not been sent. It is possible to make use of the most of these interrupt requests efficiently by transferring the transmit data to TDR in the interrupt routine. To prevent the generation of these interrupt requests (TXI and TEI), set the enable bits (TIE and TEIE) that correspond to these interrupt requests to 1, after transferring the transmit data to TDR.

chanhvinh.com



## 20.9 Usage Notes

### 20.9.1 Break Detection and Processing

When framing error detection is performed, a break can be detected by reading the RXD pin value directly. In a break, the input from the RXD pin becomes all 0s, setting the FER flag, and possibly the PER flag. Note that as the SCI3 continues the receive operation after receiving a break, even if the FER flag is cleared to 0, it will be set to 1 again.

### 20.9.2 Mark State and Break Sending

When the PMR bit corresponding to the pin selected by the PMC is 0, the TXD pin is used as an I/O port whose direction (input or output) and level are determined by PCR and PDR. This can be used to set the TXD pin to mark state (high level) or send a break during data transmission. To maintain the communication line at mark state until the PMR bit is set to 1, set both PCR and PDR to 1. As the PMR bit is cleared to 0 at this point, the TXD pin becomes an I/O port, and 1 is output from the TXD pin. To send a break during transmission, first set PCR to 1 and clear PDR to 0, and then clear the PMR bit to 0. When the PMR bit is cleared to 0, the transmitter is initialized regardless of the current transmission state, the TXD pin becomes an I/O port, and 0 is output from the TXD pin.

### 20.9.3 Receive Error Flags and Transmit Operations (Clocked Synchronous Mode Only)

Transmission cannot be started when a receive error flag (OER, PER, or FER) is set to 1, even if the TDRE flag is cleared to 0. Be sure to clear the receive error flags to 0 before starting transmission. Note also that receive error flags cannot be cleared to 0 even if the RE bit is cleared to 0.

chanhvinh.com



### 20.9.4 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode

In asynchronous mode, the SCI3 operates on a basic clock with a frequency of 16 times the transfer rate. In reception, the SCI3 samples the falling edge of the start bit using the basic clock, and performs internal synchronization. Receive data is latched internally at the rising edge of the 8th pulse of the basic clock as shown in figure 20.22. Thus, the reception margin in asynchronous mode is given by formula (1) below.

$$M = \{( 0.5 - \frac{1}{2N} ) - \frac{D - 0.5}{N} - (L - 0.5) F \} \times 100\ (\%) \quad \text{... Formula 1}$$

... Formula (1)

[Legend]
N: Ratio of bit rate to clock (N = 16)
D: Clock duty (D = 0.5 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute value of clock rate deviation

Assuming values of F (absolute value of clock rate deviation) = 0 and D (clock duty) = 0.5 in formula (1), the reception margin can be given by the formula.

$$M = \{0.5 - 1/(2 \times 16)\} \times 100\ [\%] = 46.875\%$$

However, this is only the computed value, and a margin of 20% to 30% should be allowed for in system design.

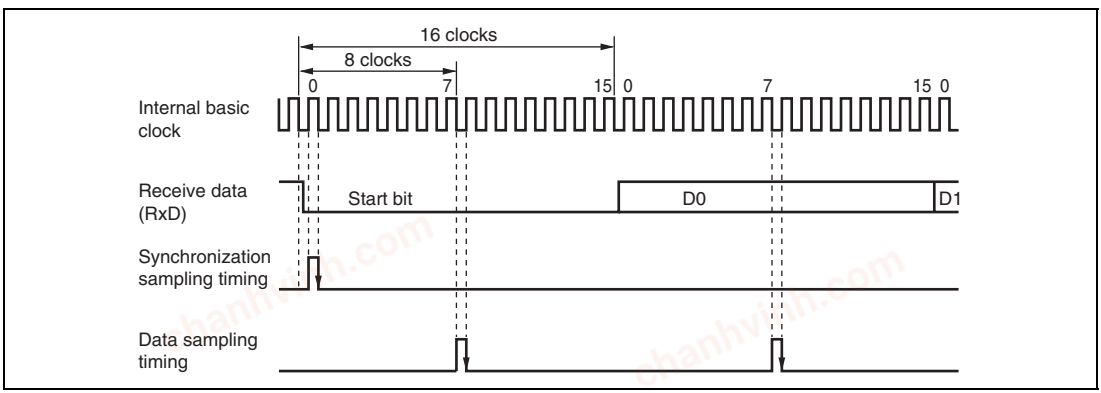



**Figure 20.22 Receive Data Sampling Timing in Asynchronous Mode**

chanhvinh.com



### 20.9.5 Relation between Writes to TDR and TDRE Flag

Data can be written to TDR regardless of the state of the TDRE flag. However, if new data is written to TDR when the TDRE flag is cleared to 0, the data stored in TDR will be lost since it has not yet been transferred to TSR. It is therefore essential to check that the TDRE flag is set to 1 before writing transmit data TDR.

### 20.9.6 Restrictions on Using DTC

When the external clock source is used as a synchronization clock, update TDR by the DTC or CPU and wait for at least five ϕ clock cycles before allowing the transmit clock to be input. If the transmit clock is input within four clock cycles after TDR modification, the SCI3 may malfunction (see figure 20.23).

When using the DTC to read RDR, be sure to set the receive end interrupt (RXI) for the relevant SCI3 as the DTC activation source.

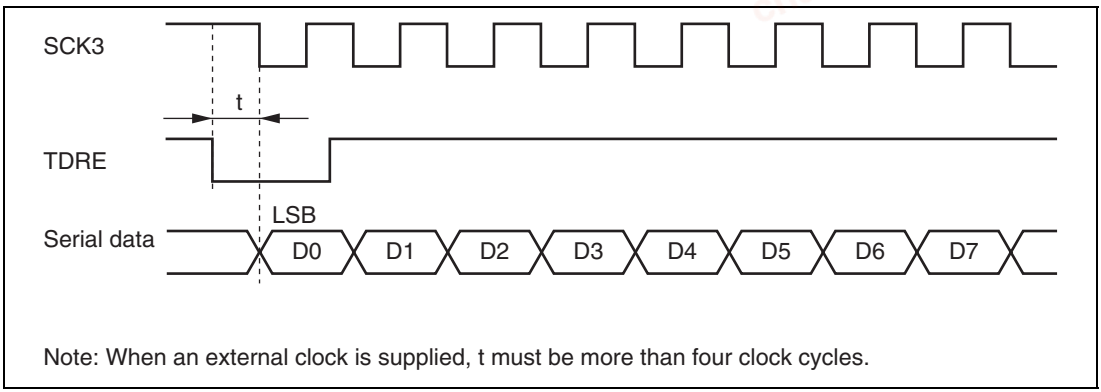



**Figure 20.23 Example of DTC Transmission in Clock Synchronous Mode**

chanhvinh.com



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

chanhvinh.com



# Section 21 I²C Bus Interface 2 (IIC2)

The I²C bus interface 2 conforms to and provides a subset of the Philips I²C bus (inter-IC bus) interface functions. The register configuration that controls the I²C bus differs partly from the Philips configuration, however. Figure 21.1 shows a block diagram of the I²C bus interface 2. Figure 21.2 shows an example of I/O pin connections to external circuits.

Either the IIC2 or SSU incorporated in this LSI can be used at a time. Accordingly, when the IIC2 function is used, the SSU function is not available.

## 21.1 Features

- Selectable for I²C bus format or clock 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
- Selectable for acknowledge output levels when receiving
- Automatic loading of acknowledge bit when transmitting
- Bit synchronization/wait function stored

  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 DTC can be activated by the transmit-data-empty and receive-data-full interrupts.
- Direct bus drive possible

  Two pins, SCL and SDA pins, function as NMOS open-drain outputs when the bus drive function is selected.

**Clock Synchronous Format:**

- Four interrupt sources

  Transmit-data-empty, transmit-end, receive-data-full, and overrun error. The DTC can be activated by the transmit-data-empty and receive-data-full interrupt sources.

chanhvinh.com



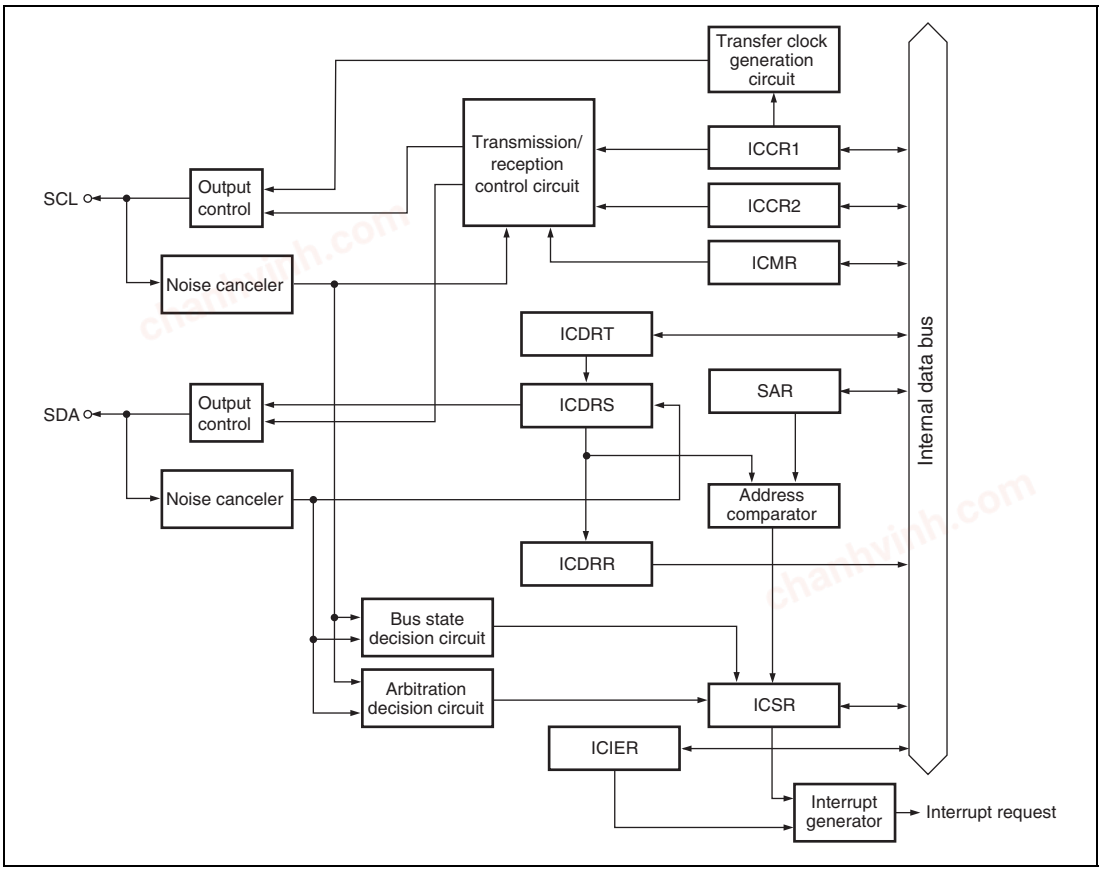



**Figure 21.1 Block Diagram of I²C Bus Interface 2**

chanhvinh.com



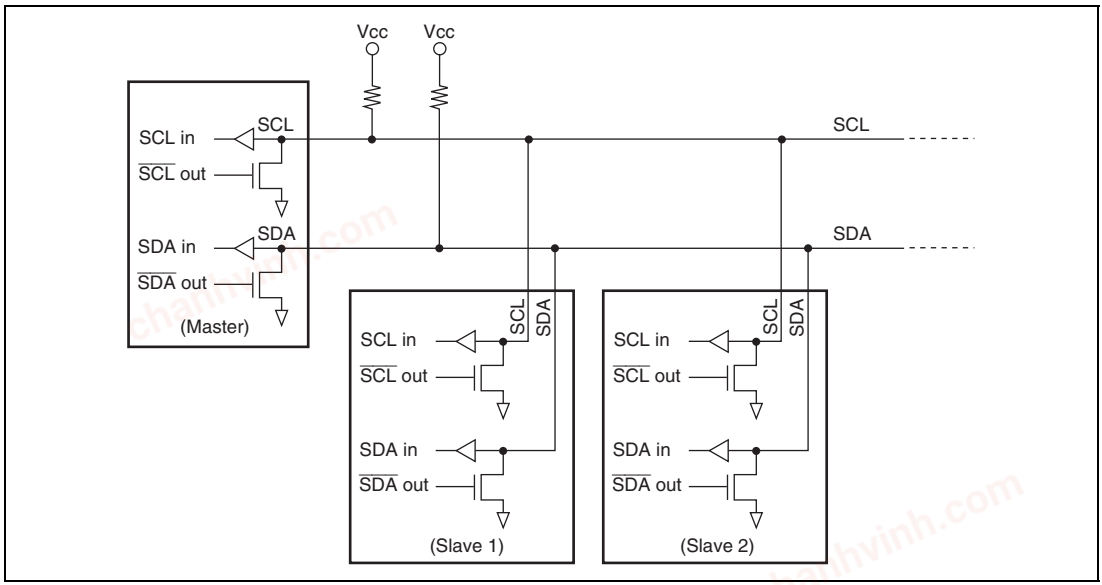



**Figure 21.2 External Circuit Connections of I/O Pins**

Table 21.1 summarizes the pin configuration used by the I²C bus interface 2.

**Table 21.1 Pin Configuration**

| Pin Name | I/O | Function |
|---|---|---|
| SCL | I/O | IIC serial clock input/output |
| SDA | I/O | IIC serial data input/output |

Note: When the IIC2 function is selected, use the PMC to allocate the SCL and SDA pin functions to P57 and P56, respectively.

chanhvinh.com



## 21.2 Register Descriptions

The IIC2 has the following registers.

- IIC2/SSU select register (ICSUSR)
- I²C bus control register 1 (ICCR1)
- I²C bus control register 2 (ICCR2)
- I²C bus mode register (ICMR)
- I²C bus interrupt enable register (ICIER)
- I²C bus status register (ICSR)
- I²C bus slave address register (SAR)
- I²C bus transmit data register (ICDRT)
- I²C bus receive data register (ICDRR)
- I²C bus shift register (ICDRS)

### 21.2.1 IIC2/SSU Select Register (ICSUSR)

Address: H'FF000B

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | SELICSU |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 to 1 | — | Reserved | These bits are read as 0. The write value should be 0. | — |
| 0 | SELICSU | IIC2/SSU module function select | 0: IIC2 function is selected.*<br>1: SSU function is selected. | R/W |

Note: To select the IIC2 function, this bit should be set to 0 without fail.

chanhvinh.com



### 21.2.2 I²C Bus Control Register 1 (ICCR1)

Address: H'FF05C8

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | ICE | RCVD | MST | TRS | CKS[3:0] | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | ICE | I²C bus interface 2 enable | 0: This module is stopped. (SCL and SDA pins are set to port function.)<br>1: This bit is enabled for transfer operations. (SCL and SDA pins are bus drive state.) | R/W |
| 6 | RCVD | Reception disable | 0: Enables next reception<br>1: Disables next reception | R/W |
| 5 | MST | Master/slave select | 00: Slave receive mode<br>01: Slave transmit mode<br>10: Master receive mode<br>11: Master transmit mode | R/W |
| 4 | TRS | Transmit/ receive select | | R/W |
| 3 to 0 | CKS[3:0] | Transfer clock select 3 to 0 | These bits should be set according to the necessary transfer rate (see table 21.2) in master mode. | R/W |

- RCVD bit (reception disable)

  Selects to enable or disable the next operation when TRS is 0 and ICDRR is read.

- MST bit (master/slave select) and TRS bit (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 performed between transfer frames.

  After data receive has been started in slave receive mode, when the first seven bits of the receive data agree with the slave address that is set to SAR and the eighth bit is 1, TRS is automatically set to 1. If an overrun error occurs in master mode with the clock synchronous serial format, MST is cleared to 0 and slave receive mode is entered.

  Operating modes are described above according to MST and TRS combination. When clock synchronous serial format is selected and MST is 1, clock is output.

chanhvinh.com



- CKS[3:0] bits (transfer clock select 3 to 0)

  These bits should be set according to the necessary transfer rate (see table 21.2) in master mode. In slave mode, these bits are used for reservation of the data setup time in transmit mode. The time is 10 $t_{cyc}$ when CKS3 = 0 and 20 $t_{cyc}$ when CKS3 = 1.

**Table 21.2 Transfer Rate**

| Bit 3 | Bit 2 | Bit 1 | Bit 0 | | Transfer Rate | | | | |
|---|---|---|---|---|---|---|---|---|---|
| CKS3 | CKS2 | CKS1 | CKS0 | Clock | φ = 5 MHz | φ = 8 MHz | φ = 10 MHz | φ = 16 MHz | φ = 20 MHz |
| 0 | 0 | 0 | 0 | φ/28 | 179 kHz | 286 kHz | 357 kHz | 571 kHz | 714 kHz |
| | | | 1 | φ/40 | 125 kHz | 200 kHz | 250 kHz | 400 kHz | 500 kHz |
| | | 1 | 0 | φ/48 | 104 kHz | 167 kHz | 208 kHz | 333 kHz | 417 kHz |
| | | | 1 | φ/64 | 78.1 kHz | 125 kHz | 156 kHz | 250 kHz | 313 kHz |
| | 1 | 0 | 0 | φ/80 | 62.5 kHz | 100 kHz | 125 kHz | 200 kHz | 250 kHz |
| | | | 1 | φ/100 | 50.0 kHz | 80.0 kHz | 100 kHz | 160 kHz | 200 kHz |
| | | 1 | 0 | φ/112 | 44.6 kHz | 71.4 kHz | 89.3 kHz | 143 kHz | 179 kHz |
| | | | 1 | φ/128 | 39.1 kHz | 62.5 kHz | 78.1 kHz | 125 kHz | 156 kHz |
| 1 | 0 | 0 | 0 | φ/56 | 89.3 kHz | 143 kHz | 179 kHz | 286 kHz | 357 kHz |
| | | | 1 | φ/80 | 62.5 kHz | 100 kHz | 125 kHz | 200 kHz | 250 kHz |
| | | 1 | 0 | φ/96 | 52.1 kHz | 83.3 kHz | 104 kHz | 167 kHz | 208 kHz |
| | | | 1 | φ/128 | 39.1 kHz | 62.5 kHz | 78.1 kHz | 125 kHz | 156 kHz |
| | 1 | 0 | 0 | φ/160 | 31.3 kHz | 50.0 kHz | 62.5 kHz | 100 kHz | 125 kHz |
| | | | 1 | φ/200 | 25.0 kHz | 40.0 kHz | 50.0 kHz | 80.0 kHz | 100 kHz |
| | | 1 | 0 | φ/224 | 22.3 kHz | 35.7 kHz | 44.6 kHz | 71.4 kHz | 89.3 kHz |
| | | | 1 | φ/256 | 19.5 kHz | 31.3 kHz | 39.1 kHz | 62.5 kHz | 78.1 kHz |

chanhvinh.com



### 21.2.3 I²C Bus Control Register 2 (ICCR2)

Address: H'FF05C9

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | BBSY | SCP | SDAO | SDAOP | SCLO | — | IICRST | — |
| Value after reset: | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | BBSY*[1]*[3] | Bus busy | This bit enables to confirm whether the I²C bus is occupied or released and to issue start/stop conditions in master mode. With the clock synchronous serial format, this bit has no meaning. 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. To issue start/stop conditions, use the MOV instruction. | R/W |
| 6 | SCP*[3] | Start/stop condition issue disable | The SCP bit 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. If 1 is written, the data is not stored. | R/W |
| 5 | SDAO*[3] | SDA output value control | This bit is used with SDAOP (bit 4) when modifying output level of SDA. This bit should not be manipulated during transfer. Writing 1 to the IICRST bit also sets this bit to 1.<br>0: When reading, SDA pin outputs low.<br>When writing, SDA pin is changed to output low.<br>1: When reading, SDA pin outputs high.<br>When writing, SDA pin is changed to output Hi-Z (outputs high by external pull-up resistance). | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 4 | SDAOP | SDAO write protect | This bit 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 by the MOV instruction. This bit is always read as 1. | R/W |
| 3 | SCLO | SCL output level monitor | This bit monitors SCL output level. When reading and SCLO is 1, SCL pin outputs high. When reading and SCLO is 0, SCL pin outputs low. Writing 1 to the IICRST bit also sets this bit to 1. | R |
| 2 | — | Reserved | This bit is read as 1. The write value should be 1. | — |
| 1 | IICRST*[2] | IIC control part reset | This bit resets the control part except for I²C registers. If this bit is set to 1 when hang-up occurs because of communication failure during I²C operation, I²C control part can be reset without setting ports and initializing registers. | R/W |
| 0 | — | Reserved | This bit is read as 1. The write value should be 1. | — |

Notes: 1. In standby mode, the BBSY bit in ICCR2 is reset.
2. Clear IICRST to 0 by software since this bit is not cleared automatically.
3. Writing to this bit is invalid during a reset due to setting the IICRST bit in ICCR2 to 1.

chanhvinh.com



### 21.2.4 I²C Bus Mode Register (ICMR)

Address: H'FF05CA

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | MLS | WAIT | — | — | BCWP | BC[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | MLS | MSB-first/LSB-first select | 0: Transfer in MSB-first*<br>1: Transfer in LSB-first | R/W |
| 6 | WAIT | Wait insertion | 0: Data and acknowledge bits are transferred consecutively with no wait inserted.<br>1: After the fall of the clock for the final data bit, low period is extended for two transfer clocks. | R/W |
| 5 | — | Reserved | This bit is read as 0. The write value should be 0. | — |
| 4 | — | Reserved | This bit is read as 1. The write value should be 1. | — |
| 3 | BCWP | BC write protect | 0: When writing, modifying BC2 to BC0 values is valid.<br>1: When writing, modifying BC2 to BC0 values is invalid. | R/W |
| 2 to 0 | BC[2:0] | Bit counter 2 to 0 | I²C Bus Format / Clock Synchronous Serial Format<br>000: 9 bits / 000: 8 bits<br>001: 2 bits / 001: 1 bits<br>010: 3 bits / 010: 2 bits<br>011: 4 bits / 011: 3 bits<br>100: 5 bits / 100: 4 bits<br>101: 6 bits / 101: 5 bits<br>110: 7 bits / 110: 6 bits<br>111: 8 bits / 111: 7 bits | R/W |

Note: * Set this bit to 0 when the I²C bus format is used.

chanhvinh.com



- WAIT bit (wait insertion)

  In master mode with the I²C bus format, this bit selects whether to insert a wait after data transfer except the acknowledge bit. When WAIT is set to 1, after the fall of the clock for the final data bit, low period is extended for two transfer clocks. If WAIT is cleared to 0, data and acknowledge bits are transferred consecutively with no wait inserted.

  The setting of this bit is invalid in slave mode with the I²C bus format or with the clock synchronous serial format.

- BCWP bit (BC write protect)

  Controls the BC2 to BC0 modifications. When modifying BC2 to BC0, this bit should be cleared to 0 and use the MOV instruction. In clock synchronous serial mode, BC should not be modified.

- BC[2:0] bits (bit counter 2 to 0)

  Specifies 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 additional acknowledge bit. Bit BC2 to BC0 should be set during an interval between transfer frames. If bits BC2 to BC0 are set to a value other than 000, the setting should be made while the SCL pin is low. The value automatically returns to 000 at the end of a data transfer, including the acknowledge bit. With the clock synchronous serial format, these bits should not be modified.

chanhvinh.com



### 21.2.5 I²C Bus Interrupt Enable Register (ICIER)

Address: H'FF05CB

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TIE | TEIE | RIE | NAKIE | STIE | ACKE | ACKBR | ACKBT |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | TIE | Transmit interrupt enable | 0: Transmit data empty interrupt request (TXI) is disabled.<br>1: Transmit data empty interrupt request (TXI) is enabled. | R/W |
| 6 | TEIE | Transmit end interrupt enable | 0: Transmit end interrupt request (TEI) is disabled.<br>1: Transmit end interrupt request (TEI) is enabled. | R/W |
| 5 | RIE | Receive interrupt enable | 0: Receive data full interrupt request (RXI) is disabled.<br>1: Receive data full interrupt request (RXI) is enabled. | R/W |
| 4 | NAKIE | NACK receive interrupt enable | 0: NACK receive interrupt request (NAKI) and overrun error interrupt request (ERI) with the clock synchronous format are disabled.<br>1: NACK receive interrupt request (NAKI) and overrun error interrupt request (ERI) with the clock synchronous format are enabled. | R/W |
| 3 | STIE | Stop condition detection interrupt enable | 0: Stop condition detection interrupt request (STPI) is disabled.<br>1: Stop condition detection interrupt request (STPI) is enabled. | R/W |
| 2 | ACKE | Acknowledge bit judgment select | 0: The value of the receive acknowledge bit is ignored, and continuous transfer is performed.<br>1: If the receive acknowledge bit is 1, continuous transfer is stopped. | R/W |
| 1 | ACKBR | Receive acknowledge | 0: Receive acknowledge = 0<br>1: Receive acknowledge = 1 | R |
| 0 | ACKBT | Transmit acknowledge | 0: 0 is sent at the acknowledge timing.<br>1: 1 is sent at the acknowledge timing. | R/W |

chanhvinh.com



- TIE bit (transmit interrupt enable)

  When the TDRE bit in ICSR is set to 1, this bit enables or disables the transmit data empty interrupt (TXI).

- TEIE bit (transmit end interrupt enable)

  This bit 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.

- RIE bit (receive interrupt enable)

  This bit enables or disables the receive data full interrupt request (RXI) when a receive data is transferred from ICDRS to ICDRR and the RDRF bit in ICSR is set to 1. RXI can be canceled by clearing the RDRF or RIE bit to 0.

- NAKIE bit (NACK receive interrupt enable)

  This bit enables or disables the NACK receive interrupt request (NAKI) and the overrun error (setting of the OVE bit in ICSR) interrupt request (ERI) with the clock synchronous format, when the NACKF and AL bits in ICSR are set to 1. NAKI can be canceled by clearing the NACKF, OVE, or NAKIE bit to 0.

- STIE bit (stop condition detection interrupt enable)

  This bet should be set to 1 while the STOP bit in ICSR is 0.

- ACKBR bit (receive acknowledge)

  In transmit mode, this bit stores the acknowledge data that are returned by the receive device. This bit cannot be modified.

- ACKBT bit (transmit acknowledge)

  In receive mode, this bit specifies the bit to be sent at the acknowledge timing.

chanhvinh.com



### 21.2.6 I²C Bus Status Register (ICSR)

Address: H'FF05CC

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TDRE | TEND | RDRF | NACKF | STOP | AL_OVE | AAS | ADZ |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | TDRE | Transmit data empty flag | [Setting conditions]<br>• When data is transferred from ICDRT to ICDRS and ICDRT becomes empty<br>• When TRS is set<br>• When a start condition (including re-transfer) has been issued<br>• When transmit mode is entered from receive mode in slave mode<br>• When 1 is written to IICRST in ICCR2 in master transmit mode and slave transmit mode<br>[Clearing conditions]<br>• When 0 is written in TDRE after reading TDRE = 1<br>• When data is written to ICDRT with an instruction<br>• When the DTC transfers data to ICDRT by a TXI interrupt request, and the DTC settings satisfy the flag clearing conditions. | R/W |
| 6 | TEND | Transmit end flag | [Setting conditions]<br>• When the ninth clock of SCL rises with the I²C bus format while the TDRE flag is 1<br>• When the final bit of transmit frame is sent with the clock synchronous serial format<br>[Clearing conditions]<br>• When 0 is written in TEND after reading TEND = 1<br>• When data is written to ICDRT with an instruction | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 5 | RDRF | Receive data register full flag | [Setting condition]<br>• When a receive data is transferred from ICDRS to ICDRR<br>[Clearing conditions]<br>• When 0 is written in RDRF after reading RDRF = 1<br>• When ICDRR is read with an instruction<br>• When the DTC transfers data to ICDRR by an RXI interrupt request, and the DTC settings satisfy the flag clearing conditions. | R/W |
| 4 | NACKF | No acknowledge detection flag | [Setting condition]<br>• When no acknowledge is detected from the receive device in transmission while the ACKE bit in ICIER is 1<br>[Clearing condition]<br>• When 0 is written in NACKF after reading NACKF = 1 | R/W |
| 3 | STOP | Stop condition detection flag | [Setting conditions]<br>• When a stop condition is detected after frame transfer end<br>[Clearing condition]<br>• When 0 is written in STOP after reading STOP = 1 | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 2 | AL_OVE | Arbitration lost flag/overrun error flag | [Setting conditions]<br>• If the internal SDA and SDA pin disagree at the rise of SCL in master transmit mode<br>• When the SDA pin outputs high in master mode while a start condition is detected<br>• When the final bit is received with the clock synchronous format while RDRF = 1<br>[Clearing condition]<br>• When 0 is written in AL/OVE after reading AL/OVE = 1 | R/W |
| 1 | AAS | Slave address recognition flag | [Setting conditions]<br>• When the slave address is detected in slave receive mode<br>• When the general call address is detected in slave receive mode.<br>[Clearing condition]<br>• When 0 is written in AAS after reading AAS = 1 | R/W |
| 0 | ADZ | General call address recognition flag | This bit is enabled in slave receive mode with I²C bus format.<br>[Setting condition]<br>• When the general call address is detected in slave receive mode<br>[Clearing condition]<br>• When 0 is written in ADZ after reading ADZ = 1 | R/W |

Notes: In standby mode, ICSR is reset.

* The DTC clears the peripheral module flags when all of the following three conditions are satisfied.

1. When the DISEL bit is 0.
2. When the transfer counter is not 0. (DTC transfer count register A (CRA) in normal mode and repeat mode, or DTC transfer count register B (CRB) in block mode)
3. When chain transfer is not used.

chanhvinh.com



- AL_OVE bit (arbitration lost flag/overrun error flag)

  This 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 clock synchronous format.

  When two or more master devices attempt to seize the bus at nearly the same time, if the I²C bus interface detects data differing from the data it sent, it sets AL to 1 to indicate that the bus has been taken by another master.

- AAS bit (slave address recognition flag)

  In slave receive mode, this flag is set to 1 if the first frame following a start condition matches bits SVA6 to SVA0 in SAR.

### 21.2.7 Slave Address Register (SAR)

Address: H'FF05CD

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | SVA[6:0] | | | | | | | FS |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 to 1 | SVA[6:0] | Slave address 6 to 0 | These bits set a unique address in bits SVA[6:0], differing form the addresses of other slave devices connected to the I²C bus. | R/W |
| 0 | FS | Format select | 0: I²C bus format is selected.<br>1: Clock synchronous serial format is selected. | R/W |

SAR selects the format and sets the slave address. When SAR is in slave mode with the I²C bus format, if the upper 7 bits of SAR match the upper 7 bits of the first frame received after a start condition, SAR operates as the slave device.

chanhvinh.com



### 21.2.8 I²C Bus Transmit Data Register (ICDRT)

Address: H'FF05CE

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

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 during transferring data of ICDRS, continuous transfer is possible. If the MLS bit of ICMR is set to 1 and when the data is written to ICDRT, the MSB/LSB inverted data is read. The initial value of ICDRT is H'FF. ICDRT is reset in standby mode.

### 21.2.9 I²C Bus Receive Data Register (ICDRR)

Address: H'FF05CF

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

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. The initial value of ICDRR is H'FF. ICDRR is reset in standby mode.

### 21.2.10 I²C Bus Shift Register (ICDRS)

Address: —

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | — | — | — | — | — | — | — | — |

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. ICDRS is reset in standby mode.

chanhvinh.com



## 21.3 Operation

The I²C bus interface 2 can communicate either in I²C bus mode or clock synchronous serial mode by setting FS in SAR.

### 21.3.1 I²C Bus Format

Figure 21.3 shows the I²C bus formats. Figure 21.4 shows the I²C bus timing. The first frame following a start condition always consists of 8 bits.

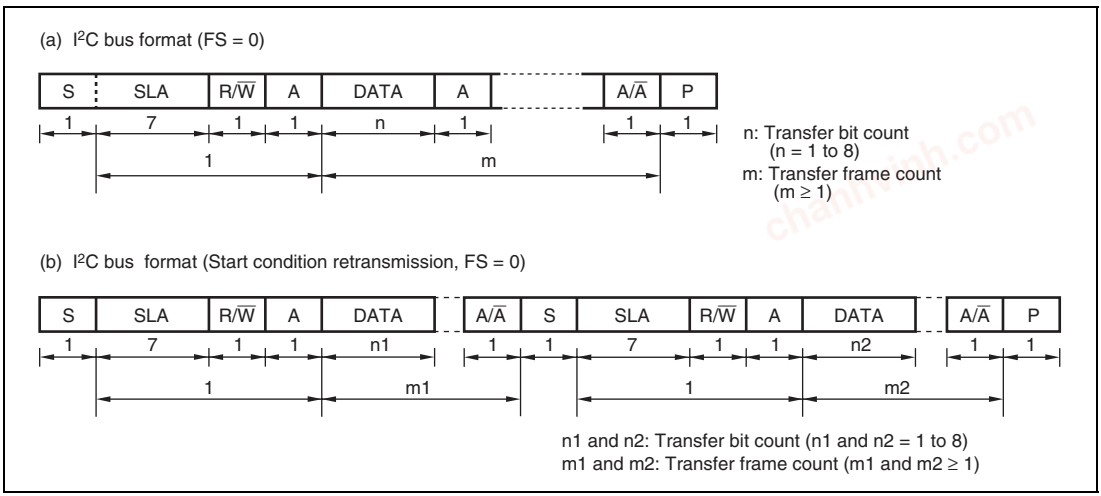



**Figure 21.3 I²C Bus Formats**

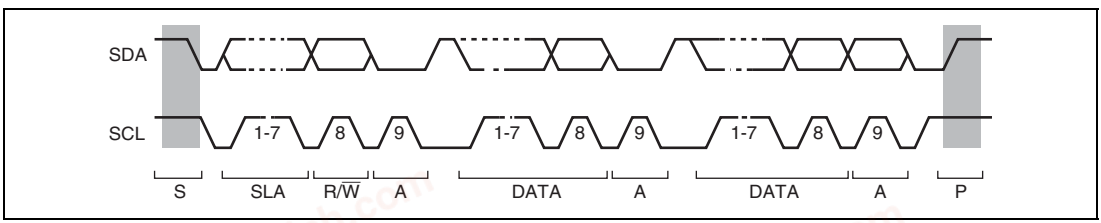



**Figure 21.4 I²C Bus Timing**

chanhvinh.com



[Legend]

S: Start condition. The master device drives SDA from high to low while SCL is high.

SLA: Slave address

R/$\overline{W}$: Indicates the direction of data transfer: from the slave device to the master device when R/$\overline{W}$ is 1, or from the master device to the slave device when R/$\overline{W}$ 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.

### 21.3.2 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, see figures 21.5 and 21.6. The transmission procedure and operations in master transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MLS and WAIT bits in ICMR and the CKS3 to CKS0 bits in ICCR1 to 1. (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 using MOV instruction. (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 using MOV instruction. 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.

chanhvinh.com



7. When the STOP bit in ICSR is set to 1, the operation returns to the slave receive mode.

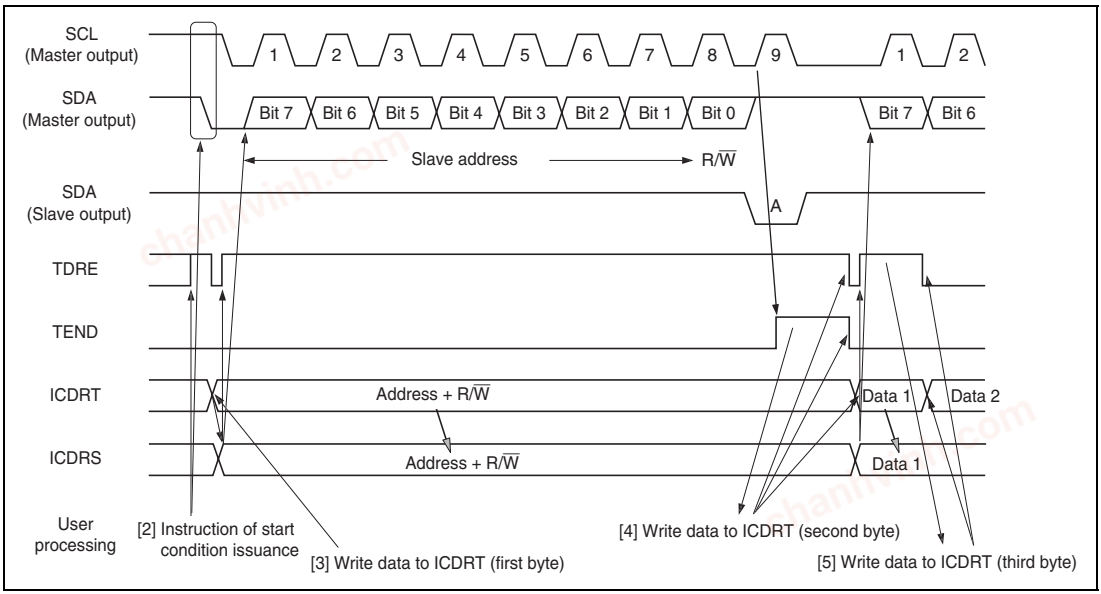



**Figure 21.5 Master Transmit Mode Operation Timing (1)**

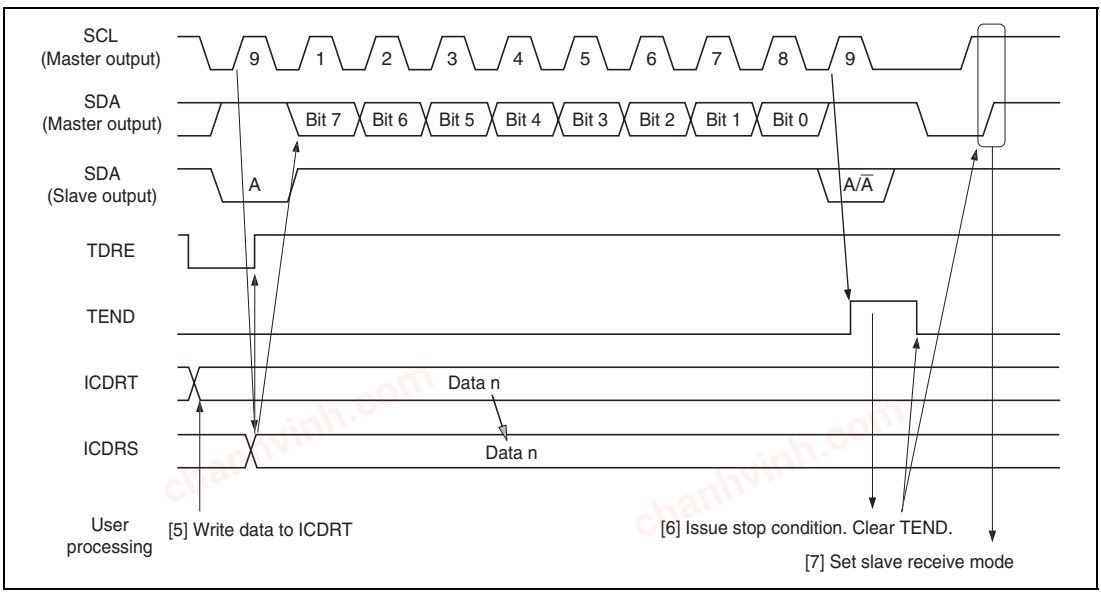



**Figure 21.6 Master Transmit Mode Operation Timing (2)**

chanhvinh.com



### 21.3.3 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, see figures 21.7 and 21.8. 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 read), reception is started. And the receive clock is output to receive data 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 stop 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.

chanhvinh.com



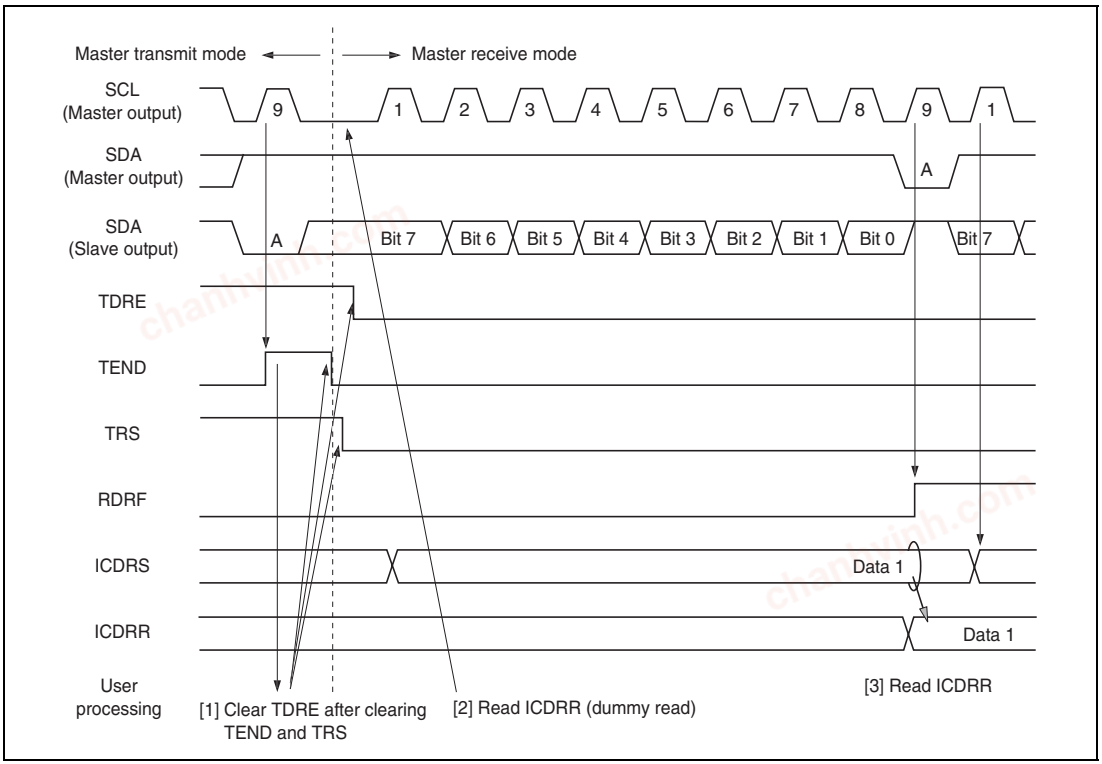



**Figure 21.7 Master Receive Mode Operation Timing (1)**

chanhvinh.com



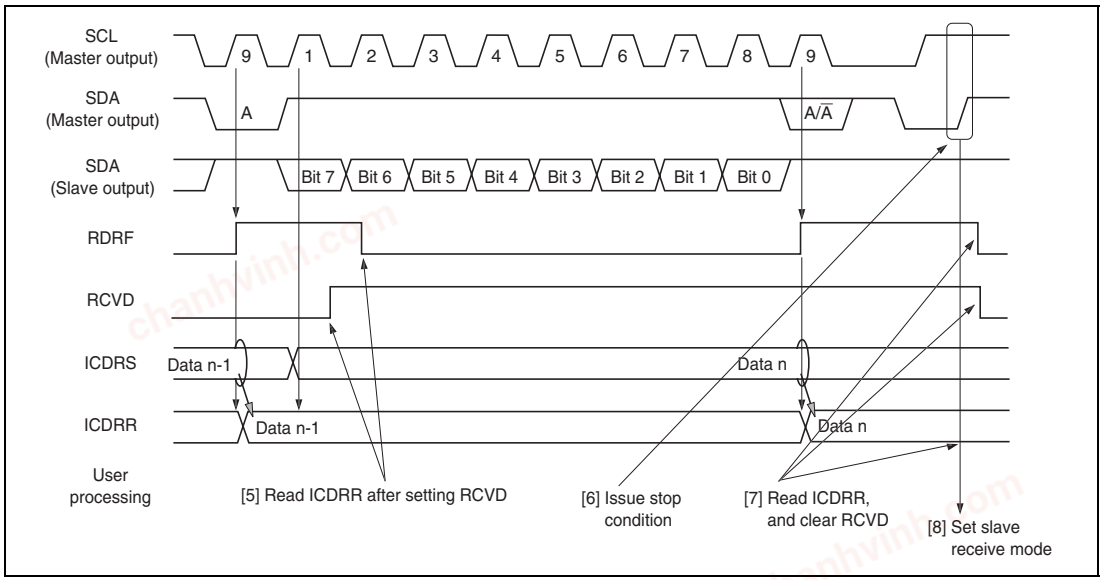



**Figure 21.8 Master Receive Mode Operation Timing (2)**

### 21.3.4 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, see figures 21.9 and 21.10.

The transmission procedure and operations in slave transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MLS and WAIT bits in ICMR and the CKS3 to CKS0 bits in ICCR1 to 1 (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/$\overline{W}$) is 1, the TRS and ICSR bits in ICCR1 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 released.
5. Clear TDRE.

chanhvinh.com



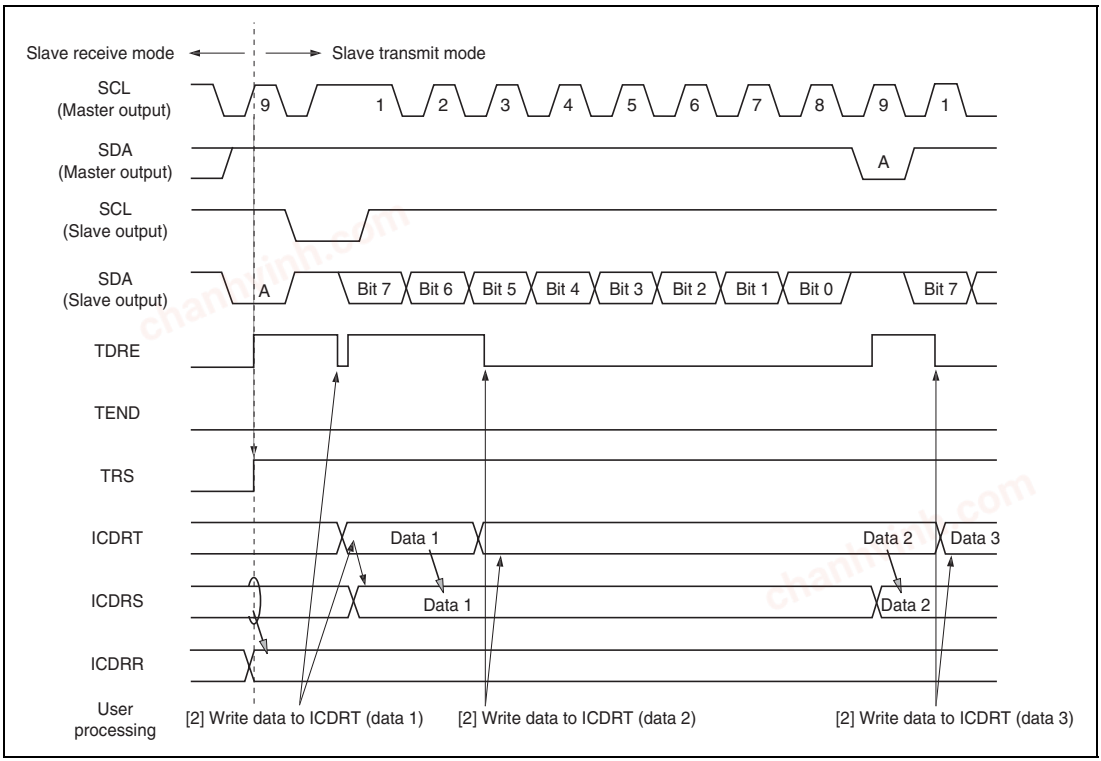



**Figure 21.9 Slave Transmit Mode Operation Timing (1)**

chanhvinh.com



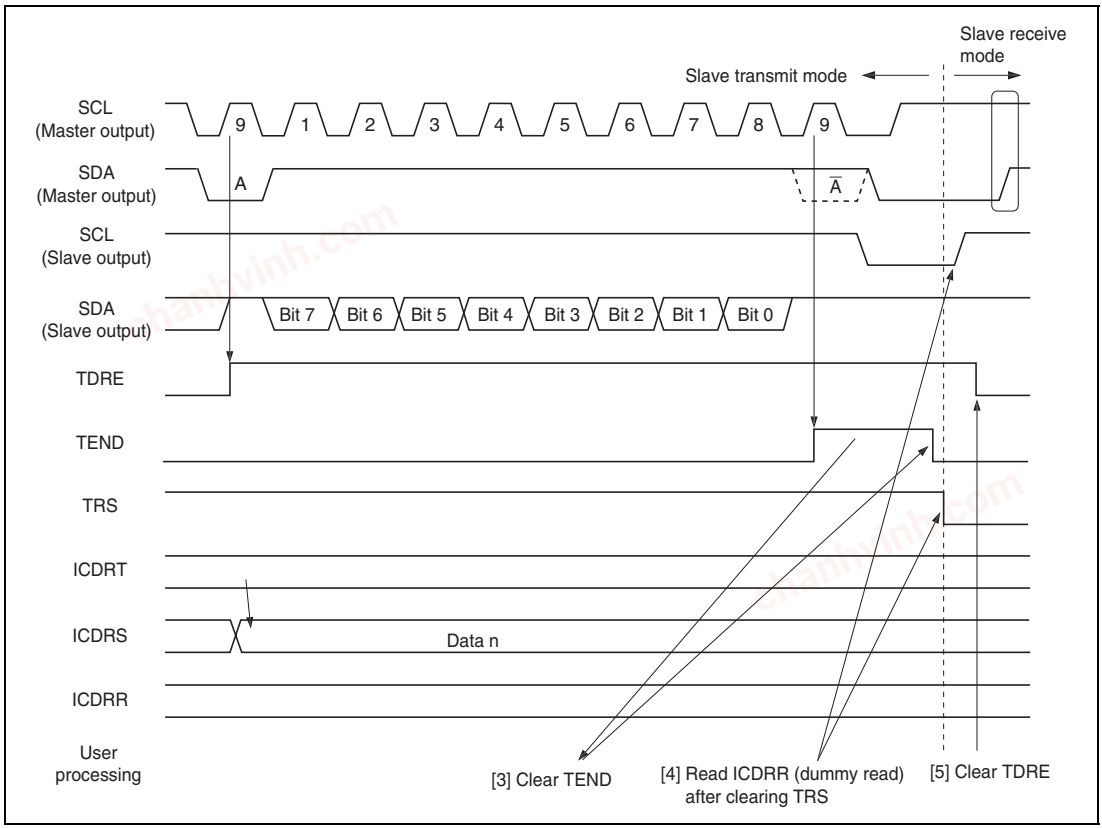



**Figure 21.10 Slave Transmit Mode Operation Timing (2)**

chanhvinh.com



### 21.3.5 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, see figures 21.11 and 21.12. The reception procedure and operations in slave receive mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MLS and WAIT bits in ICMR and the CKS3 to CKS0 bits in ICCR1 to 1. (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 + R/$\overline{W}$, it is not used.)
3. Read ICDRR every time RDRF is set. If 8th receive clock pulse falls while RDRF is set to 1, SCL is fixed low until ICDRR is read. The change of the acknowledge before reading ICDRR, which is returned to the master device, is reflected to the next transmit frame.
4. The last byte data is read by reading ICDRR.

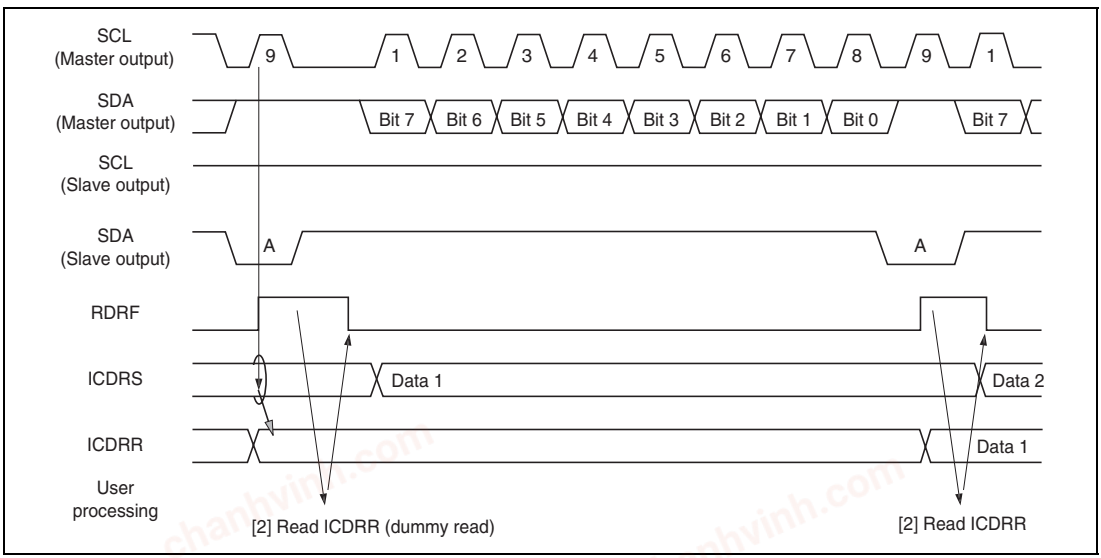



**Figure 21.11 Slave Receive Mode Operation Timing (1)**

chanhvinh.com



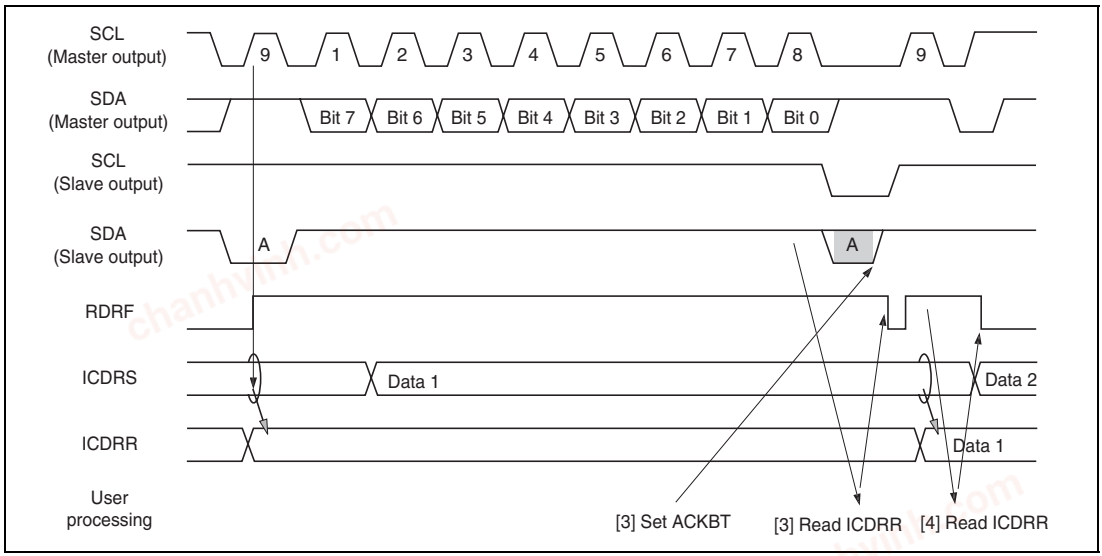



**Figure 21.12 Slave Receive Mode Operation Timing (2)**

### 21.3.6 Clock Synchronous Serial Format

This module can be operated with the clock 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 21.13 shows the clock synchronous serial transfer format.

The transfer data is output from the rise 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.

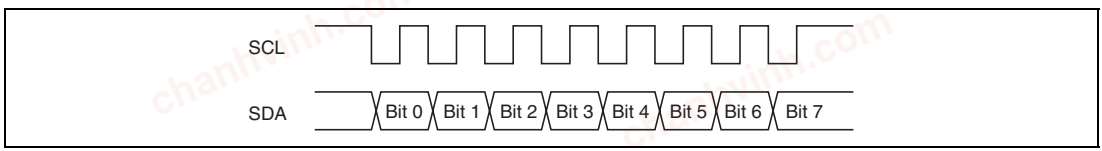



**Figure 21.13 Clock Synchronous Serial Transfer Format**

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, see figure 21.14. The transmission procedure and operations in transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MST and CKS3 to CKS0 bits in ICCR1 to 1 (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 set to 1.

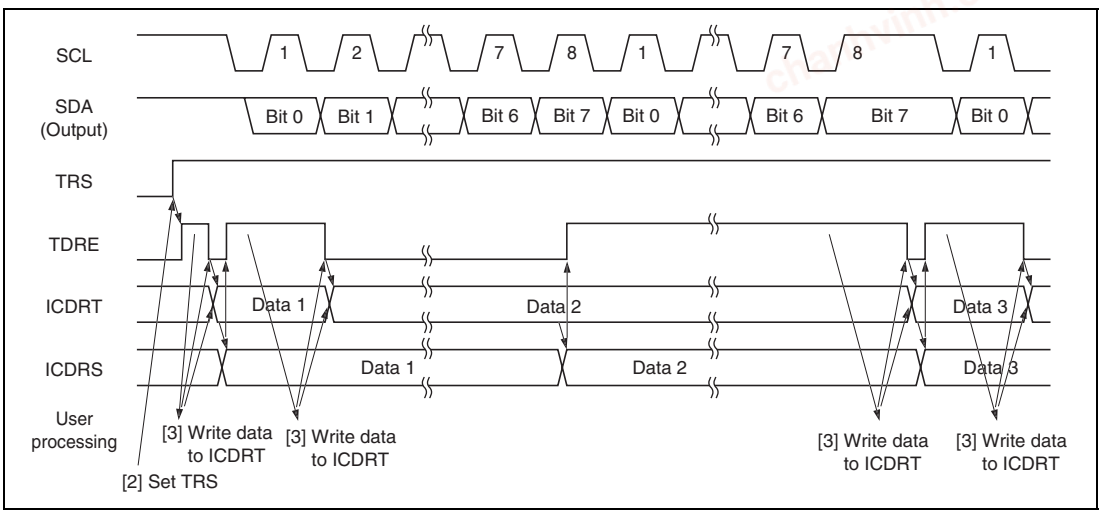



**Figure 21.14 Transmit Mode Operation Timing**

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, see figure 21.15. The reception procedure and operations in receive mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MST and CKS3 to CKS0 bits in ICCR1 to 1 (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 set to 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, read ICDRR after setting RCVD in ICCR1 to 1. Then, SCL is fixed high after receiving the next byte data.

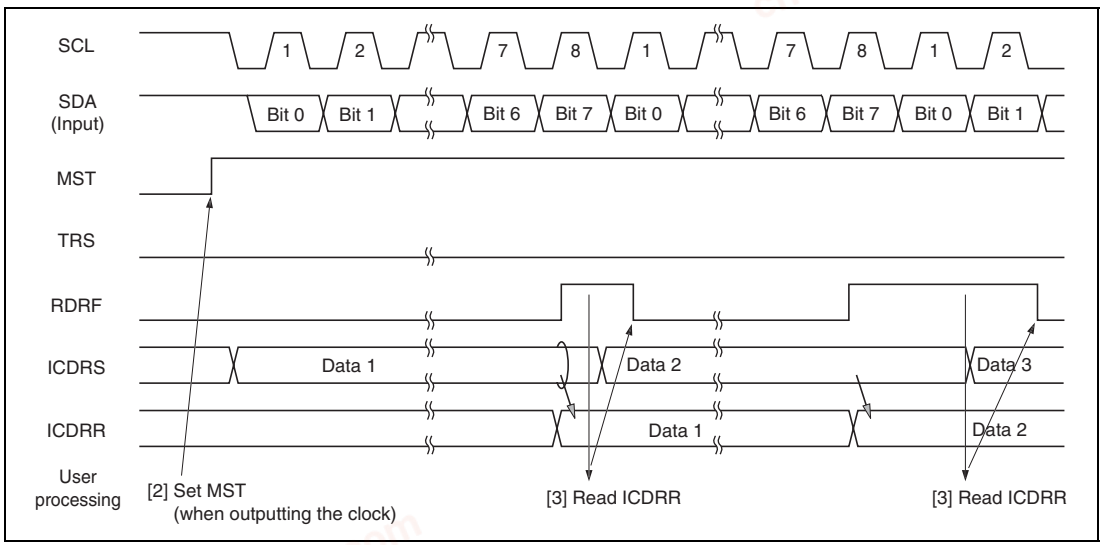



**Figure 21.15 Receive Mode Operation Timing**

chanhvinh.com



### 21.3.7 Noise Filter Circuit

The signal state on the SCL and SDA pins are internally latched via the noise filter circuit. Figure 21.16 shows a block diagram of the noise filter circuit.

The noise filter consists of two cascaded latches and a match detector. The SCL (or SDA) input signal is sampled on the system clock. When both outputs of the latches match, its level is output to other blocks by the match detector circuit. If they do not match, the previous value is held.

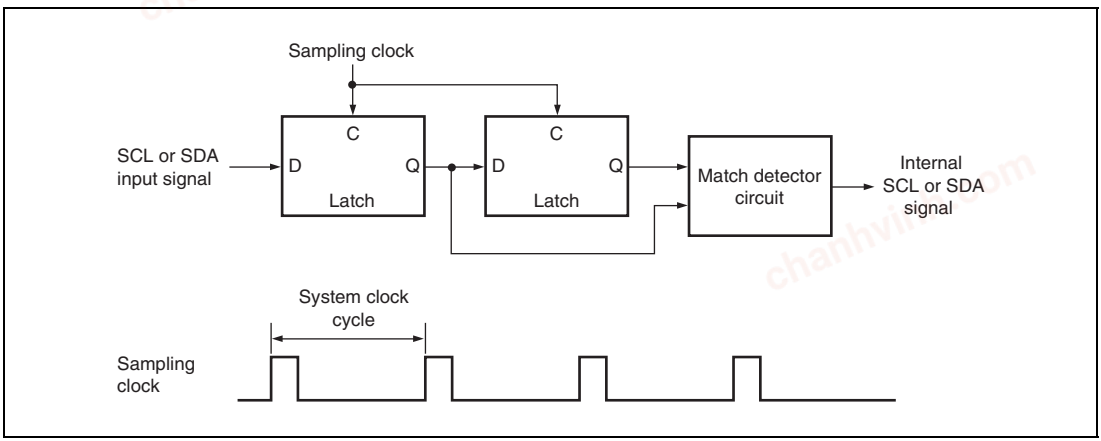



**Figure 21.16 Block Diagram of Noise Filter Circuit**

chanhvinh.com



### 21.3.8 Example of Use

Flowcharts in respective modes that use the I²C bus interface 2 are shown in figures 21.17 to 21.20.

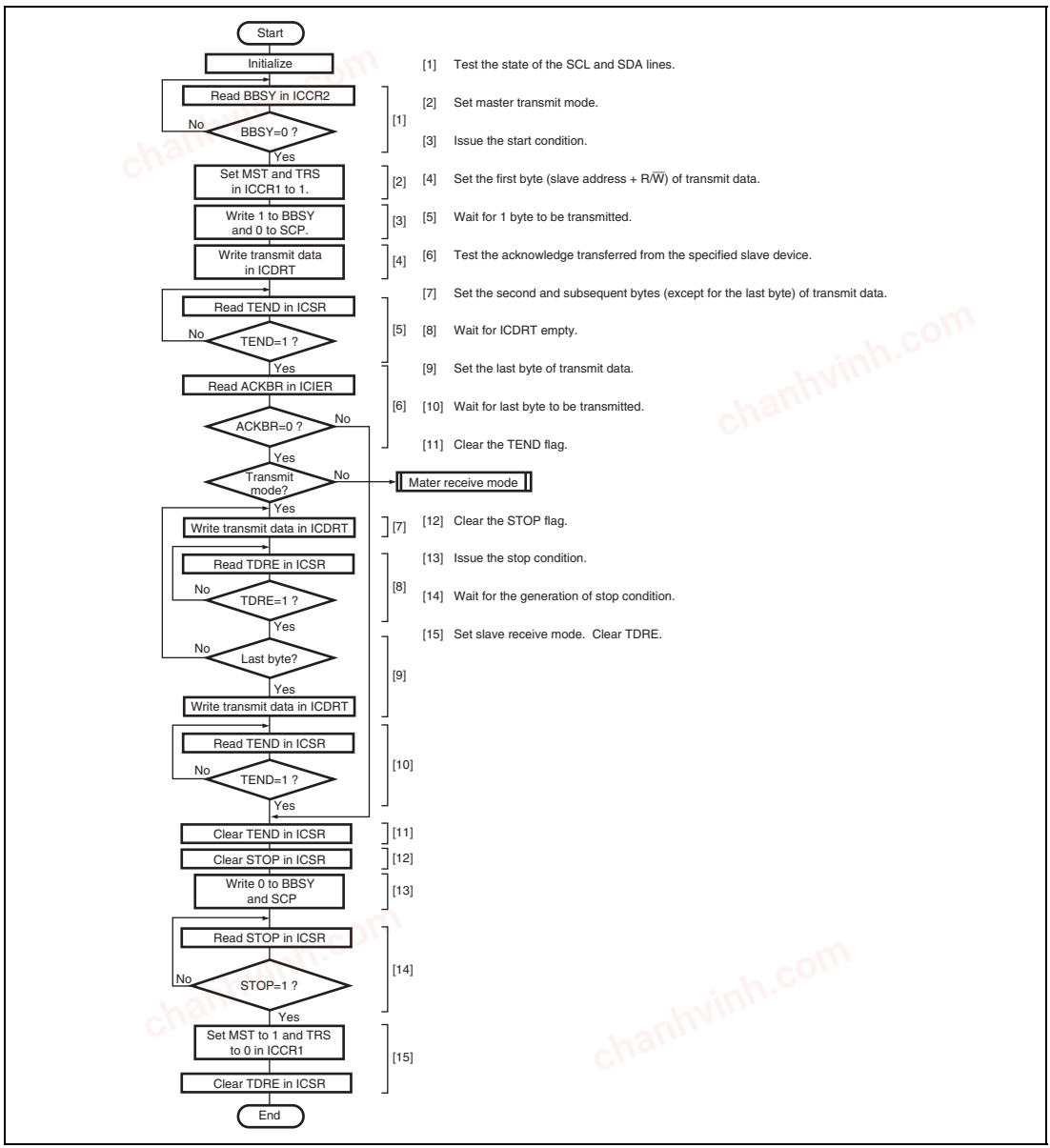
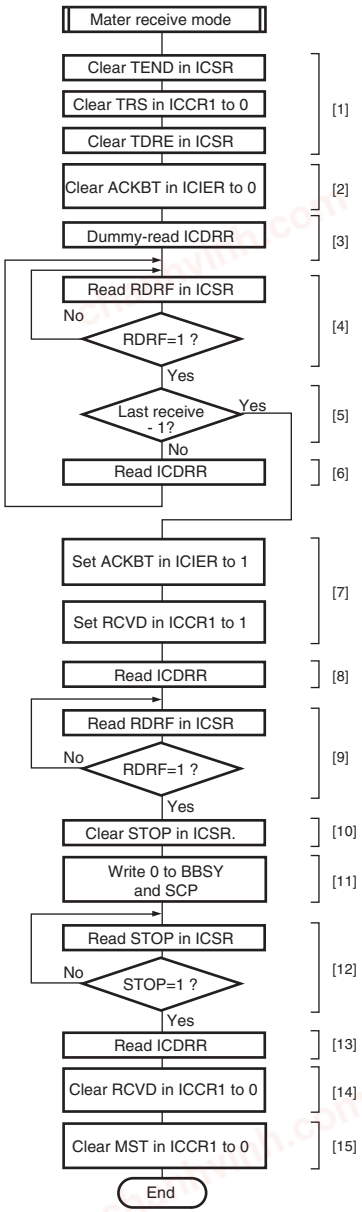



**Figure 21.17 Sample Flowchart for Master Transmit Mode**

chanhvinh.com






[1] Clear TEND, select master receive mode, and then clear TDRE.*

[2] Set acknowledge to the transmit device.*

[3] Dummy-read ICDRR.*

[4] Wait for 1 byte to be received

[5] Check whether it is the (last receive - 1).

[6] Read the receive data.

[7] Set acknowledge of the last byte. Disable continuous reception (RCVD = 1).

[8] Read the (last byte - 1) of receive data.

[9] Wait for the last byte to be receive.

[10] Clear the STOP flag.

[11] Issue the stop condition.

[12] Wait for the generation of stop condition.

[13] Read the last byte of receive data.

[14] Clear RCVD.

[15] Set slave receive mode.

Note: Do not activate an interrupt during the execution of steps [1] to [3].

Supplementary explanation: When one byte is received, steps [2] to [6] are skipped after step [1] before jumping to step [7]. The step [8] is dummy-read in ICDRR.

**Figure 21.18 Sample Flowchart for Master Receive Mode**

chanhvinh.com



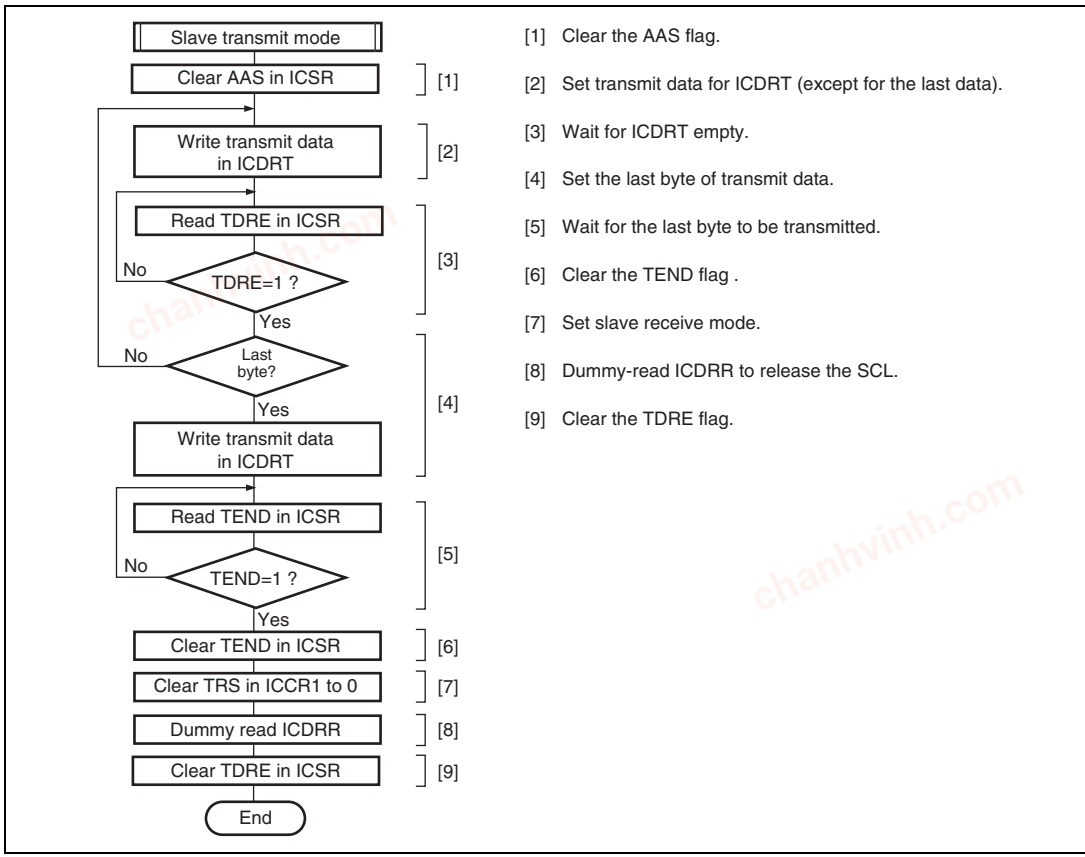
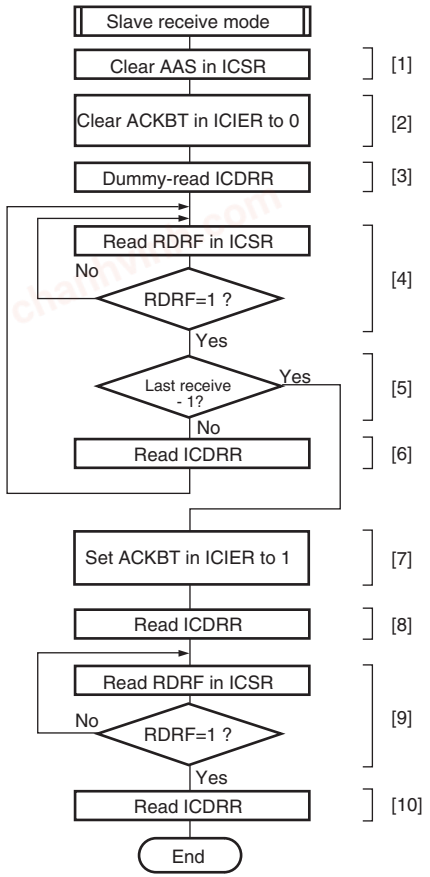



**Figure 21.19 Sample Flowchart for Slave Transmit Mode**

chanhvinh.com






[1] Clear the AAS flag.

[2] Set acknowledge to the transmit device.

[3] Dummy-read ICDRR.

[4] Wait for 1 byte to be received.

[5] Check whether it is the (last receive - 1).

[6] Read the receive data.

[7] Set acknowledge of the last byte.

[8] Read the (last byte - 1) of receive data.

[9] Wait the last byte to be received.

[10] Read for the last byte of receive data.

Supplementary explanation: When one byte is received, steps [2] to [6] are skipped after step [1] before jumping to step [7]. The step [8] is dummy-read in ICDRR.

**Figure 21.20 Sample Flowchart for Slave Receive Mode**

chanhvinh.com



## 21.4 Interrupt Request

There are six interrupt requests in this module; transmit data empty, transmit end, receive data full, NACK detection, STOP condition detection, and arbitration lost/overrun error. Table 21.3 shows the contents of each interrupt request.

**Table 21.3 Interrupt Requests**

| Interrupt Request | Abbreviation | Interrupt Condition | I²C Mode | Clock Synchronous Mode |
|---|---|---|---|---|
| Transmit Data Empty | TXI | (TDRE=1) • (TIE=1) | O | O |
| Transmit End | TEI | (TEND=1) • (TEIE=1) | O | O |
| Receive Data Full | RXI | (RDRF=1) • (RIE=1) | O | O |
| STOP Condition Detection | STPI | (STOP=1) • (STIE=1) | O | × |
| NACK Detection | NAKI | {(NACKF=1)+(AL=1)} • (NAKIE=1) | O | × |
| Arbitration Lost/ Overrun Error | | | O | O |

When an exception processing is executed under interrupt conditions described in table 21.3, interrupt sources should be cleared in the exception processing. TDRE and TEND are automatically cleared to 0 by writing the transmit data to ICDRT. RDRF are automatically cleared to 0 by reading ICDRR. TDRE is set to 1 again at the same time when transmit data is written to ICDRT. When TDRE is cleared to 0, then an excessive data of one byte may be transmitted.

The DTC can be activated by a TXI interrupt request to transfer data. The TDRE flag is automatically cleared upon data transfer by the DTC. The DTC can also be activated by an RXI interrupt request to transfer data. The RDRF flag is automatically cleared to 0 upon data transfer by the DTC.

chanhvinh.com



## 21.5 Bit Synchronous Circuit

In master mode, this module has a possibility that high level period may be shortened 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 21.21 shows the timing of the bit synchronous circuit and table 21.4 shows the time when SCL output changes from low to Hi-Z and then SCL is monitored.

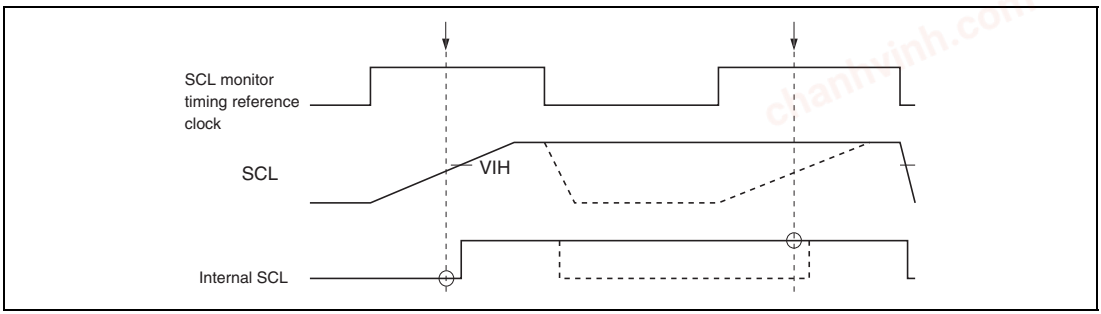



**Figure 21.21 The Timing of the Bit Synchronous Circuit**

**Table 21.4 Time for Monitoring SCL**

| CKS3 | CKS2 | Time for Monitoring SCL |
|---|---|---|
| 0 | 0 | 7.5 tcyc |
| | 1 | 19.5 tcyc |
| 1 | 0 | 17.5 tcyc |
| | 1 | 41.5 tcyc |

chanhvinh.com



## 21.6 Usage Notes

### 21.6.1 SCL and SDA pins selected by PMC

This LSI incorporates the IIC2 and SSU modules, one of which module functions should be selected by the SELICSU bit in ICSUSR. Therefore, when assigning the pin functions using the peripheral function mapping controller (PMC), the SCL and SDA pin functions should be assigned to the P56 and P57 pins when the IIC2 function is selected. If these pin functions are assigned to other pins, correct operation cannot be guaranteed.

### 21.6.2 Restriction on Use of Bit Manipulation Instructions to Set MST and TRS in Multi-Master Usage

When master transmission is selected by consecutively manipulating the MST and TRS bits in multi-master usage, an arbitration loss during execution of the bit-manipulation instruction for TRS leads to the contradictory situation where AL in ICSR is 1 in master transmit mode (MST = 1, TRS = 1).

Ways to avoid this effect are listed below.

- Use the MOV instruction to set MST and TRS when used in multi-master mode.
- When arbitration is lost, confirm that MST = 0 and TRS = 0. If the setting of MST = 0 and TRS = 0 is not confirmed, then set MST = 0 and TRS = 0 again.

### 21.6.3 Note regarding Master Receive Mode of I²C-Bus Interface Mode

When the interface is used with the I²C bus format in master receive mode, an overlap between issuing of the stop condition or re-issuing of the start condition and the falling edge of the ninth cycle of the SCL signal leads to output of the tenth cycle of SCL.

To avoid this problem, 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.

- After confirming that the receive data register full (RDRF) flag in the ICSR has become one, confirm that the SCL output level monitor (SCLO) bit in ICCR2 has become zero (indicating low-level output on the SCL pin).

chanhvinh.com



### 21.6.4 Note on Access to ICE in ICCR1 and IICRST in ICCR2 during I²C Bus Operation

Writing 0 to ICE in ICCR1 or 1 to IICRST in ICCR2 while any of the following four conditions during I²C bus operation is satisfied leads to undefined values for BBSY in ICCR2 and STOP in ICSR.

1. This module holds the I²C bus mastership and is in master transmit mode (MST = 1 and TRS = 1 in ICCR1)
2. This module holds the I²C bus mastership and is in master receive mode (MST = 1 and TRS = 0 in ICCR1)
3. This module is transmitting data in slave transmit mode (MST = 0 and TRS = 1 in ICCR1).
4. This module is transmitting an acknowledge signal in slave receive mode (MST = 0 and TRS = 0 in ICCR1).

Use any of the following workarounds to resolve the undefined state of BBSY in ICCR2.

- Input a start condition (falling edge of SDA while SCL is at the high level) so that BBSY is set to 1.
- Input a stop condition (rising edge of SDA while SCL is at the high level) so that BBSY is cleared to 0.
- In master transmit mode, write 1 to BBSY and 0 to SCP in ICCR2 while SCL and SDA are both at the high level to generate a start condition. BBSY is set to 1 when the start condition (falling edge of SDA while SCL is at the high level) is output.
- In master transmit mode or master receive mode, when SDA is at the low level and there is no device other than this module to drive SCL low, write 0 to BBSY and SCP in ICCR2 to generate a stop condition. BBSY is cleared to 0 when the stop condition (rising edge of SDA while SCL is at the high level) is output.
- Writing 1 to FS in SAR causes BBSY to be cleared to 0.

chanhvinh.com



# Section 22 Synchronous Serial Communication Unit (SSU)

Note: In this section, the synchronous serial communication unit is abbreviated as SSU for convenience.

The synchronous serial communication unit (SSU) can handle clocked synchronous serial data communication.

Figure 22.1 shows a block diagram of the SSU.

Either the SSU or IIC2 incorporated in this LSI can be used at a time. Accordingly, when the SSU function is used, the IIC2 function is not available.

## 22.1 Features

- Can be operated in clocked synchronous communication mode or four-line bus communication mode (including bidirectional communication mode)
- Can be operated as a master or a slave device
- Choice of seven internal clocks (φ/256, φ/128, φ/64, φ/32, φ/16, φ/8, φ/4) and an external clock as a clock source
- Clock polarity and phase of SSCK can be selected
- Choice of data transfer direction (MSB-first or LSB-first)
- Receive error detection: overrun error
- Multimaster error detection: conflict error
- Five interrupt sources: transmit-end, transmit-data-empty, receive-data-full, overrun error, and conflict error. The DTC can be activated by the transmit-data-empty and receive-data-full interrupts.
- The transmitter and receiver with buffer structure allow continuous transmission and reception of serial data.

chanhvinh.com



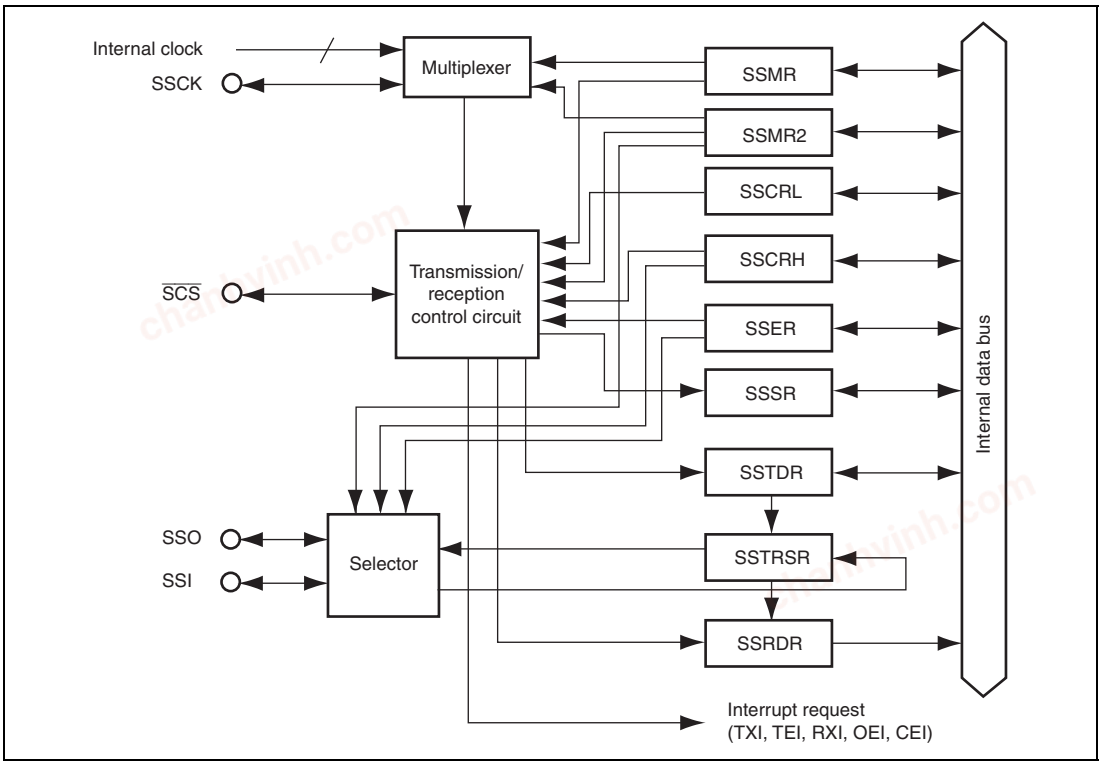



**Figure 22.1 Block Diagram of SSU**

Table 22.1 shows the pin configuration of the SSU.

**Table 22.1 Pin Configuration**

| Pin Name | I/O | Function |
|---|---|---|
| SSCK | I/O | SSU clock input/output |
| SSI | I/O | SSU data input/output |
| SSO | I/O | SSU data input/output |
| $\overline{SCS}$ | I/O | SSU chip select input/output |

chanhvinh.com



## 22.2 Register Descriptions

The SSU has the following registers.

- IIC2/SSU select register (ICSUSR)
- SS control register H (SSCRH)
- SS control register L (SSCRL)
- SS mode register (SSMR)
- SS mode register 2 (SSMR2)
- SS enable register (SSER)
- SS status register (SSSR)
- SS receive data register (SSRDR)
- SS transmit data register (SSTDR)
- SS shift register (SSTRSR)

### 22.2.1 IIC2/SSU Select Register (ICSUSR)

Address: H'FF000B

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | SELICSU |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 to 1 | — | Reserved | These bits are read as 0. The write value should be 0. | — |
| 0 | SELICSU | IIC2/SSU module function select | 0: IIC2 function is selected.<br>1: SSU function is selected.* | R/W |

Note: * To select the SSU function, this bit should be set to 1 without fail.

chanhvinh.com



### 22.2.2 SS Control Register H (SSCRH)

Address: H'FF05C8

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | RSSTP | MSB | — | — | CKS[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is read as 0. The write value should be 0. | — |
| 6 | RSSTP | Receive single stop | 0: After receiving 1 byte of data, reception continues.<br>1: After receiving 1 byte of data, reception ends.* | R/W |
| 5 | MSS | Master/slave device select | 0: Operates as a slave device<br>1: Operates as a master device | R/W |
| 4, 3 | — | Reserved | These bits are read as 0. The write value should be 0. | — |
| 2 to 0 | CKS[2:0] | Transfer clock rate select | 000: φ/256<br>001: φ/128<br>010: φ/64<br>011: φ/32<br>100: φ/16<br>101: φ/8<br>110: φ/4<br>111: Reserved | R/W |

Note: * The setting of the RSSTP bit is invalid when the MSS bit is cleared to 0.

- MSS bit (master/slave device select)
  Selects whether this module is used as a master device or a slave device. When this module is used as a master device, transfer clock is output from the SSCK pin. When the CE bit in SSSR is set, this bit is automatically cleared.

- CKS[2:0] bits (transfer clock rate select)
  Sets transfer clock rate (prescaler division ratio) when the internal clock is selected.

chanhvinh.com



### 22.2.3 SS Control Register L (SSCRL)

Address: H'FF05C9

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | SOL | SOLP | — | — | SRES | — |
| Value after reset: | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is read as 0. The write value should be 0. | — |
| 6 | — | Reserved | This bit is read as 1. The write value should be 1. | — |
| 5 | SOL*[1] | Serial data output level setting | 0: When reading, serial data output level is low. When writing, serial data output level is changed to low.<br>1: When reading, serial data output level is high. When writing, serial data output level is changed to high. | R/W |
| 4 | SOLP | SOL write protect | 0: When writing, output level can be changed according to the value of the SOL bit.<br>1: When reading, this bit is always read as 1. When writing, modifying to the SOL bit is invalid. | — |
| 3, 2 | — | Reserved | These bits are read as 1. The write value should be 1. | — |
| 1 | SRES | Software reset | 0: Does not reset.<br>1: The SSU internal sequencer is forcibly reset.*[2] | |
| 0 | — | Reserved | This bit is read as 1. The write value should be 1. | — |

Notes: 1. When the output level is changed, the SOLP bit (bit4) should be cleared to 0 and the MOV instruction should be used. If this bit is written during data transfer, erroneous operation may occur. Therefore this bit must not be manipulated during transmission.
2. This bit should always be cleared by software as this bit is not cleared automatically.

chanhvinh.com



- SOL bit (serial data output level setting)

  Although the value in the last bit of transmit data is retained in the serial data output after the end of transmission, the output level of serial data can be changed by manipulating this bit before or after transmission.

- SOLP bit (SOL write protect)

  When output level of serial data is changed, the MOV instruction is used to set the SOL bit to 1 and clear this bit to 0 or to clear the SOL bit and this bit to 0.

- SRES bit (software reset)

  When this bit is set to 1, the SSU internal sequencer is forcibly reset. The register value in the SSU is retained.

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

chanhvinh.com



### 22.2.4 SS Mode Register (SSMR)

Address: H'FF05CA

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | MLS | CPOS | CPHS | — | — | BC[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | MLS | MSB-first/LSB-first select | 0: Transfer by MSB-first<br>1: Transfer by LSB-first | R/W |
| 6 | CPOS | Clock polarity select | 0: SSCK clock idling state = high<br>1: SSCK clock idling state = low | R/W |
| 5 | CPHS | Clock phase select | 0: Data change at first edge<br>1: Data latch at first edge | R/W |
| 4, 3 | — | Reserved | This bit is read as 1. The write value should be 1. | — |
| 2 to 0 | BC[2:0] | Bit counter 2 to 0 | 000: 8 bits<br>001: 1 bit<br>010: 2 bits<br>011: 3 bits<br>100: 4 bits<br>101: 5 bits<br>110: 6 bits<br>111: 7 bits | R/W |

- BC[2:0] bits (bit counter 2 to 0)
  When read, the remaining number of transfer bits is indicated.

chanhvinh.com



### 22.2.5 SS Mode Register 2 (SSMR2)

Address: H'FF05CD

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | BIDE | SCKS | CSS[1:0] | | SCKOS | SOOS | CSOS | SSUMS |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | BIDE | Bidirectional mode enable | 0: Normal mode.<br>Communication is performed by using two pins.<br>1: Bidirectional mode.<br>Communication is performed by using only one pin. | R/W |
| 6 | SCKS | SSCK pin select | 0: Functions as a port*[1]<br>1: Functions as a serial clock pin | R/W |
| 5, 4 | CSS[1:0] | $\overline{SCS}$ pin select | 00: Functions as a port*[1]<br>01: Functions as an $\overline{SCS}$ input<br>1X: Functions as an $\overline{SCS}$ output (however, functions as an $\overline{SCS}$ input before starting transfer) | R/W |
| 3 | SCKOS | SSCK pin open-drain output select | 0: CMOS output<br>1: NMOS open-drain output*[2] | R/W |
| 2 | SOOS | SSO/SSI pins open-drain output select | SSO pin<br>0: CMOS output<br>1: NMOS open-drain output*[2]<br>SSI pin<br>0: CMOS output or NMOS push-pull output*[3]<br>1: NMOS open-drain output*[2] | R/W |
| 1 | CSOS | $\overline{SCS}$ pin open-drain output select | 0: CMOS output or NMOS push-pull output*[3]<br>1: NMOS open-drain output*[2] | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 0 | SSUMS | SSU mode select | 0: Clocked synchronous communication mode<br>Data input: SSI pin, Data output: SSO pin<br>1: Four-line bus communication mode<br>When MSS = 1 in SSCRH and BIDE = 0 in SSMR2:<br>Data input: SSI pin, Data output: SSO pin<br>When MSS = 0 in SSCRH and BIDE = 0 in SSMR2:<br>Data input: SSO pin, Data output: SSI pin<br>When BIDE = 1 in SSMR2:<br>Data input and output: SSO pin | R/W |

[Legend]

X: Don't care.

Notes: 1. To function these pins as ports, clear the PMR bit corresponding to the pin to 0.

2. If the NMOS open-drain output is selected, use the PMC to allocate the pin from port 5. If the pin is allocated from a port other than port 5, only the CMOS output can be selected.

3. When the functions of SSI output and $\overline{SCS}$ output are allocated to P57 and P56, respectively, the output with this setting is NMOS push-pull. This differs from the CMOS output in its high-level output characteristics. On the other hand, if these functions are allocated to other pins, the output with this setting is CMOS.

- BIDE bit (bidirectional mode enable)

  Selects whether the serial data input pin and the output pin are both used or only one pin is used. For details, see section 22.3.3, Relationship between Data Input/Output Pin and Shift Register. When the SSUMS bit in SSMR2 is 0, this setting is invalid.

- SCKS bit (SSCK pin select)

  Selects whether the SSCK pin functions as a port or a serial clock pin.

- CSS[1:0] bits ($\overline{SCS}$ pin select)

  Selects whether the $\overline{SCS}$ pin functions as a port, an $\overline{SCS}$ input, or $\overline{SCS}$ output. When the SSUMS bit in SSMR2 is 0, the $\overline{SCS}$ pin functions as a port regardless of the setting of this bit.

chanhvinh.com



- SOOS bit (SSO/SSI pins open-drain output select)

  Selects whether the serial data output pin is CMOS output or NMOS open-drain output. However, when the SSI output function is allocated to P57, this bit selects between NMOS push-pull output and NMOS open-drain output. The serial data output pin is changed according to the register setting value. For details, see section 22.3.3, Relationship between Data Input/Output Pin and Shift Register.

- CSOS bit ($\overline{\text{SCS}}$ pin open-drain output select)

  Selects whether the serial data output pin is CMOS output or NMOS open-drain output. However, when the $\overline{\text{SCS}}$ output function is allocated to P56, this bit selects between NMOS push-pull output and NMOS open-drain output.

- SSUMS bit (SSU mode select)

  Selects which combination of the serial data input pin and serial data output pin is used.

  For details, see section 22.3.3, Relationship between Data Input/Output Pin and Shift Register.

chanhvinh.com



### 22.2.6 SS Enable Register (SSER)

Address: H'FF05CB

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TIE | TEIE | RIE | TE | RE | — | — | CEIE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | TIE | Transmit interrupt enable | 0: A TXI interrupt request is disabled.<br>1: A TXI interrupt request is enabled. | R/W |
| 6 | TEIE | Transmit end interrupt enable | 0: A TEI interrupt request is disabled.<br>1: A TEI interrupt request is enabled. | R/W |
| 5 | RIE | Receive interrupt enable | 0: An RXI and an OEI interrupt requests are disabled.<br>1: An RXI and an OEI interrupt requests are enabled. | R/W |
| 4 | TE* | Transmit enable | 0: Transmit operation is disabled.<br>1: Transmit operation is enabled. | R/W |
| 3 | RE* | Receive enable | 0: Receive operation is disabled.<br>1: Receive operation is enabled. | R/W |
| 2, 1 | — | Reserved | These bits are read as 0. The write value should be 0. | — |
| 0 | CEIE | Conflict error interrupt enable | 0: A CEI interrupt request is disabled.<br>1: A CEI interrupt request is enabled. | R/W |

Note: * The TE and RE bits are reset in standby mode.

chanhvinh.com



### 22.2.7 SS Status Register (SSSR)

Address: H'FF05CC

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TDRE | TEND | RDRF | — | — | ORER | — | CE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | TDRE | Transmit data empty flag | [Setting conditions]<br>• When the TE bit in SSER is 0<br>• When data transfer is performed from SSTDR to SSTRSR and data can be written in SSTDR<br>[Clearing conditions]<br>• When 0 is written to this bit after reading 1<br>• When data is written in SSTDR<br>• When the DTC transfers data to SSTDR by a TXI interrupt request, and the DTC settings satisfy the flag clearing conditions.* | R/W |
| 6 | TEND | Transmit end flag | [Setting condition]<br>• When the last bit of data is transmitted, the TDRE bit is 1<br>[Clearing conditions]<br>• When 0 is written to this bit after reading 1<br>• When data is written in SSTDR | R/W |
| 5 | RDRF | Receive data register full flag | [Setting condition]<br>• When serial reception is completed normally and receive data is transferred from SSTRSR to SSRDR<br>[Clearing conditions]<br>• When 0 is written to this bit after reading 1<br>• When data is read from SSRDR<br>• When the DTC transfers data to SSRDR by an RXI interrupt request, and the DTC settings satisfy the flag clearing conditions.* | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 4, 3 | — | Reserved | These bits are read as 0. The write value should be 0. | — |
| 2 | ORER | Overrun error flag | [Setting condition]<br>• When the next serial reception is completed while RDRF = 1<br>[Clearing condition]<br>• When 0 is written to this bit after reading 1 | R/W |
| 1 | — | Reserved | These bits are read as 0. The write value should be 0. | — |
| 0 | CE | Conflict error flag | [Setting conditions]<br>• When serial communication is started while SSUMS = 1 in SSMR2 and MSS = 1 in SSCRH, the $\overline{SCS}$ pin input is low<br>• When the $\overline{SCS}$ pin level changes from low to high during transfer while SSUMS = 1 in SSMR2 and MSS = 0 in SSCRH<br>[Clearing condition]<br>• When 0 is written to this bit after reading 1 | R/W |

Notes: In standby mode, SSSR is reset.

* The DTC clears the peripheral module flags when all of the following three conditions are satisfied.

1. When the DISEL bit is 0.
2. When the transfer counter (DTC transfer count register A (CRA) in normal mode and repeat mode, or DTC transfer count register B (CRB) in block mode) is not 0.
3. When chain transfer is not used.

- ORER bit (overrun error flag)

  Indicates that the RDRF bit is abnormally terminated in reception because an overrun error has occurred. SSRDR retains received data before the overrun error occurs and the received data after the overrun error occurs is lost. When this bit is set to 1, subsequent serial reception cannot be continued. When the MSS bit in SSCRH is 1, this is also applied to serial transmission.

chanhvinh.com



### 22.2.8 SS Receive Data Register (SSRDR)

Address: H'FF05CF

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

SSRDR is an 8-bit register that stores received serial data. When the SSU has received one byte of serial data, it transfers the received serial data from SSTRSR to SSRDR to end receive operation. After this, SSTRSR is receive-enabled. As SSTRSR and SSRDR function as a double buffer in this way, continuous receive operations are possible. SSRDR is a read-only register and cannot be written to by the CPU. SSRDR is initialized to H'FF. In standby mode, SSRDR is initialized.

### 22.2.9 SS Transmit Data Register (SSTDR)

Address: H'FF05CE

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

SSTDR is an 8-bit register that stores serial data for transmission. SSTDR can be read or written to by the CPU at all times. When the SSU detects that SSTRSR is empty, it transfers the transmit data written in SSTDR to SSTRSR and starts serial transmission. If the next transmit data has already been written to SSTDR during serial transmission, continuous serial transmission is possible. If the MLS bit in SSMR is set to 1 and when the data is written to SSTDR, the MSB/LSB inverted data is read. SSTDR is initialized to H'FF. In standby mode, SSTDR is initialized.

### 22.2.10 SS Shift Register (SSTRSR)

Address: —

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | — | — | — | — | — | — | — | — |

SSTRSR is a shift register that transmits and receives serial data. SSTRSR is not directly readable or writable from the CPU. In standby mode, SSTRSR is initialized.

chanhvinh.com



## 22.3 Operation

### 22.3.1 Transfer Clock

Transfer clock can be selected from seven internal clocks and an external clock. When this module is used, the SSCK pin must be selected as a serial clock by setting the SCKS bit in SSMR2 to 1. When the MSS bit in SSCRH is 1, an internal clock is selected and the SSCK pin is in the output state. If transfer is started, the SSCK pin outputs clocks of the transfer rate set in the CKS2 to CKS0 bits in SSCRH. When the MSS bit is 0, an external clock is selected and the SSCK pin is in the input state.

### 22.3.2 Relationship between Clock Polarity and Phase, and Data

Relationship between clock polarity and phase, and transfer data changes according to a combination of the SSUMS bit in SSMR2 and the CPOS and CPHS bits in SSMR. Figure 22.2 shows the relationship.

MSB-first transfer or LSB first transfer can be selected by the setting of the MLS bit in SSMR. When the MLS bit is 1, transfer is started from LSB to MSB. When the MLS bit is 0, transfer is started from MSB to LSB.

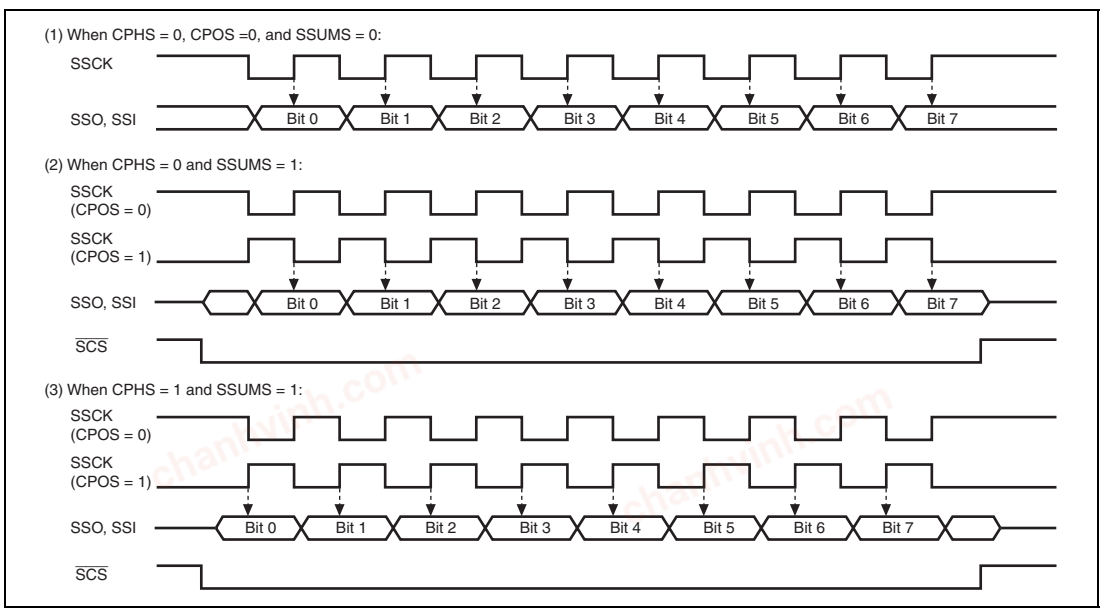



**Figure 22.2 Relationship between Clock Polarity and Phase, and Data**

chanhvinh.com



### 22.3.3 Relationship between Data Input/Output Pin and Shift Register

Relationship of connection between the data input/output pin and SSTRSR changes according to a combination of the MSS bit in SSCRH and the SSUMS bit in SSMR2. It also changes by the BIDE bit in SSMR2. Figure 22.3 shows the relationship.

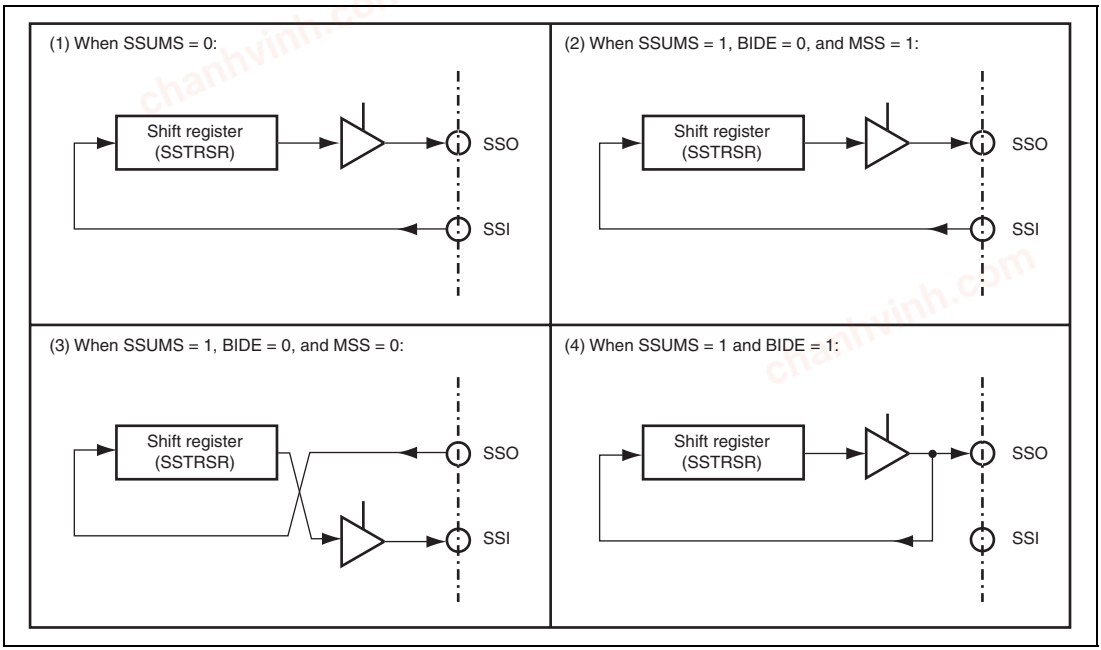



**Figure 22.3 Relationship between Data Input/Output Pin and Shift Register**

chanhvinh.com



### 22.3.4 Communication Modes and Pin Functions

The SSU switches functions of the input/output pin in each communication mode according to the settings of the MSS bit in SSCRH and the RE and TE bits in SSER. Table 22.2 shows the relationship between communication modes and the input/output pins. In bidirectional communication mode, the TE and RE bits should not be set to 1 at the same time.

**Table 22.2 Relationship between Communication Modes and Input/Output Pins**

| Communication Mode | Register State | | | | | Pin State | | |
|---|---|---|---|---|---|---|---|---|
| | SSUMS | BIDE | MSS | TE | RE | SSI | SSO | SSCK |
| Clocked Synchronous Communication Mode | 0 | * | 0 | 0 | 1 | In | — | In |
| | | | | 1 | 0 | — | Out | In |
| | | | | | 1 | In | Out | In |
| | | | 1 | 0 | 1 | In | — | Out |
| | | | | 1 | 0 | — | Out | Out |
| | | | | | 1 | In | Out | Out |
| Four-Line Bus Communication Mode | 1 | 0 | 0 | 0 | 1 | — | In | In |
| | | | | 1 | 0 | Out | — | In |
| | | | | | 1 | Out | In | In |
| | | | 1 | 0 | 1 | In | — | Out |
| | | | | 1 | 0 | — | Out | Out |
| | | | | | 1 | In | Out | Out |
| Four-Line Bus (Bidirectional) Communication Mode | 1 | 1 | 0 | 0 | 1 | — | In | In |
| | | | | 1 | 0 | — | Out | In |
| | | | 1 | 0 | 1 | — | In | Out |
| | | | | 1 | 0 | — | Out | Out |

[Legend]

—: Can be used as a general I/O port.

chanhvinh.com



### 22.3.5 Operation in Clocked Synchronous Communication Mode

#### (1) Initialization in Clocked Synchronous Communication Mode

Figure 22.4 shows the initialization in clocked synchronous communication mode. Before transmitting and receiving data, the TE and RE bits in SSER should be cleared to 0, then the SSU should be initialized.

Note: When the operating mode, or transfer format, is changed for example, the TE and RE bits must be cleared to 0 before making the change using the following procedure. When the TE bit is cleared to 0, the TDRE flag is set to 1. Note that clearing the RE bit to 0 does not change the contents of the RDRF and ORER flags, or the contents of SSRDR.

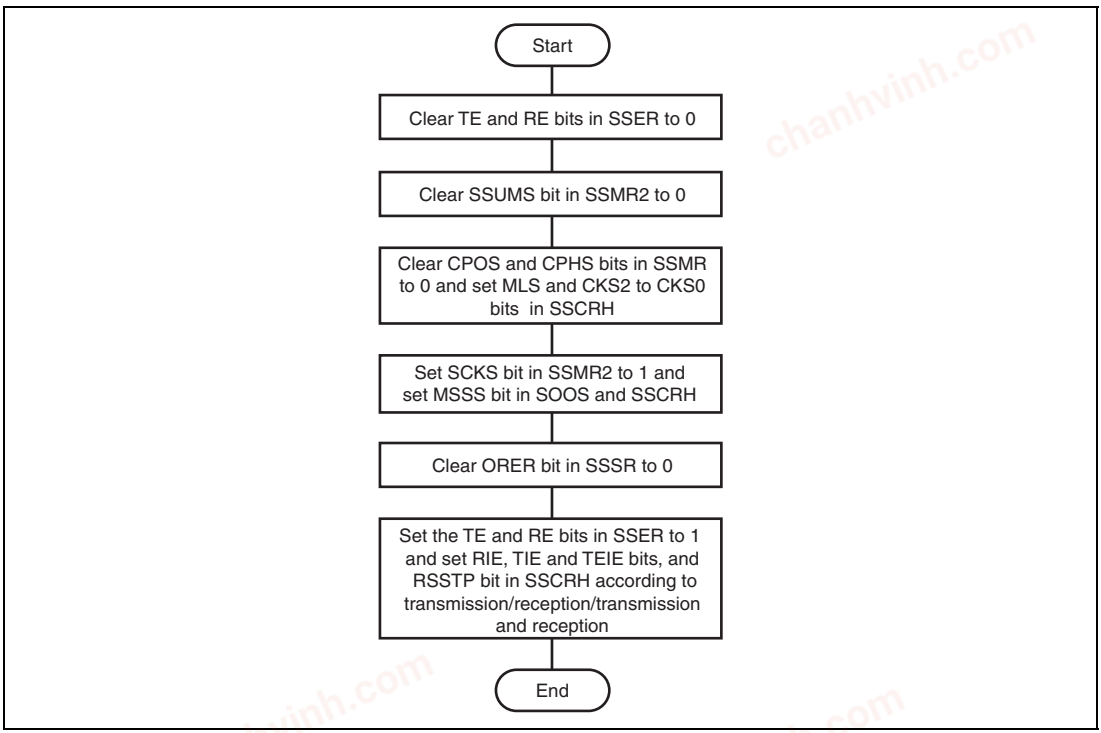



**Figure 22.4 Initialization in Clocked Synchronous Communication Mode**

chanhvinh.com



### (2) Serial Data Transmission

Figure 22.5 shows an example of the SSU operation for transmission. In serial transmission, the SSU operates as described below.

When the SSU is set as a master device, it outputs a synchronous clock and data. When the SSU is set as a slave device, it outputs data in synchronized with the input clock.

When the SSU writes transmit data in SSTDR after setting the TE bit to 1, the TDRE flag is automatically cleared to 0 and data is transferred from SSTDR to SSTRSR. Then the SSU sets the TDRE flag to 1 and starts transmission. If the TIE bit in SSER is set to 1 at this time, a TXI is generated.

When the TDRE flag is 0 and one frame of data has transferred, data is transferred from SSTDR to SSTRSR and serial transmission of the next frame is started. If the eighth bit is transmitted while the TDRE flag is 1, the TEND bit in SSSR is set to 1 and the state is retained. If the TEIE bit in SSER is set to 1 at this time, a TEI is generated. After transmission is ended, the SSCK pin is fixed high.

While the ORER bit in SSSR is set to 1, transmission cannot be performed. Therefore confirm that the ORER bit is cleared to 0 before transmission.

Figure 22.6 shows a sample flowchart for serial data transmission.

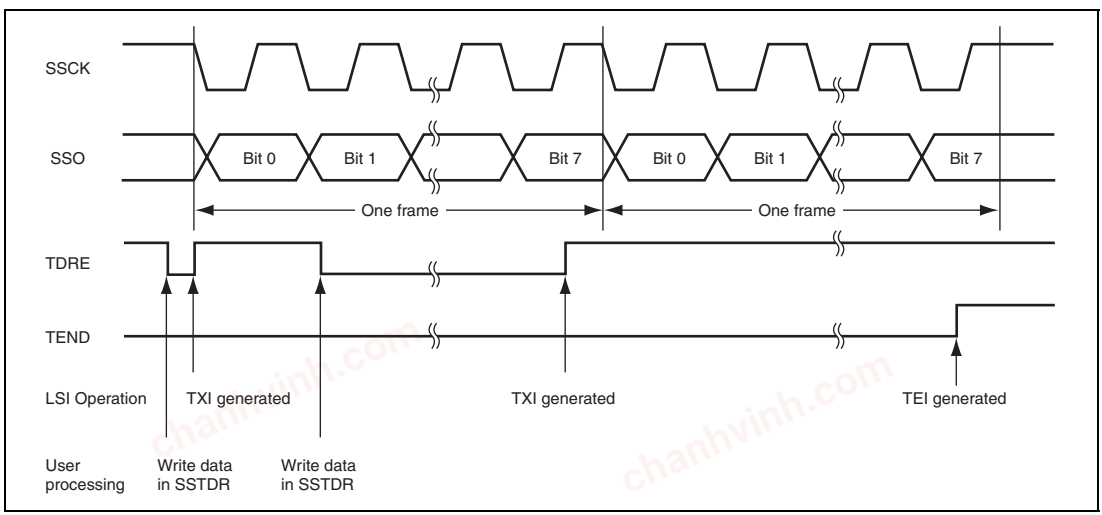



**Figure 22.5 Example of Operation in Data Transmission**

chanhvinh.com



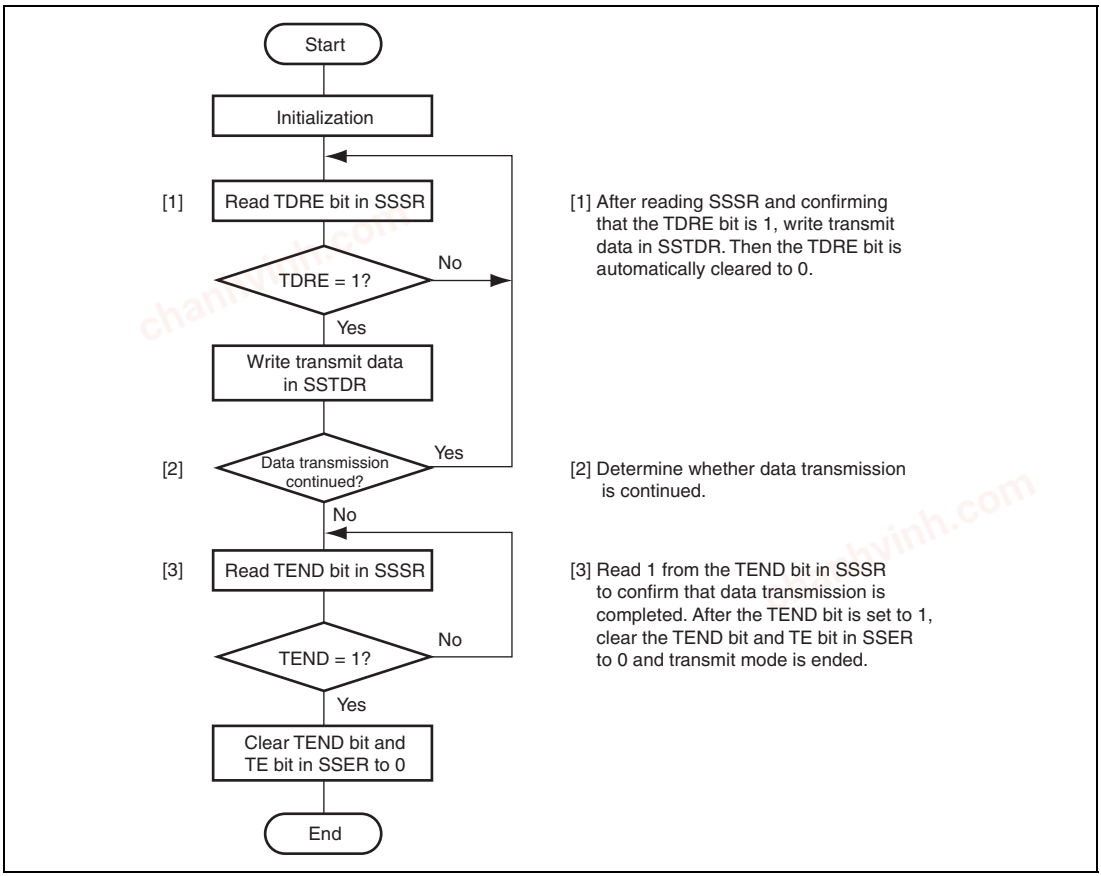



**Figure 22.6 Sample Serial Transmission Flowchart**

chanhvinh.com



### (3) Serial Data Reception

Figure 22.7 shows an example of the SSU operation for reception. In serial reception, the SSU operates as described below.

When the SSU is set as a master device, it outputs a synchronous clock and inputs data. When the SSU is set as a slave device, it inputs data in synchronized with the input clock. When the SSU is set as a master device, it outputs a receive clock and starts reception by performing dummy read on SSRDR.

After eight bits of data is received, the RDRF bit in SSSR is set to 1 and received data is stored in SSRDR. If the RIE bit in SSER is set to 1 at this time, a RXI is generated. If SSRDR is read, the RDRF bit is automatically cleared to 0.

When the SSU is set as a master device and reception is ended, received data is read after setting the RSSTP bit in SSCRH to 1. Then the SSU outputs eight bits of clocks and operation is stopped. After that, the RE and RSSTP bits are cleared to 0 and the last received data is read. Note that if SSRDR is read while the RE bit is set to 1, received clock is output again.

When the eighth clock rises while the RDRF bit is 1, the ORER bit in SSSR is set. Then an overrun error (OEI) is generated and operation is stopped. When the ORER bit in SSSR is set to 1, reception cannot be performed. Therefore confirm that the ORER bit is cleared to 0 before reception.

Figure 22.8 shows a sample flowchart for serial data reception.

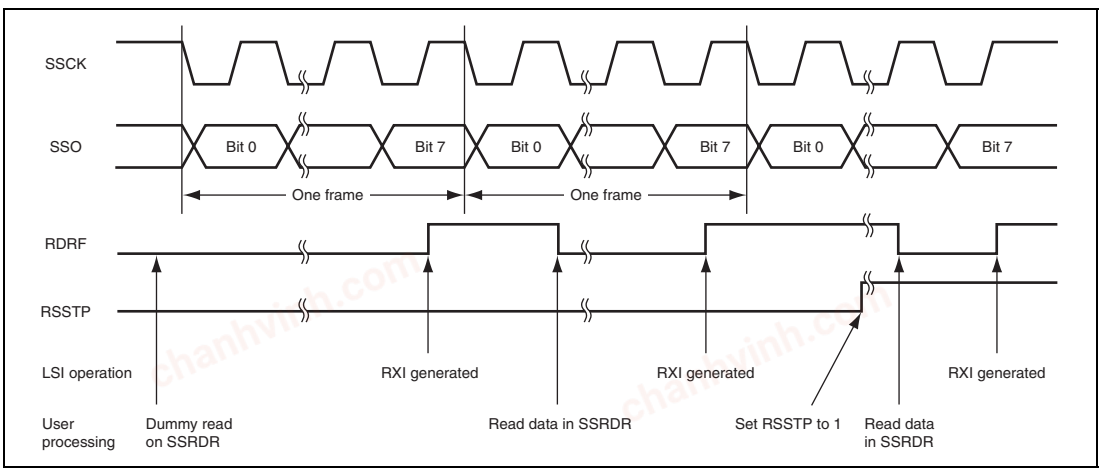



**Figure 22.7 Example of Operation in Data Reception (MSS = 1)**

chanhvinh.com



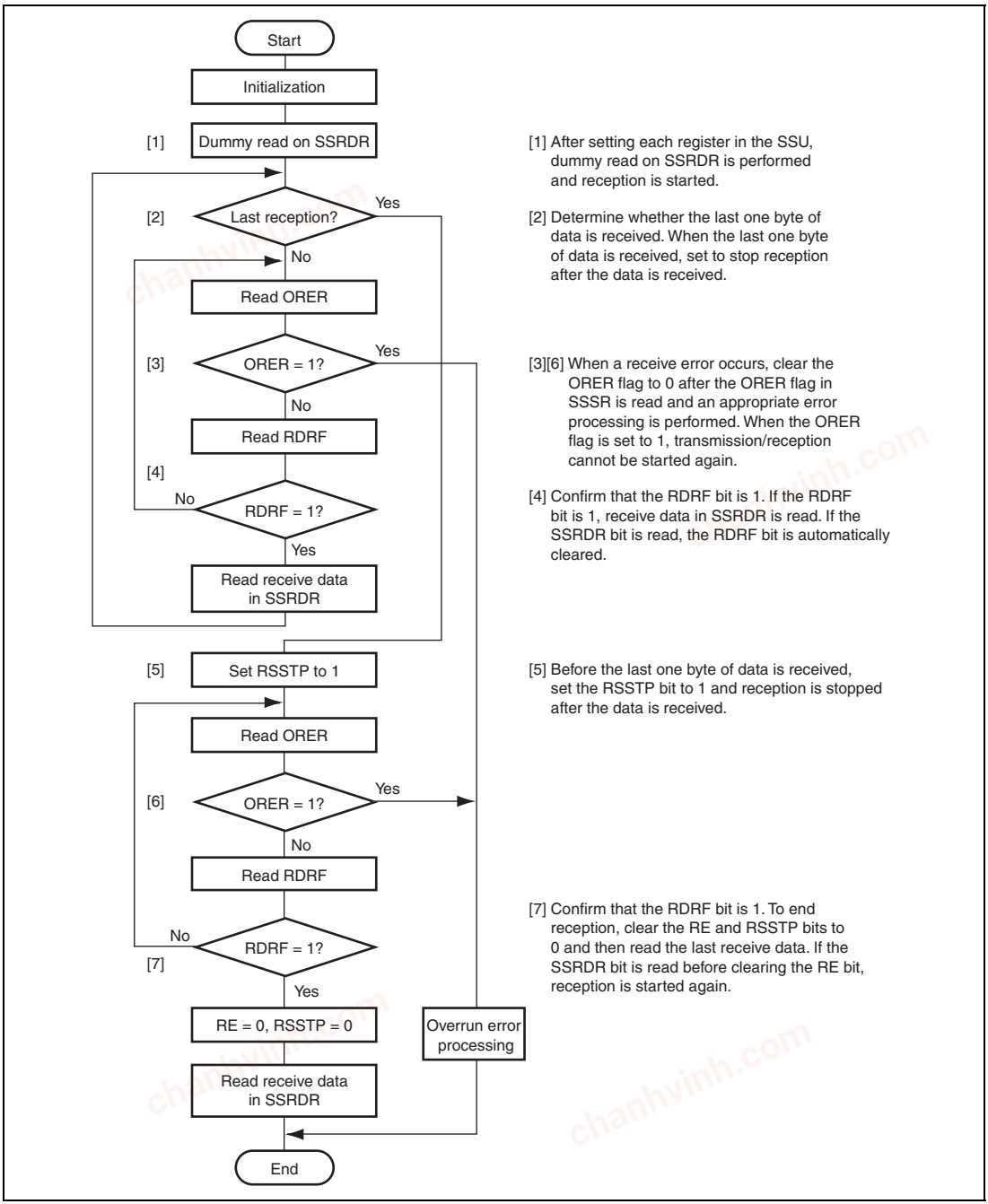



**Figure 22.8 Sample Serial Reception Flowchart (MSS = 1)**

chanhvinh.com



### (4) Serial Data Transmission and Reception

Data transmission and reception is a combined operation of data transmission and reception which are described before. Transmission and reception is started by writing data in SSTDR. When the eighth clock rises while the TDRE bit is set to 1 or the ORER bit is set to 1, transmission and reception is stopped.

To switch from transmit mode (TE = 1) or receive mode (RE = 1) to transmit and receive mode (TE = RE = 1), the TE and RE bits should be cleared to 0. After confirming that the TEND, RDRF, and ORER bits are cleared to 0, set the TE and RE bits to 1.

When the module is released from transmit and receive mode (TE = 1 and RE = 1), setting TE = 0 (and RE = 1) after the SSRDR has been read can cause output of the clock signal. For this reason, start by setting RE = 0 and only set TE = 0 after that (or set both RE = 0 and TE = 0 at the same time). When TE = 0 and RE = 1 is subsequently set, only set RE = 1 after changing SRES from 1 to 0.

chanhvinh.com



Figure 22.9 shows a sample flowchart for serial transmit and receive operations.

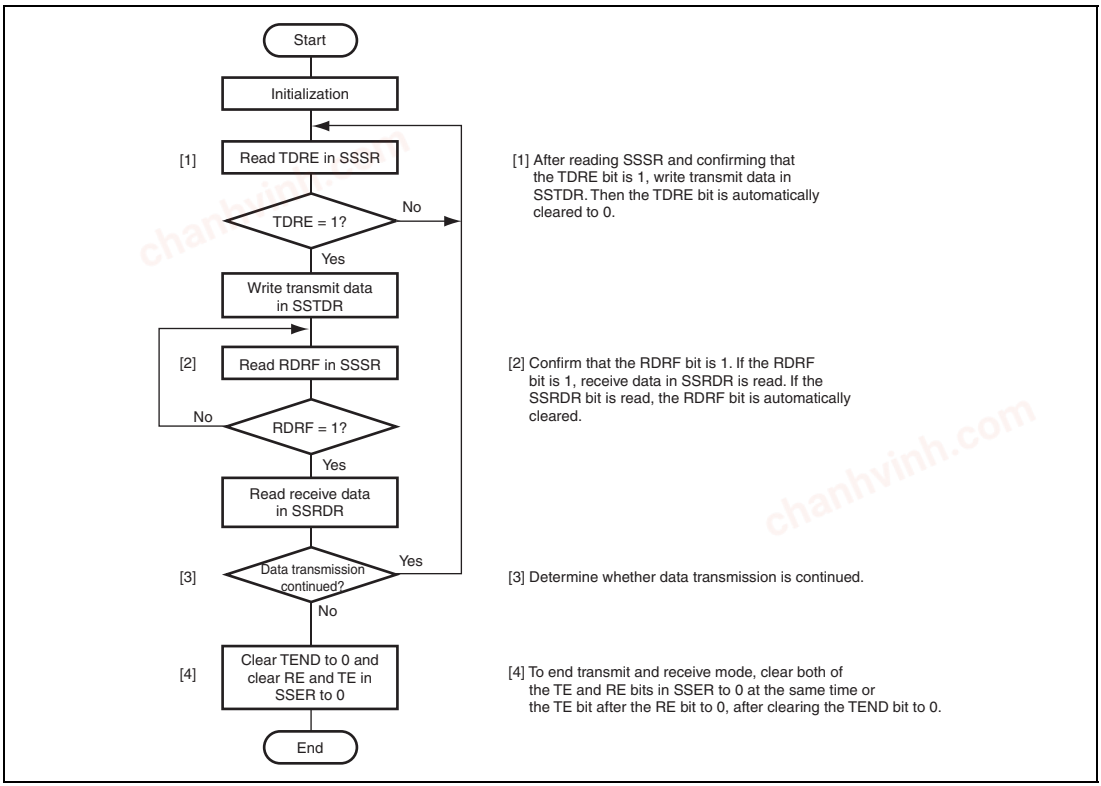



**Figure 22.9 Sample Flowchart for Serial Transmit and Receive Operations**

chanhvinh.com



### 22.3.6 Operation in Four-Line Bus Communication Mode

Four-line bus communication mode is a mode which communicates with the four-line bus; a clock line, a data input line, a data output line, and a chip select line. This mode includes bidirectional mode in which the data input line and the data output line function as a single pin. The data input line and the data output line are changed according to the settings of the MSS bit in SSCRH and BIDE bit in SSMR2. For details, see section 22.3.3, Relationship between Data Input/Output Pin and Shift Register. In this mode, relationship between clock polarity and phase, and data can be set by the CPOS and CPHS bits in SSMR. For details, see section 22.3.2, Relationship between Clock Polarity and Phase, and Data.

When the SSU is set as a master device, the chip select line controls output. When the SSU is set as a slave device, the chip select line controls input. When the SSU is set as a master device, the chip select line controls output of the $\overline{\text{SCS}}$ pin or controls output of a general port by setting the CSS1 bit in SSMR2 to 1. When the SSU is set as a slave device, the chip select line sets the $\overline{\text{SCS}}$ pin as an input pin by setting the CSS1 and CSS0 bits in SSMR2 to 01.

In four-line bus communication mode, transfer is performed in MSB-first order while the MLS bit in SSMR is 0.

#### (1) Initialization in Four-Line Bus Communication Mode

Figure 22.10 shows the initialization in four-line bus communication mode. Before transmitting and receiving data, the TE and RE bits in SSER should be cleared to 0, then the SSU should be initialized.

Note: When the operating mode, or transfer format, is changed for example, the TE and RE bits must be cleared to 0 before making the change using the following procedure. Note that clearing the RE bit to 0 does not change the contents of the RDRF and ORER flags, or the contents of SSRDR.

chanhvinh.com



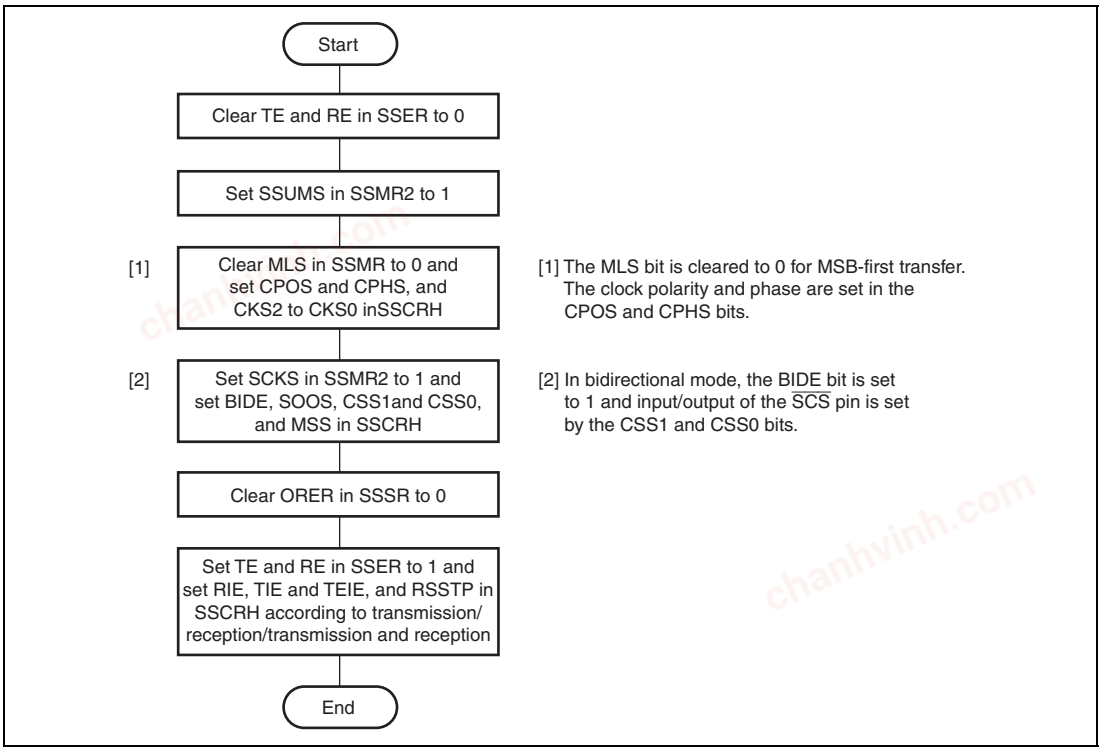



**Figure 22.10 Initialization in Four-Line Bus Communication Mode**

chanhvinh.com



### (2) Serial Data Transmission

Figure 22.11 shows an example of the SSU operation for transmission. In serial transmission, the SSU operates as described below.

When the SSU is set as a master device, it outputs a synchronous clock and data. When the SSU is set as a slave device, the $\overline{\text{SCS}}$ pin is in the low-input state and the SSU outputs data in synchronized with the input clock.

When the SSU writes transmit data in SSTDR after setting the TE bit to 1, the TDRE flag is automatically cleared to 0 and data is transferred from SSTDR to SSTRSR. Then the SSU sets the TDRE flag to 1 and starts transmission. If the TIE bit in SSER is set to 1 at this time, a TXI is generated.

When the TDRE flag is 0 and one frame of data has transferred, data is transferred from SSTDR to SSTRSR and serial transmission of the next frame is started. If the eighth bit is transmitted while the TDRE flag is 1, the TEND bit in SSSR is set to 1 and the state is retained. If the TEIE bit in SSER is set to 1 at this time, a TEI is generated. After transmission is ended, the SSCK pin is fixed high and the $\overline{\text{SCS}}$ pin goes high. When continuous transmission is performed with the $\overline{\text{SCS}}$ pin low, the next data should be written to SSTDR before transmitting the eighth bit of the frame.

While the ORER bit in SSSR is set to 1, transmission cannot be performed. Therefore confirm that the ORER bit is cleared to 0 before transmission.

The difference between this mode and clocked synchronous communication mode is as follows: when the SSU is set as a master device, the SSO pin is in the high impedance state if the $\overline{\text{SCS}}$ pin is in the high impedance state and when the SSU is set as a slave device, the SSI pin is in the high impedance state if the $\overline{\text{SCS}}$ pin is in the high-input state. The sample flowchart for serial data transmission is the same as that in clocked synchronous communication mode.

chanhvinh.com



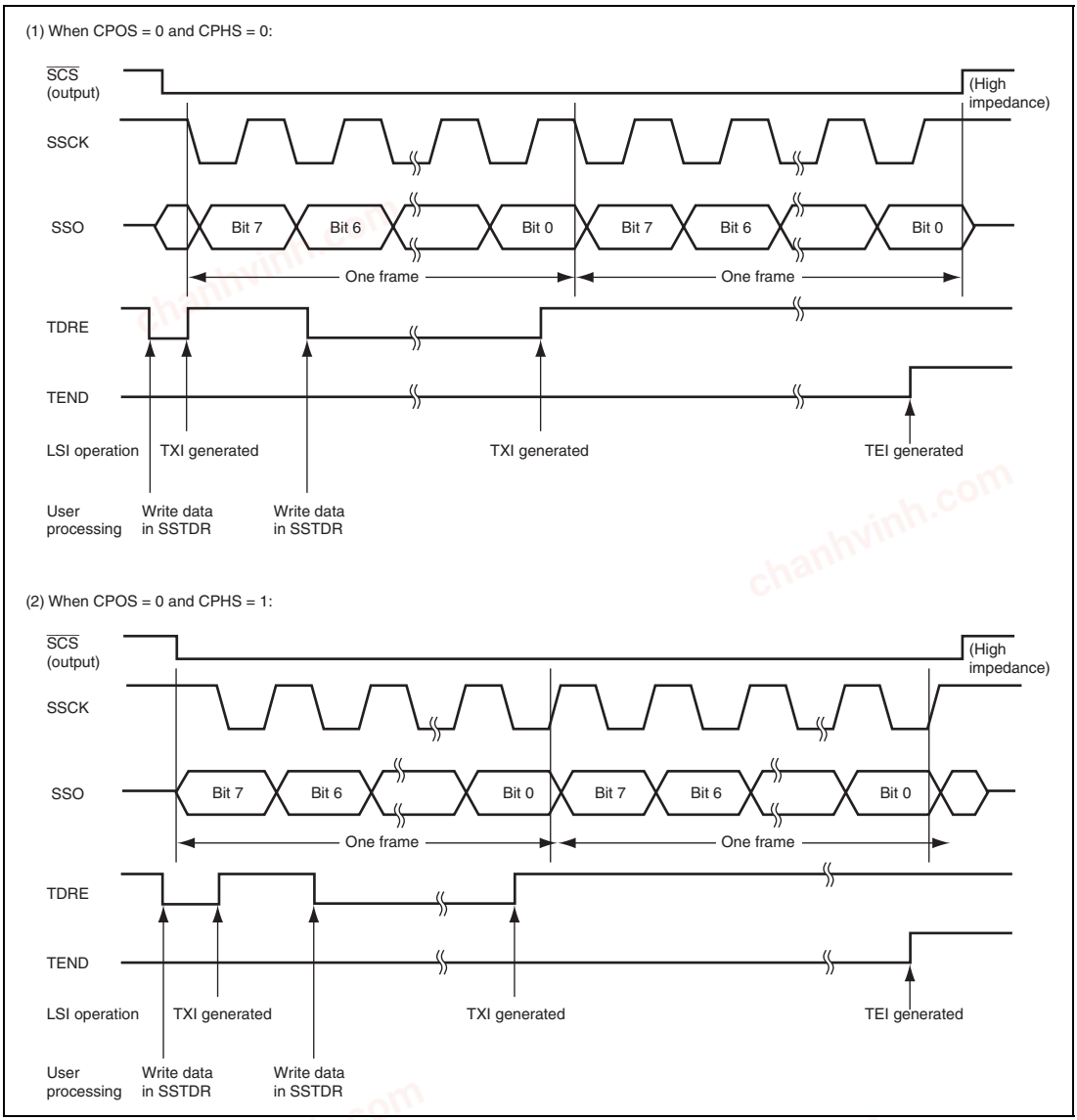



**Figure 22.11 Example of Operation in Data Transmission (MSS = 1)**

chanhvinh.com



### (3) Serial Data Reception

Figure 22.12 shows an example of the SSU operation for reception. In serial reception, the SSU operates as described below.

When the SSU is set as a master device, it outputs a synchronous clock and inputs data. When the SSU is set as a slave device, the $\overline{\text{SCS}}$ pin is in the low-input state and inputs data in synchronized with the input clock. When the SSU is set as a master device, it outputs a receive clock and starts reception by performing dummy read on SSRDR.

After eight bits of data is received, the RDRF bit in SSSR is set to 1 and received data is stored in SSRDR. If the RIE bit in SSER is set to 1 at this time, an RXI is generated. If SSRDR is read, the RDRF bit is automatically cleared to 0.

When the SSU is set as a master device and reception is ended, received data is read after setting the RSSTP bit in SSER to 1. Then the SSU outputs eight bits of clocks and operation is stopped. After that, the RE and RSSTP bits are cleared to 0 and the last received data is read. Note that if SSRDR is read while the RE bit is set to 1, received clock is output again.

When the eighth clock rises while the RDRF bit is 1, the ORER bit in SSSR is set. Then an overrun error (OEI) is generated and operation is stopped. When the ORER bit in SSSR is set to 1, reception cannot be performed. Therefore confirm that the ORER bit is cleared to 0 before reception.

The set timings of the RDRF and ORER flags differ according to the CPHS setting. These timings are shown in figure 22.2. When the CPHS bit is set to 1, the flag is set during the frame. Therefore care should be taken at the end of reception.

The sample flowchart for serial data reception is the same as that in clocked synchronous communication mode.

chanhvinh.com



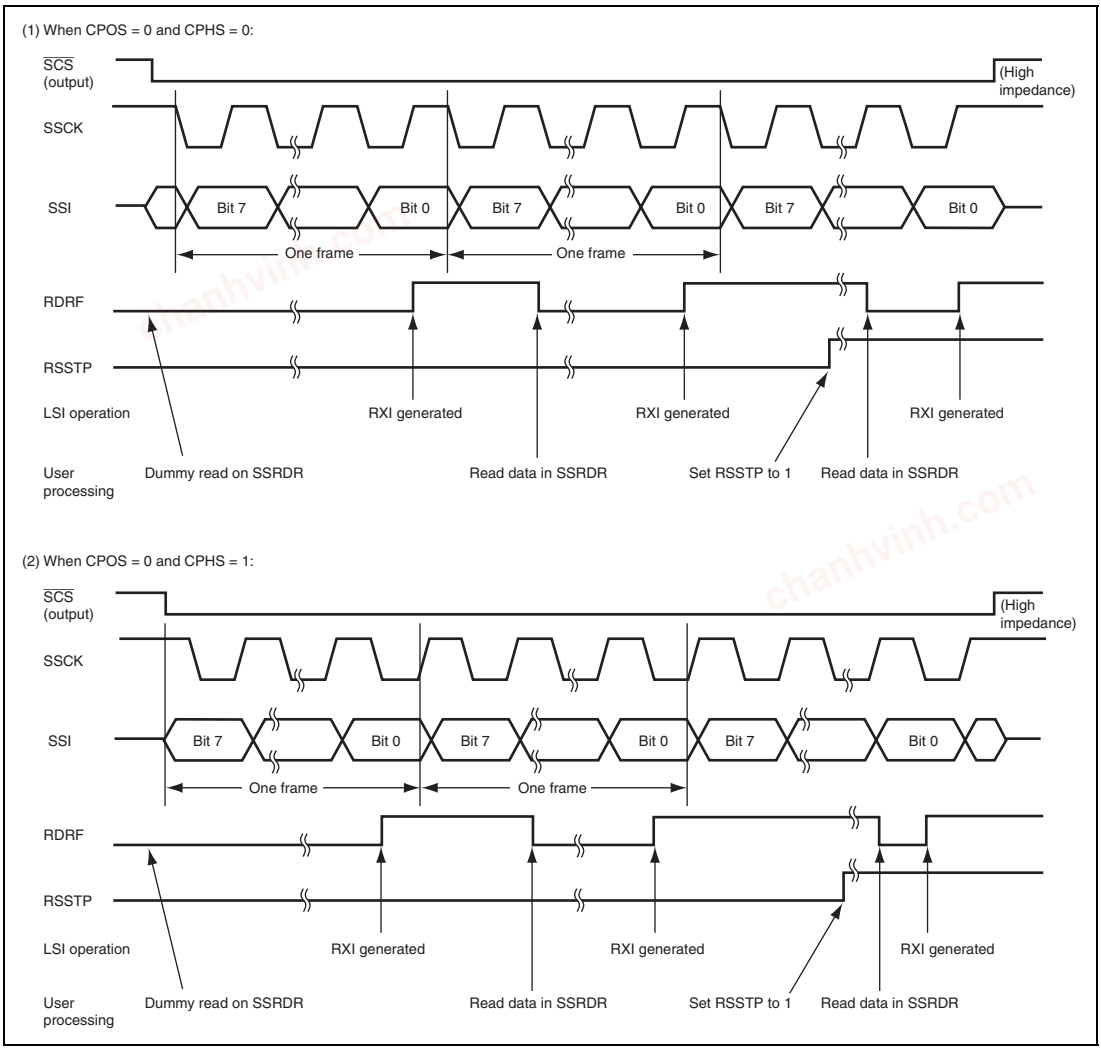



**Figure 22.12 Example of Operation in Data Reception (MSS = 1)**

chanhvinh.com



### 22.3.7 $\overline{\text{SCS}}$ Pin Control and Arbitration

When the SSUMS bit in SSMR2 is set to 1 and the CSS1 bit is set to 1, the MSS bit in SSCRH is set to 1 and then the arbitration of the $\overline{\text{SCS}}$ pin is checked before starting serial transfer. If the SSU detects that the synchronized internal SCS pin goes low in this period, the CE bit in SSSR is set and the MSS bit in SSCRH is cleared.

Note: When a conflict error is set, subsequent transmit operation is not possible. Therefore the CE bit must be cleared to 0 before starting transmission.
When the multimaster error is used, the CSOS bit in SSMR2 should be set to 1.

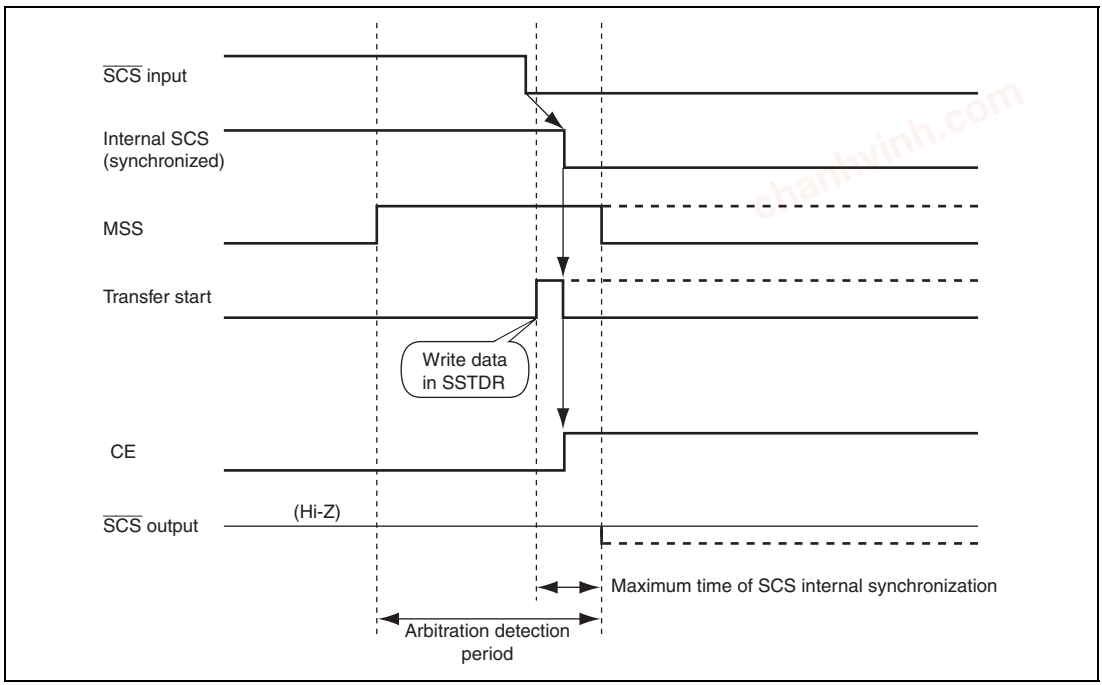



**Figure 22.13 Arbitration Check Timing**

chanhvinh.com



## 22.4 Interrupt Requests

The SSU has five interrupt requests: transmit data empty, transmit end, receive data full, overrun error, and conflict error. Since these interrupt requests are assigned to the common vector address, interrupt sources must be determined by flags. Table 22.3 lists the interrupt requests.

**Table 22.3 Interrupt Requests**

| Interrupt Request | Abbreviation | Interrupt Condition | DTC Activation |
|---|---|---|---|
| Transmit data empty | TXI | (TIE = 1), (TDRE = 1) | Possible |
| Transmit end | TEI | (TEIE = 1), (TEND = 1) | Impossible |
| Receive data full | RXI | (RIE = 1), (RDRF = 1) | Possible |
| Overrun error | OEI | (RIE = 1), (ORER = 1) | Impossible |
| Conflict error | CEI | (CEIE = 1), (CE = 1) | Impossible |

When an interrupt exception handling by an interrupt source shown in table 22.4 is executed, each interrupt source must be cleared during the exception handling. Note that the TDRE and TEND bits are automatically cleared by writing transmit data in SSTDR and the RDRF bit is automatically cleared by reading SSRDR. When transmit data is written in SSTDR, the TDRE bit is set again at the same time. Then if the TDRE bit is cleared, additional one byte of data may be transmitted. The DTC can be activated by a TXI interrupt to transfer data. The TDRE flag is automatically cleared upon data transfer by the DTC. The DTC can also be activated by an RXI interrupt to transfer data. The RDRF flag is automatically cleared upon data transfer by the DTC.

## 22.5 Usage Notes

**(1)** If the NMOS open-drain output is selected for the SSCK output pin, the SSO output pin, and the $\overline{\text{SCS}}$ output pin, use the PMC to allocate that pin from port 5. If the pins are allocated from a port other than port 5, only the CMOS output is available.

chanhvinh.com



# Section 23 Hardware LIN

The hardware LIN works in cooperation with timer RA and SCI3_1 to provide LIN communications.

## 23.1 Overview

- Master mode
  Generates Sync Break.
  Detects bus conflicts.
- Slave mode
  Detects Sync Break.
  Measures Sync Field.
  Controls Sync Break and Sync Field signal inputs to SCI3_1.
  Detects bus conflicts.

Figure 23.1 shows a block diagram of the hardware LIN interface.

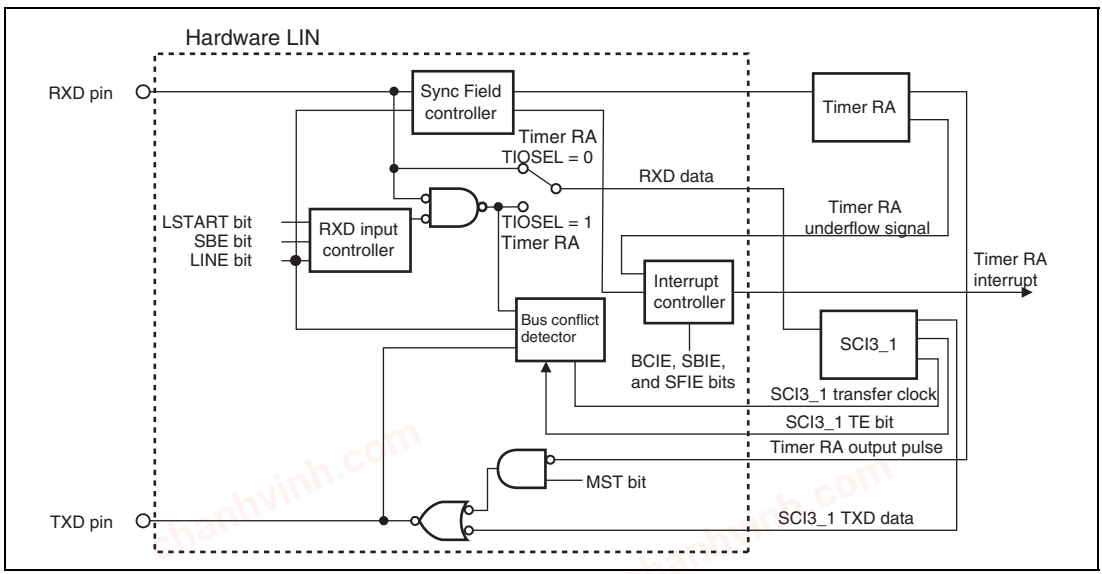



**Figure 23.1 Block Diagram of Hardware LIN**

chanhvinh.com



Table 23.1 shows the hardware LIN pins.

**Table 23.1 Pin Configuration**

| Pin Symbol | I/O | Description |
|---|---|---|
| RXD | Input | Receive-data input to the hardware LIN |
| TXD | Output | Transmit-data output from the hardware LIN |

## 23.2 Register Configuration

The hardware LIN interface has the following registers.

- LIN control register (LINCR)
- LIN status register (LINST)

### 23.2.1 LIN Control Register (LINCR)

Address: H'FF0518

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | LINE | MST | SBE | LSTART | RXDSF | BCIE | SBIE | SFIE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | LINE | LIN start | 0: Enables LIN operation.<br>1: Disables LIN operation.*[1] | R/W |
| 6 | MST | LIN operating mode setting*[2] | 0: Slave mode (Enables the Sync Break detector.)<br>1: Master mode (Takes the OR between timer RA output and TXD data.) | R/W |
| 5 | SBE | RXD input mask cancellation timing select | (Valid only in slave mode)<br>0: Cancels the mask upon Sync Break detection.<br>1: Cancels the mask upon completion of Sync Field measurement. | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 4 | LSTART | Sync Break detection start | 0: Don't care.<br>1: Enables timer RA input and disables RXD input. | R/W |
| 3 | RXDSF | RXD input status flag | 0: Indicates that RXD input has been enabled.<br>1: Indicates that RXD input has been disabled. | R |
| 2 | BCIE | Bus conflict detection interrupt enable | 0: Disables a bus conflict detection interrupt.<br>1: Enables a bus conflict detection interrupt. | R/W |
| 1 | SBIE | Sync Break detection interrupt enable | 0: Disables a Sync Break detection interrupt.<br>1: Enables a Sync Break detection interrupt. | R/W |
| 0 | SFIE | Sync Field measurement end interrupt enable | 0: Disables a Sync Field measurement end interrupt.<br>1: Enables a Sync Field measurement end interrupt. | R/W |

Notes: 1. Immediately after setting this bit to 1, inputs to timer RA and SCI3_1 are prohibited.
2. Before switching the LIN operating modes, temporarily disable the LIN (LINE = 0).
3. After setting LSTART and then checking that the RXDSF flag is 1, start inputting Sync Break.

chanhvinh.com



### 23.2.2 LIN Status Register (LINST)

Address: H'FF0519

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | B2CLR | B1CLR | B0CLR | BCDCT | SBDCT | SFDCT |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7, 6 | — | Reserved | These bits are read as 0. The write value should be 0. | — |
| 5 | B2CLR | BCDCT flag clear | The BCDCT flag is cleared when 1 is written to this bit.<br>This bit is always read as 0. | R/W |
| 4 | B1CLR | SBDCT flag clear | The SBDCT flag is cleared when 1 is written to this bit.<br>This bit is always read as 0. | R/W |
| 3 | B0CLR | SFDCT flag clear | The SFDCT flag is cleared when 1 is written to this bit.<br>This bit is always read as 0. | R/W |
| 2 | BCDCT | Bus conflict detection flag | 0: No bus conflict is detected.<br>1: Indicates that bus conflict has been detected. | R |
| 1 | SBDCT | Sync Break detection flag | 0: Sync Bread is not detected.<br>1: Indicates that Sync Break has been detected. | R |
| 0 | SFDCT | Sync Field measurement end flag | 0: Sync Field measurement is not ended.<br>1: Indicates that Sync Field measurement has been ended. | R |

chanhvinh.com



## 23.3 Operation

### 23.3.1 Master Mode

Figure 23.2 shows the example of hardware LIN interface operation for transmitting the header field in master mode. Figures 23.3 and 23.4 show the flowcharts for header field transmission.

The hardware LIN interface operates as follows for header field transmission.

1. When 1 is written to the TSTART bit in TRACR register of timer RA, the hardware LIN keeps outputting a low level from the TXD pin for the period specified by the TRAPRE and TRATR registers of timer RA.
2. When timer RA underflows, the hardware LIN inverts the TXD pin output, thus setting the SBDCT flag in the LINST register to 1. In this case, if the SBIE bit in the LINCR register is set to 1, the timer RA/HW-LIN interrupt occurs.
3. The hardware LIN interface transmits H'55 using SCI3_1.
4. After completing H'55 transmission, the hardware LIN interface transmits the ID field using SCI3_1.
5. After completing ID field transmission, the hardware LIN interface performs response field communications.

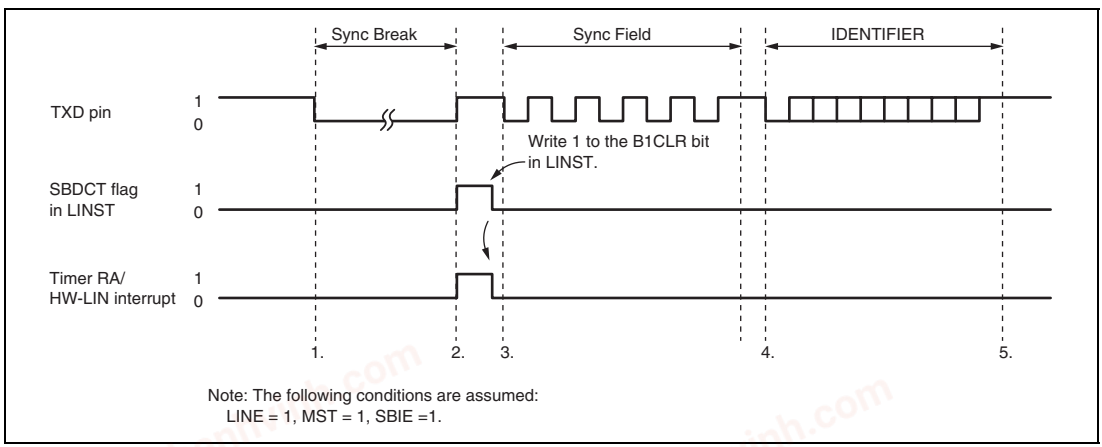
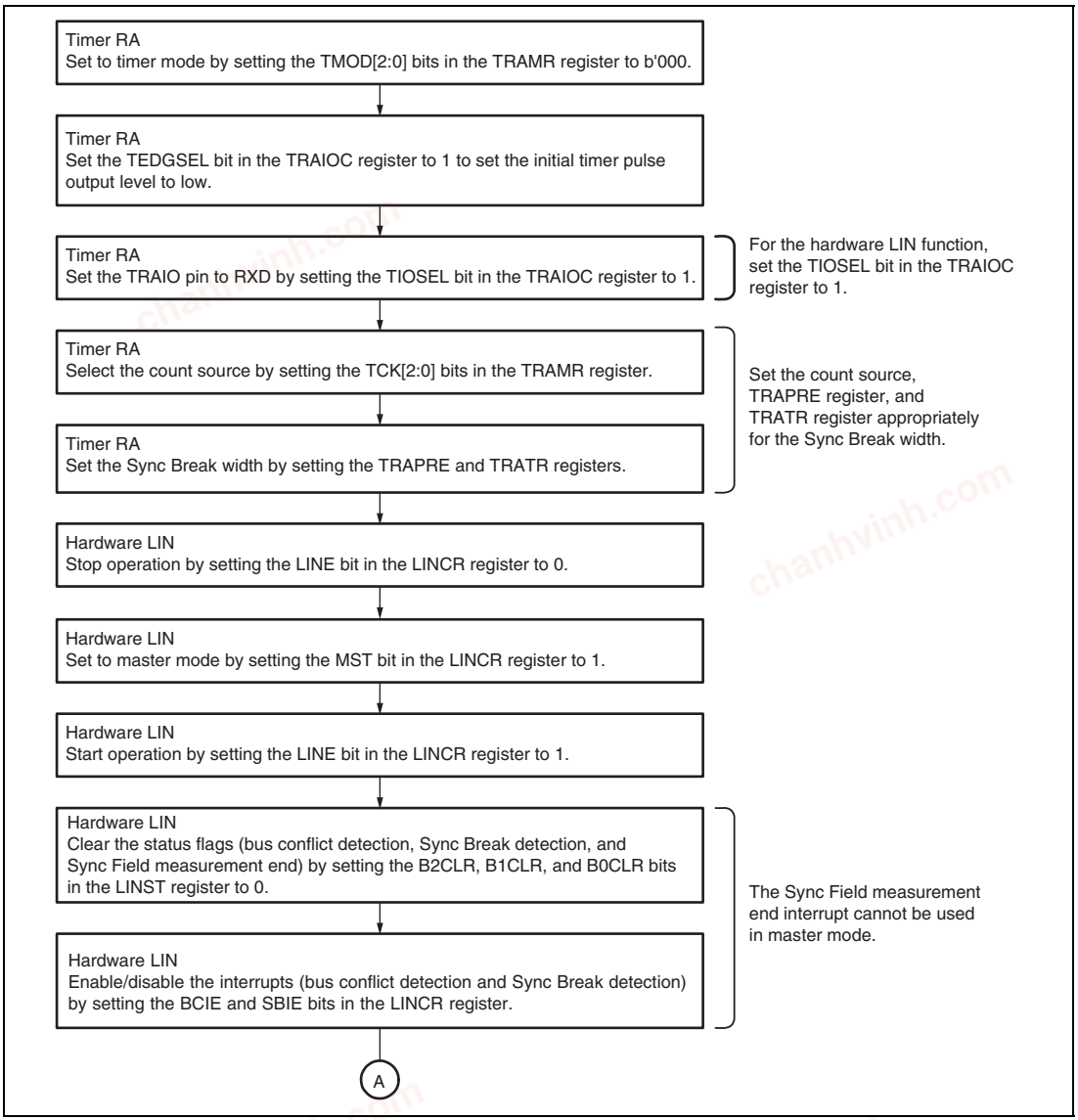



**Figure 23.2 Example of LIN Operation for Transmitting Header Field**

chanhvinh.com

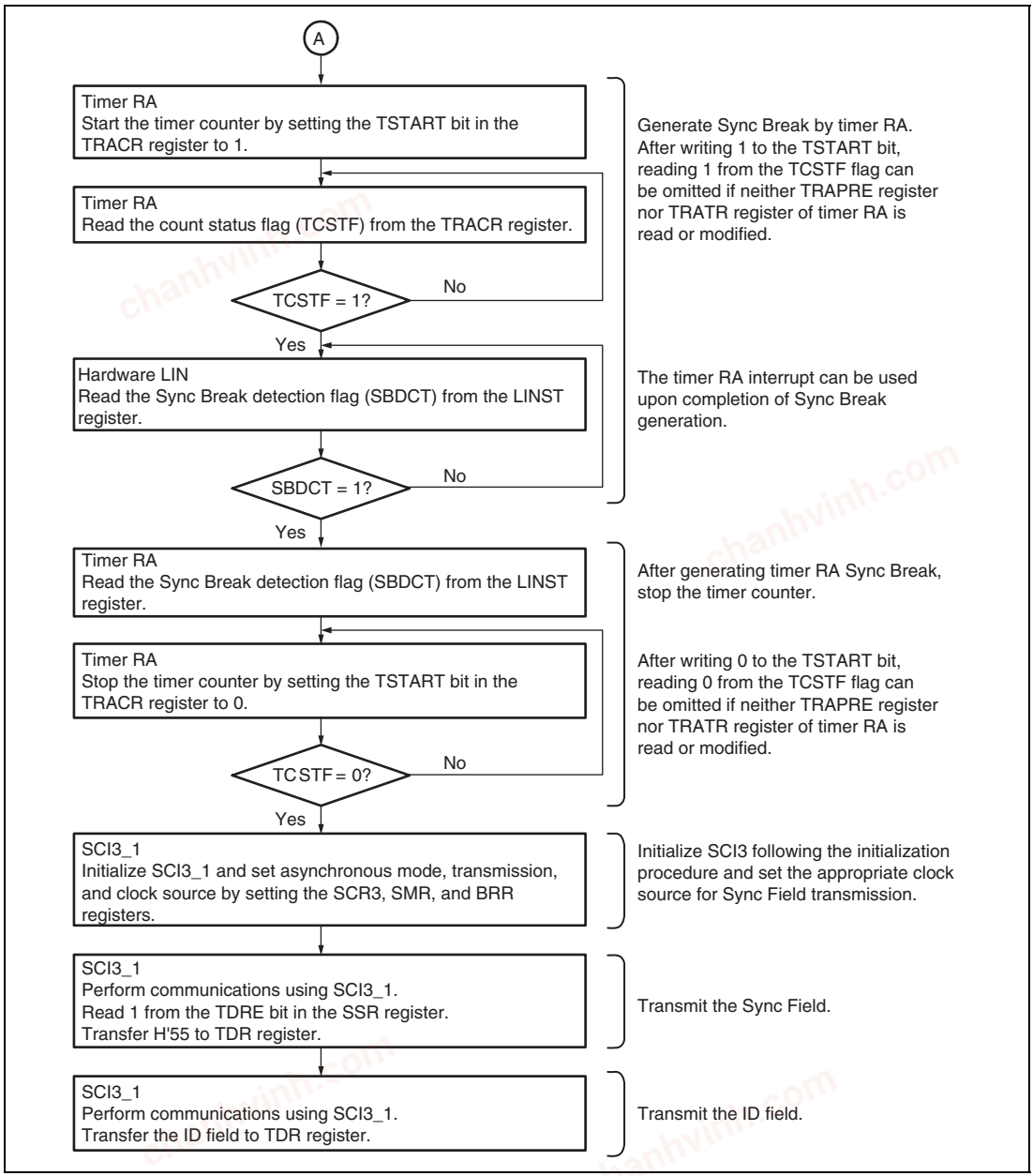





**Figures 23.3 Header Field Transmission Flowchart (1)**

chanhvinh.com






**Figures 23.4 Header Field Transmission Flowchart (2)**

chanhvinh.com



### 23.3.2 Slave Mode

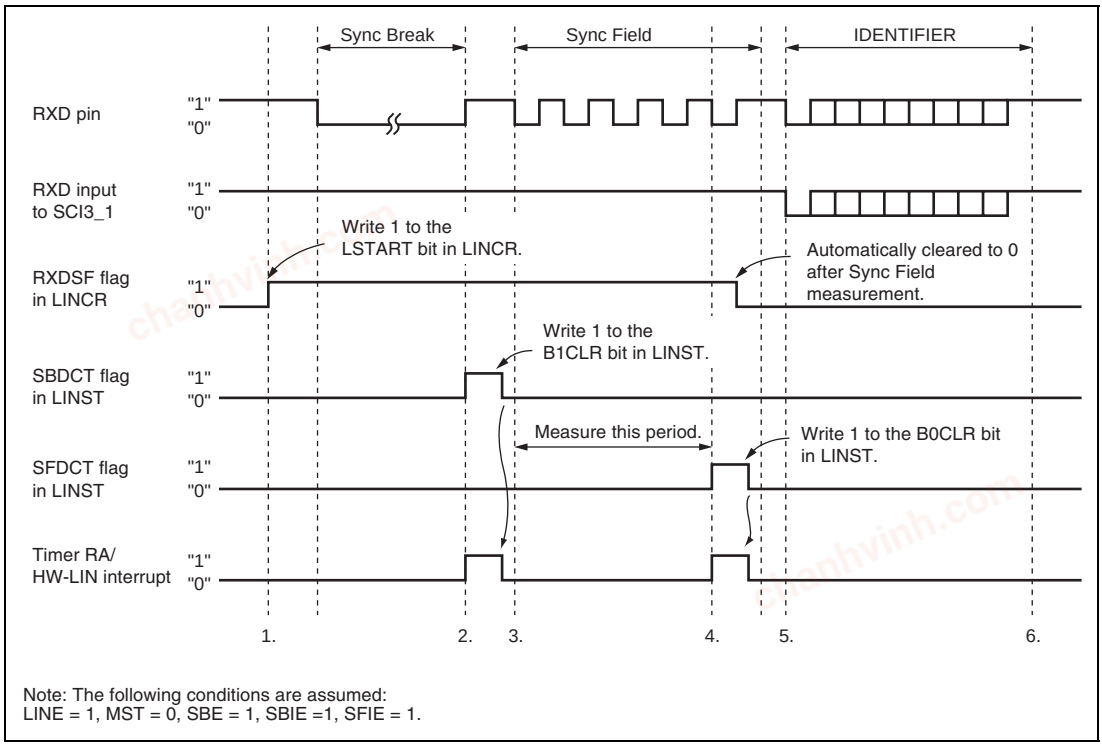
Figure 23.5 shows the example of hardware LIN interface operation for receiving the header field in slave mode. Figures 23.6 to 23.8 show the flowcharts for header field reception.

The hardware LIN interface operates as follows for header field reception.

1. When 1 is written to the LSTART bit in LINCR register of the hardware LIN interface, Sync Break detection is enabled.
2. When a low level input is longer than the time set in timer RA, it is detected as Sync Break, thus setting the SBDCT flag in the LINST register to 1. In this case, if the SBIE bit in the LINCR register is set to 1, the timer RA/HW-LIN interrupt occurs. The hardware LIN interface then measures the Sync Field.
3. The hardware LIN interface receives the Sync Field (H'55). During reception, the hardware LIN interface measures the time from the start bit through bit 6. Here, the Sync Field input to the SCI3 RXD can be either enabled or disabled depending on the SBE bit setting in the LINCR register.
4. Completion of Sync Field measurement sets the SFDCT flag in the LINST register to 1. In this case, if the SFIE bit in the LINCR register is 1, the timer RA/HW-LIN interrupt occurs.
5. After completing Sync Field measurement, the hardware LIN interface calculates the transfer rate from the timer RA count value and sets the rate in SCI3_1, and also updates the TRAPRE and TRATR registers in timer RA. Then the hardware LIN interface receives the ID field using SCI3_1.
6. After completing ID field reception, the hardware LIN interface performs response field communications.

chanhvinh.com



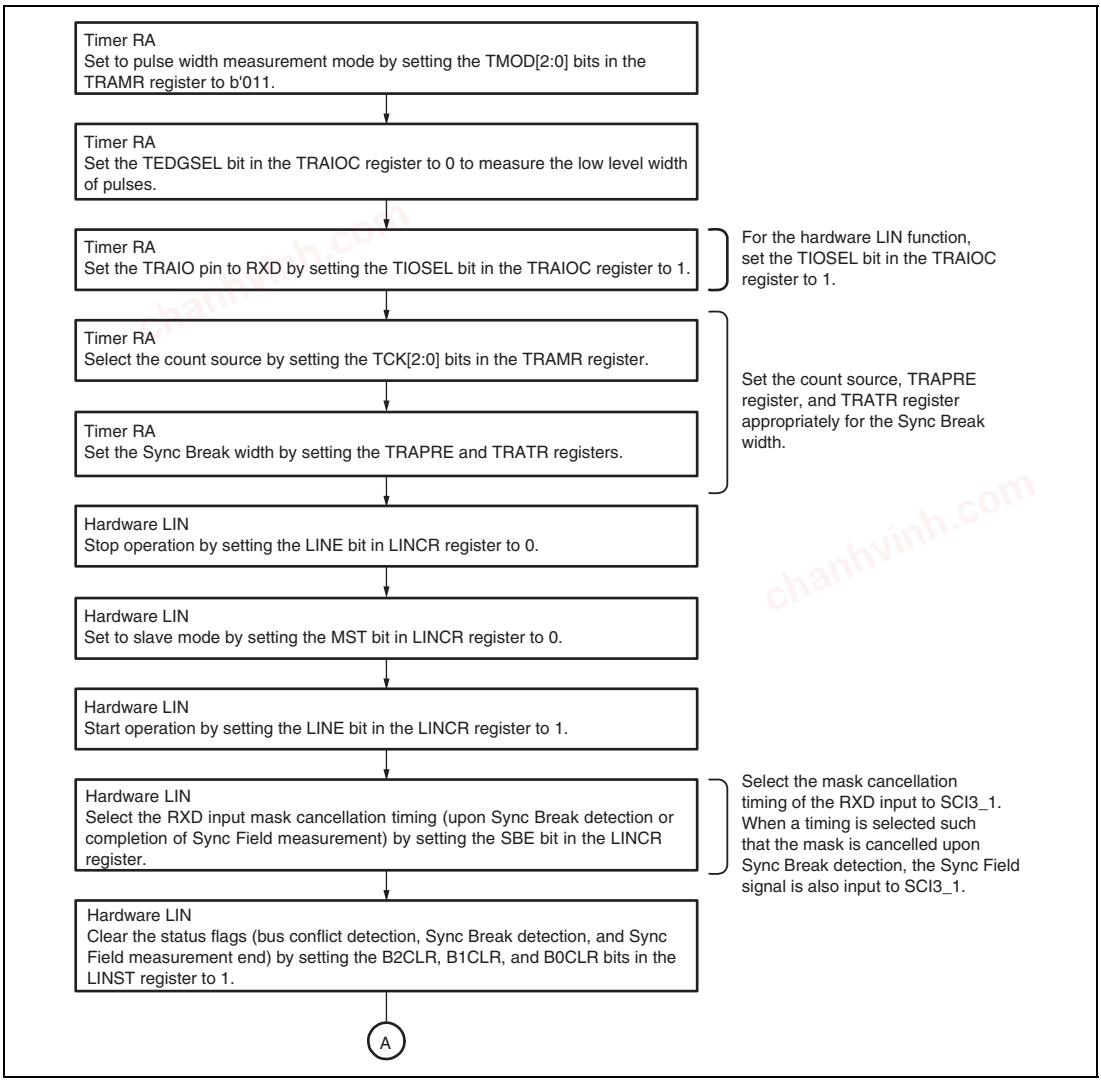



Note: The following conditions are assumed:
LINE = 1, MST = 0, SBE = 1, SBIE =1, SFIE = 1.

**Figure 23.5 Example of LIN Operation for Receiving Header Field**

chanhvinh.com






**Figures 23.6 Header Field Reception Flowchart (1)**

chanhvinh.com



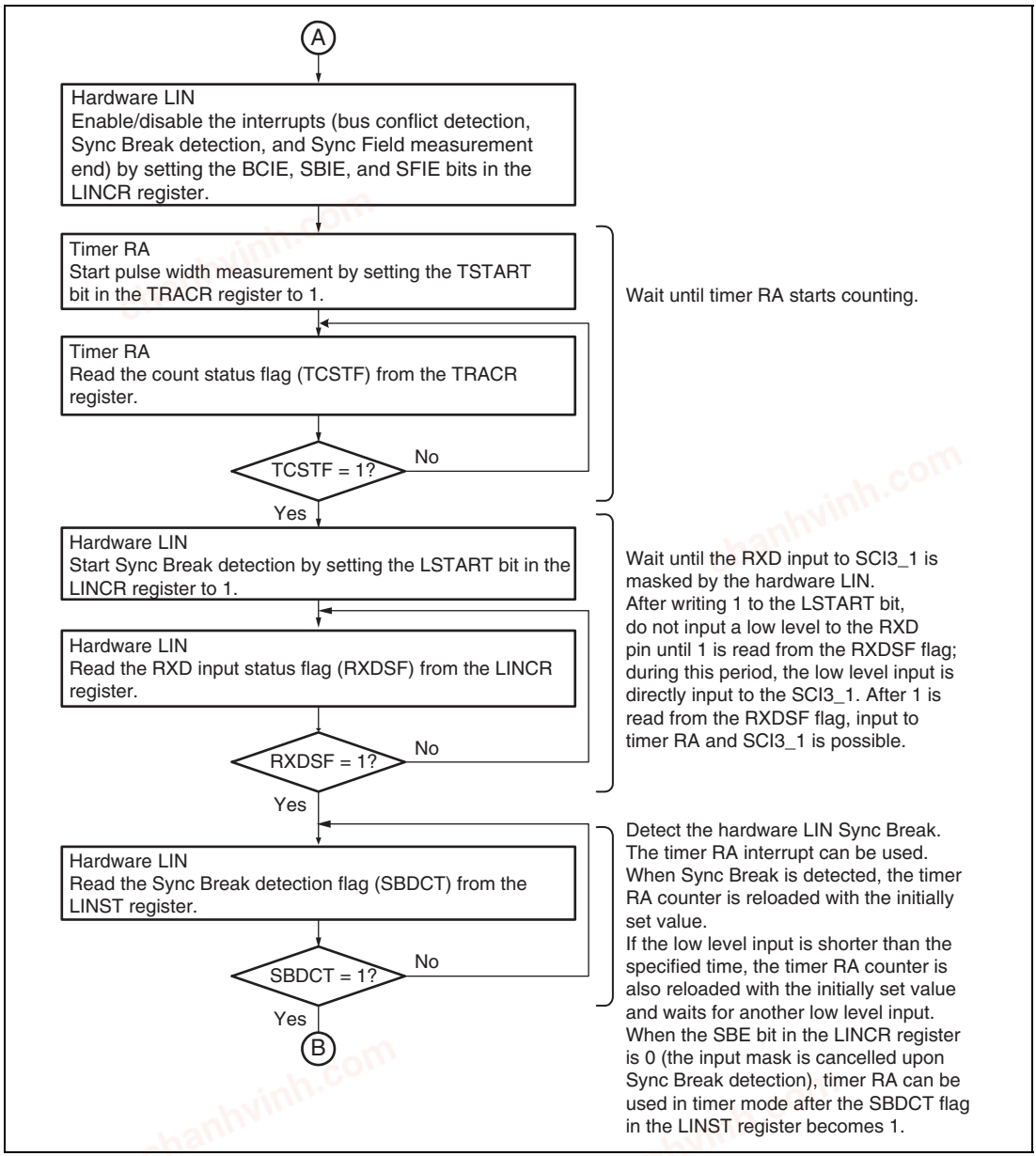



**Figure 23.7 Header Field Reception Flowchart (2)**

chanhvinh.com



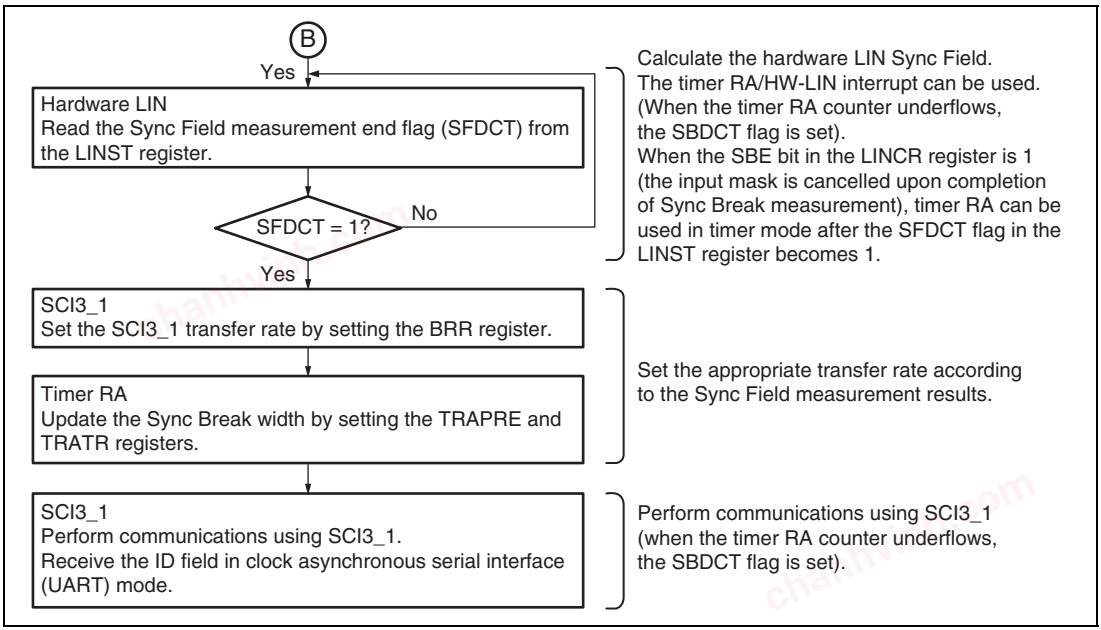



**Figure 23.8 Header Field Reception Flowchart (3)**

chanhvinh.com



### 23.3.3 Bus Conflict Detection Function

The hardware LIN interface can detect bus conflicts if SCI3_1 is enabled for transmission (TE bit in SCR3 register is 1).

Figure 23.9 shows the example of hardware LIN interface operation for detecting bus conflicts.

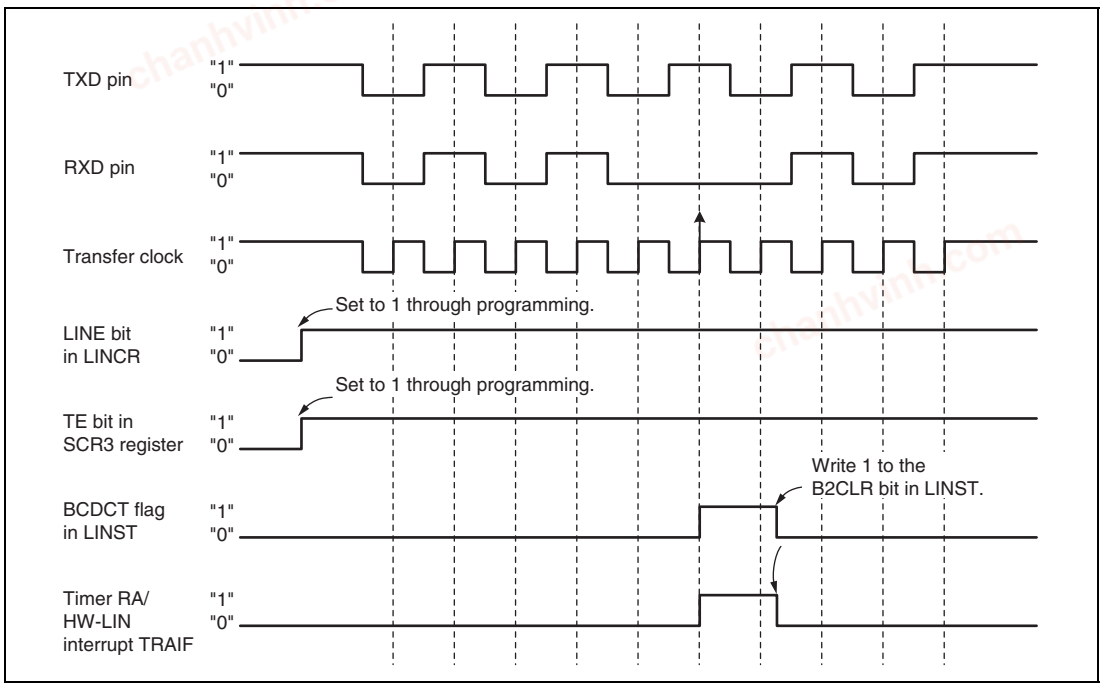



**Figure 23.9 Example of LIN Operation for Detecting Bus Conflicts**

chanhvinh.com



### 23.3.4 Terminating Hardware LIN

Figure 23.10 shows the flowchart for terminating hardware LIN communications. The hardware LIN interface should be terminated at the following timing.

- Case 1: When the bus conflict detection function is used
  After checksum field transmission
- Case 2: When the bus conflict detection function is not used
  After header field transmission/reception

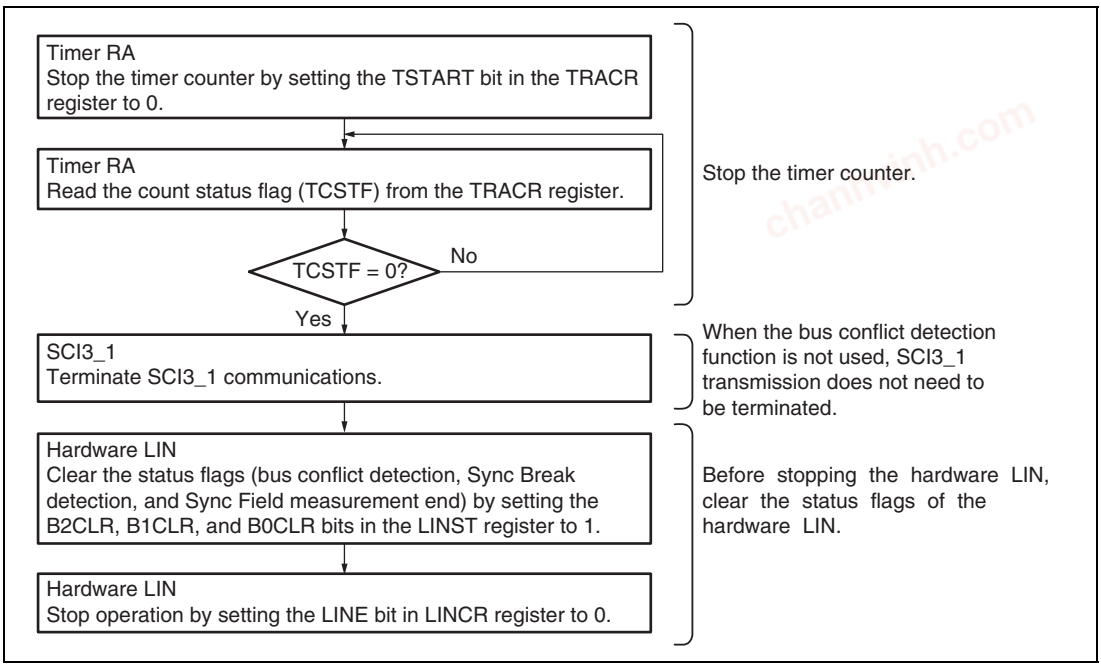



**Figure 23.10 Flowchart for Terminating Hardware LIN Communications**

chanhvinh.com



## 23.4 Interrupt Requests

The hardware LIN interface can request four types of interrupts: Sync Break detection, Sync Break generation end, Sync Field measurement end, and bus conflict detection. All these interrupts are requested as the timer RA interrupt. Table 23.2 describes these interrupt requests.

**Table 23.2 Interrupt Requests by Hardware LIN**

| Interrupt Request | Status Flag | Interrupt Source |
|---|---|---|
| Sync Break detection | SBDCT | • The low-level period of the RXD input is measured using timer RA and the counter underflows<br>• The low-level period of the RXD input is longer than the Sync Break period during communications. |
| Sync Break generation end | | The low level has been output via TXD for the period specified by timer RA. |
| Sync Field measurement end | SFDCT | Measuring the 8-bit Sync Field period has been completed using timer RA. |
| Bus conflict detection | BCDCT | The RXD input and TXD output values differ from each other when data is latched while SCI3_1 is enabled for transmission. |

## 23.5 Usage Note

For processing the header and response field timeout, measure the time from the Sync Break detection interrupt using another timer.

chanhvinh.com



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

chanhvinh.com



# Section 24 A/D Converter

This LSI includes a successive approximation type 10-bit A/D converter (one unit or two units) that allows up to sixteen analog input channels to be selected. Figures 24.1 and 24.2 show the block diagrams of A/D converters unit 1 and unit 2, respectively.

The differences between unit 1 and unit 2 are the number of analog input channels and the number of data registers. The other functions of units 1 and 2 are the same.

## 24.1 Features

- 10-bit resolution
- Input channels
  - Unit 1: 12 channels for the H8S/20223, H8S/20203, H8S/20215, and H8S/20235 Groups and 8 channels for the H8S/20103 and H8S/20115 Groups
  - Unit 2: 4 channels for the H8S/20223 and H8S/20235 Groups
- Conversion time: 2 µs per channel (at 20 MHz operation)
- Operating modes: Two
  - A/D conversion mode: A selected analog input is A/D converted
  - Compare mode: A selected analog input is compared with the voltage specified by the user
- Channel select modes
  - Single mode: Single-channel A/D conversion or comparison
  - Scan mode: Continuous A/D conversion on 1 to 4 channels, or 1 to 8 channels
- Data registers: 8 data registers for unit 1 and 4 data registers for unit 2
  - Conversion results are held in a 16-bit data register for each channel
- Sample and hold function
- Three kinds of conversion start
  - Conversion can be started by software, conversion start trigger by 16-bit timer (timer RC or RD), or external trigger signal.
- Interrupt request
  - A/D conversion end interrupt (ADI) request can be generated
  - Compare result change interrupt (CMPI) request can be generated
- Module standby function can be set

chanhvinh.com



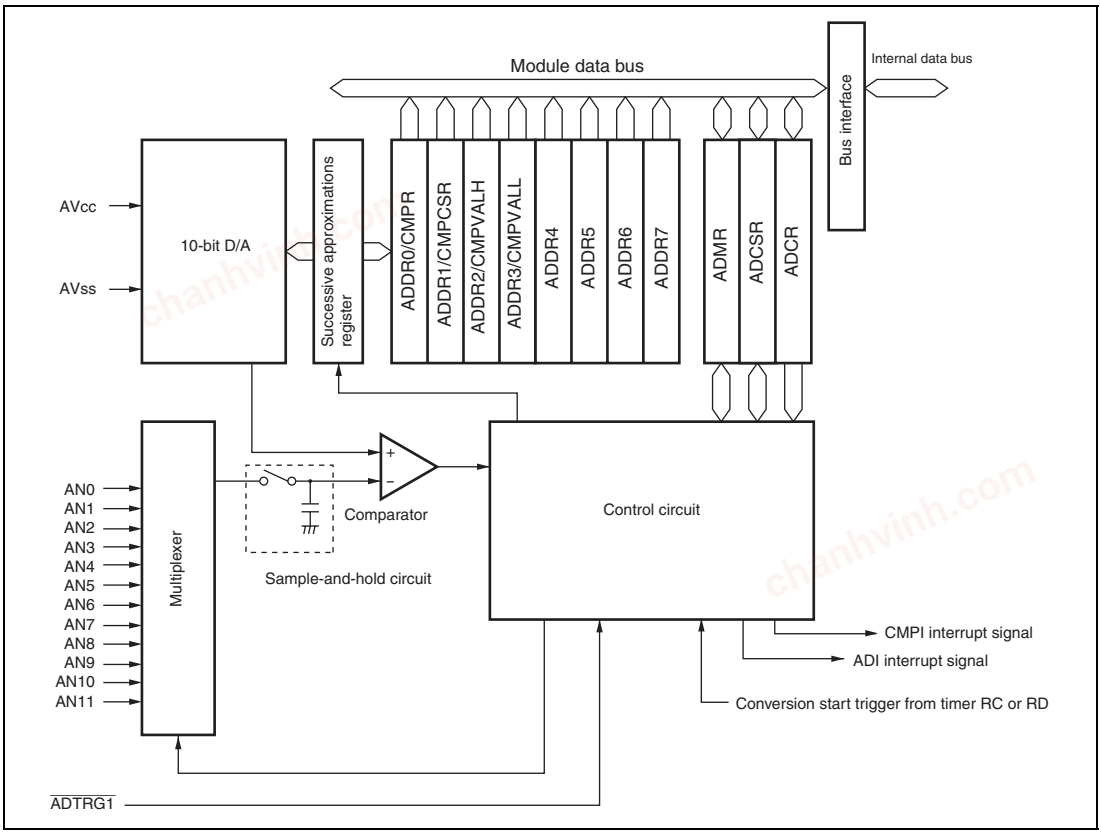



**Figure 24.1 Block Diagram of A/D Converter (Unit 1)**

chanhvinh.com



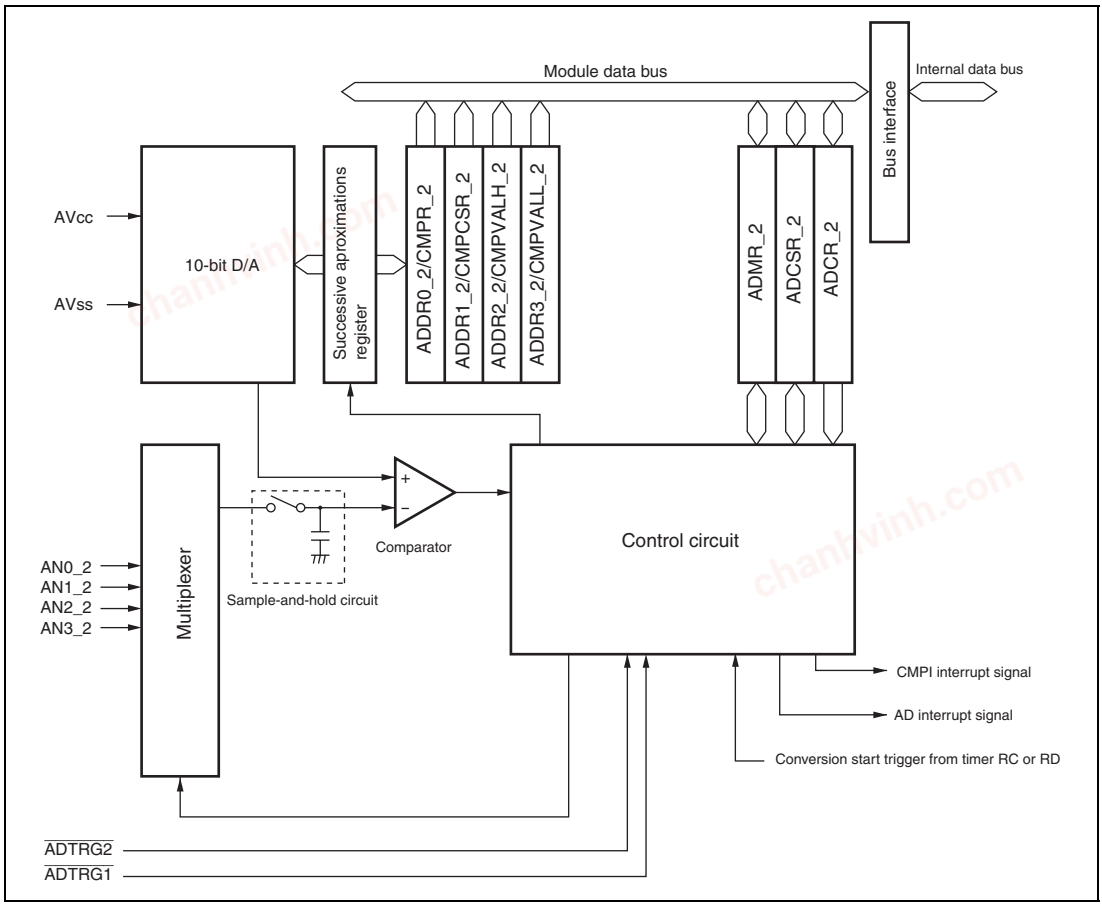



**Figure 24.2 Block Diagram of A/D Converter (Unit 2)**

chanhvinh.com



Table 24.1 shows the pin configuration of the A/D converter.

The $AV_{CC}$ and $AV_{SS}$ pins are the power supply pins for the analog block in the A/D converter.

Unit 1 has 12 analog input pins; unit 2 has four analog input pins. Note that the actual number of analog inputs in units 1 and 2 depends on the product group.

**Table 24.1 Pin Configuration**

| Unit | Pin Name | I/O | Function |
|---|---|---|---|
| Common | $AV_{CC}$ | Input | Analog block power supply |
| | $AV_{SS}$ | Input | Analog block ground |
| Unit 1 | AN0 | Input | Unit 1 group 0 analog inputs |
| | AN1 | Input | |
| | AN2 | Input | |
| | AN3 | Input | |
| | AN4 | Input | Unit 1 group 1 analog inputs |
| | AN5 | Input | |
| | AN6 | Input | |
| | AN7 | Input | |
| | AN8 | Input | Unit 1 group 2 analog inputs*1 |
| | AN9 | Input | |
| | AN10 | Input | |
| | AN11 | Input | |
| | $\overline{\text{ADTRG1}}$ | Input | External trigger input 1 for starting A/D conversion*3 |
| Unit 2 | AN0_2 | Input | Unit 2 group 0 analog inputs*2 |
| | AN1_2 | Input | |
| | AN2_2 | Input | |
| | AN3_2 | Input | |
| | $\overline{\text{ADTRG2}}$ | Input | External trigger input 2 for starting A/D conversion*2 |

Notes: 1. Not supported in the H8S/20103 and H8S/20115 Groups.
2. Supported only in the H8S/20223 and H8S/20235 Groups.
3. This pin can also be used as the trigger-signal input for unit 2.

chanhvinh.com



## 24.2 Register Description

The A/D converter has the following registers.

**Unit 1:**

- A/D data register 0 (ADDR0)
- A/D data register 1 (ADDR1)
- A/D data register 2 (ADDR2)
- A/D data register 3 (ADDR3)
- A/D data register 4 (ADDR4)
- A/D data register 5 (ADDR5)
- A/D data register 6 (ADDR6)
- A/D data register 7 (ADDR7)
- A/D control/status register (ADCSR)
- A/D control register (ADCR)
- A/D mode register (ADMR)
- Compare data register (CMPR)
- Compare control/status register (CMPCSR)
- Compare voltage register H (CMPVALH)
- Compare voltage register L (CMPVALL)

**Unit 2:**

- A/D data register 0_2 (ADDR0_2)
- A/D data register 1_2 (ADDR1_2)
- A/D data register 2_2 (ADDR2_2)
- A/D data register 3_2 (ADDR3_2)
- A/D control/status register_2 (ADCSR_2)
- A/D control register_2 (ADCR_2)
- A/D mode register_2 (ADMR_2)
- Compare data register_2 (CMPR_2)
- Compare control/status register_2 (CMPCSR_2)
- Compare voltage register H_2 (CMPVALH_2)
- Compare voltage register L_2 (CMPVALL_2)

chanhvinh.com



### 24.2.1 A/D Data Registers 0 to 7 (ADDR0 to ADDR7)

Address: H'FF05E0 to H'FF05EE, H'FF0600 to H'FF0606

| Bit: | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | – | – | – | – | – | – |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

ADDR registers are 16-bit read-only registers which are used to store the results of A/D conversion. Unit 1 incorporates eight registers ADDR0 to ADDR7. Unit 2 incorporates four registers ADDR0_2 to ADDR3_2. The ADDR registers, which store a conversion result for each channel, are shown in table 24.2.

The converted 10-bit data is stored to bits 15 to 6. The lower 6-bit data is always read as 0.

The data bus between the CPU and the A/D converter is 16-bit width. Data can be accessed in 16 bits at one time or 8 bits at two times.

**Table 24.2 Analog Input Channels and Corresponding ADDR Registers**

| Analog Input Channel | | A/D Data Register which Stores Conversion Result |
|---|---|---|
| Channel Set 0 (CH3 = 0) | Channel Set 1 (CH3 = 1) | |
| AN0 | AN8 | ADDR0 |
| AN1 | AN9 | ADDR1 |
| AN2 | AN10 | ADDR2 |
| AN3 | AN11 | ADDR3 |
| AN4 | — | ADDR4 |
| AN5 | — | ADDR5 |
| AN6 | — | ADDR6 |
| AN7 | — | ADDR7 |

chanhvinh.com



### 24.2.2 A/D Control/Status Register (ADCSR)

Address: H'FF05F0, H'FF0610

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | ADF | ADIE | ADST | — | CH[3:0] | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | ADF | A/D end flag | 0: A/D conversion or comparison is in progress.<br>1: A/D conversion or comparison has been completed.<br>[Setting conditions]<br>• When A/D conversion or comparison ends in single mode<br>• When A/D conversion or comparison ends on all specified channels in scan mode<br>[Clearing conditions]<br>• When 0 is written after reading ADF = 1<br>• When the DTC is activated by an ADI interrupt and ADDR is read | R/W* |
| 6 | ADIE | A/D interrupt enable | 0: Disables an ADF interrupt.<br>1: Enables an ADF interrupt. | R/W |
| 5 | ADST | A/D start | 0: Stops A/D conversion or comparison and places the A/D converter in the wait state.<br>1: Starts A/D conversion or comparison. | R/W |
| 4 | — | Reserved bit | This bit is read as 0. The write value should be 0. | — |
| 3 to 0 | CH[3:0] | Channel select 3 to 0 | When SCANE = 0 and SCANS = ×<br>0000: AN0<br>0001: AN1<br>0010: AN2<br>0011: AN3<br>0100: AN4<br>0101: AN5<br>0110: AN6<br>0111: AN7<br>1000: AN8<br>1001: AN9<br>1010: AN10<br>1011: AN11<br>11xx: Setting prohibited | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 3 to 0 | CH[3:0] | Channel select 3 to 0 | When SCANE = 1 and SCANS = 0<br>0000: AN0<br>0001: AN0 and AN1<br>0010: AN0 to AN2<br>0011: AN0 to AN3<br>0100: AN4<br>0101: AN4 and AN5<br>0110: AN4 to AN6<br>0111: AN4 to AN7<br>1000: AN8<br>1001: AN8 and AN9<br>1010: AN8 to AN10<br>1011: AN8 to AN11<br>11xx: Setting prohibited<br>When SCANE = 1 and SCANS = 1<br>0000: AN0<br>0001: AN0 and AN1<br>0010: AN0 to AN2<br>0011: AN0 to AN3<br>0100: AN0 to AN4<br>0101: AN0 to AN5<br>0110: AN0 to AN6<br>0111: AN0 to AN7<br>1000: AN8<br>1001: AN8 and AN9<br>1010: AN8 to AN10<br>1011: AN8 to AN11<br>11xx: Setting prohibited | R/W |

[Legend]

x: Don't care.

Notes: * Only 0 can be written in bit 7, to clear the flag.

1. The A/D converter should be stopped (ADST = 0) while the Input channels are being selected.
2. In unit 2, channels can be selected from four channels AN0_2 to AN3_2. Accordingly, the CH3 and CH2 bits should be cleared to 0 in unit 2.

- ADST bit (A/D start)

  Clearing this bit to 0 stops A/D conversion or comparison, and the A/D converter enters wait state. When this bit is set to 1 by software, timer RC, timer RD (conversion start trigger), or the $\overline{\text{ADTRG}}$ pin, A/D conversion or comparison starts. This bit remains set to 1 during A/D conversion or comparison. In single mode, this bit is cleared to 0 automatically when conversion or comparison on the specified channel ends. In scan mode, conversion continues sequentially on the specified channels until this bit is cleared to 0 by a reset, a transition to standby mode or software. ADST is cleared to 0 if A/D conversion or comparison of all the selected channels has been completed while the ADSTCLR bit is 1.

chanhvinh.com



The event link function can be used to set the ADST bit. When the event specified in ELSR10 or ELS11 of the ELC occurs, the corresponding ADST bits (in A/D converter unit 1 or A/D converter unit 2, respectively) are set and the A/D conversion or comparison starts.

To select AN0 and AN0_2 as the channels for conversion, specify analog input operation for the pins by setting the PMRA2 and PMRA3 bits.

### 24.2.3 A/D Control Register (ADCR)

Address: H'FF05F1, H'FF0611

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TRGS[1:0] | | SCANE | SCANS | CKS[1:0] | | ADSTCLR | EXTRGS |
| Value after reset: | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7, 6 | TRGS[1:0] | Trigger select 1 and 0 | b0 b7 b6<br>0 0 0: A/D conversion start by external trigger is disabled.<br>0 0 1: A/D conversion start by external trigger pin 1 ($\overline{\text{ADTRG1}}$) is enabled.<br>0 1 0: A/D conversion start by external trigger pin 2 ($\overline{\text{ADTRG2}}$) is enabled.*1<br>0 1 1: A/D conversion start by external trigger (timer RC) is enabled.*2<br>1 0 0: A/D conversion start by external trigger (timer RD_0) is enabled.<br>1 0 1: A/D conversion start by external trigger (timer RD_1) is enabled.*3<br>1 1 x: Reserved (setting prohibited) | R/W |
| 5<br>4 | SCANE<br>SCANS | Channel selection mode | 0x: Single mode<br>10: Scan mode (A/D conversion is performed continuously for channels 1 to 4)<br>11: Scan mode (A/D conversion is performed continuously for channels 1 to 8.) | R/W |
| 3, 2 | CKS[1:0]*4 | Clock select 1 to 0 | 00: Setting prohibited<br>01: Setting prohibited<br>10: A/D conversion time = 84 states (max) (initial value)<br>11: A/D conversion time = 43 states (max) | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 1 | ADSTCLR | ADST clear | If ADSTCLR is set to 1 in scan mode, the ADST bit is automatically cleared to 0 when A/D conversion of all the selected channels has been completed. | R/W |
| 0 | EXTRGS | External trigger select | EXTRGS combined with the TRGS1 and TRGS0 bits selects a trigger signal. For details, see the above description for the TRGS1 and TRGS0 bits. | R/W |

[Legend]

x: Don't care.

Notes: 1. Selected only for the H8S/20223 and H8S/20235 Groups.
2. Selected only for the H8S/20103 and H8S/20115 Groups.
3. Not selected only for the H8S/20103 and H8S/20115 Groups.
4. Select these bits to fall the conversion time within the specified time.

- TRGS[1:0] bits (trigger select 1 and 0)

  These bits combined with the EXTRGS bit select enable or disable the A/D conversion start by a trigger signal.

- CKS[1:0] bits (clock select 1 to 0)

  These bits the A/D conversion time.

  The conversion time should be set while the A/D conversion is stopped (ADST = 0).

chanhvinh.com



### 24.2.4 A/D Mode Register (ADMR)

Address: H'FF05F4, H'FF0614

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | ADM1 | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7, 6 | — | Reserved | These bits are read as 0. The write value should be 0. | — |
| 5 | ADM1 | A/D converter operating mode selection | 0: A/D conversion mode<br>1: Compare mode | R/W |
| 4 to 0 | — | All 0 | These bits are read as 0. The write value should be 0. | — |

Note: The A/D converter operating mode should be changed while the ADST bit in ADCSR is 0.

- ADM1 bit (A/D conversion mode selection)

  If the A/D converter operating mode changes from conversion mode to compare mode, CMPR, CMPCSR, and CMPVAL are initialized to H'00.

chanhvinh.com



### 24.2.5 Compare Data Register (CMPR)

Address: H'FF05E0, H'FF0600

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | CMP7 | CMP6 | CMP5 | CMP4 | CMP3 | CMP2 | CMP1 | CMP0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | CMP7 | Compare data 7 | [Setting condition]<br>When the voltage of the selected analog input channel is greater than the voltage set in the CMPVAL register in compare mode.<br>[Clearing conditions]<br>• When the A/D converter operating mode is changed from A/D conversion mode to compare mode according to the ADM bit in ADMR setting.<br>• When the voltage of the selected analog input channel is equal to or lower than the voltage set in the CMPVAL register in compare mode. | R |
| 6 | CMP6 | Compare data 6 | | R |
| 5 | CMP5 | Compare data 5 | | R |
| 4 | CMP4 | Compare data 4 | | R |
| 3 | CMP3 | Compare data 3 | | R |
| 2 | CMP2 | Compare data 2 | | R |
| 1 | CMP1 | Compare data 1 | | R |
| 0 | CMP0 | Compare data 0 | | R |

[Legend]
x: Don't care.
Note: * Only 0 can be written to clear the flag.

CMPR holds the comparison result. CMPR is a read-only register that is assigned to the same address as ADDR0 and ADDR0_2. CMPR is valid in compare mode.

chanhvinh.com



CMP bits and the corresponding analog input channels are shown in table 24.3.

**Table 24.3 Relationship between CMP Bits and Corresponding Analog Input Channels**

| Unit | Channel | | Corresponding Compare Data Bit |
|---|---|---|---|
| Unit 1 | AN0 | AN8 | CMP0 |
| | AN1 | AN9 | CMP1 |
| | AN2 | AN10 | CMP2 |
| | AN3 | AN11 | CMP3 |
| | AN4 | — | CMP4 |
| | AN5 | — | CMP5 |
| | AN6 | — | CMP6 |
| | AN7 | — | CMP7 |
| Unit 2 | AN0_2 | — | CMP0 |
| | AN1_2 | — | CMP1 |
| | AN2_2 | — | CMP2 |
| | AN3_2 | — | CMP3 |

chanhvinh.com



### 24.2.6 Compare Control/Status Register (CMPCSR)

Address: H'FF05E2, H'FF0602

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | CMPF | CMPIE | CMPFC1 | CMPFC0 | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | CMPF | CMPI interrupt status | [Setting condition]<br>If the condition specified by the CMPFC1 or CMPFC0 bit is satisfied when comparison has been completed.<br>[Clearing conditions]<br>• When the A/D converter operating mode is changed from A/D conversion mode to compare mode according to the ADM bit in ADMR setting.<br>• When 0 is written to this bit after this bit is read as 1.<br>• When the DTC is activated by a CMPI interrupt and the DISEL bit in MRB of the DTC is 0.<br>• When this LSI enters standby mode or module standby mode. | R/W |
| 6 | CMPIE | CMPI interrupt enable | 0: Disables a compare match interrupt (CMPI).<br>1: Enables a compare match interrupt (CMPI). | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 5 | CMPFC1 | CMPI interrupt condition 1 | 0: Does not generate an interrupt by a comparison result change.<br>1:<br>In single compare mode:<br>Sets the CMPF bit to 1 if the comparison result of the selected channel changes from 0 to 1.<br>In scan compare mode:<br>Sets the CMPF bit to 1 if the comparison result of any of the selected channels changes from 0 to 1. | R/W |
| 4 | CMPFC0 | CMPI interrupt condition 0 | 0: Does not generate an interrupt by a comparison result change.<br>1:<br>In single compare mode:<br>Sets the CMPF bit to 1 if the comparison result of the selected channel changes from 0 to 1.<br>In scan compare mode:<br>Sets the CMPF bit to 1 if the comparison result of any of the selected channels changes from 0 to 1. | R/W |
| 3 to 0 | — | Reserved | These bits are read as 0. The write value should be 0. | — |

chanhvinh.com



### 24.2.7 Compare Voltage Registers H and L (CMPVALH and CMPVALL)

• CMPVALH

Address: H'FF05E4, H'FF0604

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | VAL9 | VAL8 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

• CMPVALL

Address: H'FF05E6, H'FF0606

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | VAL7 | VAL6 | VAL5 | VAL4 | VAL3 | VAL2 | VAL1 | VAL0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- CMPVALH

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 to 2 | — | Reserved | This bit is read as 0. The write value should be 0. | — |
| 1 | VAL9 | — | These bits set the compare voltage VAL[9:8]. | R/W |
| 0 | VAL8 | — | | R/W |

- CMPVALL

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | VAL7 | — | These bits set the compare voltage VAL[7:0]. | R/W |
| 6 | VAL6 | — | | R/W |
| 5 | VAL5 | — | | R/W |
| 4 | VAL4 | — | | R/W |
| 3 | VAL3 | — | | R/W |
| 2 | VAL2 | — | | R/W |
| 1 | VAL1 | — | | R/W |
| 0 | VAL0 | — | | R/W |

chanhvinh.com



CMPVALL and the lower 2 bits of CMPVALH specify the voltage to be compared.

CMPVALH and CMPVALL are assigned to the same addresses as ADDR2 (ADDR2_2) and ADDR3 (ADDR3_2), respectively. CMPVALH and CMPVALL become valid in compare mode.

Table 24.4 shows the correspondence between VAL[9:0] setting and the voltage to be compared.

**Table 24.4 VAL[9:0] Setting and Corresponding Voltage to be Compared**

| VAL[9:0] Setting | Voltage to be Compared |
|---|---|
| B'0000000000 | AVss |
| B'0000000001 | AVcc × 1/1024 |
| B'0000000010 | AVcc × 2/1024 |
| : | : |
| B'111111100 | AVcc × 1020/1024 |
| B'111111101 | AVcc × 1021/1024 |
| B'111111110 | AVcc × 1022/1024 |
| B'111111111 | AVcc × 1023/1024 |

chanhvinh.com



## 24.3 Operation

The A/D converter operates in two operating modes as shown in table 24.5. In A/D conversion mode, the A/D converter converts the analog input of the selected channel by successive approximation with 10-bit resolution. In compare mode, the analog input of the selected channel is compared with the voltage to be specified.

Each operating mode has two operating modes: single mode and scan mode. When changing the analog input channel, to prevent incorrect operation, first clear the ADST bit in ADCSR to 0. The ADST bit can be set at the same time as the operating mode or analog input channel is changed.

**Table 24.5 A/D Converter Operating Mode**

| Operating Mode | Channel Selection Mode | Register Setting |
|---|---|---|
| A/D conversion mode | Single mode | ADM1 = 0, SCANE = 0 |
| | Scan mode | ADM1 = 0, SCANE = 1 |
| Compare mode | Single mode | ADM1 = 1, SCANE = 0 |
| | Scan mode | ADM1 = 1, SCANE = 1 |

chanhvinh.com



## 24.4 A/D Conversion Mode Operation

### 24.4.1 Single Mode in A/D Conversion Mode

In single mode in A/D conversion mode, A/D conversion is to be performed only once on the specified single channel. Operations are as follows.

1. A/D conversion is started when the ADST bit in ADCSR is set to 1, according to the software or external trigger input.
2. When A/D conversion is completed, the result is transferred to the A/D data register corresponding to the channel.
3. On completion of conversion, 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 remains set to 1 during A/D conversion, and is automatically cleared to 0 when conversion ends. When the ADST bit is cleared to 0 during A/D conversion, A/D conversion stops and the A/D converter enters wait state.

chanhvinh.com



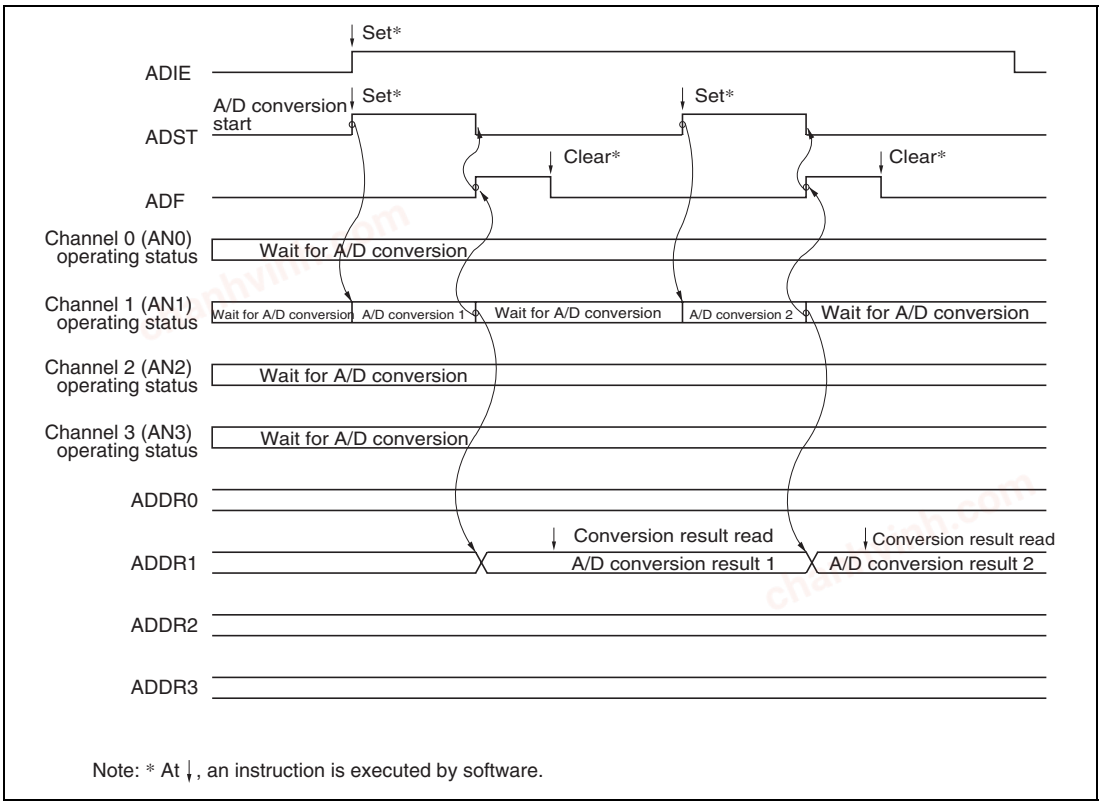



**Figure 24.3 A/D Converter Operation in A/D Conversion Mode (When Channel 1 Is Selected in Single Mode)**

chanhvinh.com



### 24.4.2 Scan Mode in A/D Conversion Mode

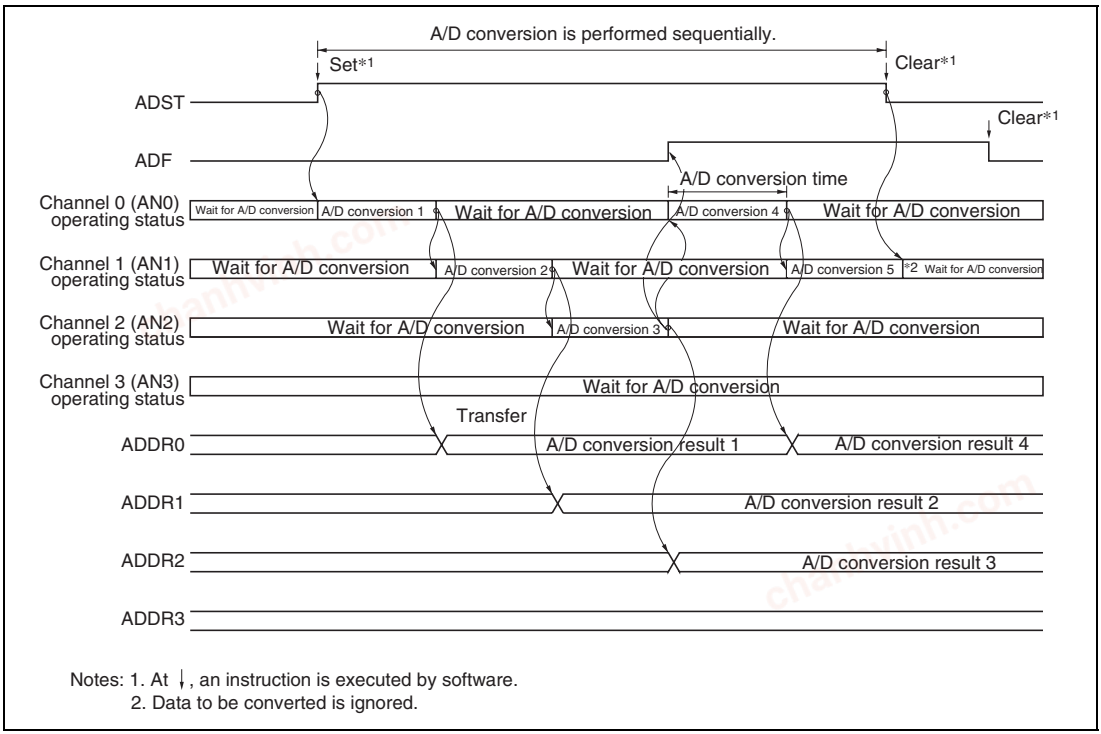
In scan mode in A/D conversion mode, A/D conversion is to be performed sequentially on the specified channels: maximum four channels or maximum eight channels. Operations are as follows.

1. When the ADST bit in ADCSR is set to 1 by a software, timer RC, timer RD or external trigger input, A/D conversion starts on the first channel of the channel set.

   The consecutive A/D conversion on maximum four channels (SCANE = 1 and SCANS = 0) or on maximum eight channels (SCANE = 1 and SCANS = 1) can be selected. When the consecutive A/D conversion is performed on the four channels, the A/D conversion starts on AN0 when CH3 = 0 and CH2 =0, AN4 when CH3 = 0 and CH2 = 1, or AN8 when CH3 = 1 and CH2 = 0. When the consecutive A/D conversion is performed on the eight channels, the A/D conversion starts on AN0 when CH3 = 0 and CH2 = 0.
2. When A/D conversion for each channel is completed, the result is sequentially transferred to the A/D data register corresponding to each channel.
3. When conversion of all the selected channels is completed, the ADF bit in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt is requested. The A/D conversion starts again from the first channel of the channel set again.
4. The ADST bit is not cleared automatically, and steps [2] to [3] are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops and the A/D converter enters wait state. If the ADST bit is later set to 1, A/D conversion starts again from the first channel of the channel set.

chanhvinh.com






Notes: 1. At ↓, an instruction is executed by software.
2. Data to be converted is ignored.

**Figure 24.4 A/D Converter Operation in A/D Conversion Mode (When AN0 to AN2 Channels are Selected in Scan Mode)**

chanhvinh.com



## 24.5 Compare Mode Operation

### 24.5.1 Single Mode in Compare Mode

In single mode in compare mode, the analog input of one selected channel is compared with the specified voltage. Operations are as follows. The setting of the channel by the CH[3:0] bits in ADCSR is the same as that in A/D conversion mode.

1. Comparison between the analog input of the selected channel and the voltage specified by the VAL[9:0] bits is started when the ADST bit in ADCSR is set to 1 by software or external trigger input.
2. When the comparison is completed, the result is transferred to a bit corresponding to the channel.
3. On completion of comparison, 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. In addition, if a condition specified by the CMPFC1 or CMPFC0 bit is satisfied, the CMPF bit in CMPCSR is set to 1. If the CMPIE bit is set to 1 at this time, a CMPI interrupt is requested.
4. The ADST bit remains set to 1 during comparison, and is automatically cleared to 0 when comparison ends. When the ADST bit is cleared to 0 during comparison, the A/D converter stops operation and enters wait state.

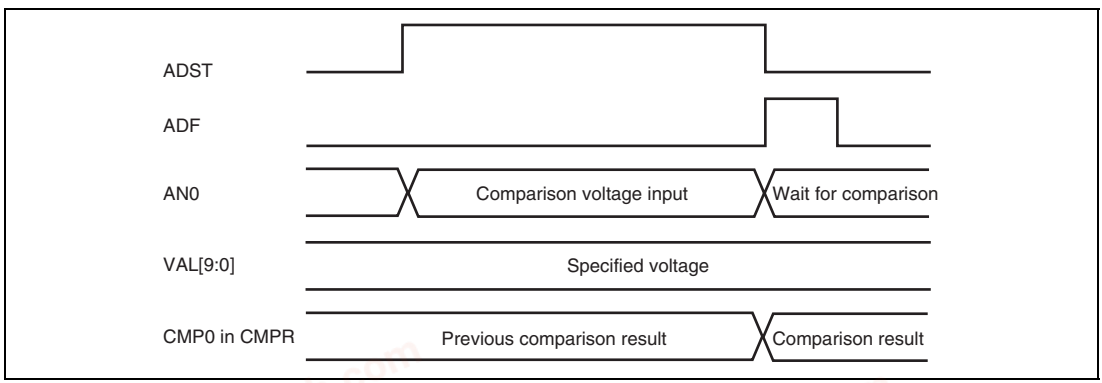



**Figure 24.5 A/D Converter Operation in Compare Mode (When Channel 0 Is Selected in Single Mode)**

chanhvinh.com



### 24.5.2 Scan Mode in Comparison Mode

In scan mode in comparison mode, the analog input of the selected channels (four or eight maximum) are compared sequentially with the specified voltage. Operations are as follows.

1. When the ADST bit in ADCSR is set to 1 by a software, timer RC, timer RD or external trigger input, comparison between the analog input of the selected channels and the voltage specified by the VAL[9:0] bits is started.

   The comparison on maximum four channels (SCANE = 1 and SCANS= 0) or on maximum eight channels (SCANE = 1 and SCANS= 1) can be selected. When the consecutive comparison is performed on the four channels, the comparison starts on AN0 when CH3 = 0 and CH2 = 0, AN4 when CH3 = 0 and CH2 = 1, or AN8 when CH3 = 1 and CH2 = 0. When the consecutive comparison is performed on the eight channels, the comparison starts on AN0 when CH3 = 0 and CH2 = 0.
2. When comparison for each channel is completed, the result is sequentially transferred to a bit corresponding to each channel.
3. When comparison of all the selected channels is completed, the ADF bit in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt is requested. In addition, if a condition specified by the CMPFC1 or CMPFC0 bits is satisfied in any of the selected channels, the CMPF bit in CMPCSR is set to 1. If the CMPIE bit is set to 1 at this time, a CMPI interrupt is requested. The A/D converter starts comparison from the first channel of the channel set.
4. The ADST bit is not cleared automatically when ADSTCLR = 0, and steps [2] to [3] are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0 during comparison, the A/D converter stops operation and enters wait state. If the ADST bit is later set to 1, the A/D converter starts comparison from the first channel of the channel set.

chanhvinh.com



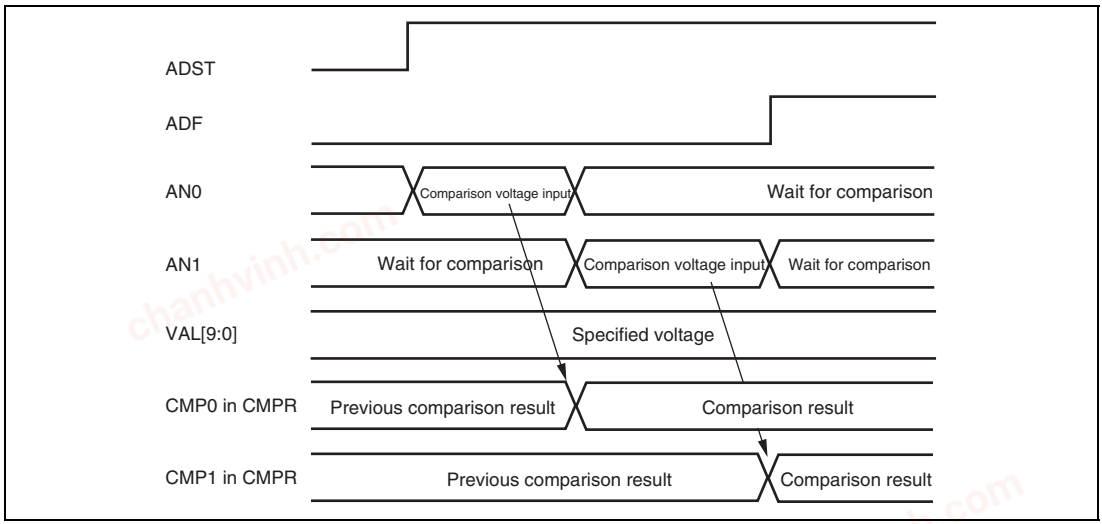



**Figure 24.6 A/D Converter Operation in Compare Mode (When AN0 to AN2 Channels are Selected in Scan Mode)**

### 24.5.3 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 when A/D conversion start delay time ($t_D$) passes after the ADST bit is set to 1, then starts conversion. Figure 24.7 shows the A/D conversion timing. Table 24.6 indicates the A/D conversion time.

As indicated in figure 24.7, 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 tables 24.6.

In scan mode, the values given in table 24.6 apply to the first conversion time. The values given in table 24.7 apply to the second and subsequent conversions. In any conversions, the CKS[1:0] bits in ADCR should be set so that the conversion time should fall within the specified A/D conversion characteristics range.

chanhvinh.com



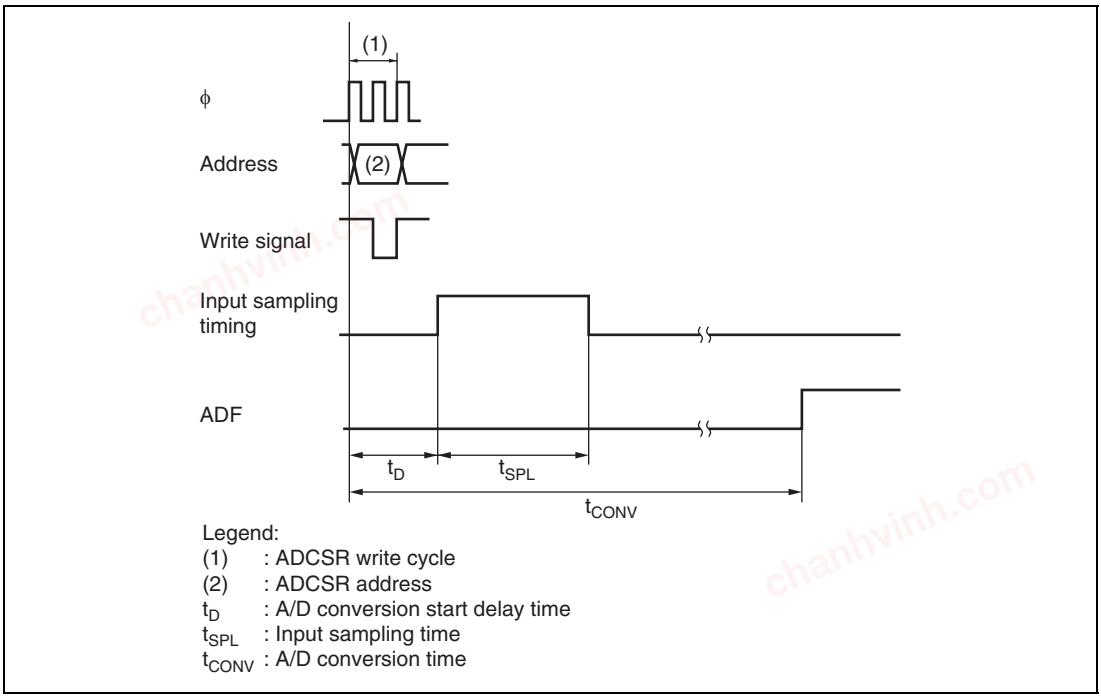



**Figure 24.7 A/D Conversion Timing**

**Table 24.6 A/D Conversion Time (Single Mode)**

| | | CKS1 = 1 | | | | | |
|---|---|---|---|---|---|---|---|
| | | CKS0 = 0 | | | CKS0 = 1 | | |
| **Item** | **Symbol** | **Min** | **Typ** | **Max** | **Min** | **Typ** | **Max** |
| A/D conversion start delay time | $t_D$ | 3 | — | 4 | — | 3 | — |
| Input sampling time | $t_{SPL}$ | — | 30 | — | — | 15 | — |
| A/D conversion time | $t_{CONV}$ | 83 | — | 84 | — | 43 | — |

Note: Values in the table are the number of states.

chanhvinh.com



**Table 24.7 A/D Conversion Time (Scan Mode)**

| CKS1 | CKS0 | Conversion Time (State) |
|---|---|---|
| 1 | 0 | 80 |
| | 1 | 40 |

### 24.5.4 External Trigger Input Timing

A/D conversion can be externally triggered. When the EXTRGS, TRGS1, and TRGS0 bits are set to B'001 in ADCR, external trigger input is enabled at the $\overline{\text{ADTRG}}$ pin. A falling edge of the $\overline{\text{ADTRG}}$ pin sets the ADST bit in ADCSR to 1, starting A/D conversion. Other operations, in both single and scan modes, are the same as when the bit ADST has been set to 1 by software. Figure 24.8 shows the timing.

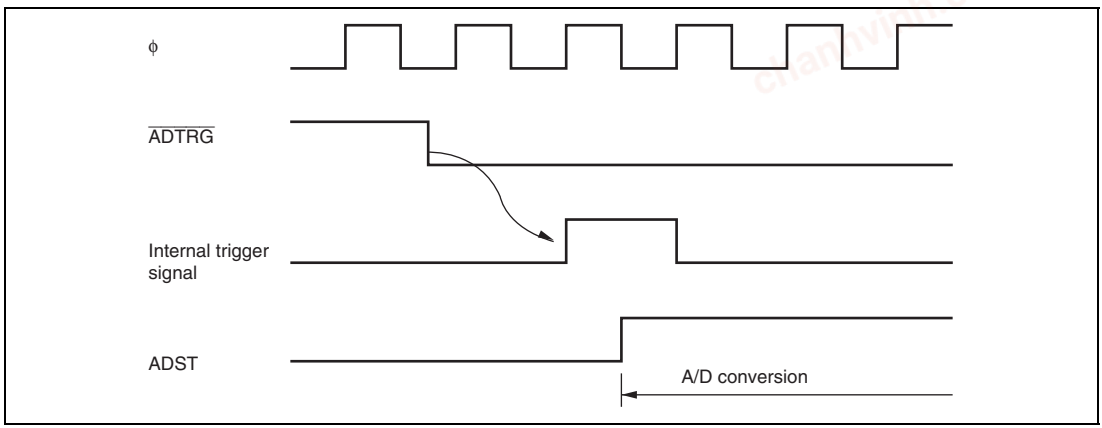



**Figure 24.8 External Trigger Input Timing**

chanhvinh.com



## 24.6 Interrupt Source

In A/D conversion mode, an A/D conversion end interrupt (ADI) occurs at the end of A/D conversion. Specifically, the ADF bit in ADCSR is set to 1 when A/D conversion is completed; and if the ADIE bit is 1 at this time, the A/D converter generates an ADI interrupt.

In compare mode, a compare result change interrupt (CMPI) occurs if the comparison result of the specified channel changes (in three cases: from 1 to 0, from 0 to 1, and both). Specifically, the CMPF bit is set when the comparison result between the specified channel and the specified voltage satisfies the specified condition; and if the CMPIE bit is 1 at this time, the A/D converter generates a CMPI interrupt.

The DTC can be activated by an ADI or CMPI interrupt. Having the converted data read by the DTC in response to an ADI or CMPI interrupt enables continuous conversion to be achieved without imposing a load on software.

**Table 24.8 A/D Converter Interrupt Source**

| Name | Interrupt Source | Interrupt Flag | DTC Activation |
|---|---|---|---|
| ADI | End of A/D conversion or comparison | ADF | Possible |
| CMPI | Comparison result change | CMPF | Possible |

chanhvinh.com



## 24.7 A/D Conversion Accuracy Definitions

This LSI's A/D conversion accuracy definitions are given below.

- Resolution

  The number of A/D converter digital output codes
- Quantization error

  The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 24.9).
- Offset error

  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from the minimum voltage value B'0000000000 (H'000) to B'0000000001 (H'001) (see figure 24.10).
- Full-scale error

  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from B'1111111110 (H'3FE) to B'1111111111 (H'3FF) (see figure 24.10).
- Nonlinearity error

  The error with respect to the ideal A/D conversion characteristic between the zero voltage and the full-scale voltage. Does not include the offset error, full-scale error, or quantization error (see figure 24.10).
- Absolute precision

  The deviation between the digital value and the analog input value. Includes the offset error, full-scale error, quantization error, and nonlinearity error.

chanhvinh.com



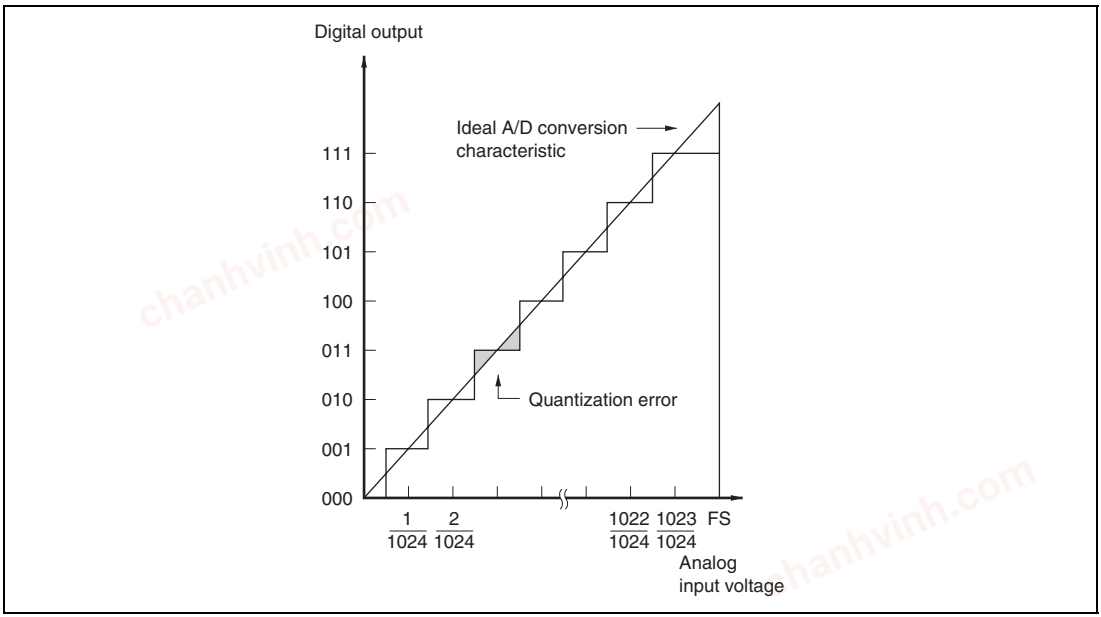



**Figure 24.9 A/D Conversion Accuracy Definitions**

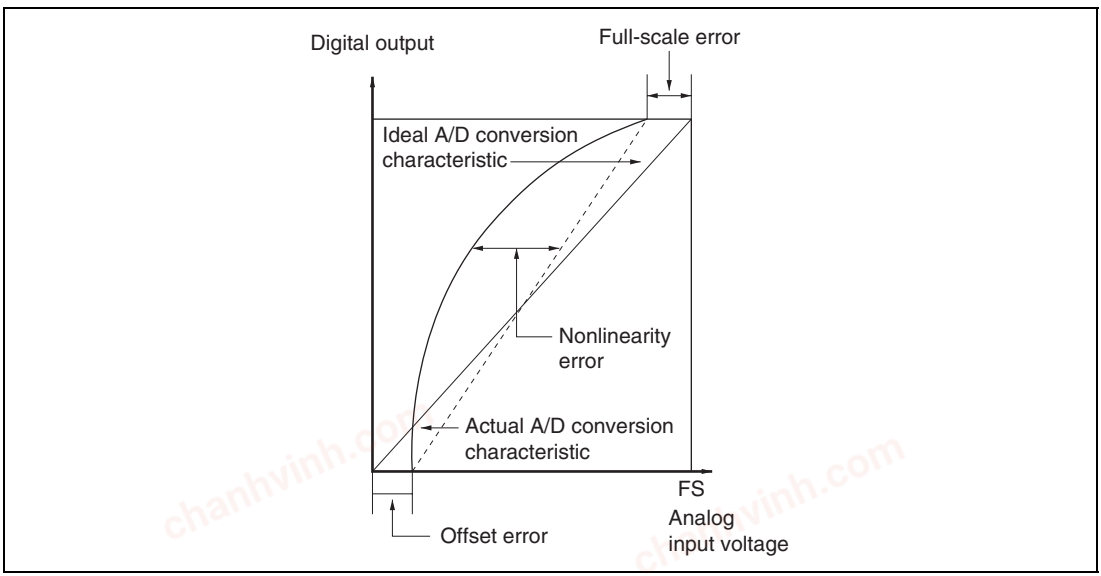



**Figure 24.10 A/D Conversion Accuracy Definitions**

chanhvinh.com



## 24.8 Usage Notes

### 24.8.1 Module Standby Mode Setting

Operation of the A/D converter can be disabled or enabled using the module 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 6, Power-Down Modes.

### 24.8.2 Permissible Signal Source Impedance

This LSI's analog input is designed so 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 the A/D conversion accuracy. However, if a large capacitance is provided externally for conversion in single mode, the input load will essentially comprise only the internal input resistance of 10 kΩ, and the signal source impedance is ignored. However, since 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 24.11). When converting a high-speed analog signal or conversion in scan mode, a low-impedance buffer should be inserted.

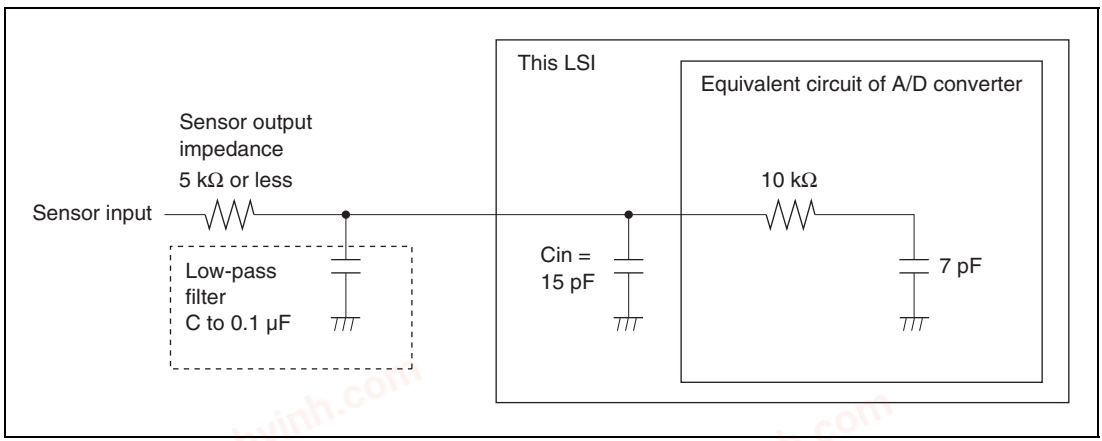



**Figure 24.11 Example of Analog Input Circuit**

chanhvinh.com



### 24.8.3 Influences on Absolute Precision

Adding capacitance results in coupling with GND, and therefore noise in GND may adversely affect absolute precision. Be sure to make the connection to an electrically stable GND such as AVss.

Care is also required to insure that filter circuits do not communicate with digital signals on the mounting board, so acting as antennas.

### 24.8.4 Setting Range of Analog Power Supply and Other Pins

If conditions shown below are not met, the reliability of the device may be adversely affected.

- Analog input voltage range
  The voltage applied to analog input pin ANn during A/D conversion should be in the range AVss ≤ ANn ≤ AVcc.
- Relation between AVcc, AVss and Vcc, Vss
  As the relationship between AVcc, AVss and Vcc, Vss, set AVcc ≤ Vcc and AVss = Vss. If the A/D converter is not used, the AVcc and AVss pins must not be left open.

### 24.8.5 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.

Also, digital circuitry must be isolated from the analog input signals (AN0 to AN1, AN0_2 to AN3_2) 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



### 24.8.6 Notes on Noise Countermeasures

A protection circuit connected to prevent damage due to an abnormal voltage such as an excessive surge at the analog input pins (AN0 to AN11, AN0_2 to AN3_2) should be connected between AVcc and AVss as shown in figure 24.12. Also, the bypass capacitors connected to AVcc and the filter capacitor connected to AN0 to AN11 or AN0_2 to AN3_2 must be connected to AVss.

If a filter capacitor is connected, the input currents at the analog input pins (AN0 to AN1, AN0_2 to AN3_2) are averaged, and so an error may arise. Also, when A/D conversion is performed frequently, as in scan mode, if the current charged and discharged by the capacitance of the sample-and-hold circuit in the A/D converter exceeds the current input via the input impedance ($R_{in}$), an error will arise in the analog input pin voltage. Careful consideration is therefore required when deciding the circuit constants.

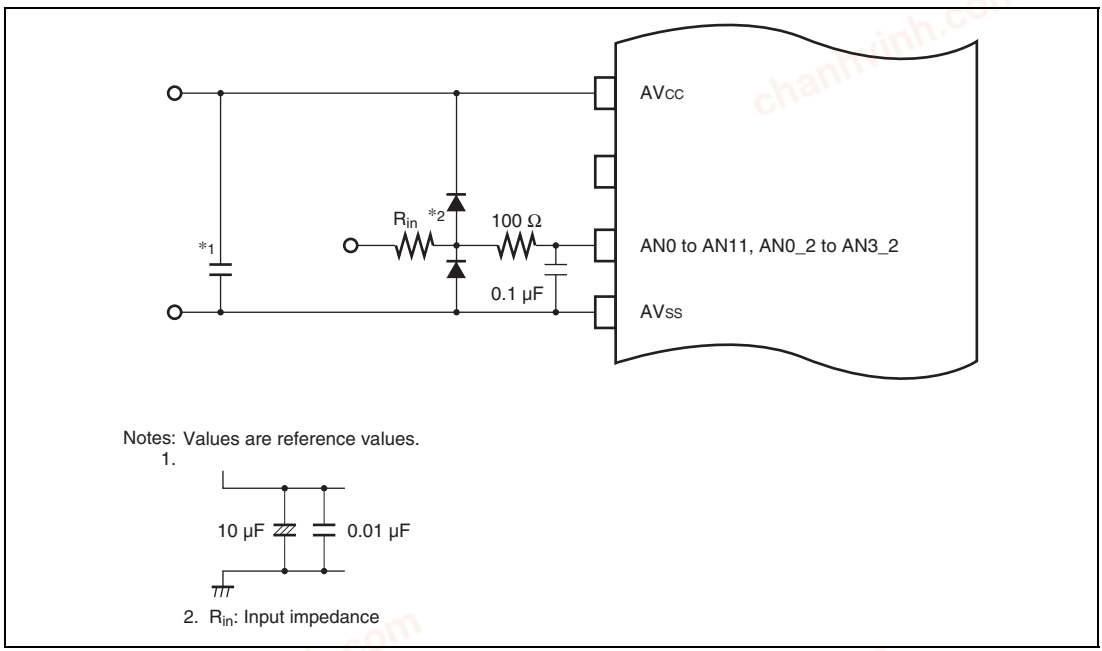



**Figure 24.12 Example of Analog Input Protection Circuit**

chanhvinh.com



### 24.8.7 Notes on Analog Input Pins

Analog input pins (AN0 to AN11, AN0_2 to AN3_2) are multiplexed with general I/O ports. Accordingly, if the direction of input or output of the general I/O port is changed or the output value of the general I/O port is changed during A/D conversion, the conversion accuracy may be affected.

Before using an analog input pin multiplexed with general I/O port as a general I/O port, influence on the A/D conversion accuracy should be evaluated carefully.

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

chanhvinh.com



# Section 25 D/A Converter

## 25.1 Features

- 8-bit resolution
- Output channels: 2 channels
- Maximum conversion time of 3 µs (with 20 pF load capacitance)
- Output voltage of 0 V to AVcc
- Settable for the module standby mode

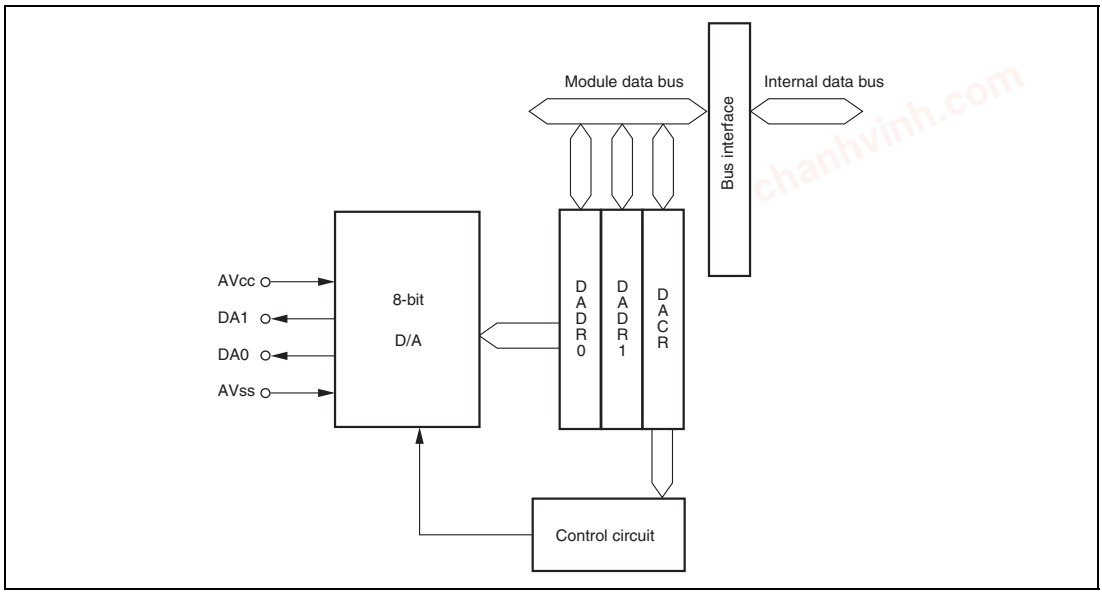



**Figure 25.1 Block Diagram of D/A Converter**

chanhvinh.com



Table 25.1 shows the input/output pin configuration of the D/A converter.

**Table 25.1 Pin Configuration**

| Pin Name | I/O | Function |
|---|---|---|
| AVcc | Input | Analog power supply |
| AVss | Input | Analog ground |
| DA0 | Output | Channel 0 analog output |
| DA1 | Output | Channel 1 analog output |

## 25.2 Register Descriptions

- D/A data register 0 (DADR0)
- D/A data register 1 (DADR1)
- D/A control register (DACR)

### 25.2.1 D/A Data Registers 0 and 1 (DADR0 and DADR1)

- DADR0 and DADR1

• DADR0, DADR1

Address: H'FF05D4, H'FF05D5

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

DADR0 and DADR1 are 8-bit readable/writable registers that store data for conversion. Whenever analog output is enabled, the values in DADR are converted and output to the analog output pins. DADR is initialized to H'00 in standby mode or module standby.

chanhvinh.com



### 25.2.2 D/A Control Register (DACR)

Address: H'FF05D6

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | DAOE1 | DAOE0 | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | DAOE1 | D/A output enable 1 | 0: Disables the analog output on channel 1 (DA1).<br>1: Enables the channel 1 D/A conversion; enables the analog output (DA1). | R/W |
| 6 | DAOE0 | D/A output enable 0 | 0: Disables the analog output on channel 0 (DA0).<br>1: Enables the channel 0 D/A conversion; enables the analog output (DA0). | R/W |
| 5 to 0 | — | Reserved | These bits are always read as 0 and cannot be modified. | — |

Note: In standby mode or module standby mode, the contents of DACR are retained.

- DAOE1 bit and DAOE0 bit (D/A output enable 1 and 0)

  These bits control the D/A conversion and analog output.

  The event link function can be used to set the DAOE[1:0] bits. When the event specified in ELSR31 or ELSR32 of the ELC occurs, the corresponding DAOE1 or DAOE0 bit is set to 1, respectively and the D/A conversion starts.

chanhvinh.com



## 25.3 Operation

The D/A converter includes D/A conversion circuits for two channels, each of which can operate independently. When DAOE bit in DACR is set to 1, D/A conversion is enabled and the conversion result is output. The operation example of D/A conversion on channel 0 is as follows. Figure 25.2 shows the timing of this operation.

1. Write the conversion data to DADR0.
2. Set the DAOE0 bit in DACR to 1 to start D/A conversion. The conversion result is output from the analog output pin DA0 after the conversion time $t_{DCONV}$ has elapsed. The conversion result is continued to be output until DADR0 is written to again or the DAOE0 bit is cleared to 0. The output value is expressed by the following formula:

$$\frac{\text{DADR contents}}{256} \times \text{AVcc}$$

3. If DADR0 is written to again, the conversion is immediately started. The conversion result is output after the conversion time $t_{DCONV}$ has elapsed.
4. If the DAOE0 bit is cleared to 0, analog output is disabled.

chanhvinh.com



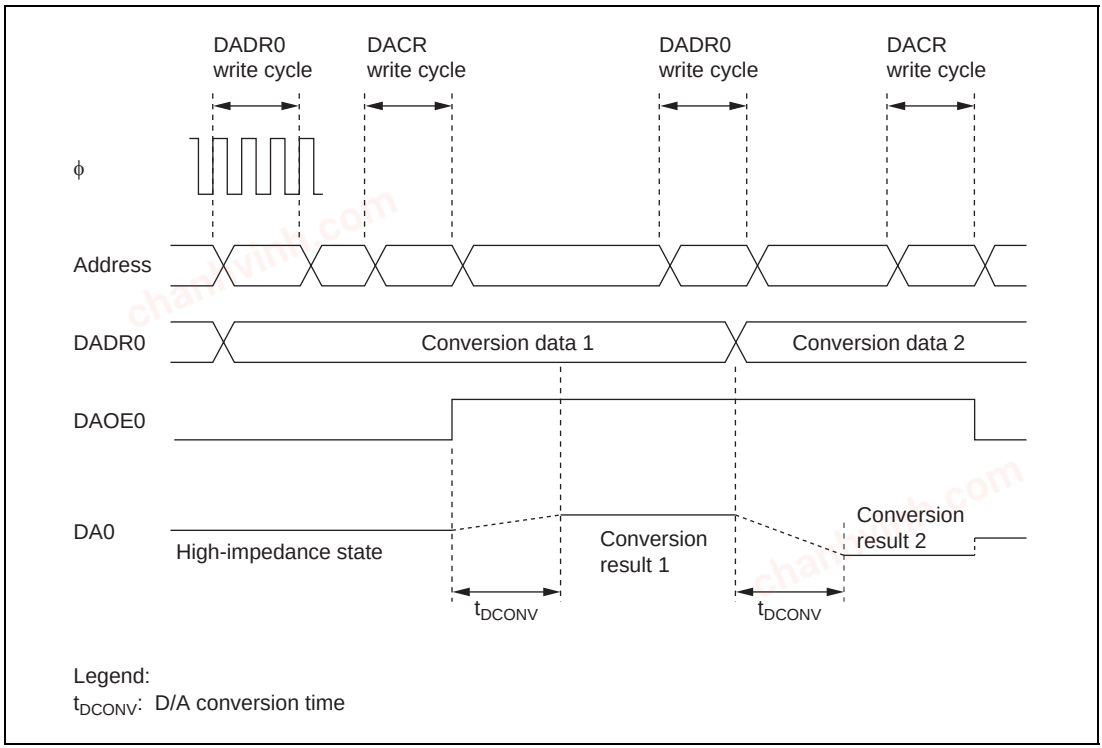



**Figure 25.2 Example of D/A Converter Operation**

chanhvinh.com



## 25.4 Usage Notes

### 25.4.1 Setting for Module Stop Mode

The module standby control can select to enable/disable the D/A converter operation. The D/A converter does not operate by the initial value of the register. The register can be accessed by releasing the module standby mode. DADR is initialized in module standby.

### 25.4.2 Operation in Standby Mode

If D/A conversion is enabled and this LSI enters standby mode, DADR is initialized while DACR is held. When the analog power supply current is required to go low in software standby mode, the DAOE1 and DAOE0 bits should be cleared to 0, and D/A output should be disabled.

chanhvinh.com



# Section 26 Low-Voltage Detection Circuits

This microcontroller includes a low-voltage detection module consisting of three circuits, LVD0, LVD1, and LVD2.

The circuits are used to prevent abnormal operation (runaway execution) from occurring due to the power supply voltage falling and to recreate the state of the microcontroller before the power supply voltage fell when the power supply voltage rises again.

If the power supply voltage falls below a threshold voltage set by the users application, a warning can be given to the application so the application can shutdown in a controlled manner. If the power supply voltage continues to fall below a second programmable threshold voltage, the device can be safely placed in the reset state. This avoids the situation where the power supply voltage falls below the guaranteed operating voltage and the microcontroller enters an unstable state. Thus, system stability can be improved. If the power supply voltage rises again, active mode is automatically entered.

The circuits monitor the power-supply voltage, and generate a reset or an interrupt when the voltage falls below or rises above a specified value.

Figure 26.1 is a block diagram of the low-voltage detection circuits. Figures 26.2, 26.3, and 26.4 are block diagrams of the LVD2, LVD1 and LVD0 interrupt/reset generation circuits, respectively.

chanhvinh.com



## 26.1 Features

- Power-on reset function

  Monitors the power-supply voltage input to the VCC pin to generate an internal reset signal when power is first supplied.

  Releases a reset when the power-supply voltage rises above the specified voltage.

- Low-voltage detection function

  Reset function: Monitors the power-supply voltage, and generates an internal reset signal when the voltage falls below a specified value.

  Interrupt function: Monitors the power-supply voltage, and generates an interrupt when the voltage falls below or rises above respective specified values.

  Detection levels: Nine levels are available for LVD1 and two levels are available for LVD0. (The level is fixed for LVD2.

  External voltage input function: For detection, external pins can be selected for a detection voltage and a reference voltage, respectively (LVD2 only).

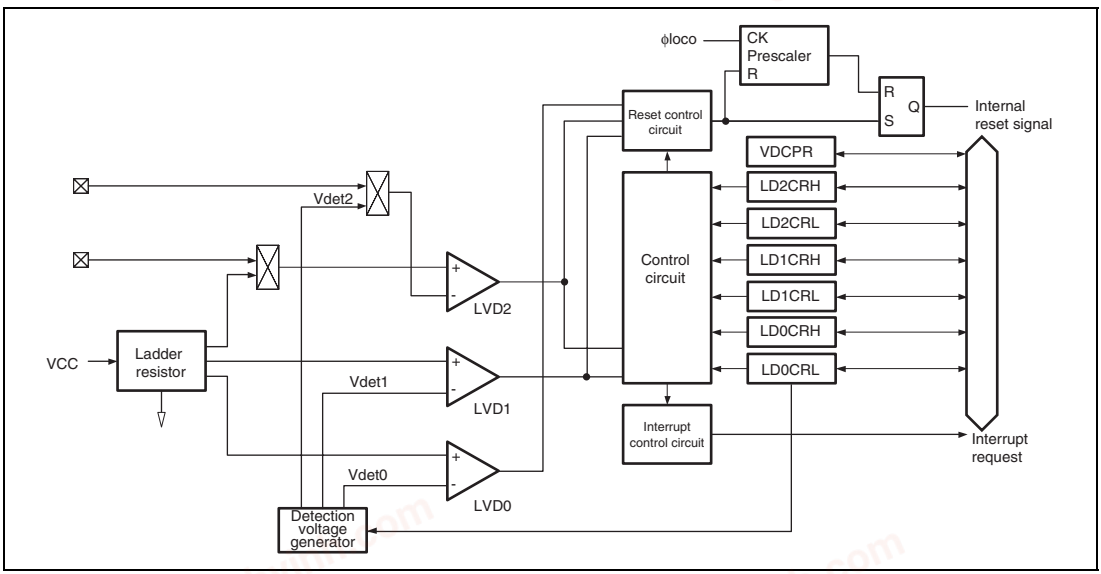



**Figure 26.1 Block Diagram of Low-Voltage Detection Circuits**

chanhvinh.com



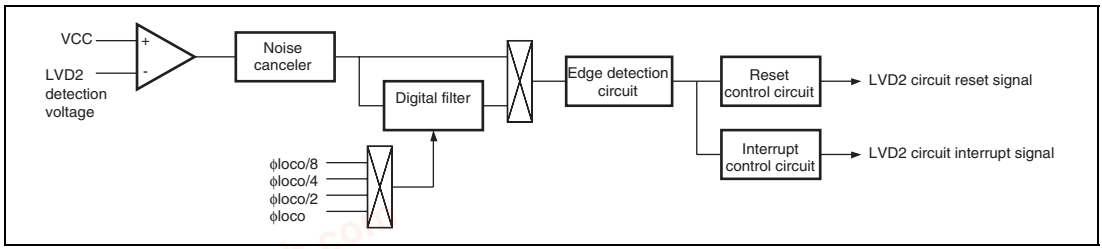



**Figure 26.2 Block Diagram of LVD2 Interrupt/Reset Generation Circuit**

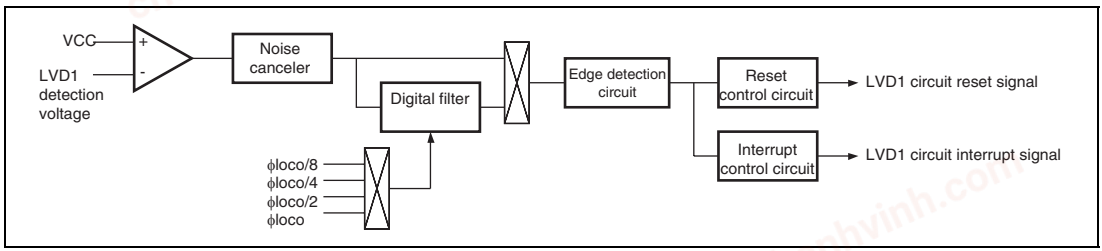



**Figure 26.3 Block Diagram of LVD1 Interrupt/Reset Generation Circuit**

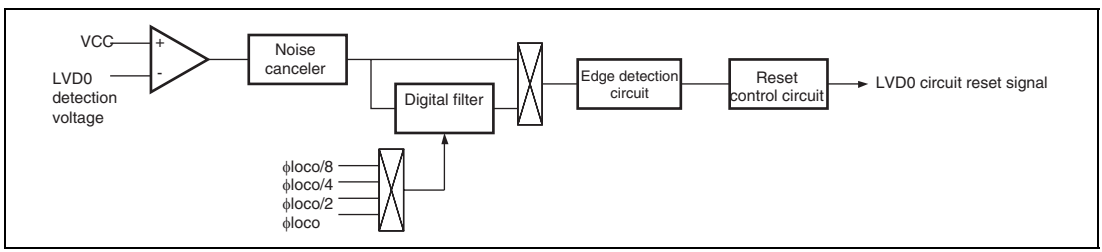



**Figure 26.4 Block Diagram of LVD0 Reset Generation Circuit**

chanhvinh.com



## 26.2 Register Descriptions

This module has the following registers.

- Low-voltage detection circuit control protect register (VDCPR)
- Low-voltage detection circuit 2 control register H (LD2CRH)
- Low-voltage detection circuit 2 control register L (LD2CRL)
- Low-voltage detection circuit 1 control register H (LD1CRH)
- Low-voltage detection circuit 1 control register L (LD1CRL)
- Low-voltage detection circuit 0 control register H (LD0CRH)
- Low-voltage detection circuit 0 control register L (LD0CRL)

chanhvinh.com
chanhvinh.com
chanhvinh.com

chanhvinh.com



### 26.2.1 Low-Voltage Detection Circuit Control Protect Register (VDCPR)

Address: H'FF0628

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | WRI | — | — | — | — | — | — | LDPRC |
| Value after reset: | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | WRI | VDCPR write disable | 0: Writing to the VDCPR bit is enabled.<br>1: Writing to the VDCPR bit is disabled. | W |
| 6 to 1 | — | Reserved | These bits are read as 0. The write value should always be 0. | — |
| 0 | LDPRC | Low-voltage detection circuit control register write enable | 0: Writing to each low-voltage detection circuit control register is disabled.<br>1: Writing to each low-voltage detection circuit control register is enabled. | R/W |

Note: Use a MOV instruction to modify this register.

- WRI bit (VDCPR write disable)

  Only when this bit is written to 0, writing to this register is enabled. This bit is always read as 1.

- LDPRC bit (Low-voltage detection circuit control register write enable)

  Only when the value of this bit is 1, writing to the low-voltage detection circuit control register (LD2CRH, LD2CRL, LD1CRH, LD1CRL, LD0CRH and LD0CRL) is enabled.

chanhvinh.com



### 26.2.2 Low-Voltage Detection Circuit 2 Control Register H (LD2CRH)

Address: H'FF0622

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | VD2DF | VD2UF | VD2DFCK[1:0] | | VD2DFS | VD2IRCS | VD2MS | VD2RE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | VD2DF | LVD2 power supply voltage drop flag | [Setting condition]<br>• When the power-supply voltage falls below Vdet2.<br>[Clearing conditions]<br>• When 1 is read from this bit and then 0 is written to this bit.<br>• When the LVD2 circuit is in standby mode. | R/W |
| 6 | VD2UF | LVD2 power supply voltage rise flag | [Setting condition]<br>• When the power supply voltage falls below Vdet2 and rises to Vdet2 or higher before falling to Vdet0 or lower.<br>[Clearing conditions]<br>• When 1 is read from this bit and then 0 is written to this bit.<br>• When the LVD2 circuit is in standby mode. | R/W |
| 5, 4 | VD2DFCK[1:0] | LVD2 digital filter sampling clock select | 00: φloco/1<br>01: φloco/2<br>10: φloco/4<br>11: φloco/8 | R/W |
| 3 | VD2DFS | LVD2 digital filter function select | 0: Disables the digital filter function<br>1: Enables the digital filter function | R/W |
| 2 | VD2IRCS | LVD2 interrupt request generation condition select | 0: when VCC rises to Vdet2 or higher.<br>1: when VCC falls to Vdet2 or lower.<br>When VD2DFS = 1, an interrupt request is generated when the voltage reaches Vdet2, regardless of this bit setting.<br>When VD2MS = 1, a reset request is generated when the voltage falls below Vdet2, regardless of this bit setting. | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 1 | VD2MS | LVD2 mode select | 0: Generates an interrupt request when the voltage reaches Vdet2.<br>1: Generates a reset request when the voltage reaches Vdet2. | R/W |
| 0 | VD2RE | LVD2 interrupt/ reset request enable | This bit is enabled when the VD2E bit is 1.<br>0: Disables interrupt/reset requests when the specified voltage level is detected.<br>1: Enables interrupt/reset requests when the specified voltage level is detected. | R/W |

Note: This register is not initialized by an LVD2 reset and LVD1 reset.

Table 26.1 shows the relationship between LD2CRH settings and the selection function. LD2CRH should be used according to table 26.1.

**Table 26.1 LD2CRH Settings and Select Functions**

| LD2CRH Settings | | | Selection Function | | |
|---|---|---|---|---|---|
| VD2MS | VD2DFS | VD2IRCS | LVD2 Falling Reset | LVD2 Falling Interrupt | LVD2 Rising Interrupt |
| 1 | x | x | O | — | — |
| 0 | 1 | x | — | O | O |
| 0 | 0 | 1 | — | O | — |
| 0 | 0 | 0 | — | — | O |

[Legend] x: Don't care.

chanhvinh.com



### 26.2.3 Low-Voltage Detection Circuit 2 Control Register L (LD2CRL)

Address: H'FF0623

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | VD2E | VD2CVS | VD2RVS | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | VD2E | LVD2 circuit enable | 0: The LVD2 circuit is not used. (In standby mode)<br>1: The LVD2 circuit is used. | R/W |
| 6 | VD2CVS | LVD2 circuit reference voltage input select | 0: VCC voltage is used as the reference voltage.<br>1: Externally input (PA7) voltage is used as the reference voltage.* | R/W |
| 5 | VD2RVS | LVD2 circuit detection voltage input select | 0: Internally generated voltage is used as the detection voltage.<br>1: Externally input (PA6) voltage is used as the detection voltage.* | R/W |
| 4 to 0 | — | Reserved | These bits are read as 0. The write value should be 0. | — |

Notes: When an externally input voltage is used as the reference voltage or detection voltage, the externally input voltage must not exceed ½ Vcc.

* This register is not initialized by an LVD2 reset and LVD1 reset.

chanhvinh.com



### 26.2.4 Low-Voltage Detection Circuit 1 Control Register H (LD1CRH)

Address: H'FF0624

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | VD1DF | VD1UF | VD1DFCK[1:0] | | VD1DFS | VD1IRCS | VD1MS | VD1RE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | VD1DF | LVD1 power supply voltage drop flag | [Setting condition]<br>• When the power-supply voltage falls below Vdet1.<br>[Clearing conditions]<br>• When 1 is read from this bit and then 0 is written to.<br>• When the LVD1 circuit is in standby mode. | R/W |
| 6 | VD1UF | LVD1 power supply voltage rise flag | [Setting condition]<br>• When the power supply voltage falls below Vdet1 and rises to Vdet1 or higher before falling to Vdet0 or lower<br>[Clearing conditions]<br>• When 1 is read from this bit and then 0 is written to.<br>• When the LVD1 circuit is in standby mode. | R/W |
| 5, 4 | VD1DFCK [1:0] | LVD1 digital filter sampling clock select | 00: φloco/1<br>01: φloco/2<br>10: φloco/4<br>11: φloco/8 | R/W |
| 3 | VD1DFS | LVD1 digital filter function select | 0: Disables the digital filter function<br>1: Enables the digital filter function | R/W |
| 2 | VD1IRCS | LVD1 interrupt request generation condition select | 0: Generates an interrupt request when VCC rises to Vdet1 or more.<br>1: Generates an interrupt request when VCC falls to Vdet1 or less.<br>When VD1DFS = 1, an interrupt request is generated when the voltage reaches Vdet1, regardless of this bit setting.<br>When VD1MS = 1, a reset request is generated when the voltage falls below Vdet1, regardless of this bit setting. | R/W |

chanhvinh.com



| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 1 | VD1MS | LVD1 mode select | 0: Generates an interrupt request when the voltage reaches Vdet1.<br>1: Generates a reset request when the voltage reaches Vdet1. | R/W |
| 0 | VD1RE | LVD1 interrupt/ reset request enable | This bit is enabled when the VD1E bit is 1.<br>0: Disables interrupt/reset requests generated when the specified voltage level is detected.<br>1: Enables interrupt/reset requests generated when the specified voltage level is detected. | R/W |

Note: This register is not initialized by an LVD2 reset and LVD1 reset.

Table 26.2 shows the relationship between the LD1CRH settings and selection function. LD1CRH should be set according to table 26.2.

**Table 26.2 LD1CRH Settings and Select Functions**

| LD1CRH | | | Select Functions | | |
|---|---|---|---|---|---|
| VD1MS | VD1DFS | VD1IRCS | LVD1 Falling Reset | LVD1 Falling Interrupt | LVD1 Rising Interrupt |
| 1 | x | x | O | — | — |
| 0 | 1 | x | — | O | O |
| 0 | 0 | 1 | — | O | — |
| 0 | 0 | 0 | — | — | O |

[Legend] x: Don't care.

chanhvinh.com



### 26.2.5 Low-Voltage Detection Circuit 1 Control Register L (LD1CRL)

Address: H'FF0625

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | VD1E | — | — | — | VD1LS[3:0] | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | VD1E | LVD1 circuit enable | 0: The LVD1 circuit is not used. (In standby mode)<br>1: The LVD1 circuit is used. | R/W |
| 6 to 4 | — | Reserved | These bits are read as 0. The write value should be 0. | — |
| 3 to 0 | VD1LS [3:0] | LVD1 detection voltage level select 3 to 0 | 0000: Setting prohibited<br>0001: Setting prohibited<br>0010: Setting prohibited<br>0011: Setting prohibited<br>0100: Setting prohibited<br>0101: Setting prohibited<br>0110: Setting prohibited<br>0111: 3.07 V (typ.)<br>1000: 3.22 V (typ.)<br>1001: 3.37 V (typ.)<br>1010: 3.52 V (typ.)<br>1011: 3.67 V (typ.)<br>1100: 3.82 V (typ.)<br>1101: 3.97 V (typ.)<br>1110: 4.12 V (typ.)<br>1111: 4.27 V (typ.) | R/W |

Note: This register is not initialized by an LVD2 reset and LVD1 reset.

chanhvinh.com



### 26.2.6 Low-Voltage Detection Circuit 0 Control Register H (LD0CRH)

Address: H'FF0626

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | VD0DFCK[1:0] | | VD0DFS | — | — | — |
| Value after reset: | — | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is read as undefined value. The write value should be 0. | — |
| 6 | — | Reserved | This bit is read as 0. The write value should be 0. | — |
| 5, 4 | VD0DFCK[1:0] | LVD0 digital filter sampling clock select | 00: ϕloco/1<br>01: ϕloco/2<br>10: ϕloco/4<br>11: ϕloco/8 | R/W |
| 3 | VD0DFS | LVD0 digital filter function select | 0: Disables the digital filter function.<br>1: Enables the digital filter function. | R/W |
| 2, 1 | — | Reserved | These bits are read as 0. The write value should always be 0. | — |
| 0 | — | Reserved | This bit is read as 1. The write value should be 1. | — |

Note: This register is not initialized by an LVD2 reset and LVD1 reset.

chanhvinh.com



### 26.2.7 Low-Voltage Detection Circuit 0 Control Register L (LD0CRL)

Address: H'FF0627

| Bit: | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | VD0LS1 | — |
| Value after reset: | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 | — | Reserved | This bit is read as 1. The write value should be 1. | — |
| 6 to 2 | — | Reserved | This bit is read as 0. The write value should be 0. | — |
| 1 | VD0LS1 | LVD0 detection voltage level select | 0: 2.35 V (typ.)<br>1: 3.80 V (typ.) | R/W |
| 0 | — | Reserved | The write value should always be 0. | — |

Note: This register is not initialized by an LVD2 reset and LVD1 reset.

chanhvinh.com



## 26.3 Operation

### 26.3.1 Power-On Reset Function

Figure 26.5 shows the operation timing of the power-on reset function. During the power-on reset function, the LVD0 circuit monitors the power-supply voltage level to initialize the entire chip.

When the power-supply voltage level rises above Vdet0, the prescaler is released from its reset state and it starts counting. The OVF signal is generated to release the internal reset signal after the prescaler has counted 128 ϕloco cycles. After a power-on reset, the LVD0 reset function is always enabled.

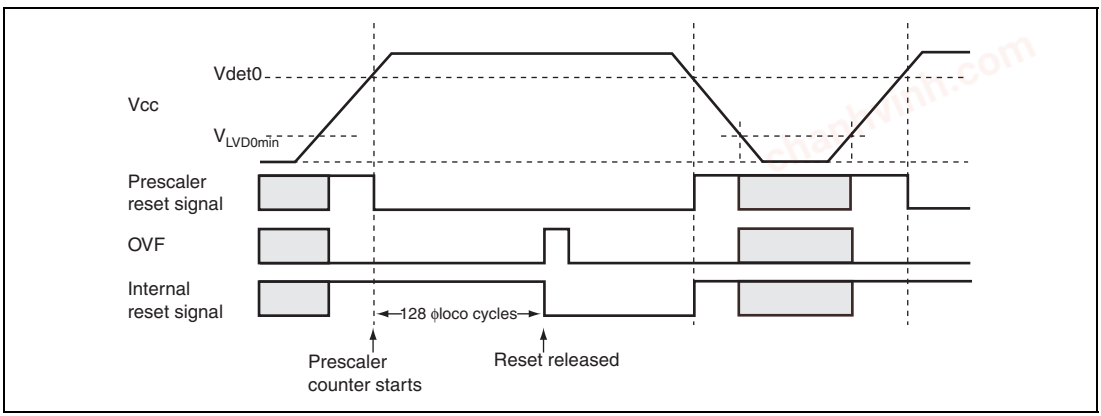



**Figure 26.5 Operational Timing of Power-On Reset**

chanhvinh.com



### 26.3.2 Low-Voltage Detection Circuit

#### (1) Low Voltage Detect Reset 2 (LVDR2)

LVDR2 is a reset generated by the LVD2 circuit. Figure 26.6 shows the operation timing of the LVDR2.

The LVD2 enters the module-standby state after release from a power-on reset. To operate the LVDR2, set the VD2E bit in LD2CRL to 1, wait for 50 µs ($t_{d(E-A)}$) until the detection voltage and the low-voltage detection circuit 2 operation have stabilized using a software timer, etc., then set the VD2MS and VD2RE bits in LD2CRH to 1. After that, the output settings of I/O ports must be made. To cancel the LVDR2, first the VD2RE bit in LD2CRH should be cleared to 0 and then the VD2E bit in LD2CRL should be cleared to 0. Figure 26.7 shows the procedure to set the LVDR2.

When the power-supply voltage falls below Vdet2, the LVDR2 clears the $\overline{\text{LVDRES2}}$ signal to 0, and resets the prescaler. The low-voltage detection reset state remains in place until a power-on reset is generated. When the power-supply voltage rises above the Vdet2 voltage again, the prescaler starts counting. It counts 32 ϕloco cycles, and then releases the internal reset signal.

Note that if the power supply voltage falls below $V_{LVD2min}$ = 2.7 V and then rises from that point, the LVDR2 may not occur. Such a case should be evaluated thoroughly.

If the power supply voltage falls below Vdet0, a power-on reset occurs.

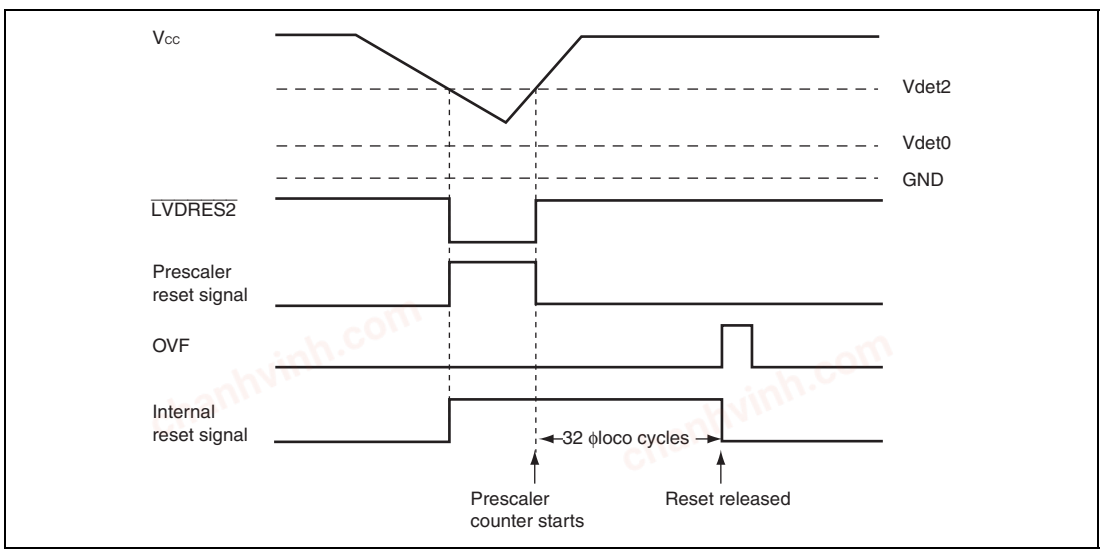



**Figure 26.6 Operation Timing of LVDR2**

chanhvinh.com



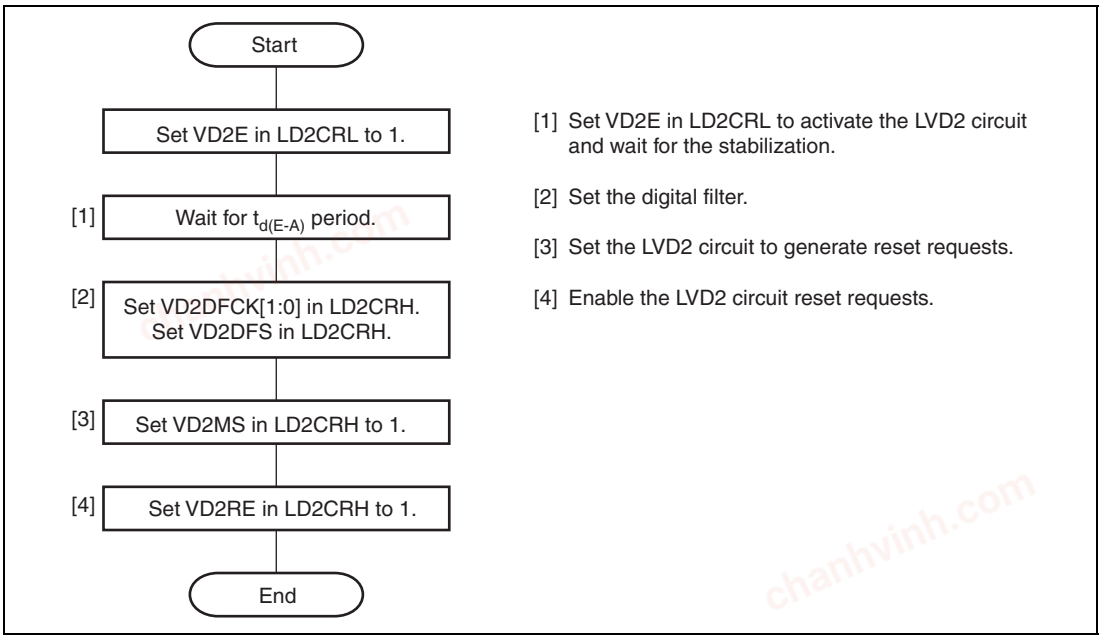



**Figure 26.7 Procedure to Set LVDR2**

chanhvinh.com



### (2) Low Voltage Detect Interrupt 2 (LVDI2)

LVDI2 is an interrupt generated by the LVD2 circuit. Figure 26.8 shows the operation timing of LVDI2.

The LVD2 enters the module-standby state after release from a power-on reset. To operate the LVDI2, set the VD2E bit in LD2CRL to 1, wait for 50 µs ($t_{d(E-A)}$) until the detection voltage and the low-voltage detection circuit 2 operation have stabilized using a software timer, etc., then clear the VD2MS bit to 0 and set the VD2RE bit to 1 in LD2CRH. After that, the output settings of I/O ports must be made. To cancel the LVDI2, first the VD2RE bit in LD2CRH should be cleared to 0 and then the VD2E bit in LD2CRL should be cleared to 0. Figure 26.9 shows the procedure to set the LVDI2.

When the power-supply voltage falls below Vdet2, the LVDI2 clears the $\overline{\text{LVDINT2}}$ signal to 0 and the VD2DFS bit in LD2CRH is set to 1. If the VD2DFS or VD2IRCS bit in LD2CRH is 1 at this time, an LVD2 interrupt request is simultaneously generated. In this case, the necessary data must be saved in the on-chip flash memory area or external EEPROM, etc, and a transition must be made to standby mode or sleep mode. Until this processing is completed, the power supply voltage must be higher than the lower limit of the guaranteed operating voltage.

When the power-supply voltage does not fall below Vdet0 but rises above Vdet2, the LVDI2 sets the $\overline{\text{LVDINT2}}$ signal to 1 and set the VD2UF bit in LD2CRH to 1. If the VD2DFS bit in LD2CRH is 1 or the VD2IRCS bit in LD2CRH is 0 at this time, an LVD2 interrupt request is simultaneously generated.

If the power supply voltage falls below Vdet0, a power-on reset occurs.

chanhvinh.com



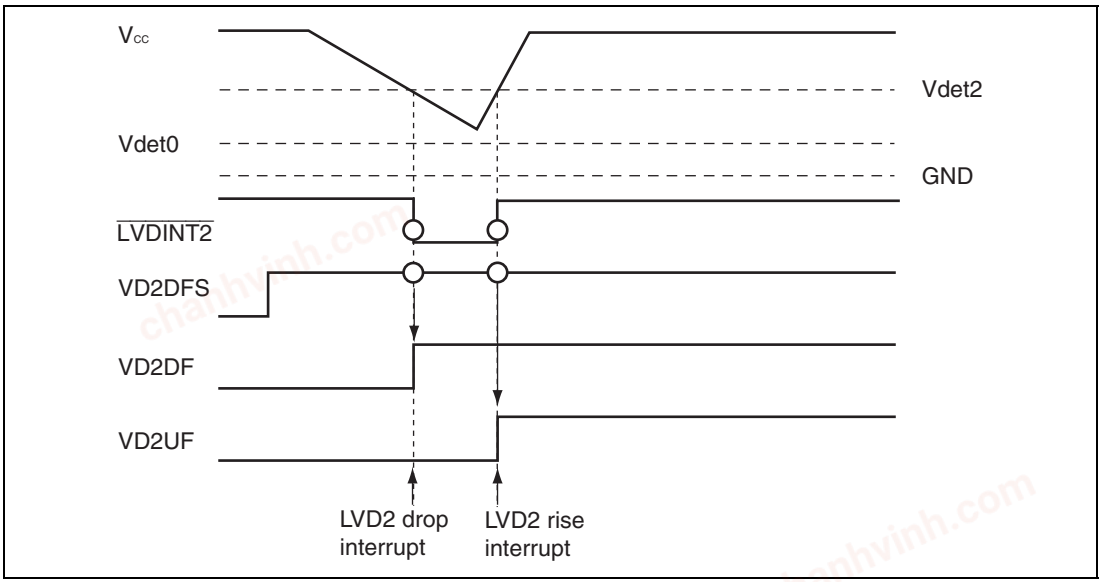



**Figure 26.8 Operation Timing of LVDI2**

chanhvinh.com



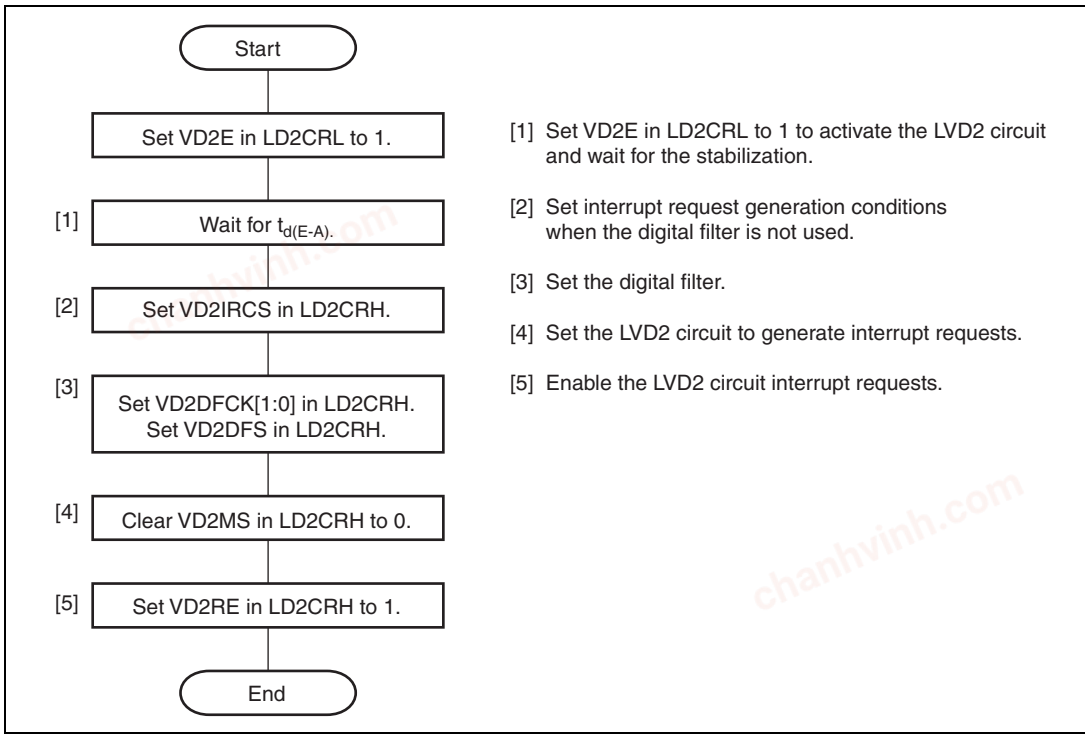



**Figure 26.9 Procedure to Set LVDI2**

chanhvinh.com



### (3) Low Voltage Detect Reset 1 (LVDR1)

LVDR1 is a reset generated by the LVD1 circuit. Figure 26.10 shows the operation timing of the LVDR1.

The LVD1 enters the module-standby state after release from a power-on reset. To operate the LVDR1, set the VD1E bit in LD1CRL to 1, wait for 50 µs ($t_{d(E-A)}$) until the detection voltage and the low-voltage detection circuit 1 operation have stabilized using a software timer, etc., then set the VD1MS and VD1RE bits in LD1CRH to 1. After that, the output settings of I/O ports must be made. To cancel the LVDR1, first the VD1RE bit in LD1CRH should be cleared to 0 and then the VD1E bit in LD1CRL should be cleared to 0. Figure 26.11 shows the procedure to set the LVDR1.

When the power-supply voltage falls below Vdet1, the LVDR1 clears the $\overline{\text{LVDRES1}}$ signal to 0, and resets the prescaler. The low-voltage detection reset state remains in place until a power-on reset is generated. When the power-supply voltage rises above the Vdet1 voltage again, the prescaler starts counting. It counts 32 ϕloco cycles, and then releases the internal reset signal.

Note that if the power supply voltage falls below $V_{LVD1min}$ = 2.7 V and then rises from that point, the LVDR1 may not occur. Such a case should be evaluated thoroughly.

If the power supply voltage falls below Vdet0, a power-on reset occurs.

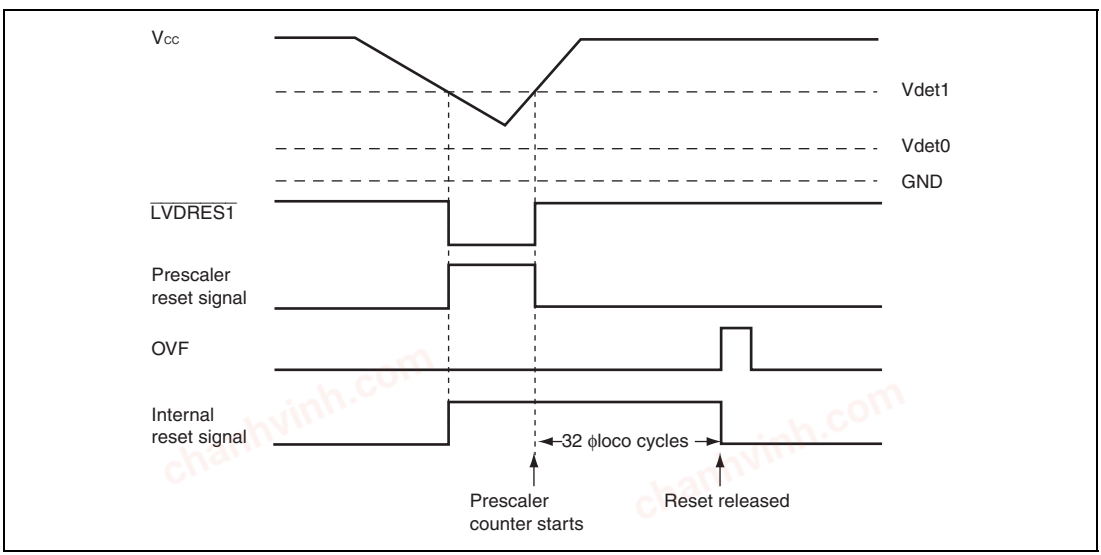



**Figure 26.10 Operation Timing of LVDR1**

chanhvinh.com



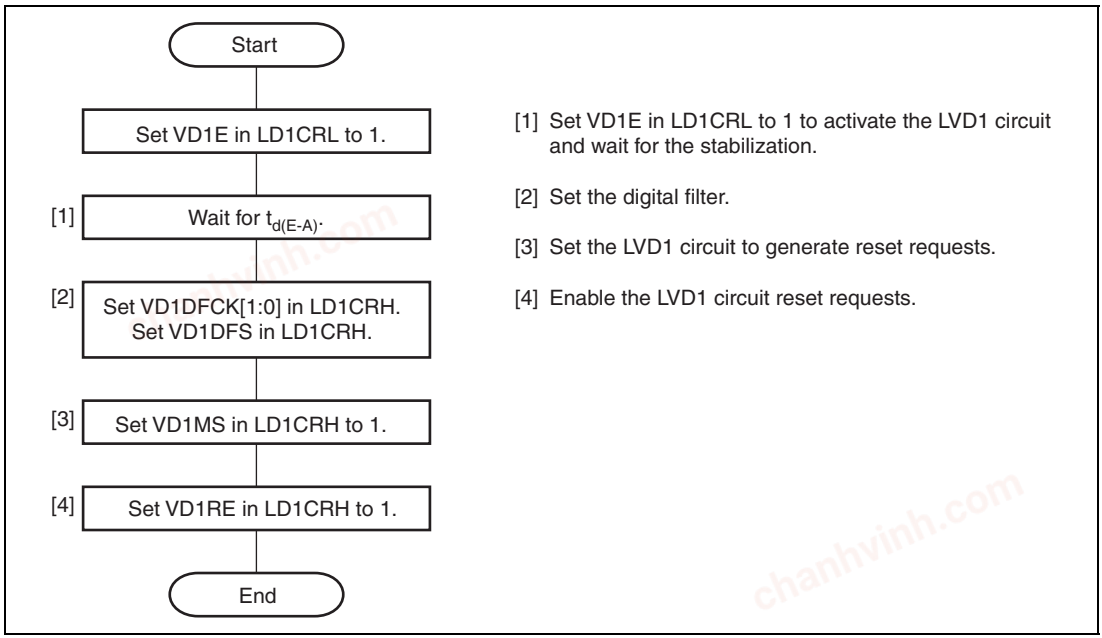



**Figure 26.11 Procedure to Set LVDR1**

chanhvinh.com



### (4) Low Voltage Detect Interrupt 1 (LVDI1)

LVDI1 is an interrupt generated by the LVD1 circuit. Figure 26.12 shows the operation timing of LVDI1.

The LVD1 enters the module-standby state after release from a power-on reset is canceled. To operate the LVDI1, set the VD1E bit in LD1CRL to 1, wait for 50 µs ($t_{d(E-A)}$) until the detection voltage and the low-voltage detection circuit 1 operation have stabilized using a software timer, etc., then clear the VD1MS bit to 0 and set the VD1RE bit to 1 in LD1CRH. After that, the output settings of I/O ports must be made. To cancel the LVDI1, first the VD1RE bit in LD1CRH should be cleared to 0 and then the VD1E bit in LD1CRL should be cleared to 0. Figure 26.13 shows the procedure to set the LVDI1.

When the power-supply voltage falls below Vdet1, the LVDI1 clears the $\overline{\text{LVDINT1}}$ signal to 0 and the VD1DFS bit in LD1CRH is set to 1. If the VD1DFS or VD1IRCS bit in LD1CRH is 1 at this time, an LVD1 interrupt request is simultaneously generated. In this case, the necessary data must be saved in the on-chip flash memory area or external EEPROM, etc, and a transition must be made to standby mode or sleep mode. Until this processing is completed, the power supply voltage must be higher than the lower limit of the guaranteed operating voltage.

When the power-supply voltage does not fall below Vdet0 but rises above Vdet1, the LVDI2 sets the $\overline{\text{LVDINT1}}$ signal to 1 and set the VD1UF bit in LD2CRH to 1. If the VD1DFS bit in LD1CRH is 1 or the VD1IRCS bit in LD1CRH is 0 at this time, an LVD1 interrupt request is simultaneously generated.

If the power supply voltage falls below Vdet0, a power-on reset occurs.

chanhvinh.com



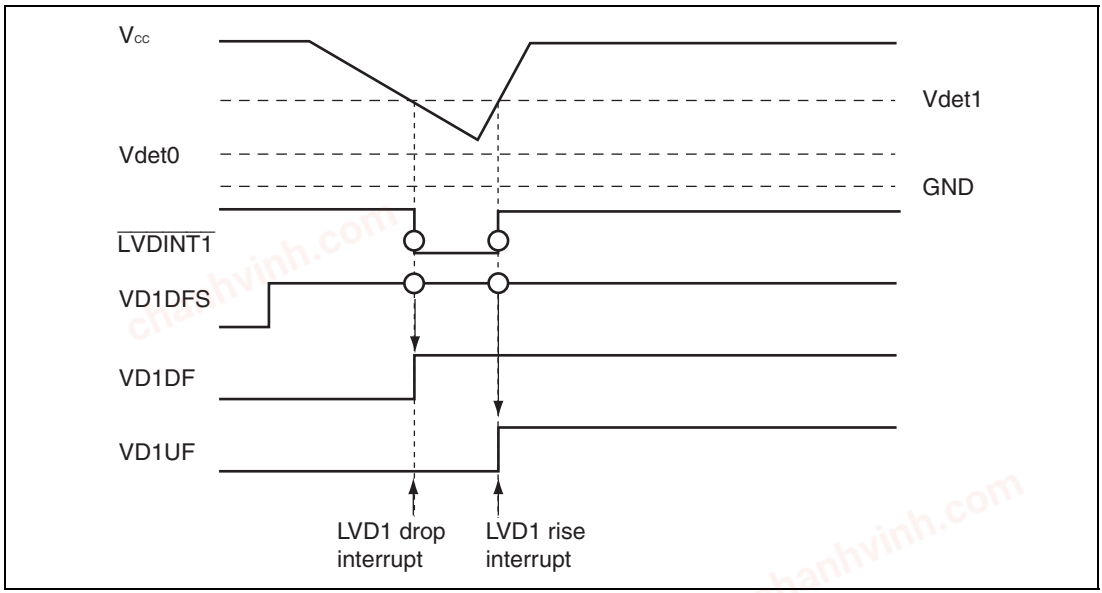



**Figure 26.12 Operational Timing of LVDI1**

chanhvinh.com



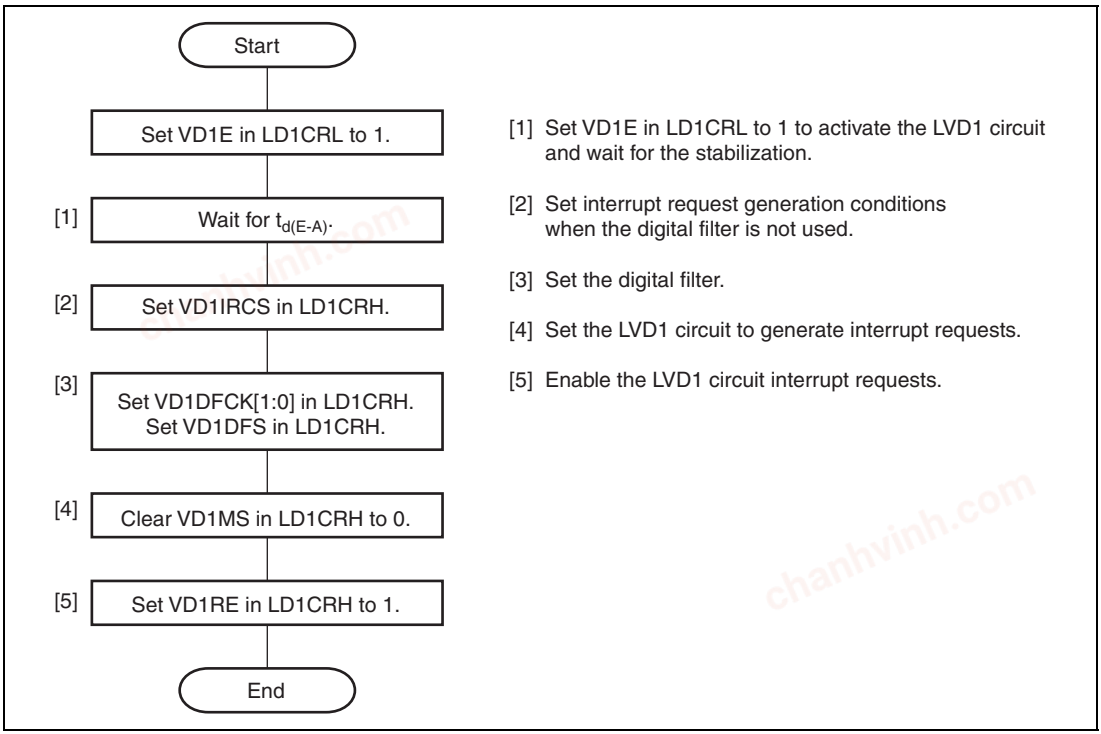



**Figure 26.13 Procedure to Set LVDI1**

chanhvinh.com



### (5) Low Voltage Detect Reset 0 (LVDR0)

LVDR0 is a reset generated by the LVD0 circuit. Figure 26.14 shows the operation timing of the LVDR0.

After a power-on reset is released, the LVD0 circuit is always enabled.

When the power-supply voltage falls below Vdet0, the LVDR0 clears the $\overline{\text{LVDRES0}}$ signal to 0, and resets the prescaler, and a power-on reset operation is enabled. When the power-supply voltage rises above the Vdet0 voltage again, the prescaler starts counting. It counts 128 ϕloco cycles, and then releases the internal reset signal.

Note that if the power supply voltage falls below $V_{LVD0min}$ = 1.8 V and then rises from that point, the LVDR0 may not occur. Such a case should be evaluated thoroughly.

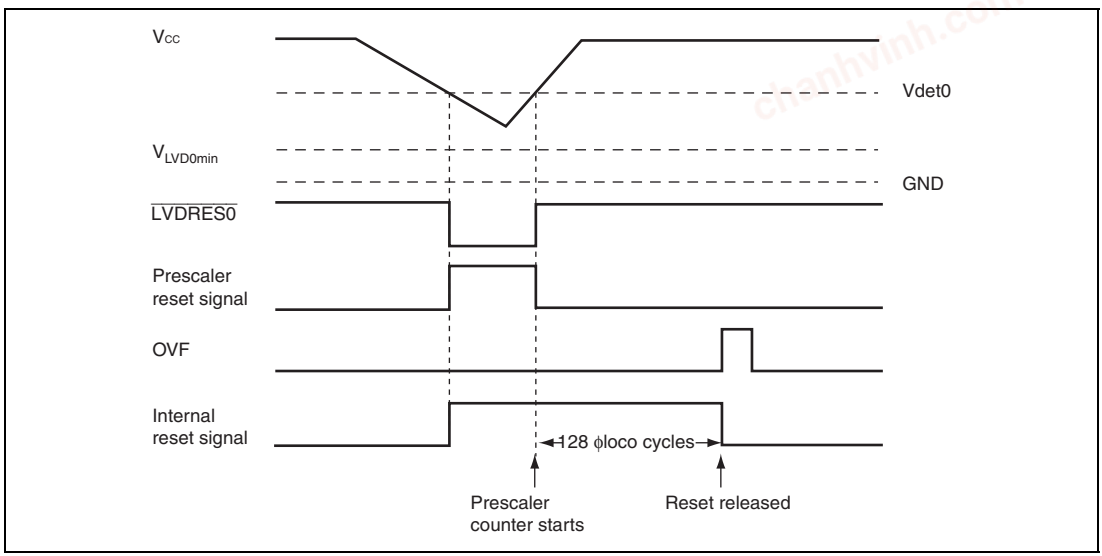



**Figure 26.14 Operation Timing of LVDR0**

chanhvinh.com



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

chanhvinh.com



# Section 27 List of Registers

The address list gives information on the on-chip I/O register addresses, how the register bits are configured, and the register states in each operation mode. The information is given as shown below.

1. Register Addresses (address order)

- Registers are listed from the lower allocation addresses.
- Registers are classified by functional modules.
- The data bus width is indicated.
- The number of access states is indicated.

2. Register Bits

- Bit configurations of the registers are described in the same order as the register addresses (address order).
- Reserved bits are indicated by — in the bit name column.
- When registers consist of 16 bits, bits are described from the MSB side.

3. Register States in Each Operating Mode

- Register states are described in the same order as the register addresses (address order).
- The register states described here are for the basic operating modes. If there is a specific reset for an on-chip peripheral module, see the section on that on-chip peripheral module.

chanhvinh.com



## 27.1 Register Addresses (Address Order)

The data-bus width column indicates the number of bits. The access-state column shows the number of states of the selected basic clock that is required for access to the register.

Note: Do not attempt to access undefined or reserved addresses. Correct operation of the access itself or later operations is not guaranteed when such an address is accessed.

| Register Name | Abbreviation | Number of Bits | Address | Module | Data Bus Width | Number of Access States |
|---|---|---|---|---|---|---|
| Port mode register 1 | PMR1 | 8 | H'FF0000 | I/O port | 8 | 2 |
| Port mode register 2 | PMR2 | 8 | H'FF0001 | I/O port | 8 | 2 |
| Port mode register 3 | PMR3 | 8 | H'FF0002 | I/O port | 8 | 2 |
| Port mode register 5 | PMR5 | 8 | H'FF0004 | I/O port | 8 | 2 |
| Port mode register 6 | PMR6 | 8 | H'FF0005 | I/O port | 8 | 2 |
| Port mode register 8 | PMR8 | 8 | H'FF0007 | I/O port | 8 | 2 |
| Port mode register 9*[1] | PMR9 | 8 | H'FF0008 | I/O port | 8 | 2 |
| Port mode register A | PMRA | 8 | H'FF0009 | I/O port | 8 | 2 |
| IIC/SSU select register | ICSUSR | 8 | H'FF000B | IIC/SSU | 8 | 2 |
| Port mode register J | PMRJ | 8 | H'FF000C | I/O port | 8 | 2 |
| Port pull-up control register 1 | PUCR1 | 8 | H'FF0010 | I/O port | 8 | 2 |
| Port pull-up control register 2 | PUCR2 | 8 | H'FF0011 | I/O port | 8 | 2 |
| Port pull-up control register 3 | PUCR3 | 8 | H'FF0012 | I/O port | 8 | 2 |
| Port pull-up control register 5 | PUCR5 | 8 | H'FF0014 | I/O port | 8 | 2 |
| Port pull-up control register 6 | PUCR6 | 8 | H'FF0015 | I/O port | 8 | 2 |
| Port pull-up control register 8 | PUCR8 | 8 | H'FF0017 | I/O port | 8 | 2 |
| Port pull-up control register 9*[1] | PUCR9 | 8 | H'FF0018 | I/O port | 8 | 2 |
| Port pull-up control register A | PUCRA | 8 | H'FF0019 | I/O port | 8 | 2 |
| Port pull-up control register B | PUCRB | 8 | H'FF001A | I/O port | 8 | 2 |
| Port pull-up control register J | PUCRJ | 8 | H'FF001C | I/O port | 8 | 2 |
| Port drive control register 1 | PDVR1 | 8 | H'FF0030 | I/O port | 8 | 2 |
| Port drive control register 2 | PDVR2 | 8 | H'FF0031 | I/O port | 8 | 2 |
| Port drive control register 3 | PDVR3 | 8 | H'FF0032 | I/O port | 8 | 2 |

chanhvinh.com



| Register Name | Abbreviation | Number of Bits | Address | Module | Data Bus Width | Number of Access States |
|---|---|---|---|---|---|---|
| Port drive control register 5 | PDVR5 | 8 | H'FF0034 | I/O port | 8 | 2 |
| Port drive control register 6 | PDVR6 | 8 | H'FF0035 | I/O port | 8 | 2 |
| Port drive control register 8 | PDVR8 | 8 | H'FF0037 | I/O port | 8 | 2 |
| Port drive control register 9*[1] | PDVR9 | 8 | H'FF0038 | I/O port | 8 | 2 |
| Port 1 peripheral function mapping register 1 | PMCR11 | 8 | H'FF0040 | PMC | 8 | 2 |
| Port 1 peripheral function mapping register 2 | PMCR12 | 8 | H'FF0041 | PMC | 8 | 2 |
| Port 1 peripheral function mapping register 3 | PMCR13 | 8 | H'FF0042 | PMC | 8 | 2 |
| Port 1 peripheral function mapping register 4 | PMCR14 | 8 | H'FF0043 | PMC | 8 | 2 |
| Port 2 peripheral function mapping register 1 | PMCR21 | 8 | H'FF0044 | PMC | 8 | 2 |
| Port 2 peripheral function mapping register 2 | PMCR22 | 8 | H'FF0045 | PMC | 8 | 2 |
| Port 2 peripheral function mapping register 3 | PMCR23 | 8 | H'FF0046 | PMC | 8 | 2 |
| Port 2 peripheral function mapping register 4 | PMCR24 | 8 | H'FF0047 | PMC | 8 | 2 |
| Port 3 peripheral function mapping register 1 | PMCR31 | 8 | H'FF0048 | PMC | 8 | 2 |
| Port 3 peripheral function mapping register 2 | PMCR32 | 8 | H'FF0049 | PMC | 8 | 2 |
| Port 3 peripheral function mapping register 3 | PMCR33 | 8 | H'FF004A | PMC | 8 | 2 |
| Port 3 peripheral function mapping register 4 | PMCR34 | 8 | H'FF004B | PMC | 8 | 2 |
| Port 5 peripheral function mapping register 1 | PMCR51 | 8 | H'FF0050 | PMC | 8 | 2 |
| Port 5 peripheral function mapping register 2 | PMCR52 | 8 | H'FF0051 | PMC | 8 | 2 |
| Port 5 peripheral function mapping register 3 | PMCR53 | 8 | H'FF0052 | PMC | 8 | 2 |
| Port 5 peripheral function mapping register 4 | PMCR54 | 8 | H'FF0053 | PMC | 8 | 2 |

chanhvinh.com



| Register Name | Abbreviation | Number of Bits | Address | Module | Data Bus Width | Number of Access States |
|---|---|---|---|---|---|---|
| Port 6 peripheral function mapping register 1 | PMCR61 | 8 | H'FF0054 | PMC | 8 | 2 |
| Port 6 peripheral function mapping register 2 | PMCR62 | 8 | H'FF0055 | PMC | 8 | 2 |
| Port 6 peripheral function mapping register 3 | PMCR63 | 8 | H'FF0056 | PMC | 8 | 2 |
| Port 6 peripheral function mapping register 4 | PMCR64 | 8 | H'FF0057 | PMC | 8 | 2 |
| Port 8 peripheral function mapping register 3 | PMCR83 | 8 | H'FF005E | PMC | 8 | 2 |
| Port 8 peripheral function mapping register 4 | PMCR84 | 8 | H'FF005F | PMC | 8 | 2 |
| Port 9 peripheral function mapping register 1*[1] | PMCR91 | 8 | H'FF0060 | PMC | 8 | 2 |
| Port 9 peripheral function mapping register 2*[1] | PMCR92 | 8 | H'FF0061 | PMC | 8 | 2 |
| Port 9 peripheral function mapping register 3*[1] | PMCR93 | 8 | H'FF0062 | PMC | 8 | 2 |
| Port 9 peripheral function mapping register 4*[1] | PMCR94 | 8 | H'FF0063 | PMC | 8 | 2 |
| Peripheral function mapping register write-protect register | PMCWPR | 8 | H'FF0065 | PMC | 8 | 2 |
| Port A peripheral function mapping register 3*[2] | PMCRA3 | 8 | H'FF0066 | PMC | 8 | 2 |
| Port A peripheral function mapping register 4*[2] | PMCRA4 | 8 | H'FF0067 | PMC | 8 | 2 |
| LIN control register | LINCR | 8 | H'FF0518 | HW–LIN | 8 | 2 |
| LIN status register | LINST | 8 | H'FF0519 | HW–LIN | 8 | 2 |
| Interrupt control register | INTCR | 8 | H'FF0520 | Interrupt | 8 | 2 |
| IRQ enable register | IER | 8 | H'FF0521 | Interrupt | 8 | 2 |
| IRQ sense control register H | ISCRH | 8 | H'FF0522 | Interrupt | 8 | 2 |
| IRQ sense control register L | ISCRL | 8 | H'FF0523 | Interrupt | 8 | 2 |
| IRQ status register | ISR | 8 | H'FF0524 | Interrupt | 8 | 2 |
| IRQ noise canceler control register | INCCR | 8 | H'FF0525 | Interrupt | 8 | 2 |

chanhvinh.com



| Register Name | Abbreviation | Number of Bits | Address | Module | Data Bus Width | Number of Access States |
|---|---|---|---|---|---|---|
| Interrupt vector offset register | VOFR | 16 | H'FF0526 | Interrupt | 8 | 2 |
| Event link interrupt control status register | ELCSR | 8 | H'FF0528 | Interrupt | 8 | 2 |
| Interrupt priority register A | IPRA | 8 | H'FF0529 | Interrupt | 8 | 2 |
| Interrupt priority register B | IPRB | 8 | H'FF052A | Interrupt | 8 | 2 |
| Interrupt priority register C | IPRC | 8 | H'FF052B | Interrupt | 8 | 2 |
| Interrupt priority register D | IPRD | 8 | H'FF052C | Interrupt | 8 | 2 |
| Interrupt priority register E | IPRE | 8 | H'FF052D | Interrupt | 8 | 2 |
| Interrupt priority register F | IPRF | 8 | H'FF052E | Interrupt | 8 | 2 |
| Interrupt priority register G | IPRG | 8 | H'FF052F | Interrupt | 8 | 2 |
| Interrupt priority register H | IPRH | 8 | H'FF0530 | Interrupt | 8 | 2 |
| Interrupt priority register I | IPRI | 8 | H'FF0531 | Interrupt | 8 | 2 |
| DTC enable register A | DTCERA | 8 | H'FF0534 | DTC | 8 | 2 |
| DTC enable register B | DTCERB | 8 | H'FF0535 | DTC | 8 | 2 |
| DTC enable register C | DTCERC | 8 | H'FF0536 | DTC | 8 | 2 |
| DTC enable register D | DTCERD | 8 | H'FF0537 | DTC | 8 | 2 |
| DTC enable register E | DTCERE | 8 | H'FF0538 | DTC | 8 | 2 |
| DTC enable register F*[1] | DTCERF | 8 | H'FF0539 | DTC | 8 | 2 |
| DTC enable register G | DTCERG | 8 | H'FF053A | DTC | 8 | 2 |
| DTC enable register H | DTCERH | 8 | H'FF053B | DTC | 8 | 2 |
| DTC vector register | DTVECR | 8 | H'FF053D | DTC | 8 | 2 |
| Serial mode register | SMR | 8 | H'FF0550 | SCI3 | 8 | 3 |
| Bit rate register | BRR | 8 | H'FF0551 | SCI3 | 8 | 3 |
| Serial control register 3 | SCR3 | 8 | H'FF0552 | SCI3 | 8 | 3 |
| Transmit data register | TDR | 8 | H'FF0553 | SCI3 | 8 | 3 |
| Serial status register | SSR | 8 | H'FF0554 | SCI3 | 8 | 3 |
| Receive data register | RDR | 8 | H'FF0555 | SCI3 | 8 | 3 |
| Sampling mode register | SPMR | 8 | H'FF0556 | SCI3 | 8 | 3 |
| Serial mode register_2 | SMR_2 | 8 | H'FF0558 | SCI3_2 | 8 | 3 |
| Bit rate register_2 | BRR_2 | 8 | H'FF0559 | SCI3_2 | 8 | 3 |
| Serial control register 3_2 | SCR3_2 | 8 | H'FF055A | SCI3_2 | 8 | 3 |

chanhvinh.com



| Register Name | Abbreviation | Number of Bits | Address | Module | Data Bus Width | Number of Access States |
|---|---|---|---|---|---|---|
| Transmit data register_2 | TDR_2 | 8 | H'FF055B | SCI3_2 | 8 | 3 |
| Serial status register_2 | SSR_2 | 8 | H'FF055C | SCI3_2 | 8 | 3 |
| Receive data register_2 | RDR_2 | 8 | H'FF055D | SCI3_2 | 8 | 3 |
| Sampling mode register_2 | SPMR_2 | 8 | H'FF055E | SCI3_2 | 8 | 3 |
| Serial mode register_3 | SMR_3 | 8 | H'FF0560 | SCI3_3 | 8 | 3 |
| Bit rate register_3 | BRR_3 | 8 | H'FF0561 | SCI3_3 | 8 | 3 |
| Serial control register 3_3 | SCR3_3 | 8 | H'FF0562 | SCI3_3 | 8 | 3 |
| Transmit data register_3 | TDR_3 | 8 | H'FF0563 | SCI3_3 | 8 | 3 |
| Serial status register_3 | SSR_3 | 8 | H'FF0564 | SCI3_3 | 8 | 3 |
| Receive data register_3 | RDR_3 | 8 | H'FF0565 | SCI3_3 | 8 | 3 |
| Sampling mode register_3 | SPMR_3 | 8 | H'FF0566 | SCI3_3 | 8 | 3 |
| Timer RD counter_2*[1] | TRDCNT_2 | 16 | H'FF0570 | Timer RD (unit 1, channel 2) | 16*[3] | 2 |
| General register A_2*[1] | GRA_2 | 16 | H'FF0572 | | 16*[3] | 2 |
| General register B_2*[1] | GRB_2 | 16 | H'FF0574 | | 16*[3] | 2 |
| General register C_2*[1] | GRC_2 | 16 | H'FF0576 | | 16*[3] | 2 |
| General register D_2*[1] | GRD_2 | 16 | H'FF0578 | | 16*[3] | 2 |
| Timer RD counter_3*[1] | TRDCNT_3 | 16 | H'FF057A | Timer RD (unit 1, channel 3) | 16*[3] | 2 |
| General register A_3*[1] | GRA_3 | 16 | H'FF057C | | 16*[3] | 2 |
| General register B_3*[1] | GRB_3 | 16 | H'FF057E | | 16*[3] | 2 |
| General register C_3*[1] | GRC_3 | 16 | H'FF0580 | | 16*[3] | 2 |
| General register D_3*[1] | GRD_3 | 16 | H'FF0582 | | 16*[3] | 2 |
| Timer RD control register_2*[1] | TRDCR_2 | 8 | H'FF0584 | Timer RD (unit 1, channel 2) | 8 | 2 |
| Timer RD I/O control register A_2*[1] | TRDIORA_2 | 8 | H'FF0585 | | 8 | 2 |
| Timer RD I/O control register C_2*[1] | TRDIORC_2 | 8 | H'FF0586 | | 8 | 2 |
| Timer RD status register_2*[1] | TRDSR_2 | 8 | H'FF0587 | | 8 | 2 |
| Timer RD interrupt enable register_2*[1] | TRDIER_2 | 8 | H'FF0588 | | 8 | 2 |

chanhvinh.com



| Register Name | Abbreviation | Number of Bits | Address | Module | Data Bus Width | Number of Access States |
|---|---|---|---|---|---|---|
| PWM mode output level control register_2*[1] | POCR_2 | 8 | H'FF0589 | Timer RD (unit 1, channel 2) | 8 | 2 |
| Timer RD digital filtering function select register_2*[1] | TRDDF_2 | 8 | H'FF058A | | 8 | 2 |
| Timer RD control register_3*[1] | TRDCR_3 | 8 | H'FF058B | Timer RD (unit 1, channel 3) | 8 | 2 |
| Timer RD I/O control register A_3*[1] | TRDIORA_3 | 8 | H'FF058C | | 8 | 2 |
| Timer RD I/O control register C_3*[1] | TRDIORC_3 | 8 | H'FF058D | | 8 | 2 |
| Timer RD status register_3*[1] | TRDSR_3 | 8 | H'FF058E | | 8 | 2 |
| Timer RD interrupt enable register_3*[1] | TRDIER_3 | 8 | H'FF058F | | 8 | 2 |
| PWM mode output level control register_3*[1] | POCR_3 | 8 | H'FF0590 | | 8 | 2 |
| Timer RD digital filtering function select register_3*[1] | TRDDF_3 | 8 | H'FF0591 | | 8 | 2 |
| Timer RD status register_23*[1] | TRDSTR_23 | 8 | H'FF0592 | Timer RD (unit 1, channels 2 and 3 in common) | 8 | 2 |
| Timer RD mode register_23*[1] | TRDMDR_23 | 8 | H'FF0593 | | 8 | 2 |
| Timer RD PWM mode register_23*[1] | TRDPMR_23 | 8 | H'FF0594 | | 8 | 2 |
| Timer RD function control register_23*[1] | TRDFCR_23 | 8 | H'FF0595 | | 8 | 2 |
| Timer RD output master enable register 1_23*[1] | TRDOER1_23 | 8 | H'FF0596 | | 8 | 2 |
| Timer RD output master enable register 2_23*[1] | TRDOER2_23 | 8 | H'FF0597 | | 8 | 2 |
| Timer RD output control register_23*[1] | TRDOCR_23 | 8 | H'FF0598 | | 8 | 2 |
| Timer RD A/D conversion start trigger control register_23*[1] | TRDADCR_23 | 8 | H'FF0599 | | 8 | 2 |
| $I^2C$ bus control register 1 | ICCR1 | 8 | H'FF05C8 | IIC2/SSU | 8 | 2 |
| SS control register H | SSCRH | | | | 8 | 2 |
| $I^2C$ bus control register 2 | ICCR2 | 8 | H'FF05C9 | IIC2/SSU | 8 | 2 |
| SS control register L | SSCRL | | | | 8 | 2 |

chanhvinh.com



| Register Name | Abbreviation | Number of Bits | Address | Module | Data Bus Width | Number of Access States |
|---|---|---|---|---|---|---|
| I²C bus mode register | ICMR | 8 | H'FF05CA | IIC2/SSU | 8 | 2 |
| SS mode register | SSMR | | | | 8 | 2 |
| I²C bus interrupt enable register | ICIER | 8 | H'FF05CB | IIC2/SSU | 8 | 2 |
| SS enable register | SSER | | | | 8 | 2 |
| I²C bus status register | ICSR | 8 | H'FF05CC | IIC2/SSU | 8 | 2 |
| SS status register | SSSR | | | | 8 | 2 |
| Slave address register | SAR | 8 | H'FF05CD | IIC2/SSU | 8 | 2 |
| SS mode register 2 | SSMR2 | | | | 8 | 2 |
| I²C bus transmit data register | ICDRT | 8 | H'FF05CE | IIC2/SSU | 8 | 2 |
| SS transmit data register | SSTDR | | | | 8 | 2 |
| I²C bus receive data register | ICDRR | 8 | H'FF05CF | IIC2/SSU | 8 | 2 |
| SS receive data register | SSRDR | | | | 8 | 2 |
| D/A data register 0 | DADR0 | 8 | H'FF05D4 | D/A converter | 8 | 2 |
| D/A data register 1 | DADR1 | 8 | H'FF05D5 | D/A converter | 8 | 2 |
| D/A control register | DACR | 8 | H'FF05D6 | D/A converter | 8 | 2 |
| IrDA control register | IrCR | 8 | H'FF05DE | SCI3_2 | 8 | 2 |
| A/D data register 0 | ADDR0 | 16 | H'FF05E0 | A/D converter (unit 1) | 16 | 2 |
| Compare data register | CMPR | 8 | H'FF05E0 | A/D converter (unit 1) | 16*[4] | 2 |
| A/D data register 1 | ADDR1 | 16 | H'FF05E2 | A/D converter (unit 1) | 16 | 2 |
| Compare control/status register | CMPCSR | 8 | H'FF05E2 | A/D converter (unit 1) | 16*[4] | 2 |
| A/D data register 2 | ADDR2 | 16 | H'FF05E4 | A/D converter (unit 1) | 16 | 2 |

chanhvinh.com



| Register Name | Abbreviation | Number of Bits | Address | Module | Data Bus Width | Number of Access States |
|---|---|---|---|---|---|---|
| Compare voltage register H | CMPVALH | 8 | H'FF05E4 | A/D converter (unit 1) | 16*4 | 2 |
| A/D data register 3 | ADDR3 | 16 | H'FF05E6 | A/D converter (unit 1) | 16 | 2 |
| Compare voltage register L | CMPVALL | 8 | H'FF05E6 | A/D converter (unit 1) | 16*4 | 2 |
| A/D data register 4 | ADDR4 | 16 | H'FF05E8 | A/D converter (unit 1) | 16 | 2 |
| A/D data register 5 | ADDR5 | 16 | H'FF05EA | A/D converter (unit 1) | 16 | 2 |
| A/D data register 6 | ADDR6 | 16 | H'FF05EC | A/D converter (unit 1) | 16 | 2 |
| A/D data register 7 | ADDR7 | 16 | H'FF05EE | A/D converter (unit 1) | 16 | 2 |
| A/D control/status register | ADCSR | 8 | H'FF05F0 | A/D converter (unit 1) | 8 | 2 |
| A/D control register | ADCR | 8 | H'FF05F1 | A/D converter (unit 1) | 8 | 2 |
| A/D mode register | ADMR | 8 | H'FF05F4 | A/D converter (unit 1) | 8 | 2 |
| A/D data register 0_2*5 | ADDR0_2 | 16 | H'FF0600 | A/D converter (unit 2) | 16 | 2 |
| Compare data register_2*5 | CMPR_2 | 8 | H'FF0600 | A/D converter (unit 2) | 16*4 | 2 |
| A/D data register 1_2*5 | ADDR1_2 | 16 | H'FF0602 | A/D converter (unit 2) | 16 | 2 |

chanhvinh.com



| Register Name | Abbreviation | Number of Bits | Address | Module | Data Bus Width | Number of Access States |
|---|---|---|---|---|---|---|
| Compare control/status register_2*5 | CMPCSR_2 | 8 | H'FF0602 | A/D converter (unit 2) | 16*4 | 2 |
| A/D data register 2_2*5 | ADDR2_2 | 16 | H'FF0604 | A/D converter (unit 2) | 16 | 2 |
| Compare voltage register H_2*5 | CMPVALH_2 | 8 | H'FF0604 | A/D converter (unit 2) | 16*4 | 2 |
| A/D data register 3_2*5 | ADDR3_2 | 16 | H'FF0606 | A/D converter (unit 2) | 16 | 2 |
| Compare voltage register L_2*5 | CMPVALL_2 | 8 | H'FF0606 | A/D converter (unit 2) | 16*4 | 2 |
| A/D control/status register_2*5 | ADCSR_2 | 8 | H'FF0610 | A/D converter (unit 2) | 8 | 2 |
| A/D control register_2*5 | ADCR_2 | 8 | H'FF0611 | A/D converter (unit 2) | 8 | 2 |
| A/D mode register_2*5 | ADMR_2 | 8 | H'FF0614 | A/D converter (unit 2) | 8 | 2 |
| Reset source flag register | RSTFR | 8 | H'FF0620 | Exception handling | 8 | 2 |
| Low-voltage detection circuit 2 control register H | LD2CRH | 8 | H'FF0622 | Low-voltage-detection circuit | 8 | 2 |
| Low-voltage detection circuit 2 control register L | LD2CRL | 8 | H'FF0623 | | 8 | 2 |
| Low-voltage detection circuit 1 control register H | LD1CRH | 8 | H'FF0624 | | 8 | 2 |
| Low-voltage detection circuit 1 control register L | LD1CRL | 8 | H'FF0625 | | 8 | 2 |
| Low-voltage detection circuit 0 control register H | LD0CRH | 8 | H'FF0626 | | 8 | 2 |
| Low-voltage detection circuit 0 control register L | LD0CRL | 8 | H'FF0627 | | 8 | 2 |

chanhvinh.com



| Register Name | Abbreviation | Number of Bits | Address | Module | Data Bus Width | Number of Access States |
|---|---|---|---|---|---|---|
| Low-voltage detection circuit control protect register | VDCPR | 8 | H'FF0628 | Low-voltage-detection circuit | 8 | 2 |
| Timer RG counter | TRGCNT | 16 | H'FF0640 | Timer RG | 16*3 | 2 |
| General register A | GRA | 16 | H'FF0642 | Timer RG | 16*3 | 2 |
| General register B | GRB | 16 | H'FF0644 | Timer RG | 16*3 | 2 |
| Timer RG mode register | TRGMDR | 8 | H'FF0646 | Timer RG | 8 | 2 |
| Timer RG counter control register | TRGCNTCR | 8 | H'FF0647 | Timer RG | 8 | 2 |
| Timer RG control register | TRGCR | 8 | H'FF0648 | Timer RG | 8 | 2 |
| Timer RG I/O control register | TRGIOR | 8 | H'FF0649 | Timer RG | 8 | 2 |
| Timer RG status register | TRGSR | 8 | H'FF064A | Timer RG | 8 | 2 |
| Timer RG interrupt enable register | TRGIER | 8 | H'FF064B | Timer RG | 8 | 2 |
| GRA buffer register | BRA | 16 | H'FF064C | Timer RG | 16*3 | 2 |
| GRB buffer register | BRB | 16 | H'FF064E | Timer RG | 16*3 | 2 |
| Flash memory control register 1 | FLMCR1 | 8 | H'FF0660 | FLASH | 8 | 2 |
| Flash memory control register 2 | FLMCR2 | 8 | H'FF0661 | FLASH | 8 | 2 |
| Flash memory data flash protect register | DFPR | 8 | H'FF0662 | FLASH | 8 | 2 |
| Flash memory status register | FLMSTR | 8 | H'FF0663 | FLASH | 8 | 2 |
| Event link setting register 0 | ELSR0 | 8 | H'FF0680 | ELC | 8 | 2 |
| Event link setting register 1 | ELSR1 | 8 | H'FF0681 | ELC | 8 | 2 |
| Event link setting register 2*6 | ELSR2 | 8 | H'FF0682 | ELC | 8 | 2 |
| Event link setting register 3 | ELSR3 | 8 | H'FF0683 | ELC | 8 | 2 |
| Event link setting register 4 | ELSR4 | 8 | H'FF0684 | ELC | 8 | 2 |
| Event link setting register 8 | ELSR8 | 8 | H'FF0688 | ELC | 8 | 2 |
| Event link setting register 10 | ELSR10 | 8 | H'FF068A | ELC | 8 | 2 |
| Event link setting register 11*7 | ELSR11 | 8 | H'FF068B | ELC | 8 | 2 |
| Event link setting register 12 | ELSR12 | 8 | H'FF068C | ELC | 8 | 2 |
| Event link setting register 14 | ELSR14 | 8 | H'FF068E | ELC | 8 | 2 |
| Event link setting register 15 | ELSR15 | 8 | H'FF068F | ELC | 8 | 2 |
| Event link setting register 18 | ELSR18 | 8 | H'FF0692 | ELC | 8 | 2 |

chanhvinh.com



| Register Name | Abbreviation | Number of Bits | Address | Module | Data Bus Width | Number of Access States |
|---|---|---|---|---|---|---|
| Event link setting register 19 | ELSR19 | 8 | H'FF0693 | ELC | 8 | 2 |
| Event link setting register 21 | ELSR21 | 8 | H'FF0695 | ELC | 8 | 2 |
| Event link setting register 22 | ELSR22 | 8 | H'FF0696 | ELC | 8 | 2 |
| Event link setting register 23 | ELSR23 | 8 | H'FF0697 | ELC | 8 | 2 |
| Event link setting register 24 | ELSR24 | 8 | H'FF0698 | ELC | 8 | 2 |
| Event link setting register 29 | ELSR29 | 8 | H'FF069D | ELC | 8 | 2 |
| Event link setting register 30 | ELSR30 | 8 | H'FF069E | ELC | 8 | 2 |
| Event link setting register 31 | ELSR31 | 8 | H'FF069F | ELC | 8 | 2 |
| Event link setting register 32 | ELSR32 | 8 | H'FF06A0 | ELC | 8 | 2 |
| Port-group setting register 1 | PGR1 | 8 | H'FF06A2 | ELC | 8 | 2 |
| Port-group setting register 2 | PGR2 | 8 | H'FF06A3 | ELC | 8 | 2 |
| Port-group control register 1 | PGC1 | 8 | H'FF06A6 | ELC | 8 | 2 |
| Port-group control register 2 | PGC2 | 8 | H'FF06A7 | ELC | 8 | 2 |
| Port buffer register 1 | PDBF1 | 8 | H'FF06AA | ELC | 8 | 2 |
| Port buffer register 2 | PDBF2 | 8 | H'FF06AB | ELC | 8 | 2 |
| Event link port setting register 0 | PEL0 | 8 | H'FF06AD | ELC | 8 | 2 |
| Event link port setting register 1 | PEL1 | 8 | H'FF06AE | ELC | 8 | 2 |
| Event link port setting register 2 | PEL2 | 8 | H'FF06AF | ELC | 8 | 2 |
| Event link port setting register 3 | PEL3 | 8 | H'FF06B0 | ELC | 8 | 2 |
| Event link option setting register A | ELOPA | 8 | H'FF06B5 | ELC | 8 | 2 |
| Event link option setting register B | ELOPB | 8 | H'FF06B6 | ELC | 8 | 2 |
| Event link option setting register C | ELOPC | 8 | H'FF06B7 | ELC | 8 | 2 |
| Event-generation timer control register | ELTMCR | 8 | H'FF06B8 | ELC | 8 | 2 |
| Event-generation timer interval setting register A | ELTMSA | 8 | H'FF06B9 | ELC | 8 | 2 |
| Event-generation timer interval setting register B | ELTMSB | 8 | H'FF06BA | ELC | 8 | 2 |
| Even-generation delay time selection register | ELTMDR | 8 | H'FF06BB | ELC | 8 | 2 |
| Event link control register | ELCR | 8 | H'FF06BC | ELC | 8 | 2 |
| ELC timer counter | ELTMCNT | 16 | H'FF06C0 | ELC | 16*[3] | 2 |

chanhvinh.com



| Register Name | Abbreviation | Number of Bits | Address | Module | Data Bus Width | Number of Access States |
|---|---|---|---|---|---|---|
| System clock control register | SYSCCR | 8 | H'FF06D0 | SYSTEM | 16*[8] | 2 |
| Power-down control register 1 | LPCR1 | 8 | H'FF06D1 | SYSTEM | 16*[8] | 2 |
| Power-down control register 2 | LPCR2 | 8 | H'FF06D2 | SYSTEM | 16*[8] | 2 |
| Power-down control register 3 | LPCR3 | 8 | H'FF06D3 | SYSTEM | 16*[8] | 2 |
| Backup control register | BAKCR | 8 | H'FF06D4 | SYSTEM | 16*[8] | 2 |
| OSC oscillation settling control status register | OSCCSR | 8 | H'FF06D5 | Clock oscillator | 16*[8] | 2 |
| Reset control register | RSTCR | 8 | H'FF06DA | Exception handling | 16*[8] | 2 |
| Timer RA control register | TRACR | 8 | H'FF06F0 | Timer RA | 8 | 2 |
| Timer RA I/O control register | TRAIOC | 8 | H'FF06F1 | Timer RA | 8 | 2 |
| Timer RA mode register | TRAMR | 8 | H'FF06F2 | Timer RA | 8 | 2 |
| Timer RA prescaler register | TRAPRE | 8 | H'FF06F3 | Timer RA | 8 | 2 |
| Timer RA timer register | TRATR | 8 | H'FF06F4 | Timer RA | 8 | 2 |
| Timer RA interrupt request status register | TRAIR | 8 | H'FF06F5 | Timer RA | 8 | 2 |
| Timer RC counter*[6] | TRCCNT | 16 | H'FFFF80 | Timer RC | 16*[3] | 2 |
| General register A*[6] | GRA | 16 | H'FFFF82 | Timer RC | 16*[3] | 2 |
| General register B*[6] | GRB | 16 | H'FFFF84 | Timer RC | 16*[3] | 2 |
| General register C*[6] | GRC | 16 | H'FFFF86 | Timer RC | 16*[3] | 2 |
| General register D*[6] | GRD | 16 | H'FFFF88 | Timer RC | 16*[3] | 2 |
| Timer RC mode register*[6] | TRCMR | 8 | H'FFFF8A | Timer RC | 8 | 2 |
| Timer RC control register 1*[6] | TRCCR1 | 8 | H'FFFF8B | Timer RC | 8 | 2 |
| Timer RC interrupt enable register*[6] | TRCIER | 8 | H'FFFF8C | Timer RC | 8 | 2 |
| Timer RC status register*[6] | TRCSR | 8 | H'FFFF8D | Timer RC | 8 | 2 |
| Timer RC I/O control register 0*[6] | TRCIOR0 | 8 | H'FFFF8E | Timer RC | 8 | 2 |
| Timer RC I/O control register 1*[6] | TRCIOR1 | 8 | H'FFFF8F | Timer RC | 8 | 2 |
| Timer RC control register 2*[6] | TRCCR2 | 8 | H'FFFF90 | Timer RC | 8 | 2 |
| Timer RC digital filtering function select register*[6] | TRCDF | 8 | H'FFFF91 | Timer RC | 8 | 2 |
| Timer RC output enable register*[6] | TRCOER | 8 | H'FFFF92 | Timer RC | 8 | 2 |

chanhvinh.com



| Register Name | Abbreviation | Number of Bits | Address | Module | Data Bus Width | Number of Access States |
|---|---|---|---|---|---|---|
| Timer RC A/D conversion start trigger control register*[6] | TRCADCR | 8 | H'FFFF93 | Timer RC | 8 | 2 |
| Timer counter WD | TCWD | 8 | H'FFFF98 | WDT | 8 | 2 |
| Timer mode register WD | TMWD | 8 | H'FFFF99 | WDT | 8 | 2 |
| Timer control/status register WD | TCSRWD | 8 | H'FFFF9A | WDT | 8 | 2 |
| Timer interrupt control register WD | TICRWD | 8 | H'FFFF9B | WDT | 8 | 2 |
| Timer interrupt flag register WD | TIFRWD | 8 | H'FFFF9C | WDT | 8 | 2 |
| Timer RB control register | TRBCR | 8 | H'FFFFA0 | Timer RB | 8 | 2 |
| Timer RB one-shot control register | TRBOCR | 8 | H'FFFFA1 | Timer RB | 8 | 2 |
| Timer RB I/O control register | TRBIOC | 8 | H'FFFFA2 | Timer RB | 8 | 2 |
| Timer RB mode register | TRBMR | 8 | H'FFFFA3 | Timer RB | 8 | 2 |
| Timer prescaler register | TRBPRE | 8 | H'FFFFA4 | Timer RB | 8 | 2 |
| Timer RB secondary register | TRBSC | 8 | H'FFFFA5 | Timer RB | 8 | 2 |
| Timer RB primary register | TRBPR | 8 | H'FFFFA6 | Timer RB | 8 | 2 |
| Timer RB interrupt request status register | TRBIR | 8 | H'FFFFA7 | Timer RB | 8 | 2 |
| Timer RE second data register/counter data register | TRESEC | 8 | H'FFFFA8 | Timer RE | 8 | 2 |
| Timer RE minute data register/compare data register | TREMIN | 8 | H'FFFFA9 | Timer RE | 8 | 2 |
| Timer RE hour data register | TREHR | 8 | H'FFFFAA | Timer RE | 8 | 2 |
| Timer RE day-of-week data register | TREWK | 8 | H'FFFFAB | Timer RE | 8 | 2 |
| Timer RE control register 1 | TRECR1 | 8 | H'FFFFAC | Timer RE | 8 | 2 |
| Timer RE control register 2 | TRECR2 | 8 | H'FFFFAD | Timer RE | 8 | 2 |
| Timer RE interrupt flag register | TREIFR | 8 | H'FFFFAE | Timer RE | 8 | 2 |
| Time RE clock source select register | TRECSR | 8 | H'FFFFAF | Timer RE | 8 | 2 |
| Timer RD counter_0 | TRDCNT_0 | 16 | H'FFFFB0 | Timer RD unit 0 (channel 0) | 16*[3] | 2 |
| General register A_0 | GRA_0 | 16 | H'FFFFB2 | | 16*[3] | 2 |
| General register B_0 | GRB_0 | 16 | H'FFFFB4 | | 16*[3] | 2 |
| General register C_0 | GRC_0 | 16 | H'FFFFB6 | | 16*[3] | 2 |
| General register D_0 | GRD_0 | 16 | H'FFFFB8 | | 16*[3] | 2 |

chanhvinh.com



| Register Name | Abbreviation | Number of Bits | Address | Module | Data Bus Width | Number of Access States |
|---|---|---|---|---|---|---|
| Timer RD counter_1 | TRDCNT_1 | 16 | H'FFFFBA | Timer RD unit 0 (channel 1) | 16*3 | 2 |
| General register A_1 | GRA_1 | 16 | H'FFFFBC | | 16*3 | 2 |
| General register B_1 | GRB_1 | 16 | H'FFFFBE | | 16*3 | 2 |
| General register C_1 | GRC_1 | 16 | H'FFFFC0 | | 16*3 | 2 |
| General register D_1 | GRD_1 | 16 | H'FFFFC2 | | 16*3 | 2 |
| Timer RD control register_0 | TRDCR_0 | 8 | H'FFFFC4 | Timer RD unit 0 (channel 0) | 8 | 2 |
| Timer RD I/O control register A_0 | TRDIORA_0 | 8 | H'FFFFC5 | | 8 | 2 |
| Timer RD I/O control register C_0 | TRDIORC_0 | 8 | H'FFFFC6 | | 8 | 2 |
| Timer RD status register_0 | TRDSR_0 | 8 | H'FFFFC7 | | 8 | 2 |
| Timer RD interrupt enable register_0 | TRDIER_0 | 8 | H'FFFFC8 | | 8 | 2 |
| PWM mode output level control register_0 | POCR_0 | 8 | H'FFFFC9 | | 8 | 2 |
| Timer RD digital filtering function select register_0 | TRDDF_0 | 8 | H'FFFFCA | | 8 | 2 |
| Timer RD control register_1 | TRDCR_1 | 8 | H'FFFFCB | Timer RD unit 0 (channel 1) | 8 | 2 |
| Timer RD I/O control register A_1 | TRDIORA_1 | 8 | H'FFFFCC | | 8 | 2 |
| Timer RD I/O control register C_1 | TRDIORC_1 | 8 | H'FFFFCD | | 8 | 2 |
| Timer RD status register_1 | TRDSR_1 | 8 | H'FFFFCE | | 8 | 2 |
| Timer RD interrupt enable register_1 | TRDIER_1 | 8 | H'FFFFCF | | 8 | 2 |
| PWM mode output level control register_1 | POCR_1 | 8 | H'FFFFD0 | | 8 | 2 |
| Timer RD digital filtering function select register_1 | TRDDF_1 | 8 | H'FFFFD1 | | 8 | 2 |
| Timer RD start register_01 | TRDSTR_01 | 8 | H'FFFFD2 | Timer RD unit 0 (channels 0 and 1 in common) | 8 | 2 |
| Timer RD mode register_01 | TRDMDR_01 | 8 | H'FFFFD3 | | 8 | 2 |
| Timer RD PWM mode register_01 | TRDPMR_01 | 8 | H'FFFFD4 | | 8 | 2 |
| Timer RD function control register_01 | TRDFCR_01 | 8 | H'FFFFD5 | | 8 | 2 |

chanhvinh.com



| Register Name | Abbreviation | Number of Bits | Address | Module | Data Bus Width | Number of Access States |
|---|---|---|---|---|---|---|
| Timer RD output master enable register 1_01 | TRDOER1_01 | 8 | H'FFFFD6 | Timer RD unit 0 (channels 0 and 1 in common) | 8 | 2 |
| Time RD output master enable register 2_01 | TRDOER2_01 | 8 | H'FFFFD7 | | 8 | 2 |
| Timer RD output control register_01 | TRDOCR_01 | 8 | H'FFFFD8 | | 8 | 2 |
| Timer RC A/D conversion start trigger control register_01 | TRDADCR_01 | 8 | H'FFFFD9 | | 8 | 2 |
| Module standby control register 1 | MSTCR1 | 8 | H'FFFFDC | SYSTEM | 8 | 2 |
| Module standby control register 2 | MSTCR2 | 8 | H'FFFFDD | SYSTEM | 8 | 2 |
| Module standby control register 3 | MSTCR3 | 8 | H'FFFFDE | SYSTEM | 8 | 2 |
| Port data register 1 | PDR1 | 8 | H'FFFFE0 | I/O Port | 8 | 2 |
| Port data register 2 | PDR2 | 8 | H'FFFFE1 | I/O Port | 8 | 2 |
| Port data register 3 | PDR3 | 8 | H'FFFFE2 | I/O Port | 8 | 2 |
| Port data register 5 | PDR5 | 8 | H'FFFFE4 | I/O Port | 8 | 2 |
| Port data register 6 | PDR6 | 8 | H'FFFFE5 | I/O Port | 8 | 2 |
| Port data register 8 | PDR8 | 8 | H'FFFFE7 | I/O Port | 8 | 2 |
| Port data register 9*[1] | PDR9 | 8 | H'FFFFE8 | I/O Port | 8 | 2 |
| Port data register A | PDRA | 8 | H'FFFFE9 | I/O Port | 8 | 2 |
| Port data register B | PDRB | 8 | H'FFFFEA | I/O Port | 8 | 2 |
| Port data register J | PDRJ | 8 | H'FFFFEC | I/O Port | 8 | 2 |
| Port control register 1 | PCR1 | 8 | H'FFFFF0 | I/O Port | 8 | 2 |
| Port control register 2 | PCR2 | 8 | H'FFFFF1 | I/O Port | 8 | 2 |
| Port control register 3 | PCR3 | 8 | H'FFFFF2 | I/O Port | 8 | 2 |
| Port control register 5 | PCR5 | 8 | H'FFFFF4 | I/O Port | 8 | 2 |
| Port control register 6 | PCR6 | 8 | H'FFFFF5 | I/O Port | 8 | 2 |
| Port control register 8 | PCR8 | 8 | H'FFFFF7 | I/O Port | 8 | 2 |
| Port control register 9*[1] | PCR9 | 8 | H'FFFFF8 | I/O Port | 8 | 2 |
| Port control register A | PCRA | 8 | H'FFFFF9 | I/O Port | 8 | 2 |
| Port control register B | PCRB | 8 | H'FFFFFA | I/O Port | 8 | 2 |
| Port control register J | PCRJ | 8 | H'FFFFFC | I/O Port | 8 | 2 |

Notes: 1. Not provided for the H8S/20103 and H8S/20115 Groups. These addresses are reserved.

chanhvinh.com



2. Not provided for the H8S/20223 and H8S/20235 Groups. These addresses are reserved.
3. Only 16-bit access is allowed.
4. Access in 8-bit unit.
5. Not provided for the H8S/20103, H8S/20203, H8S/20115, and H8S/20215 Groups. These addresses are reserved.
6. Provided only for the H8S/20103 and H8S/20115 Groups. Addresses for the other groups are reserved.
7. Provided only for the H8S/20223 and H8S/20235 Groups. Addresses for the other groups are reserved.
8. Although these addresses are connected to the 16-bit bus, rewriting proceeds in byte units.

chanhvinh.com



## 27.2 Register Bits

The addresses and bit names of the registers in the on-chip peripheral modules are listed below. The 16-bit register is indicated in two rows, 8 bits for each row.

| Register Abbreviation | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| PMR1 | PMR17 | PMR16 | PMR15 | PMR14*1 | PMR13 | PMR12 | PMR11 | PMR10*1 | I/O Port |
| PMR2 | PMR27 | PMR26 | PMR25 | PMR24 | PMR23 | PMR22 | PMR21 | PMR20 | |
| PMR3 | PMR37 | PMR36 | PMR35 | PMR34 | PMR33 | PMR32 | PMR31 | PMR30 | |
| PMR5 | PMR57 | PMR56 | PMR55 | PMR54 | PMR53 | PMR52 | PMR51 | PMR50 | |
| PMR6 | PMR67 | PMR66 | PMR65 | PMR64 | PMR63 | PMR62 | PMR61 | PMR60 | |
| PMR8 | PMR87 | PMR86 | PMR85 | — | — | — | — | — | |
| PMR9*1 | PMR97 | PMR96 | PMR95 | PMR94 | PMR93 | PMR92 | PMR91 | PMR90 | |
| PMRA | PMRA7 | PMRA6 | PMRA5 | PMRA4 | PMRA3*2 | PMRA2 | — | — | |
| ICSUSR | — | — | — | — | — | — | — | SELICSU | IIC2/SSU |
| PMRJ | — | — | — | — | — | — | PMRJ[1:0] | | I/O Port |
| PUCR1 | PUCR17 | PUCR16 | PUCR15 | PUCR14*1 | PUCR13 | PUCR12 | PUCR11 | PUCR10*1 | |
| PUCR2 | PUCR27 | PUCR26 | PUCR25 | PUCR24 | PUCR23 | PUCR22 | PUCR21 | PUCR20 | |
| PUCR3 | PUCR37 | PUCR36 | PUCR35 | PUCR34 | PUCR33 | PUCR32 | PUCR31 | PUCR30 | |
| PUCR5 | — | — | PUCR55 | PUCR54 | PUCR53 | PUCR52 | PUCR51 | PUCR50 | |
| PUCR6 | PUCR67 | PUCR66 | PUCR65 | PUCR64 | PUCR63 | PUCR62 | PUCR61 | PUCR60 | |
| PUCR8 | PUCR87 | PUCR86 | PUCR85 | — | — | — | — | — | |
| PUCR9*1 | PUCR97 | PUCR96 | PUCR95 | PUCR94 | PUCR93 | PUCR92 | PUCR91 | PUCR90 | |
| PUCRA | PUCRA7 | PUCRA6 | PUCRA5 | PUCRA4 | PUCRA3*1 | PUCRA2*1 | PUCRA1*1 | PUCRA0*1 | |
| PUCRB | PUCRB7 | PUCRB6 | PUCRB5 | PUCRB4 | PUCRB3 | PUCRB2 | PUCRB1 | PUCRB0 | |
| PUCRJ | — | — | — | — | — | — | PUCRJ1 | PUCRJ0 | |
| PDVR1 | PDVR17 | PDVR16 | PDVR15 | PDVR14*1 | PDVR13 | PDVR12 | PDVR11 | PDVR10*1 | |
| PDVR2 | PDVR27 | PDVR26 | PDVR25 | PDVR24 | PDVR23 | PDVR22 | PDVR21 | PDVR20 | |
| PDVR3 | PDVR37 | PDVR36 | PDVR35 | PDVR34 | PDVR33 | PDVR32 | PDVR31 | PDVR30 | |
| PDVR5 | — | — | PDVR55 | PDVR54 | PDVR53 | PDVR52 | PDVR51 | PDVR50 | |

chanhvinh.com



| Register Abbreviation | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| PDVR6 | PDVR67 | PDVR66 | PDVR65 | PDVR64 | PDVR63 | PDVR62 | PDVR61 | PDVR60 | I/O Port |
| PDVR8 | PDVR87 | PDVR86 | PDVR85 | — | — | — | — | — | |
| PDVR9*[1] | PDVR97 | PDVR96 | PDVR95 | PDVR94 | PDVR93 | PDVR92 | PDVR91 | PDVR90 | |
| PMCR11 | — | P11MD[2:0] | | — | | P10MD[2:0]*[1] | | | PMC |
| PMCR12 | — | P13MD[2:0] | | — | | P12MD[2:0] | | | |
| PMCR13 | — | P15MD[2:0] | | — | | P14MD[2:0]*[1] | | | |
| PMCR14 | — | P17MD[2:0] | | — | | P16MD[2:0] | | | |
| PMCR21 | — | P21MD[2:0] | | — | | P20MD[2:0] | | | |
| PMCR22 | — | P23MD[2:0] | | — | | P22MD[2:0] | | | |
| PMCR23 | — | P25MD[2:0] | | — | | P24MD[2:0] | | | |
| PMCR24 | — | P27MD[2:0] | | — | | P26MD[2:0] | | | |
| PMCR31 | — | P31MD[2:0] | | — | | P30MD[2:0] | | | |
| PMCR32 | — | P33MD[2:0] | | — | | P32MD[2:0] | | | |
| PMCR33 | — | P35MD[2:0] | | — | | P34MD[2:0] | | | |
| PMCR34 | — | P37MD[2:0] | | — | | P36MD[2:0] | | | |
| PMCR51 | — | P51MD[2:0] | | — | | P50MD[2:0] | | | |
| PMCR52 | — | P53MD[2:0] | | — | | P52MD[2:0] | | | |
| PMCR53 | — | P55MD[2:0] | | — | | P54MD[2:0] | | | |
| PMCR54 | — | P57MD[2:0] | | — | | P56MD[2:0] | | | |
| PMCR61 | — | P61MD[2:0] | | — | | P60MD[2:0] | | | |
| PMCR62 | — | P63MD[2:0] | | — | | P62MD[2:0] | | | |
| PMCR63 | — | P65MD[2:0] | | — | | P64MD[2:0] | | | |
| PMCR64 | — | P67MD[2:0] | | — | | P66MD[2:0] | | | |
| PMCR83 | — | P85MD[2:0] | | — | | — | — | — | |
| PMCR84 | — | P87MD[2:0] | | — | | P86MD[2:0] | | | |
| PMCR91*[1] | — | P91MD[2:0] | | — | | P90MD[2:0] | | | |
| PMCR92*[1] | — | P93MD[2:0] | | — | | P92MD[2:0] | | | |
| PMCR93*[1] | — | P95MD[2:0] | | — | | P94MD[2:0] | | | |
| PMCR94*[1] | — | P97MD[2:0] | | — | | P96MD[2:0] | | | |
| PMCWPR | B0WI | PMCRWE | — | — | — | — | — | — | |
| PMCRA3 | — | PA5MD[2:0] | | — | | PA4MD[2:0] | | | |

chanhvinh.com



| Register Abbreviation | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| PMCRA4 | — | PA7MD[2:0] | | | — | PA6MD[2:0] | | | PMC |
| LINCR | LINE | MST | SBE | LSTART | RXDSF | BCIE | SBIE | SFIE | HW-LIN |
| LINST | — | — | B2CLR | B1CLR | B0CLR | BCDCT | SBDCT | SFDCT | |
| INTCR | — | — | INTM[1:0] | | NMIEG | ADTRG1 | ADTRG0 | — | Interrupt |
| IER | IRQ7E | IRQ6E | IRQ5E | IRQ4E | IRQ3E | IRQ2E | IRQ1E | IRQ0E | |
| ISCRH | [IRQ7SCB:IRQ7SCA] | | [IRQ6SCB:IRQ6SCA] | | [IRQ5SCB:IRQ5SCA] | | [IRQ4SCB:IRQ4SCA] | | |
| ISCRL | [IRQ3SCB:IRQ3SCA] | | [IRQ2SCB:IRQ2SCA] | | [IRQ1SCB:IRQ1SCA] | | [IRQ0SCB:IRQ0SCA] | | |
| ISR | IRQ7F | IRQ6F | IRQ5F | IRQ4F | IRQ3F | IRQ2F | IRQ1F | IRQ0F | |
| INCCR | — | — | INCCR[5:4] | | INCCR[3:2] | | INCCR[1:0] | | |
| VOFR | | | | | | | | | |
| | | | | | | — | — | — | |
| ELCSR | — | — | — | — | ELIE2 | ELIE1 | ELF2 | ELF1 | |
| IPRA | IPRA[7:6] | | IPRA[5:4] | | IPRA[3:2] | | IPRA[1:0] | | |
| IPRB | IPRB[7:6] | | IPRB[5:4] | | IPRB[3:2] | | IPRB[1:0] | | |
| IPRC | IPRC[7:6] | | IPRC[5:4] | | IPRC[3:2] | | IPRC[1:0] | | |
| IPRD | IPRD[7:6] | | IPRD[5:4]*2 | | IPRD[3:2] | | IPRD[1:0] | | |
| IPRE | IPRE[7:6] | | IPRE[5:4] | | IPRE[3:2] | | — | — | |
| IPRF | — | — | — | — | IPRF[3:2] | | — | — | |
| IPRG | — | — | IPRG[5:4] | | IPRG[3:2] | | IPRG[1:0]*3 | | |
| IPRH | IPRH[7:6] | | IPRH[5:4] | | IPRH[3:2]*4 | | IPRH[1:0]*4 | | |
| IPRI | IPRI[7:6] | | — | — | IPRI[3:2] | | — | — | |
| DTCERA | DTCEA7 | DTCEA6 | DTCEA5 | DTCEA4 | DTCEA3 | DTCEA2 | DTCEA1 | DTCEA0 | DTC |
| DTCERB | DTCEB7 | DTCEB6 | DTCEB5 | DTCEB4 | DTCEB3 | DTCEB2 | DTCEB1 | DTCEB0 | |
| DTCERC | DTCEC7 | DTCEC6 | DTCEC5 | DTCEC4 | — | — | — | — | |
| DTCERD | DTCED7 | DTCED6 | — | — | DTCED3 | DTCED2 | DTCED1 | DTCED0 | |
| DTCERE | DTCEE7 | DTCEE6 | DTCEE5 | DTCEE4 | DTCEE3 | DTCEE2 | DTCEE1 | DTCEE0 | |
| DTCERF | DTCEF7*5 | DTCEF6*5 | DTCEF5*5 | DTCEF4*5 | DTCEF3*5 | DTCEF2*5 | DTCEF1*5 | DTCEF0*5 | |
| DTCERG | — | — | — | DTCEG4 | DTCEG3 | DTCEG2 | DTCEG1 | DTCEG0 | |
| DTCERH | — | — | — | — | DTCEH3 | DTCEH2 | — | — | |
| DTVECR | SWDTE | DTVEC6 | DTVEC5 | DTVEC4 | DTVEC3 | DTVEC2 | DTVEC1 | DTVEC0 | |

chanhvinh.com



| Register Abbreviation | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| SMR | COM | CHR | PE | PM | STOP | MP | | CKS[1:0] | SCI3_1 |
| BRR | | | | | | | | | |
| SCR3 | TIE | RIE | TE | RE | MPIE | TEIE | | CKE[1:0] | |
| TDR | | | | | | | | | |
| SSR | TDRE | RDRF | OER | FER | PER | TEND | MPBR | MPBT | |
| RDR | | | | | | | | | |
| SPMR | — | — | — | — | — | NFEN | — | — | |
| SMR_2 | COM | CHR | PE | PM | STOP | MP | | CKS[1:0] | SCI3_2 |
| BRR_2 | | | | | | | | | |
| SCR3_2 | TIE | RIE | TE | RE | MPIE | TEIE | | CKE[1:0] | |
| TDR_2 | | | | | | | | | |
| SSR_2 | TDRE | RDRF | OER | FER | PER | TEND | MPBR | MPBT | |
| RDR_2 | | | | | | | | | |
| SPMR_2 | — | — | — | — | — | NFEN | — | — | |
| SMR_3 | COM | CHR | PE | PM | STOP | MP | | CKS[1:0] | SCI3_3 |
| BRR_3 | | | | | | | | | |
| SCR3_3 | TIE | RIE | TE | RE | MPIE | TEIE | | CKE[1:0] | |
| TDR_3 | | | | | | | | | |
| SSR_3 | TDRE | RDRF | OER | FER | PER | TEND | MPBR | MPBT | |
| RDR_3 | | | | | | | | | |
| SPMR_3 | — | — | — | — | — | NFEN | — | — | |
| TRDCNT_2*5 | | | | | | | | | Timer RD Unit 1 (channel 2) |
| GRA_2*5 | | | | | | | | | |
| GRB_2*5 | | | | | | | | | |
| GRC_2*5 | | | | | | | | | |
| GRD_2*5 | | | | | | | | | |

chanhvinh.com



| Register Abbreviation | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| TRDCNT_3*5 | | | | | | | | | Timer RD Unit 1 (channel 3) |
| GRA_3*5 | | | | | | | | | |
| GRB_3*5 | | | | | | | | | |
| GRC_3*5 | | | | | | | | | |
| GRD_3*5 | | | | | | | | | |
| TRDCR_2*5 | | CCLR[2:0] | | | CKEG[1:0] | | TPSC[2:0] | | Timer RD Unit 1 (channel 2) |
| TRDIORA_2*5 | — | IOB2 | | IOB[1:0] | — | IOA2 | IOA[1:0] | | |
| TRDIORC_2*5 | IOD3 | IOD2 | | IOD[1:0] | IOC3 | IOC2 | IOC[1:0] | | |
| TRDSR_2*5 | — | — | — | OVF | IMFD | IMFC | IMFB | IMFA | |
| TRDIER_2*5 | — | — | — | OVIE | IMIED | IMIEC | IMIEB | IMIEA | |
| POCR_2*5 | — | — | — | — | — | POLD | POLC | POLB | |
| TRDDF_2*5 | | DFCK[1:0] | — | — | DFD | DFC | DFB | DFA | |
| TRDCR_3*5 | | CCLR[2:0] | | | CKEG[1:0] | | TPSC[2:0] | | Timer RD Unit 1 (channel 3) |
| TRDIORA_3*5 | — | IOB2 | | IOB[1:0] | — | IOA2 | IOA[1:0] | | |
| TRDIORC_3*5 | IOD3 | IOD2 | | IOD[1:0] | IOC3 | IOC2 | IOC[1:0] | | |
| TRDSR_3*5 | — | — | UDF | OVF | IMFD | IMFC | IMFB | IMFA | |
| TRDIER_3*5 | — | — | — | OVIE | IMIED | IMIEC | IMIEB | IMIEA | |
| POCR_3*5 | — | — | — | — | — | POLD | POLC | POLB | |
| TRDDF_3*5 | | DFCK[1:0] | — | — | DFD | DFC | DFB | DFA | |
| TRDSTR_23*5 | — | — | — | — | CSTPN1 | CSTPN0 | STR1 | STR0 | Timer RD Unit 1 (channels 2 and 3 in common) |
| TRDMDR_23*5 | BFD1 | BFC1 | BFD0 | BFC0 | — | — | — | SYNC | |
| TRDPMR_23*5 | — | PWMD1 | PWMC1 | PWMB1 | — | PWMD0 | PWMC0 | PWMB0 | |
| TRDFCR_23*5 | PWM3 | STCLK | ADEG | ADTRG | OLS1 | OLS0 | CMD[1:0] | | |
| TRDOER1_23*5 | ED1 | EC1 | EB1 | EA1 | ED0 | EC0 | EB0 | EA0 | |

chanhvinh.com



| Register Abbreviation | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| TRDOER2_23*5 | PTO | — | — | — | — | — | — | — | Timer RD Unit 1 (channels 2 and 3 in common) |
| TRDOCR_23*5 | TOD1 | TOC1 | TOB1 | TOA1 | TOD0 | TOC0 | TOB0 | TOA0 | |
| TRDADCR_23*5 | ADTRGD1E | ADTRGC1E | ADTRGB1E | ADTRGA1E | ADTRGD0E | ADTRGC0E | ADTRGB0E | ADTRGA0E | |
| ICCR1 | ICE | RCVD | MST | TRS | CKS[3:0] | | | | IIC2/SSU |
| SSCRH | — | RSSTP | MSS | — | — | CKS[2:0] | | | |
| ICCR2 | BBSY | SCP | SDAO | SDAOP | SCLO | — | IICRST | — | |
| SSCRL | — | — | SOL | SOLP | — | — | SRES | — | |
| ICMR | MLS | WAIT | — | — | BCWP | BC[2:0] | | | |
| SSMR | MLS | CPOS | CPHS | — | — | BC[2:0] | | | |
| ICIER | TIE | TEIE | RIE | NAKIE | STIE | ACKE | ACKBR | ACKBT | |
| SSER | TIE | TEIE | RIE | TE | RE | — | — | CEIE | |
| ICSR | TDRE | TEND | RDRF | NACKF | STOP | AL_OVE | AAS | ADZ | |
| SSSR | TDRE | TEND | RDRF | — | — | ORER | — | CE | |
| SAR | SVA[6:0] | | | | | | | FS | |
| SSMR2 | BIDE | SCKS | CSS[1:0] | | SCKOS | SOOS | CSOS | SSUMS | |
| ICDRT | | | | | | | | | |
| SSTDR | | | | | | | | | |
| ICDRR | | | | | | | | | |
| SSRDR | | | | | | | | | |
| DADR0 | | | | | | | | | D/A converter |
| DADR1 | | | | | | | | | |
| DACR | DAOE1 | DAOE0 | — | — | — | — | — | — | |
| IrCR | IrE | | IrCK[2:0] | | IrTXINV | IrRXINV | — | — | SCI3_2 (IrDA) |
| ADDR0 | | | | | | | | | A/D converter (unit 1) |
| | | | — | — | — | — | — | — | |
| CMPR | CMP7 | CMP6 | CMP5 | CMP4 | CMP3 | CMP2 | CMP1 | CMP0 | |
| ADDR1 | | | | | | | | | |
| | | | — | — | — | — | — | — | |

chanhvinh.com



| Register Abbreviation | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| CMPCSR | CMPF | CMPIE | CMPFC1 | CMPFC0 | — | — | — | — | A/D converter (unit 1) |
| ADDR2 | | | | | | | | | |
| | | | — | — | — | — | — | — | |
| CMPVALH | — | — | — | — | — | — | VAL9 | VAL8 | |
| ADDR3 | | | | | | | | | |
| | | | — | — | — | — | — | — | |
| CMPVALL | VAL7 | VAL6 | VAL5 | VAL4 | VAL3 | VAL2 | VAL1 | VAL0 | |
| ADDR4 | | | | | | | | | |
| | | | — | — | — | — | — | — | |
| ADDR5 | | | | | | | | | |
| | | | — | — | — | — | — | — | |
| ADDR6 | | | | | | | | | |
| | | | — | — | — | — | — | — | |
| ADDR7 | | | | | | | | | |
| | | | — | — | — | — | — | — | |
| ADCSR | ADF | ADIE | ADST | — | | | CH[3:0] | | |
| ADCR | | TRGS[1:0] | SCANE | SCANS | | CKS[1:0] | ADSTCLR | EXTRGS | |
| ADMR | — | — | ADM1 | — | — | — | — | — | |
| ADDR0_2*6 | | | | | | | | | A/D converter (unit 2)*6 |
| | | | — | — | — | — | — | — | |
| CMPR_2*6 | — | — | — | — | CMP3 | CMP2 | CMP1 | CMP0 | |
| ADDR1_2*6 | | | | | | | | | |
| | | | — | — | — | — | — | — | |
| CMPCSR_2*6 | CMPF | CMPIE | CMPFC1 | CMPFC0 | — | — | — | — | |
| ADDR2_2*6 | | | | | | | | | |
| | | | — | — | — | — | — | — | |
| CMPVALH_2*6 | — | — | — | — | — | — | VAL9 | VAL8 | |

chanhvinh.com



| Register Abbreviation | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| ADDR3_2*6 | | | | | | | | | A/D converter (unit2) *6 |
| | | | — | — | — | — | — | — | |
| CMPVALL_2*6 | VAL7 | VAL6 | VAL5 | VAL4 | VAL3 | VAL2 | VAL1 | VAL0 | |
| ADCSR_2*6 | ADF | ADIE | ADST | — | — | | CH[2:0] | | |
| ADCR_2*6 | TRGS[1:0] | | SCANE | SCANS | CKS[1:0] | | ADSTCLR | EXTRGS | |
| ADMR_2*6 | — | — | ADM1 | — | — | — | — | — | |
| RSTFR | — | — | SWRST | PRST | LVD2RST | LVD1RST | PORRST | WRST | Exception handling |
| LD2CRH | VD2DF | VD2UF | VD2DFCK[1:0] | | VD2DFS | VD2IRCS | VD2MS | VD2RE | Low voltage detection circuit |
| LD2CRL | VD2E | VD2CVS | VD2RVS | — | — | — | — | — | |
| LD1CRH | VD1DF | VD1UF | VD1DFCK[1:0] | | VD1DFS | VD1IRCS | VD1MS | VD1RE | |
| LD1CRL | VD1E | — | — | — | | | VD1LS[3:0] | | |
| LD0CRH | — | — | VD0DFCK[1:0] | | VD0DFS | — | — | — | |
| LD0CRL | — | — | — | — | — | — | VD0LS1 | — | |
| VDCPR | WRI | — | — | — | — | — | — | LDPRC | |
| TRGCNT | | | | | | | | | Timer RG |
| GRA | | | | | | | | | |
| GRB | | | | | | | | | |
| TRGMDR | STR | — | DFCK[1:0] | | DFB | DFA | MDF | PWM | |
| TRGCNTCR | CNTEN7 | CNTEN6 | CNTEN5 | CNTEN4 | CNTEN3 | CNTEN2 | CNTEN1 | CNTEN0 | |
| TRGCR | — | CCLR[1:0] | | CKEG[1:0] | | | TPSC[2:0] | | |
| TRGIOR | BUFB | IOB2 | IOB[1:0] | | BUFA | IOA2 | IOA[1:0] | | |
| TRGSR | — | — | — | DIRF | OVF | UDF | IMFB | IMFA | |
| TRGIER | — | — | — | — | OVIE | UDIE | IMIEB | IMIEA | |
| BRA | | | | | | | | | |
| BRB | | | | | | | | | |

chanhvinh.com



| Register Abbreviation | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| FLMCR1 | — | — | — | — | FMLBD | FMWUS | FMEWMOD | FMCMDEN | FLASH |
| FLMCR2 | — | — | — | FMRDYIE | FMBSYRDIE | FMISPE | FMSPREQ | FMSPEN | |
| DFPR | — | — | — | — | — | — | DFPR1 | DFPR0 | |
| FLMSTR | FMRDYIF | FMBSYRDIF | FMEBSF | FMERSF | FMPRSF | — | — | FMRDY | |
| ELSR0 | ELS07 | ELS06 | ELS05 | ELS04 | ELS03 | ELS02 | ELS01 | ELS00 | ELC |
| ELSR1 | ELS17 | ELS16 | ELS15 | ELS14 | ELS13 | ELS12 | ELS11 | ELS10 | |
| ELSR2※7 | ELS27 | ELS26 | ELS25 | ELS24 | ELS23 | ELS22 | ELS21 | ELS20 | |
| ELSR3 | ELS37 | ELS36 | ELS35 | ELS34 | ELS33 | ELS32 | ELS31 | ELS30 | |
| ELSR4 | ELS47 | ELS46 | ELS45 | ELS44 | ELS43 | ELS42 | ELS41 | ELS40 | |
| ELSR8 | ELS87 | ELS86 | ELS85 | ELS84 | ELS83 | ELS82 | ELS81 | ELS80 | |
| ELSR10 | ELS107 | ELS106 | ELS105 | ELS104 | ELS103 | ELS102 | ELS101 | ELS100 | |
| ELSR11※6 | ELS117 | ELS116 | ELS115 | ELS114 | ELS113 | ELS112 | ELS111 | ELS110 | |
| ELSR12 | ELS127 | ELS126 | ELS125 | ELS124 | ELS123 | ELS122 | ELS121 | ELS120 | |
| ELSR14 | ELS147 | ELS146 | ELS145 | ELS144 | ELS143 | ELS142 | ELS141 | ELS140 | |
| ELSR15 | ELS157 | ELS156 | ELS155 | ELS154 | ELS153 | ELS152 | ELS151 | ELS150 | |
| ELSR18 | ELS187 | ELS186 | ELS185 | ELS184 | ELS183 | ELS182 | ELS181 | ELS180 | |
| ELSR19 | ELS197 | ELS196 | ELS195 | ELS194 | ELS193 | ELS192 | ELS191 | ELS190 | |
| ELSR21 | ELS217 | ELS216 | ELS215 | ELS214 | ELS213 | ELS212 | ELS211 | ELS210 | |
| ELSR22 | ELS227 | ELS226 | ELS225 | ELS224 | ELS223 | ELS222 | ELS221 | ELS220 | |
| ELSR23 | ELS237 | ELS236 | ELS235 | ELS234 | ELS233 | ELS232 | ELS231 | ELS230 | |
| ELSR24 | ELS247 | ELS246 | ELS245 | ELS244 | ELS243 | ELS242 | ELS241 | ELS240 | |
| ELSR29 | ELS297 | ELS296 | ELS295 | ELS294 | ELS293 | ELS292 | ELS291 | ELS290 | |
| ELSR30 | ELS307 | ELS306 | ELS305 | ELS304 | ELS303 | ELS302 | ELS301 | ELS300 | |
| ELSR31 | ELS317 | ELS316 | ELS315 | ELS314 | ELS313 | ELS312 | ELS311 | ELS310 | |
| ELSR32 | ELS327 | ELS326 | ELS325 | ELS324 | ELS323 | ELS322 | ELS321 | ELS320 | |

chanhvinh.com



| Register Abbreviation | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| PGR1 | PGR17 | PGR16 | PGR15 | PGR14 | PGR13 | PGR12 | PGR11 | PGR10 | ELC |
| PGR2 | PGR27 | PGR26 | PGR25 | PGR24 | PGR23 | PGR22 | PGR21 | PGR20 | |
| PGC1 | — | | PGCO1[2:0] | | — | PGCOVE1 | PGCI1[1:0] | | |
| PGC2 | — | | PGCO2[2:0] | | — | PGCOVE2 | PGCI2[1:0] | | |
| PDBF1 | PDBF17 | PDBF16 | PDBF15 | PDBF14 | PDBF13 | PDBF12 | PDBF11 | PDBF10 | |
| PDBF2 | PDBF27 | PDBF26 | PDBF25 | PDBF24 | PDBF23 | PDBF22 | PDBF21 | PDBF20 | |
| PEL0 | — | | PSM0[1:0] | | PSP0[4:3] | PSP02 | PSP01 | PSP00 | |
| PEL1 | — | | PSM1[1:0] | | PSP1[4:3] | PSP12 | PSP11 | PSP10 | |
| PEL2 | — | | PSM2[1:0] | | PSP2[4:3] | PSP22 | PSP21 | PSP20 | |
| PEL3 | — | | PSM3[1:0] | | PSP3[4:3] | PSP32 | PSP31 | PSP30 | |
| ELOPA | TMRAM[2:1] | | TMRBM[2:1] | | TMRCM[2:1]*3 | | TMRD1M[2:1] | | |
| ELOPB | TMRD2M[2:1] | | — | — | — | — | — | — | |
| ELOPC | TMRGM[2:1] | | — | — | — | — | — | — | |
| ELTMCR | TMRSTR | — | — | — | | CLSRS[3:0] | | | |
| ELTMSA | | | C1CLS[3:0] | | | C0CLS[3:0] | | | |
| ELTMSB | | | C3CLS[3:0] | | | C2CLS[3:0] | | | |
| ELTMDR | C3DLY[1:0] | | C2DLY[1:0] | | C1DLY[1:0] | | C0DLY[1:0] | | |
| ELCR | ELCON | — | — | — | — | — | — | — | |
| ELTMCNT | | | | | | | | | |
| SYSCCR | WI | WE | PHIHSEL | PHILSEL | — | SUBNC[1:0] | | — | Clock oscillator |
| LPCR1 | WI | WE | SSBY | PSCSTP | SLEEPRS | STBYRS | — | PHIBSEL | |
| LPCR2 | WI | WE | — | — | — | PHI[2:0] | | | |
| LPCR3 | WI | WE | STBYINT | SLEEPINT | — | PHIS[2:0] | | | |
| BAKCR | WI | WE | OSCBAKE | BAKCKSEL | CKSWIE | CKSWIF | OSCHLT | — | |
| OSCCSR | OSCWEF | — | — | — | | STS[3:0] | | | |
| RSTCR | WI | WE | — | — | — | — | — | SRST | |
| TRACR | — | — | TUNDF | TEDGF | — | TSTOP | TCSTF | TSTART | Timer RA |
| TRAIOC | TIOGT[1:0] | | TIPF[1:0] | | TIOSEL | TOENA | TOPCR | TEDGSEL | |
| TRAMR | TCKCUT | | TCK[2:0] | | — | TMOD[2:0] | | | |

chanhvinh.com



| Register Abbreviation | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| TRAPRE | | | | | | | | | Timer RA |
| TRATR | | | | | | | | | |
| TRAIR | TRAIE | TRAIF | — | — | — | — | — | — | |
| TRCCNT*7 | | | | | | | | | Timer RC |
| GRA*7 | | | | | | | | | |
| GRB*7 | | | | | | | | | |
| GRC*7 | | | | | | | | | |
| GRD*7 | | | | | | | | | |
| TRCMR*7 | CTS | — | BUFEB | BUFEA | PWM2 | PWMD | PWMC | PWMB | |
| TRCCR1*7 | CCLR | | CKS[2:0] | | TOD | TOC | TOB | TOA | |
| TRCIER*7 | OVIE | — | — | — | IMIED | IMIEC | IMIEB | IMIEA | |
| TRCSR*7 | OVF | — | — | — | IMFD | IMFC | IMFB | IMFA | |
| TRCIOR0*7 | — | IOB2 | | IOB[1:0] | — | IOA2 | | IOA[1:0] | |
| TRCIOR1*7 | IOD3 | IOD2 | | IOD[1:0] | IOC3 | IOC2 | | IOC[1:0] | |
| TRCCR2*7 | TCEG[1:0] | | CSTP | — | — | POLD | POLC | POLB | |
| TRCDF*7 | DFCK[1:0] | | — | DFTRG | DFD | DFC | DFB | DFA | |
| TRCOER*7 | PTO | — | — | — | ED | EC | EB | EA | |
| TRCADCR*7 | — | — | — | — | ADTRGAE | ADTRGBE | ADTRGCE | ADTRGDE | |
| TCWD | | | | | | | | | WDT |
| TMWD | — | — | — | — | | | CKS[3:0] | | |
| TCSRWD | B6WI | TCWE | B4WI | TCSRWE | TMWLOCK | TMWI | — | — | |
| TICRWD | INTSEL[1:0] | | IWIE | — | — | — | — | — | |
| TIFRWD | IWF | — | — | — | — | — | — | — | |
| TRBCR | — | — | — | — | — | TSTOP | TCSTF | TSTART | Timer RB |
| TRBOCR | — | — | — | — | — | TOSSTF | TOSSP | TOSST | |

chanhvinh.com



| Register Abbreviation | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| TRBIOC | — | — | TIPF[1:0] | | INOSEG | INOSTG | TOCNT | TOPL | Timer RB |
| TRBMR | TCKCUT | TCK[2:0] | | | TWRC | — | TMOD[1:0] | | |
| TRBPRE | | | | | | | | | |
| TRBSC | | | | | | | | | |
| TRBPR | | | | | | | | | |
| TRBIR | TRBIE | TRBIF | — | — | — | — | — | — | |
| TRESEC | BSY | SC12 | SC11 | SC10 | SC03 | SC02 | SC01 | SC00 | Timer RE |
| TREMIN | BSY | MN12 | MN11 | MN10 | MN03 | MN02 | MN01 | MN00 | |
| TREHR | BSY | — | HR11 | HR10 | HR03 | HR02 | HR01 | HR00 | |
| TREWK | BSY | — | — | — | — | WK[2:0] | | | |
| TRECR1 | TSTART | H12_H24 | PM | TRERST | INT | TOENA | TCSTF | — | |
| TRECR2 | — | — | COMIE | WKIE | DYIE | HRIE | MNIE | SEIE | |
| TREIFR | — | — | COMF | WKF | DYF | HRF | MNF | SECF | |
| TRECSR | — | RCS[6:4] | | | RCS3 | RCS2 | RCS[1:0] | | |
| TRDCNT_0 | | | | | | | | | Timer RD Unit 0 (channel 0) |
| GRA_0 | | | | | | | | | |
| GRB_0 | | | | | | | | | |
| GRC_0 | | | | | | | | | |
| GRD_0 | | | | | | | | | |
| TRDCNT_1 | | | | | | | | | Timer RD Unit 0 (channel 1) |
| GRA_1 | | | | | | | | | |
| GRB_1 | | | | | | | | | |

chanhvinh.com



| Register Abbreviation | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| GRC_1 | | | | | | | | | Timer RD Unit 0 (channel 1) |
| GRD_1 | | | | | | | | | |
| TRDCR_0 | | CCLR[2:0] | | | CKEG[1:0] | | TPSC[2:0] | | Timer RD Unit 0 (channel 0) |
| TRDIORA_0 | — | IOB2 | IOB[1:0] | | — | IOA2 | IOA[1:0] | | |
| TRDIORC_0 | IOD3 | IOD2 | IOD[1:0] | | IOC3 | IOC2 | IOC[1:0] | | |
| TRDSR_0 | — | — | — | OVF | IMFD | IMFC | IMFB | IMFA | |
| TRDIER_0 | — | — | — | OVIE | IMIED | IMIEC | IMIEB | IMIEA | |
| POCR_0 | — | — | — | — | — | POLD | POLC | POLB | |
| TRDDF_0 | DFCK[1:0] | | — | — | DFD | DFC | DFB | DFA | |
| TRDCR_1 | | CCLR[2:0] | | | CKEG[1:0] | | TPSC[2:0] | | Timer RD Unit 0 (channel 1) |
| TRDIORA_1 | — | IOB2 | IOB[1:0] | | — | IOA2 | IOA[1:0] | | |
| TRDIORC_1 | IOD3 | IOD2 | IOD[1:0] | | IOC3 | IOC2 | IOC[1:0] | | |
| TRDSR_1 | — | — | UDF | OVF | IMFD | IMFC | IMFB | IMFA | |
| TRDIER_1 | — | — | — | OVIE | IMIED | IMIEC | IMIEB | IMIEA | |
| POCR_1 | — | — | — | — | — | POLD | POLC | POLB | |
| TRDDF_1 | DFCK[1:0] | | — | — | DFD | DFC | DFB | DFA | |
| TRDSTR_01 | — | — | — | — | CSTPN1 | CSTPN0 | STR1 | STR0 | Timer RD Unit 0 (channels 0 and 1 in common) |
| TRDMDR_01 | BFD1 | BFC1 | BFD0 | BFC0 | — | — | — | SYNC | |
| TRDPMR_01 | — | PWMD1 | PWMC1 | PWMB1 | — | PWMD0 | PWMC0 | PWMB0 | |
| TRDFCR_01 | PWM3 | STCLK | ADEG | ADTRG | OLS1 | OLS0 | CMD[1:0] | | |
| TRDOER1_01 | ED1 | EC1 | EB1 | EA1 | ED0 | EC0 | EB0 | EA0 | |
| TRDOER2_01 | PTO | — | — | — | — | — | — | — | |
| TRDOCR_01 | TOD1 | TOC1 | TOB1 | TOA1 | TOD0 | TOC0 | TOB0 | TOA0 | |
| TRDADCR_01 | ADTRGD1E | ADTRGC1E | ADTRGB1E | ADTRGA1E | ADTRGD0E | ADTRGC0E | ADTRGB0E | ADTRGA0E | |
| MSTCR1 | MSTWDT | — | MSTAD1 | MSTAD2*2 | MSTDA | MSTDTC | — | — | Power-down |
| MSTCR2 | MSTSCI3_1 | MSTSCI3_2 | MSTSCI3_3 | — | — | MSTICSU | — | — | |

chanhvinh.com



| Register Abbreviation | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| MSTCR3 | MSTTMRA | MSTTMRB | MSTTMRC*3 | MSTTMRD1 | MSTTMRD2*1 | MSTTMRG | — | MSTTMRE | Power-down |
| PDR1 | PDR17 | PDR16 | PDR15 | PDR14*1 | PDR13 | PDR12 | PDR11 | PDR10*1 | I/O port |
| PDR2 | PDR27 | PDR26 | PDR25 | PDR24 | PDR23 | PDR22 | PDR21 | PDR20 | |
| PDR3 | PDR37 | PDR36 | PDR35 | PDR34 | PDR33 | PDR32 | PDR31 | PDR30 | |
| PDR5 | PDR57 | PDR56 | PDR55 | PDR54 | PDR53 | PDR52 | PDR51 | PDR50 | |
| PDR6 | PDR67 | PDR66 | PDR65 | PDR64 | PDR63 | PDR62 | PDR61 | PDR60 | |
| PDR8 | PDR87 | PDR86 | PDR85 | — | — | — | — | — | |
| PDR9*1 | PDR97 | PDR96 | PDR95 | PDR94 | PDR93 | PDR92 | PDR91 | PDR90 | |
| PDRA | PDRA7 | PDRA6 | PDRA5 | PDRA4 | PDRA3*1 | PDRA2*1 | PDRA1*1 | PDRA0*1 | |
| PDRB | PDRB7 | PDRB6 | PDRB5 | PDRB4 | PDRB3 | PDRB2 | PDRB1 | PDRB0 | |
| PDRJ | — | — | — | — | — | — | PDRJ1 | PDRJ0 | |
| PCR1 | PCR17 | PCR16 | PCR15 | PCR14*1 | PCR13 | PCR12 | PCR11 | PCR10*1 | |
| PCR2 | PCR27 | PCR26 | PCR25 | PCR24 | PCR23 | PCR22 | PCR21 | PCR20 | |
| PCR3 | PCR37 | PCR36 | PCR35 | PCR34 | PCR33 | PCR32 | PCR31 | PCR30 | |
| PCR5 | PCR57 | PCR56 | PCR55 | PCR54 | PCR53 | PCR52 | PCR51 | PCR50 | |
| PCR6 | PCR67 | PCR66 | PCR65 | PCR64 | PCR63 | PCR62 | PCR61 | PCR60 | |
| PCR8 | PCR87 | PCR86 | PCR85 | — | — | — | — | — | |
| PCR9*1 | PCR97 | PCR96 | PCR95 | PCR94 | PCR93 | PCR92 | PCR91 | PCR90 | |
| PCRA | PCRA7 | PCRA6 | PCRA5 | PCRA4 | PCRA3*1 | PCRA2*1 | PCRA1*1 | PCRA0*1 | |
| PCRB | PCRB7 | PCRB6 | PCRB5 | PCRB4 | PCRB3 | PCRB2 | PCRB1 | PCRB0 | |
| PCRJ | — | — | — | — | — | — | PCRJ1 | PCRJ0 | |

Notes:
1. Not provided for the H8S/20103 and H8S/20115 Groups. These addresses and bits are reserved.
2. Provided for the H8S/20223 and H8S/20235 Groups. These bits for the other groups are reserved.
3. Provided for the H8S/20103 and H8S/20115 Groups. These bits for the other groups are reserved.
4. Not provided for the H8S/20103 and H8S/20115 Groups. These bits are reserved.
5. Not provided for the H8S/20103 and H8S/20115 Groups. These addresses are reserved.
6. Provided for the H8S/20223 and H8S/20235 Groups. These addresses for the other groups are reserved.
7. Provided for the H8S/20103 and H8S/20115 Groups. These addresses for the other groups are reserved.

chanhvinh.com



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

chanhvinh.com



# Section 28 Electrical Characteristics

## 28.1 Absolute Maximum Ratings

**Table 28.1 Absolute Maximum Ratings**

| Item | | Symbol | Value | Unit | Notes |
|---|---|---|---|---|---|
| Power supply voltage | | $V_{CC}$ | –0.3 to +6.5 | V | *1 |
| Analog power supply voltage | | $AV_{CC}$ | –0.3 to +6.5 | V | |
| Input voltage | All pins (other than AN pin, DA pin, OSC1, and X1) | $V_{IN}$ | –0.3 to $V_{CC}$ +0.3 | V | |
| | AN pin, DA pin | $AV_{IN}$ | –0.3 to $AV_{CC}$ +0.3 | V | |
| | OSC1, X1 | $V_{IN}$ | –0.3 to +1.65 | V | *2 |
| | OSC1 | $V_{IN}$ | –0.3 to $V_{CC}$ +0.3 | V | *3 |
| Operating temperature | | $T_{opr}$ | N version: –20 to +85 | °C | *4 |
| | | | D version: –40 to +85 | °C | |
| Storage temperature | | $T_{stg}$ | –55 to +125 | °C | |

Notes: 1. Permanent damage may result if maximum ratings are exceeded. Normal operation should be under the conditions specified in Electrical Characteristics. Exceeding these values can result in incorrect operation and reduced reliability.

2. The OSC1 pin is used when the external oscillator function is selected. (PMRJ1 = 1, PMRJ0 = 1)
3. When the external clock input function is selected. (PMRJ1 = 0, PMRJ0 = 1)
4. N-version products are used for customer products. D-version products are used for customer products and industrial equipment.

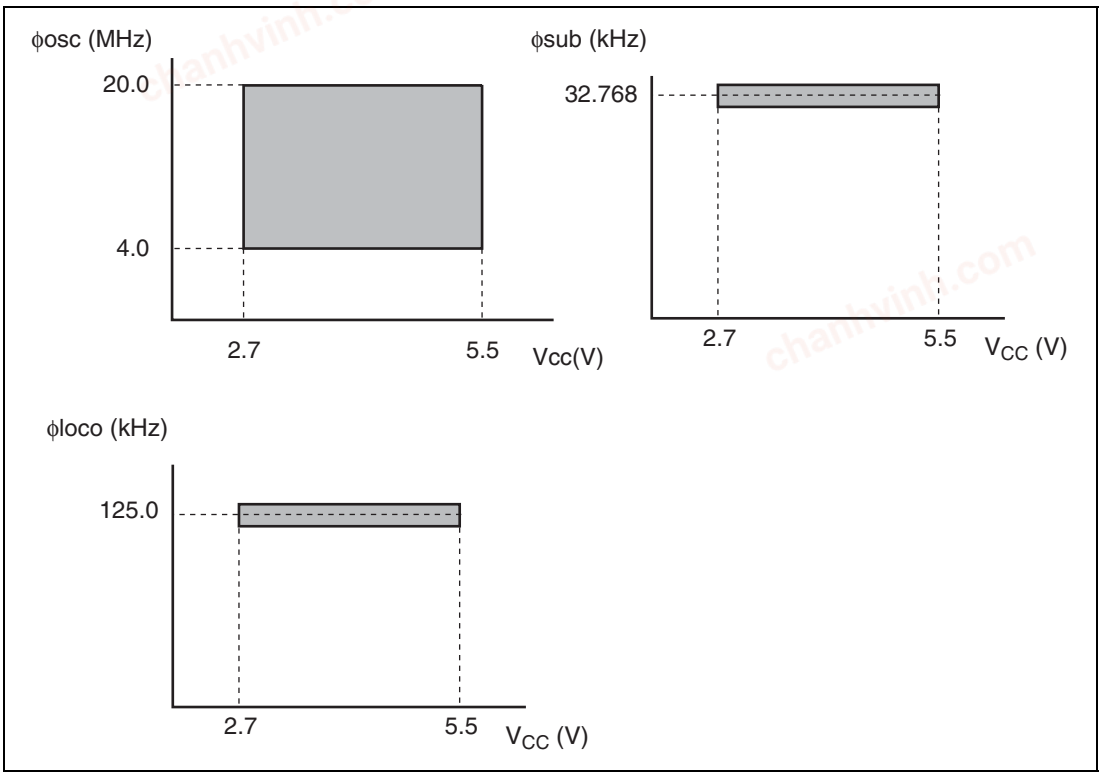
chanhvinh.com



# 28.2 Electrical Characteristics

## 28.2.1 Power Supply Voltage and Operating Ranges

### (1) Power Supply Voltage and Oscillation Frequency Range

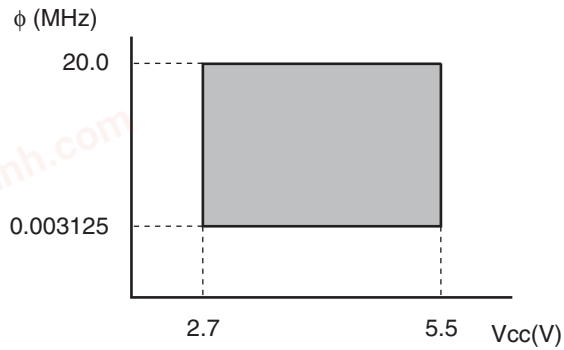



chanhvinh.com

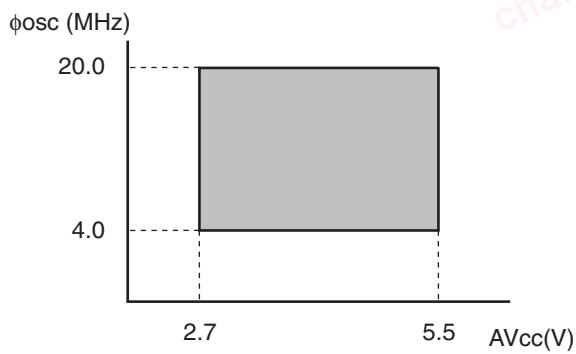
chanhvinh.com



### (2) Power Supply Voltage and Operating Frequency Range




### (3) Accuracy Guarantee Range of Analog Power Supply Voltage and A/D Converter



chanhvinh.com



## 28.3 DC Characteristics

### Table 28.2 DC Characteristics (1)

$V_{CC}$ = 2.7 to 5.5 V, $V_{SS}$ = 0.0 V, VCC ≥ AVCC, Ta = −20 to +85 °C (N version)/ −40 to +85 °C (D version), unless otherwise indicated.

| Item | Symbol | Applicable Pins | Test Condition | Values Min. | Typ. | Max. | Unit | Notes |
|---|---|---|---|---|---|---|---|---|
| Input high voltage | $V_{IH}$ | RES, NMI<br>IRQ0 to IRQ7<br>TRAIO, TRGB, FTCI,<br>TRGC, FTIOA,<br>FTIOB, FTIOC,<br>FTIOD, TRCOI,<br>FTIOA0<br>FTIOB0, FTIOC0,<br>FTIOD0, FTIOA1,<br>FTIOB1, FTIOC1,<br>FTIOD1, TRDOI_0,<br>FTIOA2, FTIOB2,<br>FTIOC2, FTIOD2,<br>FTIOA3, FTIOB3,<br>FTIOC3, FTIOD3,<br>TRDOI_1, TCLKA<br>TCLKB, TGIOA<br>TGIOB, SCK3,<br>SCK3_2, SCK3_3<br>ADTRG1, ADTRG2<br>RXD, RXD_2, RXD_3,<br>SSCK, SCS, SSI, SSO | $V_{CC}$ = 4.0 to 5.5 V | $V_{CC} \times 0.8$ | — | $V_{CC}$ + 0.3 | V | |
| | | | | $V_{CC} \times 0.9$ | — | $V_{CC}$ + 0.3 | V | |
| | | P10 to P17,<br>P20 to P27, P30 to P37<br>P50 to P57, P60 to P67<br>P85 to P87, P90 to P97<br>PA0 to PA7, PB0 to PB7<br>PJ1, PJ0, SCL, SDA | $V_{CC}$ = 4.0 to 5.5 V | $V_{CC} \times 0.7$ | — | $V_{CC}$ + 0.3 | V | |
| | | | | $V_{CC} \times 0.8$ | — | $V_{CC}$ + 0.3 | V | |
| | | OSC1 | $V_{CC}$ = 4.0 to 5.5 V<br>PMRJ[1:0] = 01 | $V_{CC}$ − 0.5 | — | $V_{CC}$ + 0.3 | V | |
| | | | PMRJ[1:0] = 01 | $V_{CC}$ − 0.3 | — | $V_{CC}$ + 0.3 | V | |

Note: Connect the TEST pin to Vss.

chanhvinh.com



| Item | Symbol | Applicable Pins | Test Condition | Values Min. | Typ. | Max. | Unit | Notes |
|---|---|---|---|---|---|---|---|---|
| Input low voltage | $V_{IL}$ | $\overline{RES}$, $\overline{NMI}$<br>$\overline{IRQ0}$ to $\overline{IRQ7}$<br>TRAIO, TRGB, FTCI,<br>TRGC, FTIOA,<br>FTIOB, FTIOC, | $V_{CC}$ = 4.0 to 5.5V | –0.3 | — | $V_{CC} \times 0.2$ | V | |
| | | FTIOD, $\overline{TRCOI}$<br>FTIOA0<br>FTIOB0, FTIOC0,<br>FTIOD0, FTIOA1,<br>FTIOB1, FTIOC1,<br>FTIOD1, $\overline{TRDOI_0}$,<br>FTIOA2, FTIOB2,<br>FTIOC2, FTIOD2,<br>FTIOA3, FTIOB3,<br>FTIOC3, FTIOD3,<br>$\overline{TRDOI_1}$, TCLKA<br>TCLKB, TGIOA<br>TGIOB, SCK3,<br>SCK3_2, SCK3_3<br>$\overline{ADTRG1}$, $\overline{ADTRG2}$,<br>RXD, RXD_2, RXD_3,<br>SSCK, $\overline{SCS}$, SSI, SSO | | –0.3 | — | $V_{CC} \times 0.1$ | V | |
| | | P10 to P17<br>P20 to P27<br>P30 to P37<br>P50 to P57<br>P60 to P67<br>P85 to P87 | $V_{CC}$ = 4.0 to 5.5V | –0.3 | — | $V_{CC} \times 0.3$ | V | |
| | | P90 to P97<br>PA0 to PA7<br>PB0 to PB7<br>PJ1, PJ0, SCL, SDA | | –0.3 | — | $V_{CC} \times 0.2$ | V | |
| | | OSC1 | $V_{CC}$ = 4.0 to 5.5V<br>PMRJ[1:0] = 01 | –0.3 | — | 0.5 | V | |
| | | | PMRJ[1:0] = 01 | –0.3 | — | 0.3 | V | |

chanhvinh.com



| Item | Symbol | Applicable Pins | Setting Condition | Test Condition | Values Min. | Values Typ. | Values Max. | Unit | Notes |
|---|---|---|---|---|---|---|---|---|---|
| Output high voltage | $V_{OH}$ | P10 to P17<br>P20 to P27<br>P30 to P37<br>P50 to P55<br>P60 to P67<br>P85 to P87<br>P90 to P97 | PDVRn0 to 7 = 0<br>(n = 1, 2, 3, 5, 6, 8, 9) | $V_{CC}$ = 4.0 to 5.5 V<br>$-I_{OH}$ = 5.0mA | $V_{CC}$ –1.0 | — | — | V | |
| | | | | $-I_{OH}$ = 0.1mA | $V_{CC}$ –0.5 | — | — | V | |
| | | | PDVRn0 to 7 = 1<br>(n = 1, 2, 3, 5, 6, 8, 9) | $V_{CC}$ = 4.0 to 5.5V<br>$-I_{OH}$ = 20.0mA | — | $V_{CC}$ –1.5 | — | V | Reference value |
| | | | | $V_{CC}$ = 4.0 to 5.5V<br>$-I_{OH}$ = 10.0mA | — | $V_{CC}$ –1.0 | — | V | Reference value |
| | | | | $V_{CC}$ = 4.0 to 5.5V<br>$-I_{OH}$ = 5.0mA | — | $V_{CC}$ –0.5 | — | V | Reference value |
| | | | | $-I_{OH}$ = 0.1mA | — | $V_{CC}$ –0.4 | — | V | Reference value |
| | | PA0 to PA7<br>PB0 to PB7<br>PJ0, PJ1 | | $V_{CC}$ = 4.0 to 5.5V<br>$-I_{OH}$ = 5.0mA | $V_{CC}$ –1.0 | — | — | V | |
| | | | | $-I_{OH}$ = 0.1mA | $V_{CC}$ –0.5 | — | — | V | |
| | | P56, P57 | | 4.0 ≤ $V_{CC}$ ≤ 5.5V<br>$-I_{OH}$ = 0.1mA | $V_{CC}$ –2.5 | — | — | V | |
| | | | | 3.0 ≤ $V_{CC}$ < 4.0V<br>$-I_{OH}$ = 0.1mA | $V_{CC}$ –2.0 | — | — | V | |

chanhvinh.com



| Item | Symbol | Applicable Pins | Setting Condition | Test Condition | Values Min. | Values Typ. | Values Max. | Unit | Notes |
|---|---|---|---|---|---|---|---|---|---|
| Output low voltage | $V_{OL}$ | P10 to P17<br>P20 to P27<br>P30 to P37<br>P50 to P57 | PDVRn0 to 7 = 0<br>(n = 1, 2, 3, 5, 6, 8, 9) | $V_{CC}$ = 4.0 to 5.5V<br>$I_{OL}$ = 1.6mA | — | — | 0.6 | V | |
| | | | | $I_{OL}$ = 0.4mA | — | — | 0.4 | V | |
| | | P60 to P67<br>P85 to P87<br>P90 to P97 | PDVRn0 to 7 = 1<br>(n = 1, 2, 3, 5, 6, 8, 9) | $V_{CC}$ = 4.0 to 5.5V<br>$I_{OL}$ = 20.0mA | — | — | 1.5 | V | |
| | | | | $I_{OL}$ = 5.0mA | — | — | 1.0 | V | |
| | | | | $V_{CC}$ = 4.0 to 5.5V<br>$I_{OL}$ = 1.6mA | — | 0.6 | — | V | Reference value |
| | | | | $I_{OL}$ = 0.4mA | — | 0.4 | — | V | Reference value |
| | | PA0 to PA7<br>PB0 to PB7<br>PJ0, PJ1 | | $V_{CC}$ = 4.0 to 5.5V<br>$I_{OL}$ = 1.6mA | — | — | 0.6 | V | |
| | | | | $I_{OL}$ = 0.4mA | — | — | 0.4 | V | |
| | | SCL, SDA | | $V_{CC}$ = 4.0 to 5.5V<br>$I_{OL}$ = 6.0mA | — | — | 0.6 | V | |
| | | | | $I_{OL}$ = 3.0mA | — | — | 0.4 | V | |

chanhvinh.com



| Item | Symbol | Applicable Pins | Test Condition | Values Min. | Typ. | Max. | Unit | Notes |
|---|---|---|---|---|---|---|---|---|
| Input/output leakage current | $\lvert I_{IL} \rvert$ | $\overline{NMI}$, $\overline{IRQ0}$ to $\overline{IRQ7}$<br>TRAIO,TRGB,<br>FTCI, TRGC, FTIOA<br>FTIOB, FTIOC,<br>FTIOD, $\overline{TRCOI}$,<br>FTIOA0, FTIOB0,<br>FTIOC0, FTIOD0,<br>FTIOA1, FTIOB1,<br>FTIOC1, FTIOD1<br>$\overline{TRDOI_0}$, FTIOA2<br>FTIOB2, FTIOC2,<br>FTIOD2<br>FTIOA3, FTIOB3,<br>FTIOC3, FTIOD3<br>$\overline{TRDOI_1}$, TCLKA<br>TCLKB, TGIOA<br>TGIOB, SCK3<br>SCK3_2, SCK3_3<br>$\overline{ADTRG1}$, $\overline{ADTRG2}$<br>RXD, RXD_2,<br>RXD_3<br>SCL, SDA<br>SSCK, $\overline{SCS}$<br>SSI, SSO, OSC1 | $V_{IN}$ = 0.5 V to ($V_{CC}$ - 0.5 V) | — | — | 1.0 | µA | |
| | | P10 to P17<br>P20 to P27<br>P30 to P37<br>P50 to P57<br>P60 to P67<br>P85 to P87<br>P90 to P97<br>PA0 to PA7<br>PB0 to PB7<br>PJ1, PJ0 | | | | | | |

chanhvinh.com



| Item | Symbol | Applicable Pins | Test Condition | Values Min. | Typ. | Max. | Unit | Notes |
|---|---|---|---|---|---|---|---|---|
| Pull-up MOS current | $-I_P$ | P10 to P17<br>P20 to P27<br>P30 to P37<br>P50 to P57<br>P60 to P67<br>P85 to P87<br>P90 to P97<br>PA0 to PA7<br>PB0 to PB7<br>PJ1, PJ0 | $V_{CC}$ = 5.0 V, $V_{IN}$ = 0.0 V | 40.0 | — | 200.0 | µA | |
| | | | $V_{CC}$ = 3.0 V, $V_{IN}$ = 0.0 V | — | 40.0 | — | µA | Reference value |
| Input capacitance | $C_{IN}$ | All input pins except power supply pins | f = 1 MHz, $V_{IN}$ = 0.0 V, Ta = 25°C | — | — | 15.0 | pF | |
| Active mode supply current | $I_{OPE1}$ | $V_{CC}$ | Active mode 1, $\phi_{OSC}$ = 20 MHz | — | 20.0 | 25.0 | mA | * |
| | | | Active mode 1, $\phi_{OSC}$ = 10 MHz | — | 12.0 | — | mA | Reference value* |
| | $I_{OPE2}$ | $V_{CC}$ | Active mode 2, $\phi_{OSC}$ = 20 MHz | — | 4.0 | 6.0 | mA | * |
| | | | Active mode 2, $\phi_{OSC}$ = 10 MHz | — | 3.5 | — | mA | Reference value* |
| | $I_{OPE3}$ | $V_{CC}$ | Active mode 3, $\phi_{OSC}$ = 20 MHz | — | 3.5 | 5.5 | mA | * |
| | | | Active mode 3, $\phi_{OSC}$ = 10 MHz | — | 3.0 | — | mA | Reference value* |
| | $I_{OPE4}$ | $V_{CC}$ | Active mode 4, $\phi_{sub}$ = 32 kHz | — | 2.0 | — | mA | Reference value* |
| | $I_{OPE5}$ | $V_{CC}$ | Active mode 5, $\phi_{sub}$ = 32 kHz | — | 2.0 | — | mA | Reference value* |

chanhvinh.com



| Item | Symbol | Applicable Pins | Test Condition | Values Min. | Typ. | Max. | Unit | Notes |
|---|---|---|---|---|---|---|---|---|
| Sleep mode supply current | $I_{SLEEP1}$ | $V_{CC}$ | Sleep mode 1, $\phi_{OSC}$ = 20MHz | — | 10.0 | 14.0 | mA | * |
| | | | Sleep mode 1, $\phi_{OSC}$ = 10MHz | — | 6.0 | — | mA | Reference value* |
| | $I_{SLEEP2}$ | $V_{CC}$ | Sleep mode 2, $\phi_{OSC}$ = 20MHz | — | 3.5 | 5.5 | mA | * |
| | | | Sleep mode 2, $\phi_{OSC}$ = 10MHz | — | 3.0 | — | mA | Reference value* |
| | $I_{SLEEP3}$ | $V_{CC}$ | Sleep mode 3, $\phi_{OSC}$ = 20MHz | — | 3.2 | 5.0 | mA | * |
| | | | Sleep mode 3, $\phi_{OSC}$ = 10MHz | — | 3.0 | — | mA | Reference value* |
| | $I_{SLEEP4}$ | $V_{CC}$ | Sleep mode 4, $\phi_{sub}$ = 32kHz | — | 1.8 | — | mA | Reference value* |
| | $I_{SLEEP5}$ | $V_{CC}$ | Sleep mode 5, $\phi_{sub}$ = 32kHz | — | 1.8 | — | mA | Reference value* |
| Standby mode supply current | $I_{STBY}$ | $V_{CC}$ | Ta ≤ 50 °C when 32-kHz crystal resonator not used | — | 10.0 | 30.0 | µA | * |
| | | | Ta > 50 °C when 32-kHz crystal resonator not used | — | 30.0 | 200.0 | µA | * |
| RAM data retaining voltage | $V_{RAM}$ | $V_{CC}$ | | 2.0 | — | — | V | |

Note: * Pin states during supply current measurement are given below (excluding current in the pull-up MOS transistors and output buffers).

chanhvinh.com



| Mode | $\overline{RES}$ Pin | Internal State | PSCSTP | Other Pins | Oscillator Pins |
|---|---|---|---|---|---|
| Active mode 1 | $V_{CC}$ | Operating ($\phi = \phi_{OSC}$) | 0 | $V_{CC}$ | Main clock oscillator: Ceramic resonator or crystal resonator<br>Subclock oscillator: Pin X1 = $V_{SS}$ |
| Active mode 2 | | Operating ($\phi = \phi_{OSC}/64$) | 0 | | |
| Active mode 3 | | Operating ($\phi = \phi_{OSC}/128$) | 0 | | |
| Sleep mode 1 | $V_{CC}$ | Only timers operating | 0 | $V_{CC}$ | |
| Sleep mode 2 | | Only timers operating ($\phi = \phi_{OSC}/64$) | 0 | | |
| Sleep mode 3 | | Only timers operating ($\phi = \phi_{OSC}/128$) | 0 | | |
| Active mode 4 | $V_{CC}$ | Operating ($\phi = \phi_{sub}$) | 1 | $V_{CC}$ | Main clock oscillator: Ceramic resonator or crystal resonator<br>Subclock oscillator: Crystal resonator |
| Active mode 5 | | Operating ($\phi = \phi_{sub}/8$) | 1 | | |
| Sleep mode 4 | $V_{CC}$ | Only timers operating ($\phi = \phi_{sub}$) | 1 | $V_{CC}$ | |
| Sleep mode 5 | | Only timers operating ($\phi = \phi_{sub}/8$) | 1 | | |
| Standby mode | $V_{CC}$ | CPU and timers both stop. | — | $V_{CC}$ | Main clock oscillator: Ceramic resonator or crystal resonator<br>Subclock oscillator: Pin X1 = $V_{SS}$ |

chanhvinh.com



## Table 28.3 DC Characteristics (2)

$V_{CC}$ = 2.7 to 5.5 V, $V_{SS}$ = 0.0 V, VCC ≥ AVCC, Ta = –20 to +85 °C (N version)/ –40 to +85 °C (D version), unless otherwise indicated.

| Item | Symbol | Applicable Pins | Setting Conditions | Test Conditions | Values Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|---|---|
| Allowable output low current (per pin) | $I_{OL}$ | P10 to P17<br>P20 to P27<br>P30 to P37 | PDVRn0 to 7 = 0<br>(n = 1, 2, 3, 5, 6, 8, 9, J) | $V_{CC}$ = 4.0 to 5.5 V | — | — | 2.0 | mA |
| | | | | | — | — | 0.5 | mA |
| | | P50 to P57<br>P60 to P67<br>P85 to P87<br>P90 to P97<br>PJ0, PJ1 | PDVRn0 to 7 = 1<br>(n = 1, 2, 3, 5, 6, 8, 9, J) | $V_{CC}$ = 4.0 to 5.5 V | — | — | 20.0 | mA |
| | | | | | — | — | 5.0 | mA |
| | | PA0 to PA7, PB0 to PB7 | — | $V_{CC}$ = 4.0 to 5.5 V | — | — | 2.0 | mA |
| | | | | | — | — | 0.5 | mA |
| | | SCL, SDA | — | $V_{CC}$ = 4.0 to 5.5 V | — | — | 6.0 | mA |
| Allowable output low current (total) | $\Sigma I_{OL}$ | All output pins | — | $V_{CC}$ = 4.0 to 5.5 V | — | — | 80 | mA |
| | | | | | — | — | 40 | mA |

chanhvinh.com



| Item | Symbol | Applicable Pins | Setting Conditions | Test Conditions | Values Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|---|---|
| Allowable output high current (per pin) | $\lvert -I_{OH} \rvert$ | P10 to P17<br>P20 to P27<br>P30 to P37 | PDVRn0 to 7 = 0<br>(n = 1, 2, 3, 5, 6, 8, 9, J) | $V_{CC}$ = 4.0 to 5.5 V | — | — | 5.0 | mA |
| | | | | | — | — | 0.2 | mA |
| | | P50 to P55<br>P60 to P67<br>P85 to P87<br>P90 to P97<br>PJ0, PJ1 | PDVRn0 to 7 = 1<br>(n = 1, 2, 3, 5, 6, 8, 9, J) | $V_{CC}$ = 4.0 to 5.5 V | — | — | 20.0 | mA |
| | | | | | — | — | 5.0 | mA |
| | | PA0 to PA7, PB0 to PB7 | | $V_{CC}$ = 4.0 to 5.5 V | — | — | 5.0 | mA |
| | | | | | — | — | 0.2 | mA |
| | | P56, P57 | | $V_{CC}$ = 4.0 to 5.5 V | — | — | 0.4 | mA |
| | | | | | — | — | 0.2 | mA |
| Allowable output high current (total) | $\lvert -\Sigma I_{OH} \rvert$ | All output pins | | $V_{CC}$ = 4.0 to 5.5 V | — | — | 80 | mA |
| | | | | | — | — | 40 | mA |
| Allowable output current (total) | $\lvert \Sigma I_O \rvert = \Sigma I_{OL} + \lvert -\Sigma I_{OH} \rvert$ | All output pins | | | — | — | 80 | mA |

chanhvinh.com



## 28.4 AC Characteristics

### Table 28.4 AC Characteristics

$V_{CC}$ = 2.7 to 5.5 V, $V_{SS}$ = 0.0 V, VCC ≥ AVCC, Ta = −20 to +85 °C (N version)/ −40 to +85 °C (D version), unless otherwise indicated.

| Item | Symbol | Applicable Pins | Test Conditions | Values | | | Unit | Reference Figure |
|---|---|---|---|---|---|---|---|---|
| | | | | Min. | Typ. | Max. | | |
| Main clock oscillator oscillation frequency | $\phi_{OSC}$ | OCS1, OSC2 | | 4.0 | — | 20.0 | MHz | |
| Subclock oscillator oscillation frequency | $\phi_{sub}$ | X1, X2 | | — | 32.768 | — | kHz | |
| Low-speed on-chip oscillator oscillation frequency | $\phi_{loco}$ | | | 25 | 125 | 225 | kHz | |
| System reference clock ($\phi_{base}$) cycle time | $t_{base}$ | | | 1 | — | 1 | $\phi_{OSC}$<br>$\phi_{sub}$<br>$\phi_{loco}$ | |
| System clock ($\phi$) cycle time | $t_{cyc}$ | | | 1 | — | 128 | $t_{base}$ | * |
| | | | | — | — | 320.0 | µs | |
| Bus master operation clock ($\phi$s) cycle time | $t_{bcyc}$ | | | 1 | — | 32 | $t_{cyc}$ | |
| | | | | — | — | 320.0 | µs | |
| Instruction cycle time | | | | 1 | — | — | $t_{cyc}$ | |
| Oscillation stabilization time (crystal resonator) | $t_{rc}$ | OSC1, OSC2 | | — | — | 6.5 | ms | Figure 28.3 |
| Oscillation stabilization time (ceramic resonator) | $t_{rc}$ | OSC1, OSC2 | | — | — | 6.5 | ms | |
| Oscillation stabilization time | $t_{rcx}$ | X1, X2 | | — | — | 2.0 | s | |
| External clock high width | $t_{CPH}$ | OSC1 | | 20.0 | — | — | ns | Figure 28.1 |
| External clock low width | $t_{CPL}$ | OSC1 | | 20.0 | — | — | ns | |

chanhvinh.com



| Item | Symbol | Applicable Pins | Test Conditions | Values Min. | Typ. | Max. | Unit | Reference Figure |
|---|---|---|---|---|---|---|---|---|
| External clock rise time | $t_{CPr}$ | OSC1 | | — | — | 10.0 | ns | Figure 28.1 |
| External clock fall time | $t_{CPf}$ | OSC1 | | — | — | 10.0 | ns | |
| $\overline{RES}$ pin low width | $t_{REL1}$ | $\overline{RES}$ | At power-on and in modes other than those below | 10.0 | — | — | ms | Figure 28.2 |
| | $t_{REL2}$ | | In active mode and sleep mode operation | 10.0 | — | — | µs | |
| Input pin high width | $t_{IH}$ | $\overline{NMI}$ | INCCR[5:4] = 00 | 300 | — | — | ns | Figure 28.4 |
| | | | INCCR[5:4] = 01 | 450 | — | — | ns | |
| | | | INCCR[5:4] = 10 | 800 | — | — | ns | |
| | | | INCCR[5:4] = 11 | 1500 | — | — | ns | |
| | | $\overline{IRQ0}$ to $\overline{IRQ3}$ | INCCR[1:0] = 00 | 300 | — | — | ns | |
| | | | INCCR[1:0] = 01 | 450 | — | — | ns | |
| | | | INCCR[1:0] = 10 | 800 | — | — | ns | |
| | | | INCCR[1:0] = 11 | 1500 | — | — | ns | |
| | | $\overline{IRQ4}$ to $\overline{IRQ7}$ | INCCR[3:2] = 00 | 300 | — | — | ns | |
| | | | INCCR[3:2] = 01 | 450 | — | — | ns | |
| | | | INCCR[3:2] = 10 | 800 | — | — | ns | |
| | | | INCCR[3:2] = 11 | 1500 | — | — | ns | |

chanhvinh.com



| Item | Symbol | Applicable Pins | Test Conditions | Values Min. | Values Typ. | Values Max. | Unit | Reference Figure |
|---|---|---|---|---|---|---|---|---|
| Input pin high width | $t_{IH}$ | FTIOA to FTIOD, FTIOA0 to FTIOD0, FTIOA1 to FTIOD1, FTIOA2 to FTIOD2, FTIOA3 to FTIOD3, FTCI, TRGC, $\overline{\text{TRCOI}}$, $\overline{\text{TRDOI_0}}$, $\overline{\text{TRDOI_1}}$, TCLKA, TCLKB, TGIOA, TGIOB, $\overline{\text{ADTRG1}}$, $\overline{\text{ADTRG_2}}$ | | 3 | — | — | $t_{cyc}$ | Figure 28.4 |
| Input pin low width | $t_{IL}$ | $\overline{\text{NMI}}$ | INCCR[5:4] = 00 | 300 | — | — | ns | Figure 28.4 |
| | | | INCCR[5:4] = 01 | 450 | — | — | ns | |
| | | | INCCR[5:4] = 10 | 800 | — | — | ns | |
| | | | INCCR[5:4] = 11 | 1500 | — | — | ns | |
| | | $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ3}}$ | INCCR[1:0] = 00 | 300 | — | — | ns | |
| | | | INCCR[1:0] = 01 | 450 | — | — | ns | |
| | | | INCCR[1:0] = 10 | 800 | — | — | ns | |
| | | | INCCR[1:0] = 11 | 1500 | — | — | ns | |
| | | $\overline{\text{IRQ4}}$ to $\overline{\text{IRQ7}}$ | INCCR[3:2] = 00 | 300 | — | — | ns | |
| | | | INCCR[3:2] = 01 | 450 | — | — | ns | |
| | | | INCCR[3:2] = 10 | 800 | — | — | ns | |
| | | | INCCR[3:2] = 11 | 1500 | — | — | ns | |

chanhvinh.com



| Item | Symbol | Applicable Pins | Test Conditions | Values Min. | Typ. | Max. | Unit | Reference Figure |
|---|---|---|---|---|---|---|---|---|
| Input pin low width | $t_{IL}$ | FTIOA to FTIOD, FTIOA0 to FTIOD0, FTIOA1 to FTIOD1, FTIOA2 to FTIOD2, FTIOA3 to FTIOD3, FTCI, TRGC, $\overline{\text{TRCOI}}$, $\overline{\text{TRDOI_0}}$, $\overline{\text{TRDOI_1}}$, TCLKA, TCLKB, TGIOA, TGIOB, $\overline{\text{ADTRG1}}$, $\overline{\text{ADTRG2}}$ | | 3 | — | — | $t_{cyc}$ | Figure 28.4 |

Note: * Determined by settings of the system clock control register (SYSCCR), power-down control register 1 (LPCR1), and power-down control register (LPCR2).

chanhvinh.com



## Table 28.5 Timing of I²C Bus Interface

$V_{CC}$ = 2.7 to 5.5 V, $V_{SS}$ = 0.0 V, VCC ≥ AVCC, Ta = –20 to +85 °C (N version)/ –40 to +85 °C (D version), unless otherwise indicated.

| Item | Symbol | Test Conditions | Values | | | Unit | Reference Figure |
|---|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | | |
| SCL input cycle time | $t_{SCL}$ | | $12t_{cyc}$ + 600 | — | — | ns | Figure 28.5 |
| SCL input high width | $t_{SCLH}$ | | $3t_{cyc}$ + 300 | — | — | ns | |
| SCL input low width | $t_{SCLL}$ | | $5t_{cyc}$ + 300 | — | — | ns | |
| SCL and SDA input fall time | $t_{Sf}$ | | — | — | 300 | ns | |
| SCL and SDA input spike pulse removal time | $t_{SP}$ | | — | — | $1t_{cyc}$ | ns | |
| SDA input bus-free time | $t_{BUF}$ | | $5t_{cyc}$ | — | — | ns | |
| Start condition input hold time | $t_{STAH}$ | | $3t_{cyc}$ | — | — | ns | |
| Retransmission start condition input setup time | $t_{STAS}$ | | $3t_{cyc}$ | — | — | ns | |
| Setup time for stop condition input | $t_{STOS}$ | | $3t_{cyc}$ | — | — | ns | |
| Data-input setup time | $t_{SDAS}$ | | $1t_{cyc}$ + 20 | — | — | ns | |
| Data-input hold time | $t_{SDAH}$ | | 0 | — | — | ns | |
| Capacitive load of SCL and SDA | Cb | | 0 | — | 400 | pF | |
| SCL and SDA output fall time | $t_{Sf}$ | Vcc = 4.0 to 5.5V | — | — | 250 | ns | |
| | | | — | — | 300 | ns | |

chanhvinh.com



### Table 28.6 Timing of Serial Communication Interface (SCI)

$V_{CC}$ = 2.7 to 5.5 V, $V_{SS}$ = 0.0 V, Ta = –20 to +85 °C (N version)/ –40 to +85 °C (D version), unless otherwise indicated.

| Item | | Symbol | Applicable Pins | Test Conditions | Values | | | | Reference Figure |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | Min. | Typ. | Max. | Unit | |
| Input clock cycle | Asynchronous | $t_{scyc}$ | SCK3 | | 4 | — | — | $t_{cyc}$ | Figure 28.6 |
| | Clocked synchronous | | | | 6 | — | — | $t_{cyc}$ | |
| Input clock pulse width | | $t_{SCKW}$ | SCK3 | | 0.4 | — | 0.6 | $t_{scyc}$ | |
| Transmit data delay time (clocked synchronous) | | $t_{TXD}$ | TXD | | — | — | 1 | $t_{cyc}$ | Figure 28.7 |
| Receive data setup time (clocked synchronous) | | $t_{RXS}$ | RXD | | 50.0 | — | — | ns | |
| Receive data hold time (clocked synchronous) | | $t_{RXH}$ | RXD | | 50.0 | — | — | ns | |

chanhvinh.com



## Table 28.7 Timing of Synchronous Serial Communication Unit (SSU)

$V_{CC}$ = 2.7 to 5.5 V, $V_{SS}$ = 0.0 V, Ta = –20 to +85 °C (N version)/ –40 to +85 °C (D version), $C_L$ = 100 pF, unless otherwise indicated.

| Item | | Symbol | Applicable Pins | Test Conditions | Values Min. | Typ. | Max. | Unit | Reference Figure |
|---|---|---|---|---|---|---|---|---|---|
| Clock cycle | | $t_{SUCYC}$ | SSCK | | 4 | — | — | $t_{CYC}$ | Figures 28.8 to 28.12 |
| Clock high pulse width | | $t_{HI}$ | SSCK | | 0.4 | — | 0.6 | $t_{SUCYC}$ | |
| Clock low pulse width | | $t_{LO}$ | SSCK | | 0.4 | — | 0.6 | $t_{SUCYC}$ | |
| Clock rise time | Master | $t_{RISE}$ | SSCK | | — | — | 1 | $t_{CYC}$ | |
| | Slave | | | | — | — | 1.0 | µs | |
| Clock fall time | Master | $t_{FALL}$ | SSCK | | — | — | 1 | $t_{CYC}$ | |
| | Slave | | | | — | — | 1.0 | µs | |
| Data input setup time | | $t_{SU}$ | SSO<br>SSI | | 100 | — | — | ns | |
| Data input hold time | | $t_H$ | SSO<br>SSI | | 1 | — | — | $t_{CYC}$ | |
| $\overline{SCS}$ setup time | Slave | $t_{LEAD}$ | $\overline{SCS}$ | | $1t_{CYC}$ + 50 | — | — | ns | |
| $\overline{SCS}$ hold time | Slave | $t_{LAG}$ | $\overline{SCS}$ | | $1t_{CYC}$ + 50 | — | — | ns | |
| Data output delay time | | $t_{OD}$ | SSO<br>SSI | | — | — | 1 | $t_{CYC}$ | |
| Slave access time | | $t_{SA}$ | SSI | | — | — | $1.5t_{CYC}$ + 100 | ns | |
| Slave out release time | | $t_{OR}$ | SSI | | — | — | $1.5t_{CYC}$ + 100 | ns | |

chanhvinh.com



## 28.5 A/D Converter Characteristics

### Table 28.8 A/D Converter Characteristics

$V_{CC}$ = 2.7 to 5.5 V, $V_{SS}$ = $AV_{SS}$ = 0.0 V, VCC ≥ AVCC, Ta = –20 to +85 °C (N version)/ –40 to +85 °C (D version), unless otherwise indicated.

| Item | Symbol | Applicable Pins | Test Conditions | Values | | | Unit | Notes |
|---|---|---|---|---|---|---|---|---|
| | | | | Min. | Typ. | Max. | | |
| Analog power supply voltage | AVcc | AVcc | | 2.7 | Vcc | 5.5 | V | *1 |
| Analog input voltage | $AV_{IN}$ | AN0 to AN11<br>AN0_2 to AN3_2 | | Vss – 0.3 | — | Avcc + 0.3 | V | |
| Analog power supply current | $AI_{OPE}$ | AVcc | AVcc = 5.0V<br>$f_{osc}$ = 20MHz | — | — | 2.0 | mA | |
| | $AI_{STOP1}$ | AVcc | | — | 50.0 | — | µA | *2 Reference value |
| | $AI_{STOP2}$ | AVcc | | — | — | 5.0 | µA | *3 |
| Analog input capacitance | $C_{AIN}$ | AN0 to AN11<br>AN0_2 to AN3_2 | | — | — | 30.0 | pF | |
| Allowable signal source impedance | $R_{AIN}$ | AN0 to AN11<br>AN0_2 to AN3_2 | | — | — | 5.0 | kΩ | |
| Resolution (data length) | | | | 10 | 10 | 10 | bit | |
| Conversion time | $t_{conv}$ | | | 2.0 | — | 43 | µs | |
| A/D conversion mode (single mode) | | AN0 to AN11<br>AN0_2 to AN3_2 | AVcc = 2.7 to 5.5V | | | | | |
| Nonlinearity error | | | | — | — | ±3.5 | LSB | |
| Offset error | | | | — | — | ±3.5 | LSB | |
| Full-scale error | | | | — | — | ±3.5 | LSB | |
| Quantization error | | | | — | — | ±0.5 | LSB | |
| Absolute accuracy | | | | — | — | ±4.0 | LSB | |

chanhvinh.com



Notes: 1. Set AVcc = Vcc when the A/D converter is not used.
2. $AI_{STOP1}$ is the current in active and sleep modes while the A/D converter is idle.
3. $AI_{STOP2}$ is the current at reset and in standby, subactive, and subsleep modes while the A/D converter is idle.

## 28.6 D/A Converter Characteristics

### Table 28.9 D/A Converter Characteristics

$V_{CC}$ = 2.7 to 5.5 V, $V_{SS}$ = $AV_{SS}$ = 0.0 V, VCC ≥ AVCC, Ta = –20 to +85 °C (N version)/ –40 to +85 °C (D version), unless otherwise indicated.

| Item | Symbol | Applicable Pins | Test Conditions | Values | | | Unit | Notes |
|---|---|---|---|---|---|---|---|---|
| | | | | Min. | Typ. | Max. | | |
| Analog power supply voltage | AVcc | AVcc | | 2.7 | Vcc | 5.5 | V | |
| Resolution | | | | 8 | 8 | 8 | bit | |
| Conversion time | | DA0 to DA1 | Load capacitance = 20 pF | — | — | 10.0 | µs | |
| Absolute accuracy | | DA0 to DA1 | Load resistance = 2 MΩ | — | ±2.0 | ±3.0 | LSB | |

chanhvinh.com



## 28.7 Flash Memory Characteristics

### Table 28.10 Flash Memory Characteristics

$V_{CC}$ = 2.7 to 5.5 V, $V_{SS}$ = 0.0 V, VCC ≥ AVCC, Ta = −20 to +85 °C (N version)/ −40 to +85 °C (D version), unless otherwise indicated.

| Item | Symbol | Test Conditions | Target Area | Values Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|---|
| Program/erase endurance*[1] | | | Programmable ROM | 1000*[2] | — | — | Times |
| | | | Data Flash | 10000*[2] | — | — | |
| Program time (per 4 bytes) | | | Programmable ROM | — | 150 | 3000 | µs |
| | | | Data Flash | — | 300 | 3000 | |
| Lock-bit program time | | | Programmable ROM | — | 70 | 3000 | µs |
| Erase time (per 1-block) | | | Programmable ROM | — | 300 | 3000 | ms |
| | | | Data Flash | — | 300 | 3000 | |
| Transition time to erase-suspend mode | $t_{d(SR-ES)}$ | | Programmable ROM | — | — | $3t_{bcyc}$ + 50 | µs |
| | | | Data Flash | | | | |
| Interval between start of erasure and request for suspension | | | Programmable ROM | 0 | — | — | µs |
| | | | Data Flash | | | | |
| Interval between restart of erasure and next request for suspension | | | Programmable ROM | 150 | — | — | µs |
| | | | Data Flash | | | | |
| Interval between suspension and restart of erasure | | | Programmable ROM | — | — | 50 | µs |
| | | | Data Flash | | | | |
| Program/erase voltage | | | Programmable ROM | 2.7 | — | 5.5 | V |
| | | | Data Flash | | | | |

chanhvinh.com



| Item | Symbol | Test Conditions | Target Area | Values Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|---|
| Read voltage | | | Programmable ROM | 2.7 | — | 5.5 | V |
| | | | Data Flash | | | | |
| Access states | | | Programmable ROM | 1 | — | — | $t_{bcyc}$ |
| | | | Data Flash | 2 | — | — | |
| Program/erase temperature | | | Programmable ROM | 0 | — | 60 | °C |
| | | | Data Flash | –20*[3] | — | 85 | |

Notes: 1. Definition of programming/erasure endurance
The programming and erasure endurance is defined on a per-block basis.
If the programming and erasure endurance is n (n = 1000, 10000), each block can be erased n times.
For example, if 1024 4-byte writes are performed to different addresses in data flash A, a 4-Kbyte block, and then the block is erased, the programming/erasure endurance still stands at one.
However, the same address must not be programmed more than once per erase operation (overwriting prohibited).
2. Endurance to guarantee all electrical characteristics after program and erase (1 to Min. value can be guaranteed).
3. –40°C for D version.

chanhvinh.com



## 28.8 Low-Voltage Detection Circuits Characteristics

### Table 28.11 Electrical Characteristics for Low-Voltage Detection Circuit 0

$V_{CC}$ = 2.7 to 5.5 V, $V_{SS}$ = 0.0 V, VCC ≥ AVCC, Ta = −20 to +85 °C (N version)/ −40 to +85 °C (D version), unless otherwise indicated.

| | | | Values | | | |
|---|---|---|---|---|---|---|
| **Item** | **Symbol** | **Test Conditions** | **Min.** | **Typ.** | **Max.** | **Unit** |
| Voltage-detection level | $V_{det0}$ | VD0LS1 = 0 | 2.10 | 2.35 | 2.50 | V |
| | | VD0LS1 = 1 | 3.50 | 3.80 | 4.10 | |
| Detection minimum pulse width of low-voltage detection circuit 0 | $t_{WLVD0}$ | | 20.0 | — | — | µs |
| Operating voltage minimum value for low-voltage detection circuit 0 | $V_{LVD0min}$ | | 1.8 | — | — | V |

chanhvinh.com



## Table 28.12 Electrical Characteristics for Low-Voltage Detection Circuit 1

$V_{CC}$ = 2.7 to 5.5 V, $V_{SS}$ = 0.0 V, VCC ≥ AVCC, Ta = –20 to +85 °C (N version)/ –40 to +85 °C (D version), unless otherwise indicated.

| Item | Symbol | Test Conditions | | Values Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|---|
| Voltage-detection level | $V_{det1}$ | VD1LS[3:0] = 0111 | Falling voltage | 2.70 | 3.07 | 3.45 | V |
| | | | Rising voltage | 3.00 | 3.30 | 3.70 | |
| | | VD1LS[3:0] = 1000 | Falling voltage | 2.80 | 3.22 | 3.60 | |
| | | | Rising voltage | 3.05 | 3.45 | 3.85 | |
| | | VD1LS[3:0] = 1001 | Falling voltage | 2.95 | 3.37 | 3.75 | |
| | | | Rising voltage | 3.20 | 3.60 | 4.00 | |
| | | VD1LS[3:0] = 1010 | Falling voltage | 3.10 | 3.52 | 3.90 | |
| | | | Rising voltage | 3.35 | 3.75 | 4.15 | |
| | | VD1LS[3:0] = 1011 | Falling voltage | 3.25 | 3.67 | 4.05 | |
| | | | Rising voltage | 3.50 | 3.90 | 4.30 | |
| | | VD1LS[3:0] = 1100 | Falling voltage | 3.40 | 3.82 | 4.20 | |
| | | | Rising voltage | 3.65 | 4.05 | 4.45 | |
| | | VD1LS[3:0] = 1101 | Falling voltage | 3.55 | 3.97 | 4.35 | |
| | | | Rising voltage | 3.80 | 4.20 | 4.60 | |
| | | VD1LS[3:0] = 1110 | Falling voltage | 3.70 | 4.12 | 4.50 | |
| | | | Rising voltage | 3.95 | 4.35 | 4.75 | |
| | | VD1LS[3:0] = 1111 | Falling voltage | 3.85 | 4.27 | 4.65 | |
| | | | Rising voltage | 4.10 | 4.50 | 4.90 | |
| Voltage hysteresis between detection for rising and falling cases | $V_{LVD1HYS}$ | | | — | 0.22 | — | V |
| Low-voltage detection circuit 1 self supply current | | Vcc = 5.0V | | — | 2.0 | — | µA |
| Waiting time until low-voltage detection circuit 1 operation starts | $t_{d(E-A)}$ | | | — | — | 50.0 | µs |
| Detection minimum pulse width of low-voltage detection circuit 1 | $t_{WLVD1}$ | | | 20.0 | — | — | µs |
| Operating voltage minimum value for low-voltage detection circuit 1 | $V_{LVD1min}$ | | | 2.7 | — | — | V |

chanhvinh.com



### Table 28.13 Electrical Characteristics for Low-Voltage Detection Circuit 2

$V_{CC}$ = 2.7 to 5.5 V, $V_{SS}$ = 0.0 V, VCC ≥ AVCC, Ta = –20 to +85 °C (N version)/ –40 to +85 °C (D version), unless otherwise indicated.

| Item | Symbol | Test Conditions | Values | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | |
| Voltage-detection level | $V_{det2}$ | Falling voltage<br>Rising voltage | 3.70<br>3.90 | 4.00<br>4.20 | 4.30<br>4.50 | V |
| Low-voltage detection circuit 2 self supply current | | Vcc = 5.0V | — | 2.0 | — | µA |
| Waiting time until low-voltage detection circuit 2 operation starts | $t_{d(E-A)}$ | | — | — | 50.0 | µs |
| Detection minimum pulse width of low-voltage detection circuit 2 | $t_{WLVD2}$ | | 20.0 | — | — | µs |
| Operating voltage minimum value for low-voltage detection circuit 2 | $V_{LVD2min}$ | | 2.7 | — | — | V |
| Voltage-detection level for external input pins | | | 1.20 | 1.33 | 1.45 | V |
| Input range for detection voltage of external inputs | | | 0 | — | 1/2Vcc | V |
| Input range for comparison voltage of external inputs | | | 0 | — | 1/2Vcc | V |

chanhvinh.com



## 28.9 Power-On Reset Function Characteristics

### Table 28.14 Power-On Reset Function Characteristics

$V_{CC}$ = 2.7 to 5.5 V, $V_{SS}$ = 0.0 V, VCC ≥ AVCC, Ta = −20 to +85 °C (N version)/ −40 to +85 °C (D version), unless otherwise indicated.

| Item | Symbol | Test Conditions | Values | | | Unit | Reference Figure |
|---|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | | |
| Power-on reset valid voltage | $V_{por}$ | | 0 | — | 0.2 | V | Figure 28.13 |
| External power $V_{CC}$ rise gradient | $t_{rth}$ | | 0.02 | — | 270 | V/msec | |

chanhvinh.com



## 28.10 Timing Charts

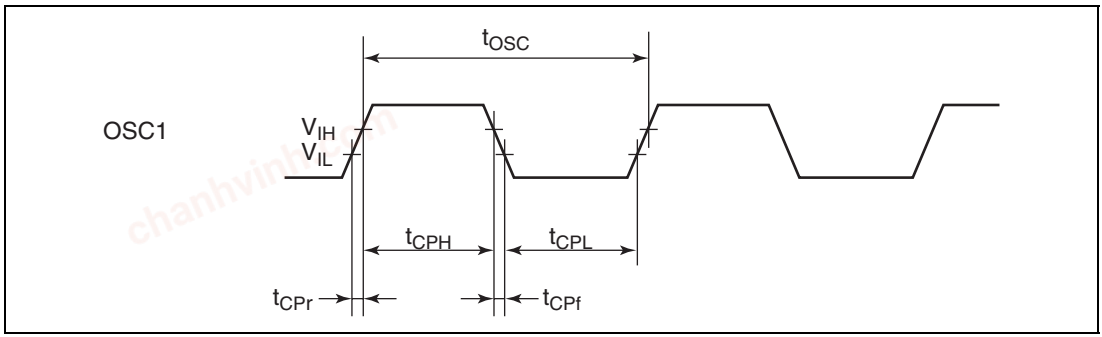



**Figure 28.1 System Clock Input Timing**

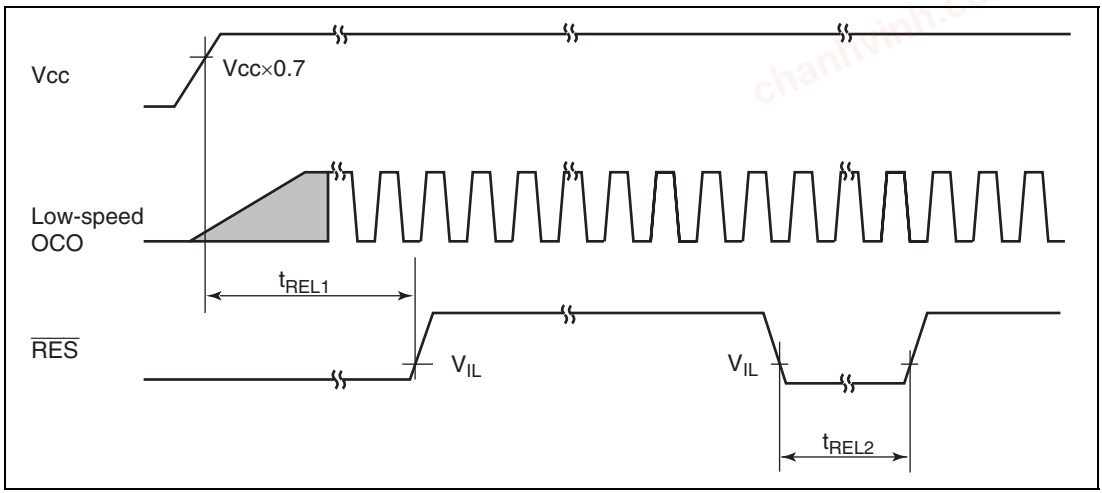



**Figure 28.2 $\overline{\text{RES}}$ Pin Low Width Timing**

chanhvinh.com



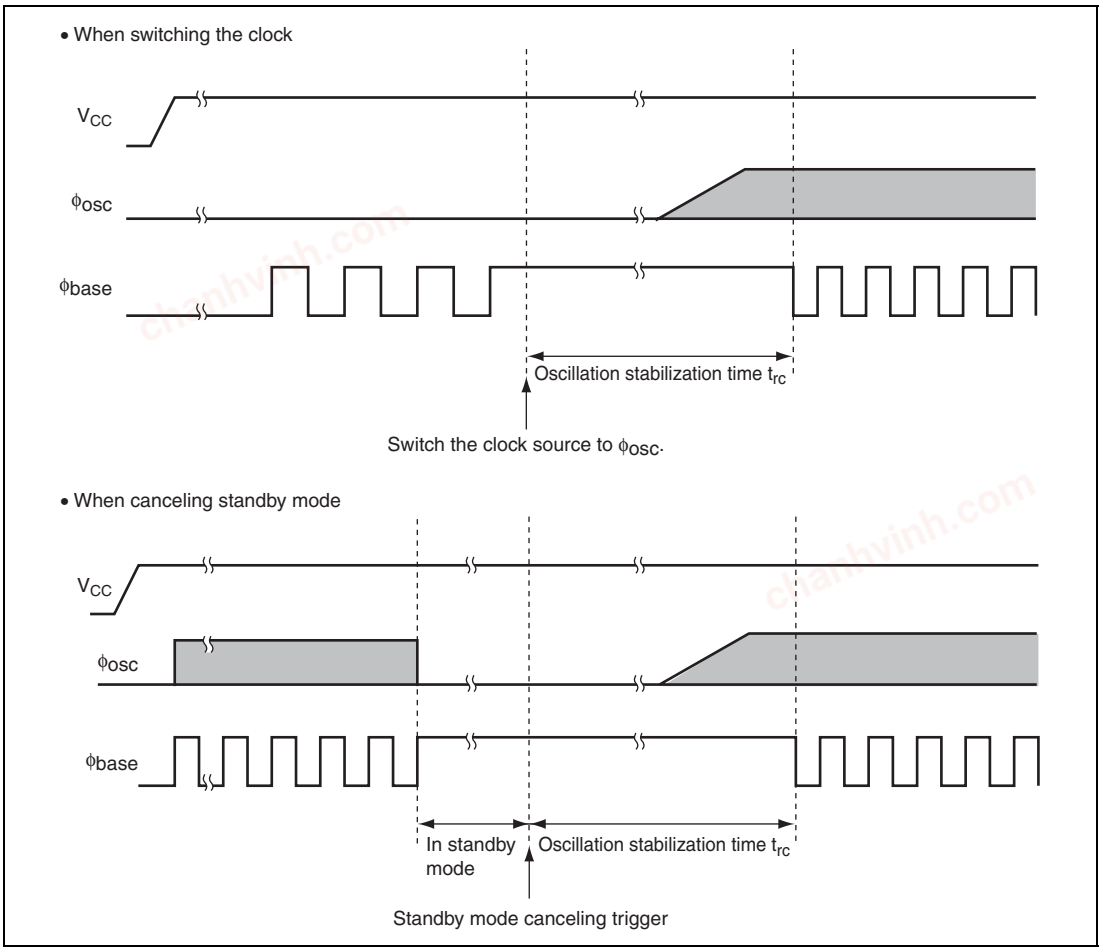



**Figure 28.3 Oscillation Stabilization Time Timing**

chanhvinh.com



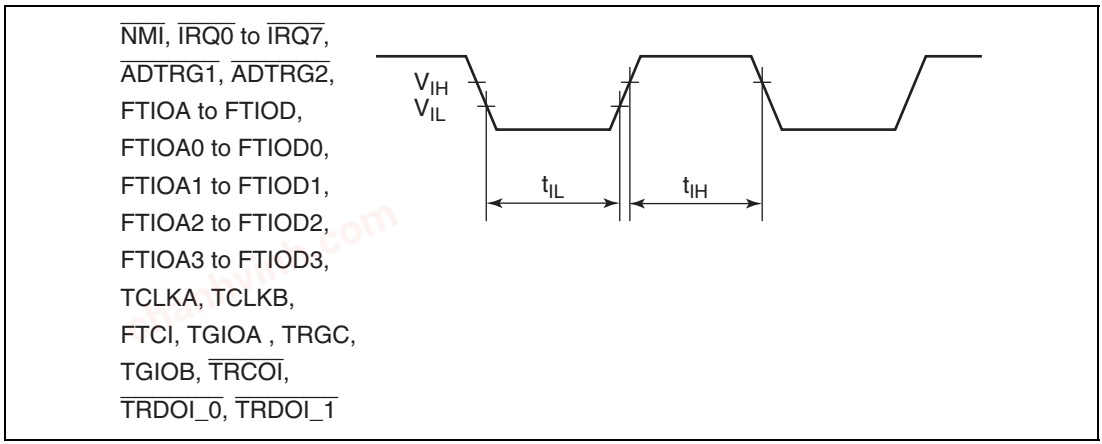



**Figure 28.4 Input Timing**

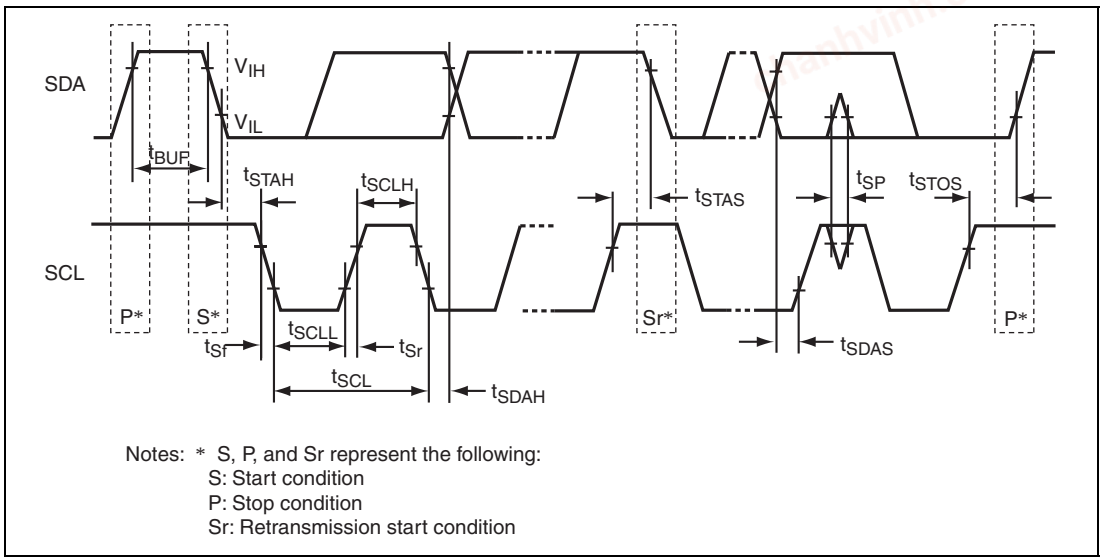



**Figure 28.5 I²C Bus Interface Input/Output Timing**

chanhvinh.com



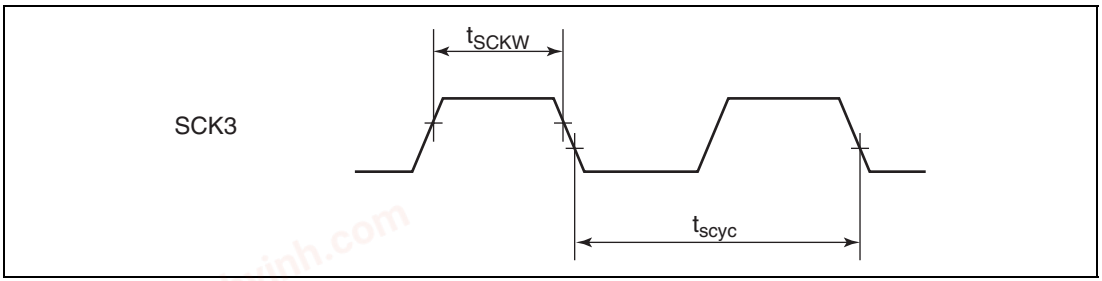
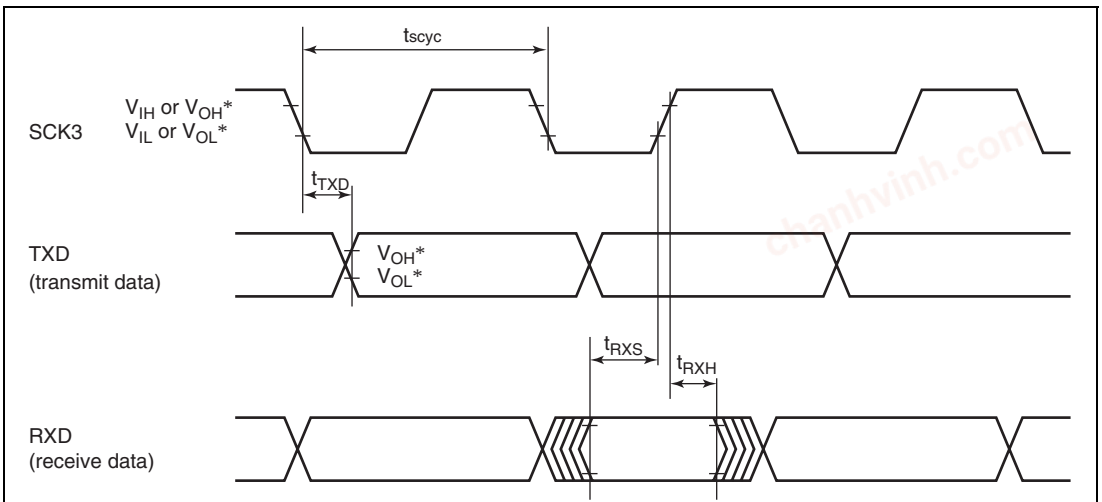



**Figure 28.6 SCK3 Input Clock Timing**




Note: * Output timing referenced levels

| | |
|---|---|
| Output high | $V_{OH}$ = 2.0V |
| Output low | $V_{OL}$ = 0.8V |

Load conditions are shown in figure 28.14.

**Figure 28.7 SCI Input/Output Timing in Clocked Synchronous Mode**

chanhvinh.com



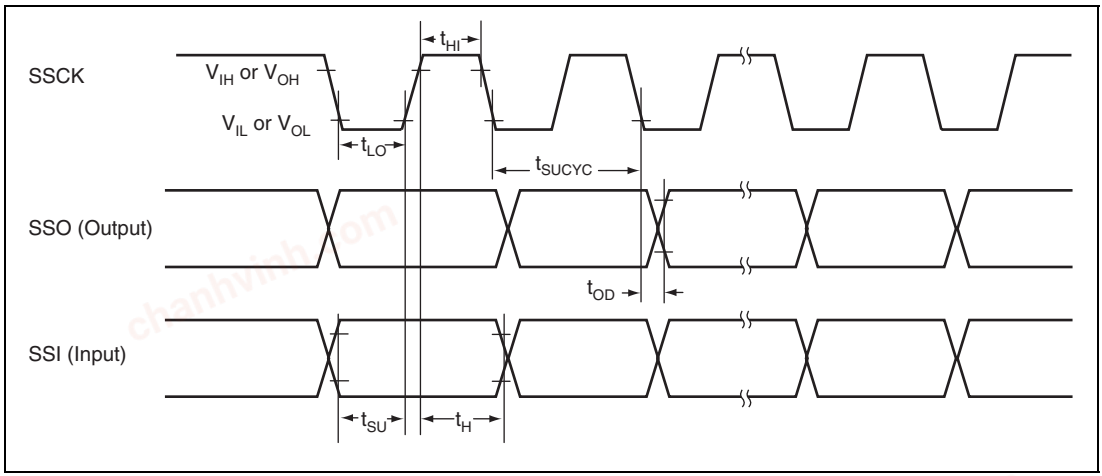



**Figure 28.8 SSU Input Timing in Clocked Synchronous Communication Mode**

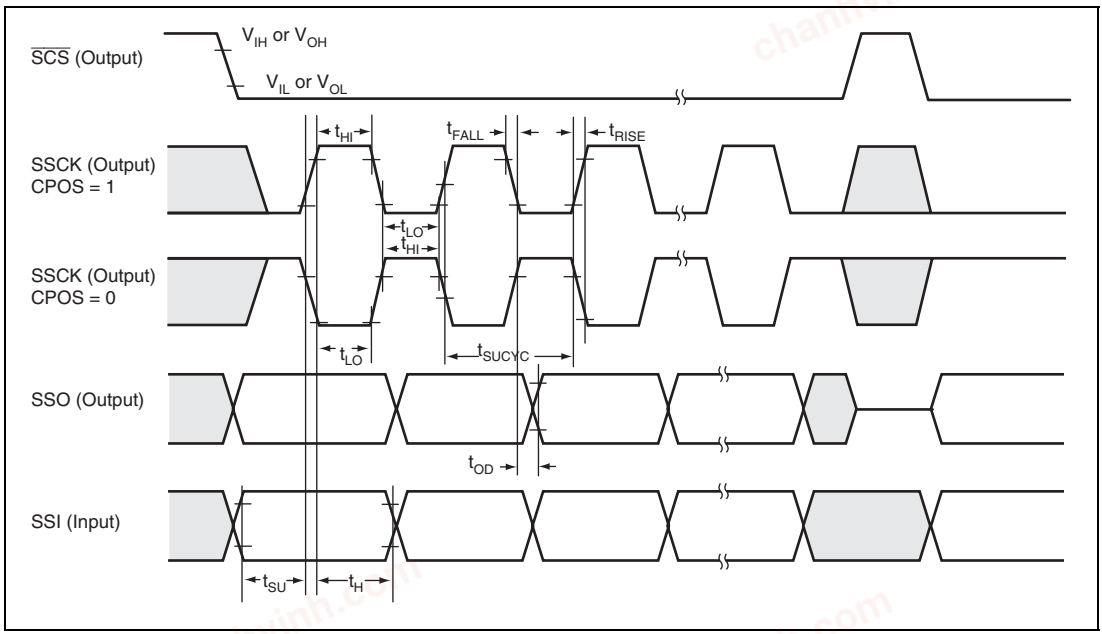



**Figure 28.9 SSU Input/Output Timing**
**(Four-Line Bus Communication Mode, Master, CPHS = 1)**

chanhvinh.com



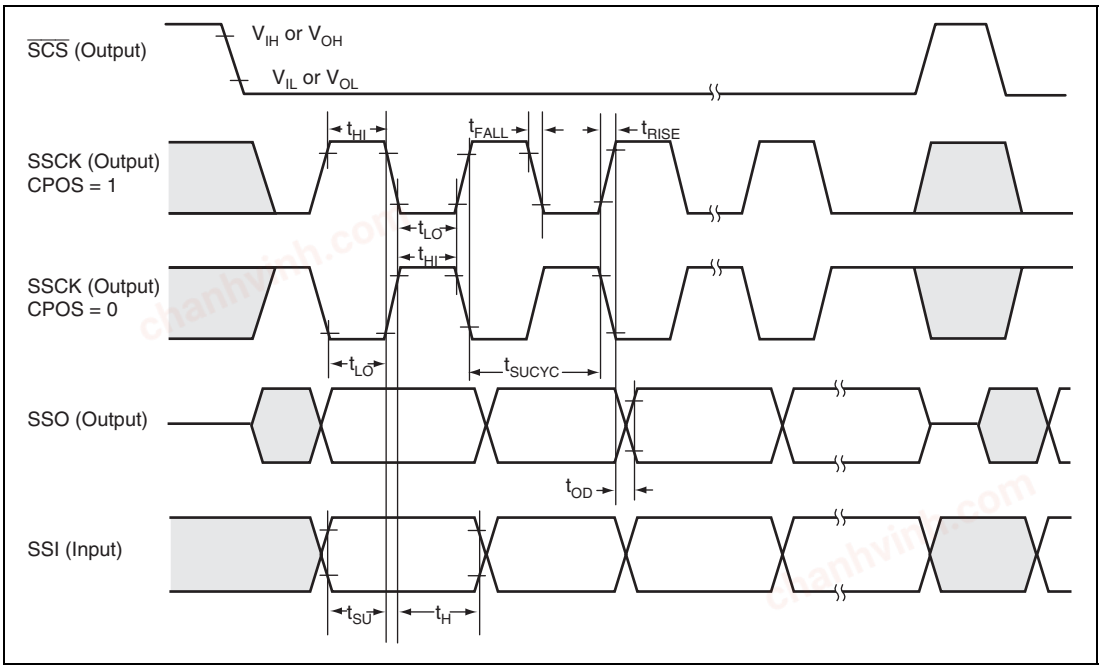



**Figure 28.10 SSU Input/Output Timing**
**(Four-Line Bus Communication Mode, Master, CPHS = 0)**

chanhvinh.com



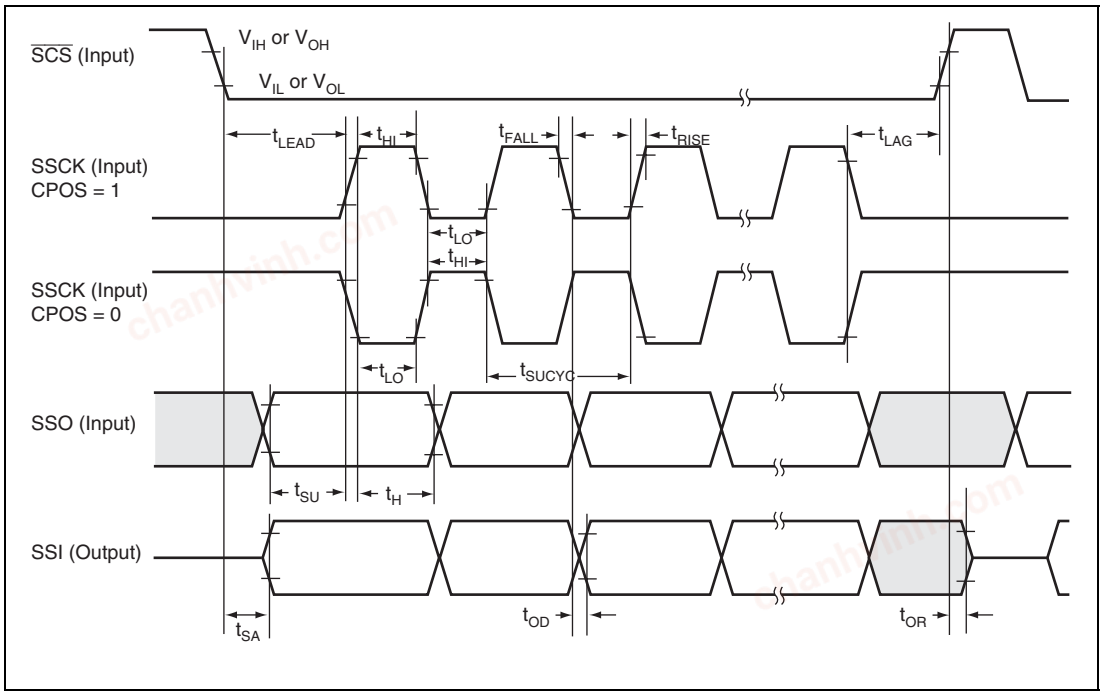



**Figure 28.11 SSU Input/Output Timing**
**(Four-Line Bus Communication Mode, Slave, CPHS = 1)**

chanhvinh.com



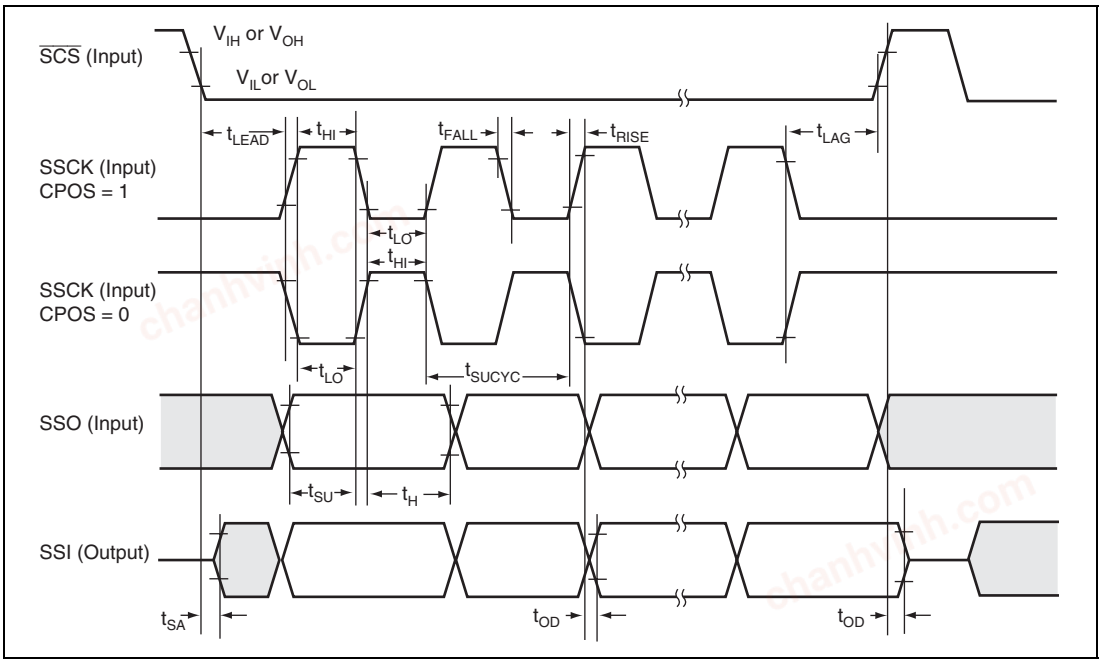



**Figure 28.12 SSU Input/Output Timing**
**(Four-Line Bus Communication Mode, Slave, CPHS = 0)**

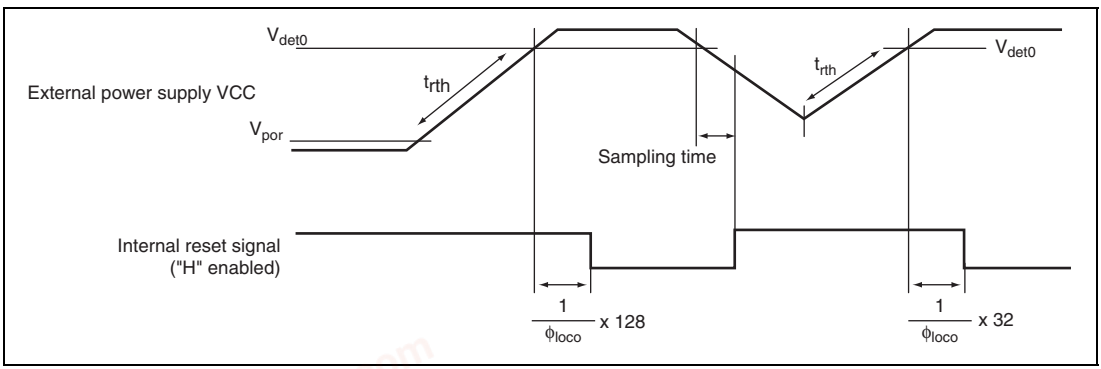



**Figure 28.13 Power-On Reset Timing**

chanhvinh.com



## 28.11 Output Load Circuit

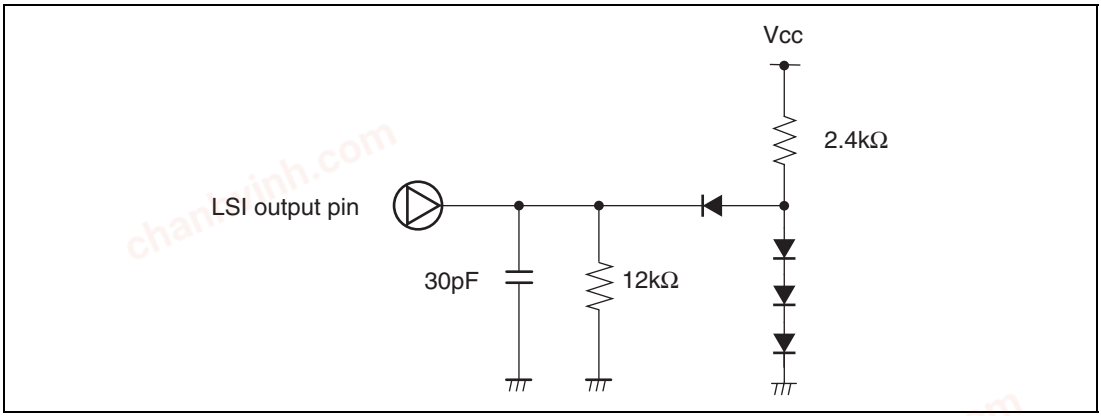



**Figure 28.14 Output Load Circuit**

chanhvinh.com



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

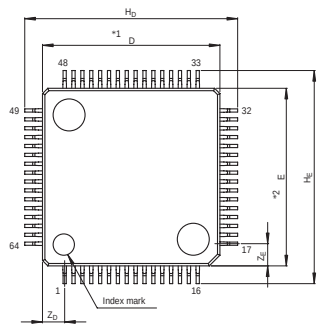
chanhvinh.com

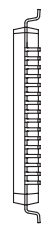


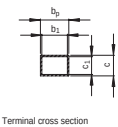
# Appendix

## A. Package Dimensions

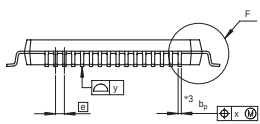
| JEITA Package Code | RENESAS Code | Previous Code | MASS[Typ.] |
|---|---|---|---|
| P-LQFP64-10x10-0.50 | PLQP0064KB-A | 64P6Q-A / FP-64K / FP-64KV | 0.3g |

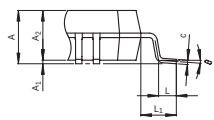




Terminal cross section



Detail F

NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

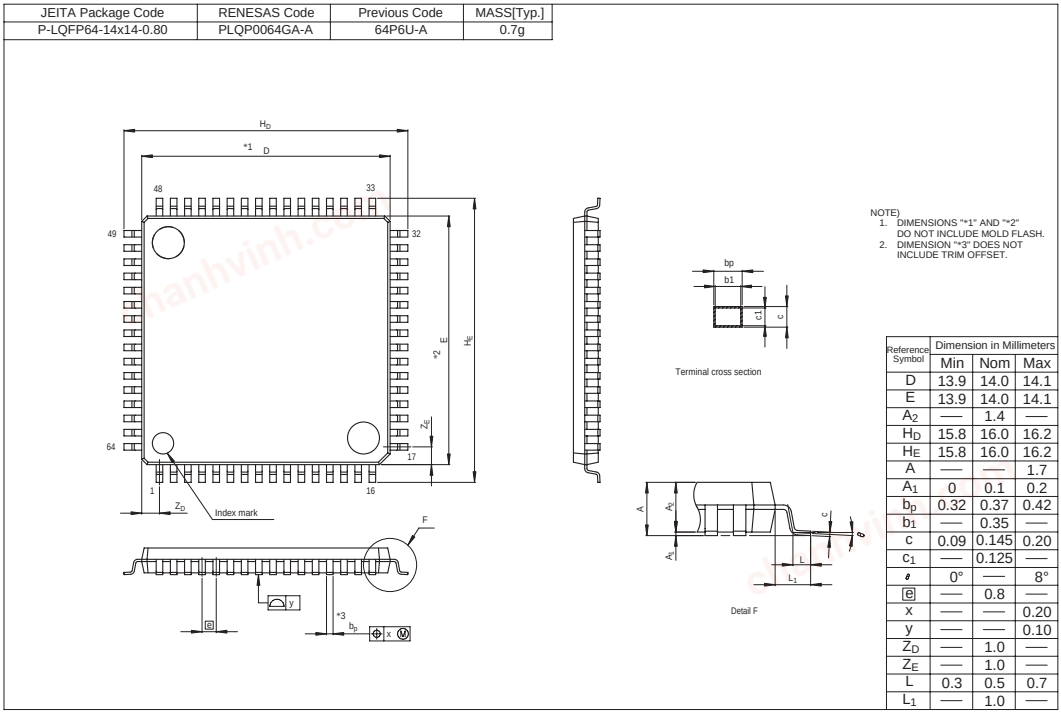
| Reference Symbol | Dimension in Millimeters | | |
|---|---|---|---|
| | Min | Nom | Max |
| D | 9.9 | 10.0 | 10.1 |
| E | 9.9 | 10.0 | 10.1 |
| $A_2$ | — | 1.4 | — |
| $H_D$ | 11.8 | 12.0 | 12.2 |
| $H_E$ | 11.8 | 12.0 | 12.2 |
| A | — | — | 1.7 |
| $A_1$ | 0.05 | 0.1 | 0.15 |
| $b_p$ | 0.15 | 0.20 | 0.25 |
| $b_1$ | — | 0.18 | — |
| c | 0.09 | 0.145 | 0.20 |
| $c_1$ | — | 0.125 | — |
| θ | 0° | — | 8° |
| e | — | 0.5 | — |
| x | — | — | 0.08 |
| y | — | — | 0.08 |
| $Z_D$ | — | 1.25 | — |
| $Z_E$ | — | 1.25 | — |
| L | 0.35 | 0.5 | 0.65 |
| $L_1$ | — | 1.0 | — |

**Figure A.1 Package Dimension (PLQP0064KB-A)**

chanhvinh.com



| JEITA Package Code | RENESAS Code | Previous Code | MASS[Typ.] |
|---|---|---|---|
| P-LQFP64-14x14-0.80 | PLQP0064GA-A | 64P6U-A | 0.7g |



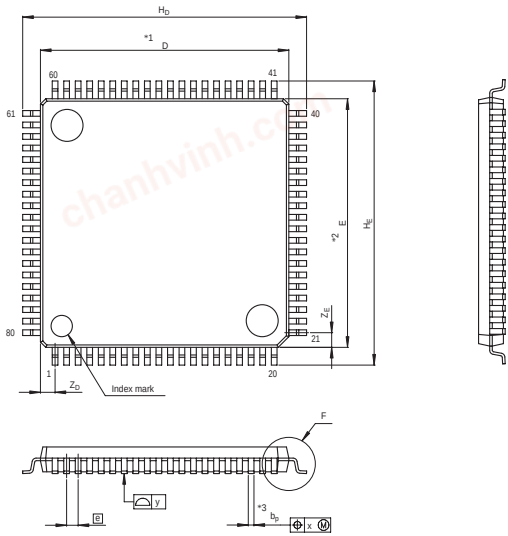

| Reference Symbol | Dimension in Millimeters | | |
|---|---|---|---|
| | Min | Nom | Max |
| D | 13.9 | 14.0 | 14.1 |
| E | 13.9 | 14.0 | 14.1 |
| $A_2$ | — | 1.4 | — |
| $H_D$ | 15.8 | 16.0 | 16.2 |
| $H_E$ | 15.8 | 16.0 | 16.2 |
| A | — | — | 1.7 |
| $A_1$ | 0 | 0.1 | 0.2 |
| $b_p$ | 0.32 | 0.37 | 0.42 |
| $b_1$ | — | 0.35 | — |
| c | 0.09 | 0.145 | 0.20 |
| $c_1$ | — | 0.125 | — |
| θ | 0° | — | 8° |
| e | — | 0.8 | — |
| x | — | — | 0.20 |
| y | — | — | 0.10 |
| $Z_D$ | — | 1.0 | — |
| $Z_E$ | — | 1.0 | — |
| L | 0.3 | 0.5 | 0.7 |
| $L_1$ | — | 1.0 | — |

**Figure A.2 Package Dimension (PLQP0064GA-A)**

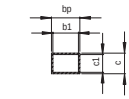
chanhvinh.com

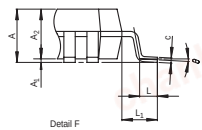


| JEITA Package Code | RENESAS Code | Previous Code | MASS[Typ.] |
|---|---|---|---|
| P-LQFP80-14x14-0.65 | PLQP0080JA-A | FP-80W / FP-80WV | 0.6g |






NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

| Reference Symbol | Dimension in Millimeters | | |
|---|---|---|---|
| | Min | Nom | Max |
| D | 13.9 | 14.0 | 14.1 |
| E | 13.9 | 14.0 | 14.1 |
| $A_2$ | — | 1.4 | — |
| $H_D$ | 15.8 | 16.0 | 16.2 |
| $H_E$ | 15.8 | 16.0 | 16.2 |
| A | — | — | 1.7 |
| $A_1$ | 0.05 | 0.1 | 0.15 |
| $b_p$ | 0.27 | 0.32 | 0.37 |
| $b_1$ | — | 0.30 | — |
| c | 0.09 | 0.145 | 0.20 |
| $c_1$ | — | 0.125 | — |
| θ | 0° | — | 8° |
| e | — | 0.65 | — |
| x | — | — | 0.13 |
| y | — | — | 0.10 |
| $Z_D$ | — | 0.825 | — |
| $Z_E$ | — | 0.825 | — |
| L | 0.35 | 0.5 | 0.65 |
| $L_1$ | — | 1.0 | — |

**Figure A.3 Package Dimension (PLQP0080JA-A)**

chanhvinh.com



# B. Handling of Unused Pins

Table B.1 shows the handling of unused pins.

**Table B.1 Handling of Unused Pins**

| Pin Name | Example of Handling Pins |
|---|---|
| $\overline{RES}$ | Connect this pin to $V_{CC}$ via a pull-up resister |
| $\overline{NMI}$ | Connect this pin to $V_{CC}$ via a pull-up resister |
| X1 | Connect this pin to $V_{SS}$ |
| X2 | Leave this pin open |
| Port 1 | Set the corresponding PMR bit or the PCR bit to 0 to set these pins in general purpose mode. Connect these pins to $V_{CC}$ via a resister (pull-up) or to $V_{SS}$ via a resister (pull-down), respectively. |
| Port 2 | |
| Port 3 | |
| Port 4 | |
| Port 5 | |
| Port 6 | |
| Port 7 | |
| Port 8 | |
| Port 9 | |
| Port A | |
| Port B | |
| Port J | |
| NC | Leave this pin open |

chanhvinh.com

# Main Revisions and Additions in this Edition

**Changes in Rev. 3.00 Compared to Rev. 2.00**

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| Section 1 Overview | 1 | Amended |
| 1.1 Features | | The on-chip peripheral function modules include a data transfer controller, event link controller, serial communication interface 3, $I^2C$ bus interface 2, synchronous serial communication unit, hardware LIN communication interface, A/D and D/A converters, low-voltage detection circuit, and versatile timers. … |
| 1.1.2 Overview of Functions | 3 | Amended |
| Table 1.1 Overview of Functions | | **Classification** / **Module/ Function**<br>Voltage detection / Low-voltage detection circuit (LVD)<br>Serial interfaces / Serial communication interface 3 (SCI3) |

chanhvinh.com

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 1.4 Pin Assignments<br>1.4.1 Pin Functions<br>Table 1.3 Pin Functions | 19 | Amended |

| | | Pin No. | |
|---|---|---|---|
| Classification | Symbol | H8S/20103 and H8S/20115 Groups | H8S/20203, H8S/20223, H8S/20215, and H8S/20235 Groups |
| I/O ports | P17 to P10*9 | 23 to 21,<br>54 to 52 | 28 to 25, 57, 56, 38,<br>37 |
| | P27 to P20 | 50 to 48, 51,<br>47 to 44 | 39 to 41, 51,<br>55 to 52 |
| | P37 to P30 | 13 to 20 | 17 to 24 |
| | P57 to P50 | 27 to 24<br>28 to 31 | 29 to 36 |
| | P67 to P60 | 40 to 37,<br>32 to 34, 36 | 49 to 42 |
| | P87 to P85 | 43 to 41 | 14 to 16 |
| | P97 to P90*10 | — | 61 to 58<br>65 to 62 |
| | PA7 to PA0*11 | 58 to 55 | 69 to 66<br>73 to 70 |
| | PB7 to PB0 | 2, 1, 64, 63,<br>59 to 62 | 2, 1, 80 to 75 |
| | PJ1 and PJ0 | 10, 11 | 10, 11 |

| | | |
|---|---|---|
| Section 2 CPU<br>2.9 Usage Notes<br>2.9.4 EEPMOV≧ Instruction | 64 | Amended<br>1. The EEPMOV≧ instruction performs a block transfer. … |

chanhvinh.com

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| Section 4 Interrupt Controller<br>4.2 Register Descriptions<br>4.2.1 Interrupt Control Register (INTCR) | 78 | Deleted<br>**Bit** — **Description**<br>2 — 0: ~~AD1 or~~ AD2 conversion is started at falling edge of $\overline{\text{ADTRG2}}$ input.<br>1: ~~AD1 or~~ AD2 conversion is started at rising edge of $\overline{\text{ADTRG2}}$ input. |
| 4.2.5 IRQ Status Register (ISR) | 85 | Description amended<br>[Setting condition]<br>• When the interrupt edge selected by ISCR occurs. |
| 4.2.7 Interrupt Vector Offset Register (VOFR) | 87 | Note added |
| Section 5 Clock Pulse Generator<br>5.2 Register Descriptions<br>5.2.5 Power-Down Control Register 3 (LPCR3) | 119, 120 | Amended<br>**Bit** — **Symbol** — **R/W**<br>5 — STBYINT — R<br>4 — SLEEPINT — R |
| Section 6 Power-Down Modes | 141 | Deleted<br>• Standby Mode<br>The CPU and all the on-chip peripheral modules are stopped. However, timer RE ~~(TMRE)~~ can operate when the realtime clock mode is selected. The watchdog timer (WDT) also operates when the low-speed OCO or subclock $\phi$sub is selected as the WDT clock source. |
| Section 7 ROM<br>7.3 CPU Reprogramming Mode<br>7.3.2 EW1 Mode | 161 | Amended<br>… If erasure has not been completed at the end of interrupt processing (FMERSF = 1 in FLMSTR), resume erasure by setting the FMSPREQ bit to 0. |

chanhvinh.com

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 7.4 Register Descriptions<br>7.4.2 Flash Memory Control Register 2 (FLMCR2) | 164 | Amended<br>Notes: 1. For programming the flash memory, set the FMSPEN bit to 1.<br>2. The FMRDYIE bit is cleared to 0 when the FMCMDEN bit changes from 0 to 1.<br>3. The FMBSYRDIE bit is cleared to 0 when the FMCMDEN bit changes from 0 to 1. |
| 7.6 Programming/Erasing<br>7.6.1 Software Commands<br>(2) Erasure | 215 | Amended<br>… To avoid this problem, add three NOP instructions immediately after the instruction that writes FMSPREQ = 0.<br>… |
| (7) Lock-Bit Reading Command | 222 | Amended<br>This command is used to read the value of the lock bit in flash memory.<br>Writing H'71 in the first command cycle and reading from the specifying block address (BA) in the second command cycle returns the value of the lock bit. If a word instruction is used for reading, the value of the lock bit will be reflected in bits 6 and 14 of the read-out word. If a byte instruction is used, the value of the lock bit will be reflected in bit 6. Execute the lock-bit reading command in EW0 mode. |
| 7.9 Usage Notes<br>(1) Prohibited Instruction | 231 | Amended<br>• TRAPA |

chanhvinh.com

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| Section 9 Peripheral I/O Mapping Controller<br>Table 9.1 Multiplexed Pin Functions (Ports 1, 2, 3, 5, and 6) | 237 | Amended |

| Group 1 | Pin Name | Function 4 |
|---|---|---|
| Port 1<br>Port 2<br>Port 3<br>Port 5<br>Port 6 | Pm7 | ~~SCL~~/SSI input/output |
| | Pm6 | ~~SDA~~/$\overline{\text{SCS}}$ input/output |

Notes: 1. The timer RC is not available on the H8S/20203, H8S/20223, H8S/20215, and H8S/20235 Groups; therefore, the function cannot be selected for these groups.

2. The SCL and SDA functions for the IIC2 module are not allocatable to pins other than P56 and P57.

| | | |
|---|---|---|
| 9.1 Register Description<br>9.1.2 Port Group 1<br>Peripheral Function Mapping Registers 1 to 4 (PMCRn1 to PMCRn4 (n = 1, 2, 3, 5, and 6))<br>(c) Port 1 Peripheral Function Mapping Register 3 (PMCR13) | 243 | Added<br>Notes: 1. For the H8S/20103 and H8S/20115 Groups, P14 is not provided and P14MD[2:0] are reserved. The initial value is B'001. The write value should be B'001. |

chanhvinh.com

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| (d) Port 1 Peripheral Function Mapping Register 4 (PMCR14)<br>(d) Port 2 Peripheral Function Mapping Register 4 (PMCR24) | 244<br>248 | Amended<br>**Bit** / **Description**<br>6 to 4: 100: SSI/~~SCL~~ input/output~~*1~~ (SSU/~~IIC2~~); 101: FTIOD1 input/output (timer RD_0); 110: $\overline{\text{ADTRG2}}$ input (AD_2); 111: Setting prohibited<br>2 to 0: 100: $\overline{\text{SCS}}$/~~SDA~~ input/output~~*1~~*2 (SSU/~~IIC2~~); 101: FTIOC1 input/output (timer RD_0); 110: $\overline{\text{ADTRG1}}$ input (AD_1); 111: Setting prohibited<br>Note~~s~~: ~~1.~~ ~~When the IICS/SSU is used as the IIC2 function, the SCL and SDA functions should be allocated to the P56 and P57 pins because SCL and SDA require buffers dedicated for IIC input/output. When the ICSU is used for the SSU function except *2, there is no restriction.~~<br>*: If the $\overline{\text{SCS}}$ output pin of the SSU is set, the NMOS open-drain output cannot be selected. |

chanhvinh.com

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| (d) Port 3 Peripheral Function Mapping Register 4 (PMCR34) | 252 | Amended<br>**Bit** / **Description**<br>6 to 4: :<br>100: SSI~~/SCL~~ input/output~~*1~~ (SSU/~~IIC2~~)<br>101: FTIOD1 input/output (timer RD_0)<br>110: $\overline{\text{ADTRG2}}$ input (AD_2)*1<br>111: Setting prohibited<br>2 to 0: :<br>100: $\overline{\text{SCS}}$/~~SDA~~ input/output~~*1~~*2 (SSU/~~IIC2~~)<br>101: FTIOC1 input/output (timer RD_0)<br>110: $\overline{\text{ADTRG1}}$ input (AD_1)<br>111: Setting prohibited<br>Notes: ~~1. When the IIC2/SSU is used as the IIC2 function, the SCL and SDA functions should be allocated to the P56 and P57 pins because SCL and SDA require buffers dedicated for IIC input/output. When the ICSSU is used for the SSU function except *3, there is no restriction.~~<br>1. This function cannot be selected for the H8S/20103, H8S/20203, H8S/20115, and H8S/20215 Groups.<br>2. If the $\overline{\text{SCS}}$ output pin of the SSU is set, the NMOS open-drain output cannot be selected. |
| (d) Port 5 Peripheral Function Mapping Register 4 (PMCR54) | 257 | Amended<br>Notes: 1. When the IIC2/SSU is used as the IIC2 function, the SCL and SDA functions can be allocated to the P56 and P57 pins only because SCL and SDA require buffers dedicated for IIC input/output. The PMC can not be used to allocate the SCL and SDA functions to other pins. When the IIC2/SSU is used for the SSU function except *2, there is no restriction. |

chanhvinh.com

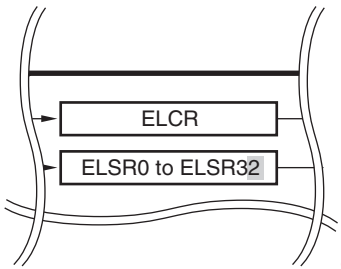
| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| (d) Port 6 Peripheral Function Mapping Register 4 (PMCR64) | 261 | Amended<br>**Bit** / **Description**<br>6 to 4: :<br>100: SSI/~~SCL~~ input/output~~*1~~ (SSU/~~IIC2~~)<br>101: FTIOD1 input/output (timer RD_0) (initial value)<br>110: $\overline{\text{ADTRG2}}$ input (AD_2) *1<br>111: Setting prohibited<br>2 to 0: :<br>100: $\overline{\text{SCS}}$/~~SDA~~ input/output~~*1~~*2 (SSU/~~IIC2~~)<br>101: FTIOC1 input/output (timer RD_0) (initial value)<br>110: $\overline{\text{ADTRG1}}$ input (AD_1)<br>111: Setting prohibited<br>Notes: ~~1. When the IIC2/SSU is used as the IIC2 function, the SCL and SDA functions should be allocated to the P56 and P57 pins because SCL and SDA require buffers dedicated for IIC input/output. When the ICSU is used for the SSU function except~~ ~~*3~~~~, there is no restriction.~~<br>1. This function cannot be selected for the H8S/20103, H8S/20203, H8S/20115, and H8S/20215 Groups.<br>2. If the $\overline{\text{SCS}}$ output pin of the SSU is set, the NMOS open-drain output cannot be selected. |
| Section 10 I/O Ports<br>10.8 Port A<br>10.8.7 Port Data Register A (PDRA) | 320 | Amended<br>PDRA is a register that stores output data for port A pins. When PCRA bits are set to 1, the values stored in PDRA are output.<br>…<br>When pins PA3 to PA0 are set as analog input channels by ADCSR and ADCR of the A/D converter, however, the corresponding PDRA bits are always read as 1 even if the respective PCRA bits are cleared to 0. |

chanhvinh.com

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 10.8.11 Port Data Register A (PDRA) | 324 | Amended<br>PDRA is a register that stores output data for port A pins. When PCRA bits are set to 1, the values stored in PDRA are output.<br>…<br>When the pins are set as analog input channels by ADCSR and ADCR of the A/D converter, however, the corresponding PDRA bits are always read as 1 even if the respective PCRA bits are cleared to 0. |
| 10.9 Port B<br>10.9.2 Port Data Register B (PDRB) | 328 | Added and amended<br>PDRB is a register that stores output data for port B pins. When PCRB bits are set to 1, the values stored in PDRB are output.<br>When PDRB is read while PCRB bits are set to 1, the values stored in PDRB are read. If PDRB is read while PCRB bits are cleared to 0, the pin states are read regardless of the value stored in PDRB.<br>When the pins are set as analog input channels by ADCSR and ADCR of the A/D converter, however, the corresponding PDRB bits are always read as 1 even if the respective PCRB bits are cleared to 0.<br>… |
| Section 11 Data Transfer Controller (DTC)<br>11.4 Location of Register Information and DTC Vector Table | 348 | Amended<br>The configuration of the vector address is a 2-byte unit. These two bytes specify the lower bits of the start address. Whenever the DTC is used, set the VOFR to H'0000 (its default value). |
| Table 11.3 Interrupt Sources, DTC Vector Addresses, and Corresponding DTCEs | 350 | Amended |

| Origin of Activation Source | Activation Source | Vector Address*1 |
|---|---|---|
| A/D converter unit 2*2 | IADEND_2 (conversion completion) | H'440 to H'441 |
| | IADCMP_2 (compare condition match) | H'442 to H'443 |

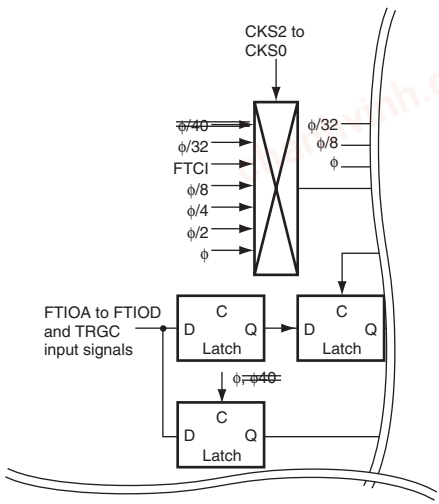
chanhvinh.com

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 11.8 Usage Notes<br>11.8.4 Limitation on Usage of the Interrupt Vector Offset Register (VOFR) | 368 | Added |
| Section 12 Event Link Controller<br>12.1 Overview<br>Figure 12.1 Block Diagram of Event Link Controller | 370 | Amended<br> |
| 12.2 Register Descriptions<br>12.2.2 Event Link Setting Registers 0 to 32 (ELSR0 to ELSR32) | 372 | Amended<br>… Table 12.1 shows the correspondence between ELSR0 to ELSR32 and the peripheral modules. … |
| 12.2.7 Port-Group Control Registers 1 and 2 (PGC1 and PGC2) | 379 | Amended<br>…<br>The correspondence between PGC and ports is shown in table 12.3. |
| 12.2.8 Port Buffer Registers 1 and 2 (PDBF1 and PDBF2) | 380 | Amended<br>PDBF is an 8-bit readable/writable register used in combination with PGR. For PDBF operations, see section 12.3, Operation. The correspondence of PDBF and PDR is shown in table 12.3. |
| 12.3 Operation<br>12.3.5 Port Operation upon Event Input and Event Generation | 392 | Amended<br>(1) Single-Ports and Port-Groups<br>… A port-group can be set by specifying any two or more bits in the port* to which an event can be connected using the PGC register. … |

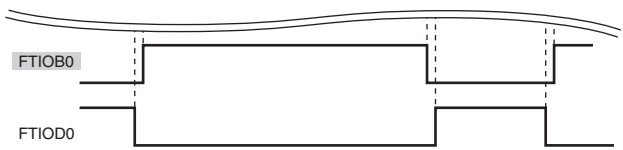
chanhvinh.com

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| Section 14 Timer RB<br>14.2 Register Descriptions<br>14.2.1 Timer RB Control Register (TRBCR) | 421 | Amended<br>**Bit** / **Symbol** / **Description**<br>1 / TCSTF / :<br>[Setting conditions]<br>• When 1 is written to TSTART and counting is started.<br>• The start of counting after ELOPA of the event link controller is selected counting by timer RB, the specified event is occurred, and the TSTART bit is set to 1.<br>: |
| 14.2.5 Timer RB Interrupt Request Status Register (TRBIR) | 426 | Section title amended |
| 14.3 Operation<br>14.3.3 Programmable One-Shot Generation Mode<br>Figure 14.3 Operation in Programmable One-Shot Generation Mode | 433 | Amended<br>Conditions for the above operation:<br>TRBPRE = H'01, TRBPR=H'01<br>TOPL = 0 and TOCNT = 0 in TRBIOC<br>INOSTG = 1 (TRGB pin one-shot trigger function is enabled)<br>INOSEG = 1 (Rising edge trigger) |
| 14.5 Usage Notes | 440 | Amended and added<br>11. When values for TRBPRE and TRBPR are successively read out from the same register, allow at least two cycles of the clock source for counting as the minimum interval for reading<br>12. When timer RB selects the timer RA underflow as a count source, be sure to set timer RA to timer mode, pulse output mode, or event counter mode. |

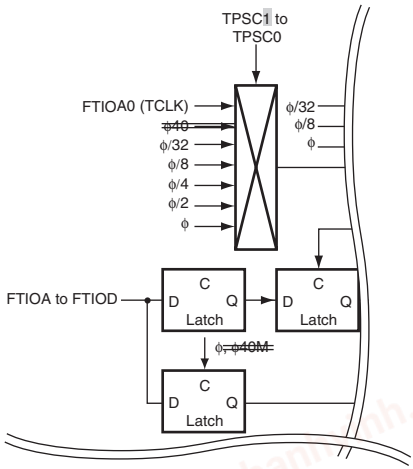
chanhvinh.com

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| Section 15 Timer RC<br>15.2 Register Descriptions<br>15.2.12 General Registers A, B, C, and D (GRA, GRB, GRC, and GRD) | 462 | Amended<br>… If the corresponding interrupt-enable bit (the IMIEA, IMIEB, IMIEC, or IMIED bit) in TRCIER is set to 1 at this time, an interrupt request is generated. … |
| 15.3 Operation<br>15.3.4 Digital Filtering Function for Input Capture Inputs | 481 | Amended<br>Input signals on the FTIOA to FTIOD and TRGC pin can be input via the digital filters. The digital filter includes three latches connected in series and a match detector circuit. |
| Figure 15.22 Block Diagram of Digital Filter | 481 | Deleted<br> |
| 15.3.7 Operation through an Event Link<br>(2) Counting Event | 486 | Added<br>… When the event specified in ELSR2 occurs, event counter operation proceeds with that event as the source to drive counting, regardless of the setting of the CKS[2:0] bits in TRCCR1 and the CTS bit in TRCMR. … |

chanhvinh.com

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| Section 16 Timer RD<br>16.2 Register Descriptions<br>16.2.12 Timer RD I/O Control Registers (TRDIORA and TRDIORC) | 526 | Deleted<br>Notes: 1. When a GR register functions as a buffer register for a paired GR register, the settings in the IOA2 and IOB2 bits in TRDIORA and the IOC2 and IOD2 bits in TRDIORC of both registers should be the same. ~~The IOA3 bit exists only in TRDIORA_0.~~<br>2. In PWM mode, PWM3 mode, complementary PWM mode, and reset synchronous PWM mode, the settings of TRDIORA are invalid. |
| 16.3.9 Buffer Operation | 581 | Amended<br>(7) Examples of Buffer Operation<br>…<br>Counter clearing by input capture B has been set for TRDCNT, and falling edges have been selected as the FTIOB pin input capture input edge. And both rising and falling edges have been selected as the FTIOA pin input capture input edge.<br>… |
| Figure 16.49 Buffer Operation (4) (Buffer Operation in Complementary PWM Mode CMD1 =1, CMD0 = 0) | 584 | Amended<br> |
| 16.3.10 Timer RD Output Timing<br>(2) Output Disable Timing of Timer RD by External Trigger | 586 | Amended<br>When $\overline{\text{TRDOI}}$ is selected for inputs and low level is input to $\overline{\text{TRDOI}}$, the master enable bit in TRDOER1 is set to 1 and the output of timer RD will be disabled. |

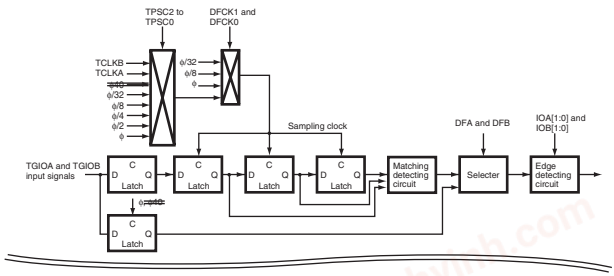
chanhvinh.com

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 16.3.11 Digital Filtering Function for Input Capture Inputs<br>Figure 16.54 Block Diagram of Digital Filter | 588 | Deleted<br> |
| 16.3.14 Operation by Event Clear<br>(2) Counting Event | 593 | Amended<br>The counting of events by timer RD can be selected by ELOPA and ELOPB of the ELC. When the event specified in ELSR3 and ELSR4 occurs, event counter operation proceeds with that event as the source to drive counting, regardless of the setting of the TPSC[2:0] bits in TRDCR~~1~~ and the STR1 and STR0 bits in TRDSTR. When the value of the counter is read, the value read out is the actual number of input events. |
| Section 17 Timer RE<br>17.3 Operation of Realtime Clock Mode<br>17.3.4 Operation in Realtime Clock Mode<br>Figure 17.5 Example of Realtime Clock Mode Operation | 622 | Amended |

chanhvinh.com

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 17.4 Operation of Output Compare Mode | 623 | Amended<br>Writing 0 to the RCS3 bit in TRECSR sets the timer RE in output compare mode and causes it to operate as a counter provided with an 8-bit compare match function. Four count sources can be selected. When used in output compare mode, the timer RE should be initialized ~~in reference to figure 17.3~~. When timer RE is initialized, first select the output-compare mode through the RCS3 bit in TRECSR and then perform the initial setting procedure shown in figure 17.3. |
| Section 18 Timer RG<br>18.3 Operation<br>• TGIOA pin<br>• TGIOB pin | 643, 644 | Tables amended |
| 18.3.1 Timer Mode<br>Figure 18.2 Example of Setting Procedure for Waveform Output by Compare Match | 645 | Replaced |
| Figure 18.8 Input Capture Input Signal Timing | 650 | Replaced |
| 18.3.2 PWM Mode<br>Figure 18.9 Example of PWM Mode Setting Procedure | 651 | Replaced |
| 18.3.3 Phase Counting Mode<br>Figure 18.17 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode | 660 | Amended<br>Pulse width: 3 states or more |
| 18.3.4 Buffer Operation | 660 | Amended<br>Buffer operation differs according to whether GR has been designated as an output compare register or an input capture register.<br>Table 18.10 shows the register combinations used in buffer operation. |
| | 661 | Amended<br>(2) When GR is an input capture register |

chanhvinh.com

<table>
<tr><th>Item</th><th>Page</th><th>Revision (See Manual for Details)</th></tr>
<tr><td>18.3.5 Operation through an Event Link<br>(2) Counting Event</td><td>663</td><td>Added<br>… When the event specified in ELSR8 occurs, event counter operation proceeds with that event as the source to drive counting, regardless of the setting of TPSC[2:0] bits in TRGCR and the STR bit in TRGMDR. …</td></tr>
<tr><td>18.3.6 Digital Filtering Function for Input Capture Inputs<br>Figure 18.22 Block Diagram of Digital Filter</td><td>664</td><td>Deleted<br>
</td></tr>
<tr><td>Section 19 Watchdog Timer (WDT)<br>19.2 Register Descriptions</td><td>667</td><td>Amended<br>• Timer interrupt control register WD (TICRWD)</td></tr>
<tr><td>19.2.1 Timer Control/Status Register WD (TCSRWD)</td><td>668</td><td>Amended<br><table>
<tr><th>Bit</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>3</td><td>Timer mode register WD lockdown</td><td>The TMWD register is write-protected when this bit is 1. Once this bit is set to 1, this bit can be cleared only by a reset.<br>:</td></tr>
<tr><td>2</td><td>Timer mode register WD write inhibit</td><td>:<br>[Clearing condition]<br>• When 0 is written to TMWI while TCSRWE is 1</td></tr>
</table></td></tr>
<tr><td>Section 20 Serial Communication Interface 3 (SCI3, IrDA)</td><td>677</td><td>Added<br>SCI3_2 provides IrDA (Infrared Data Association) communication waveform transmission/reception according IrDA standard version 1.0.</td></tr>
</table>

chanhvinh.com

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 20.2 Register Descriptions | 682 | Amended<br>Channel 2<br>• Receive shift register_2 (RSR_2)<br>• Receive data register_2 (RDR_2)<br>• Transmit shift register_2 (TSR_2)<br>• Transmit data register_2 (TDR_2)<br>• Serial mode register_2 (SMR_2)<br>• Serial control register_2 (SCR3_2)<br>• Serial status register_2 (SSR_2)<br>• Bit rate register_2 (BRR_2)<br>• Sampling mode register_2 (SPMR_2)<br>• IrDA control register_2 (IrCR_2)<br><br>Channel 3<br>• Receive shift register_3 (RSR_3)<br>• Receive data register_3 (RDR_3)<br>• Transmit shift register_3 (TSR_3)<br>• Transmit data register_3 (TDR_3)<br>• Serial mode register_3 (SMR_3)<br>• Serial control register_3 (SCR3_3)<br>• Serial status register_3 (SSR_3)<br>• Bit rate register_3 (BRR_3)<br>• Sampling mode register_3 (SPMR_3) |
| 20.6.1 Transmission | 721 | Amended<br>… The high-level pulse can be selected using the IrCKS2 to IrCKS0 bits in IrCR. |
| 20.6.3 High-Level Pulse Width Selection | 722 | Amended<br>Table 20.7 shows possible settings for bits IrCKS2 to IrCKS0 (minimum pulse width), and this LSI's operating frequencies and bit rates, … |
| Table 20.7 Settings of Bits IrCKS2 to IrCKS0 | 722 | Table title amended |

chanhvinh.com

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 20.8 Interrupt Requests | 724 | Amended<br>…<br>When the RDRF flag in SSR is set to 1, a RXI interrupt request is generated. When any of the OER, PER and FER flags is set to 1, an ERI interrupt request is generated. The DTC can be activated to perform data transfers with the RXI interrupt request. The RDRF flag is automatically cleared to 0 by the DTC data transfer. |
| 20.9 Usage Notes<br>20.9.6 Restrictions on Using DTC<br>Figure 20.23 Example of DTC Transmission in Clock Synchronous Mode | 727 | Amended<br>[Before amendment]<br>SCK<br>[After amendment]<br>SCK3 |
| Section 21 $I^2C$ Bus Interface 2 (IIC2)<br>21.1 Features<br>Table 21.1 Pin Configuration | 731 | Note added<br>Note: When the IIC2 function is selected, use the PMC to allocate the SCL and SDA pin functions to P57 and P56, respectively. |

chanhvinh.com

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 21.2 Register Descriptions<br>21.2.3 I²C Bus Control Register 2 (ICCR2) | 735, 736 | Added |

| Bit | Symbol | Description |
|---|---|---|
| 7 | BBSY*1*3 | This bit enables to confirm whether the I²C bus is… |
| 6 | SCP*3 | The SCP bit controls the issue of start/stop… |
| 5 | SDAO*3 | This bit is used with SDAOP (bit 4) when modifying output level of SDA. This bit should not be manipulated during transfer. Writing 1 to the IICRST bit also sets this bit to 1.<br>: |
| 3 | SCLO | This bit monitors SCL output level. When reading and SCLO is 1, SCL pin outputs high. When reading and SCLO is 0, SCL pin outputs low. Writing 1 to the IICRST bit also sets this bit to 1. |

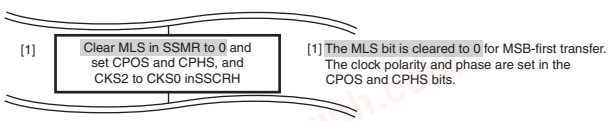
Notes: 1. In standby mode, the BBSY bit in ICCR2 is reset.
2. Clear IICRST to 0 by software since this bit is not cleared automatically.
3. Writing to this bit is invalid during a reset due to setting the IICRST bit in ICCR2 to 1.

| | | |
|---|---|---|
| 21.2.6 I²C Bus Status Register (ICSR) | 741 | Added |

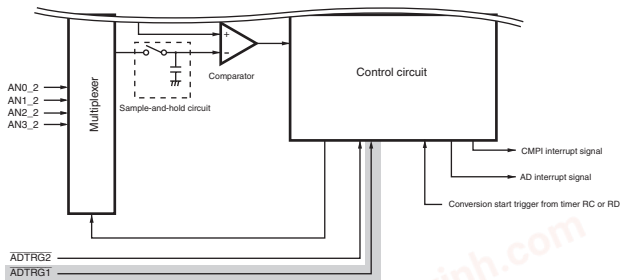
| Bit | Description |
|---|---|
| 7 | [Setting conditions]<br>:<br>• When 1 is written to IICRST in ICCR2 in master transmit mode and slave transmit mode |

| | | |
|---|---|---|
| 21.6 Usage Notes<br>21.6.3 Note regarding Master Receive Mode of I²C-Bus Interface Mode | 765 | Added |

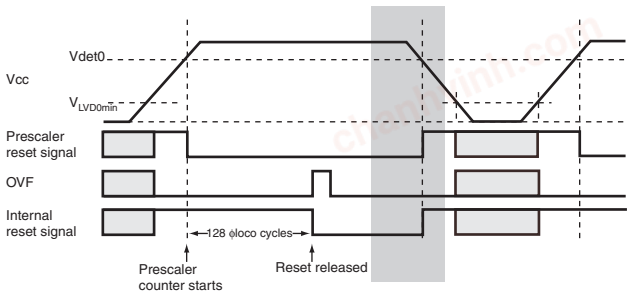
chanhvinh.com

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 21.6.4 Note on Access to ICE in ICCR1 and IICRST in ICCR2 during $I^2C$ Bus Operation | 766 | Added |
| Section 22 Synchronous Serial Communication Unit (SSU)<br>22.2 Register Descriptions<br>22.2.9 SS Transmit Data Register (SSTDR) | 780 | Added<br>… If the MLS bit in SSMR is set to 1 and when the data is written to SSTDR, the MSB/LSB inverted data is read. SSTDR is initialized to H'FF. In standby mode, SSTDR is initialized. |
| 22.2.10 SS Shift Register (SSTRSR) | 780 | Deleted<br>SSTRSR is a shift register that transmits and receives serial data. ~~When transmit data is transferred from SSTDR to SSTRSR, bit 0 in SSTDR is transferred to bit 0 in SSTRSR while the MLS bit in SSMR is 0 (LSB-first transfer) and bit 7 in SSTDR is transferred to bit 0 in SSTRSR while the MLS bit in SSMR is 1 (MSB-first transfer).~~ SSTRSR is not directly readable or writable from the CPU. In standby mode, SSTRSR is initialized. |
| 22.3 Operation<br>22.3.2 Relationship between Clock Polarity and Phase, and Data | 781 | Amended<br>…<br>MSB-first transfer or LSB first transfer can be selected by the setting of the MLS bit in SSMR. When the MLS bit is 1, transfer is started from LSB to MSB. When the MLS bit is 0, transfer is started from MSB to LSB. |
| 22.3.6 Operation in Four-Line Bus Communication Mode | 791 | Amended<br>…<br>In four-line bus communication mode, transfer is performed in MSB-first order while the MLS bit in SSMR is 0. |
| Figure 22.10 Initialization in Four-Line Bus Communication Mode | 792 | Amended<br> |

chanhvinh.com

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| Section 23 Hardware LIN<br>23.3 Operation<br>23.3.3 Bus Conflict Detection Function | 811 | Deleted<br>The hardware LIN interface can detect bus conflicts if SCI3_1 is enabled for transmission (TE bit in SCR3~~_1~~ register is 1). |
| Section 24 A/D Converter<br>24.1 Features<br>Figure 24.2 Block Diagram of A/D Converter (Unit 2) | 817 | Added<br> |
| Table 24.1 Pin Configuration | 818 | Notes amended<br>Notes: 1. Not supported in the H8S/20103 and H8S/20115 Groups.<br>2. Supported only in the H8S/20223 and H8S/20235 Groups.<br>3. This pin can also be used as the trigger-signal input for unit 2. |
| 24.2 Register Description | 819 | Amended<br>Unit 1:<br>• Compare control/status register (CMPCSR)<br>Unit 2:<br>• Compare control/status register_2 (CMPCSR_2) |
| 24.2.2 A/D Control/Status Register (ADCSR) | 823 | Added<br>• ADST bit (A/D start)<br>…<br>To select AN0 and AN0_2 as the channels for conversion, specify analog input operation for the pins by setting the PMRA2 and PMRA3 bits. |
| 24.2.7 Compare Voltage Registers H and L (CMPVALH and CMPVALL) | 830 | Register name amended |

chanhvinh.com

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 24.5 Compare Mode Operation<br>24.5.4 External Trigger Input Timing | 841 | Amended<br>A/D conversion can be externally triggered. When the EXTRGS, TRGS1, and TRGS0 bits are set to B'001 in ADCR, external trigger input is enabled at the ADTRG pin. |
| Section 26 Low-Voltage Detection Circuits<br>26.2 Register Descriptions<br>26.2.1 Low-Voltage Detection Circuit Control Protect Register (VDCPR) | 859 | Amended<br>**Bit** / **Symbol** / **R/W**<br>7 / WRI / W |
| 26.3 Operation<br>26.3.1 Power-On Reset Function<br>Figure 26.5 Operational Timing of Power-On Reset | 868 | Amended<br> |

chanhvinh.com

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| Section 27 List of Registers<br>27.1 Register Addresses (Address Order) | 885, 886 | Amended |

| Register Name | Module |
|---|---|
| Serial mode register | SCI3 |
| Bit rate register | SCI3 |
| Serial control register 3 | SCI3 |
| Transmit data register | SCI3 |
| Serial status register | SCI3 |
| Receive data register | SCI3 |
| Sampling mode register | SCI3 |
| Serial mode register_2 | SCI3_2 |
| Bit rate register_2 | SCI3_2 |
| Serial control register 3_2 | SCI3_2 |
| Transmit data register_2 | SCI3_2 |
| Serial status register_2 | SCI3_2 |
| Receive data register_2 | SCI3_2 |
| Sampling mode register_2 | SCI3_2 |
| Serial mode register_3 | SCI3_3 |
| Bit rate register_3 | SCI3_3 |
| Serial control register 3_3 | SCI3_3 |
| Transmit data register_3 | SCI3_3 |
| Serial status register_3 | SCI3_3 |
| Receive data register_3 | SCI3_3 |
| Sampling mode register_3 | SCI3_3 |

894 Amended

| Register Name | Abbreviation |
|---|---|
| Timer interrupt control/~~status~~ register WD | TICRWD |
| Timer RE second data register/counter data register | TRESEC |
| Timer RE minute data register/compare data register | TREMIN |
| Time RE clock source select register | TRECSR |

chanhvinh.com

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 27.1 Register Addresses (Address Order) | 897 | Note added<br>Notes: 1. …<br>2. Not provided for the H8S/20223 and H8S/20235 Groups. These addresses are reserved. |
| 27.2 Register Bits | — | Amended |

chanhvinh.com

# Index


## A

A/D conversion time ..... 839
A/D Converter ..... 815
Absolute Address ..... 56
Acknowledge ..... 747
Activation by Software ..... 359, 363
Addressing Mode ..... 55
ADI ..... 842
Advanced Mode ..... 24
Arithmetic Operations Instructions ..... 45, 46

## B

Bit Manipulation Instructions ..... 49, 50
Bit synchronous circuit ..... 764
Block Data Transfer Instruction ..... 53
Block Transfer Mode ..... 357
Branch Instructions ..... 51
Buffer operation ..... 577

## C

Chain Transfer ..... 358
Chain Transfer when Counter = 0 ..... 365
Clock polarity ..... 781
Clock synchronous serial format ..... 755
Clocked synchronous communication mode ..... 784
Communication mode ..... 783
Complementary PWM mode ..... 565
Condition Field ..... 53
Condition-Code Register ..... 36
CPU Operating Modes ..... 24

## D

Data Transfer Controller (DTC) ..... 335
Data Transfer Instructions ..... 44
Digital filter includes ..... 481, 664
DTC Vector Table ..... 348

## E

Effective Address ..... 55
Effective Address Extension ..... 53
Exception Handling ..... 65
Extended Control Register (EXR) ..... 35
External Trigger ..... 841

## G

General Registers ..... 34

## I

I/O ports ..... 271
$I^2C$ bus format ..... 746
$I^2C$ bus interface 2 (IIC2) ..... 729
Immediate ..... 57
Initial setting procedure ..... 619
Input capture function ..... 551
Instruction Set ..... 42
Interrupt Control Modes ..... 98
Interrupt Exception Handling ..... 72
Interrupt Exception Handling Vector Table ..... 91
Interrupt Priority Register (IPR) ..... 75
Interrupt Request Mask Level ..... 35

## L

Logic Operations Instructions ..... 47

chanhvinh.com


## M

Memory Indirect ..... 58

## N

NMI ..... 106
NMI Interrupt ..... 89
Noise canceler ..... 758
Normal Mode ..... 355

## O

Operation Field ..... 53

## P

Pin functions ..... 14
Program Counter ..... 35
Program-Counter Relative ..... 57
PWM mode ..... 470, 555
PWM2 mode ..... 475

## R

Register Direct ..... 55
Register Field ..... 53
Register Indirect ..... 55
Register Indirect with Displacement ..... 56
Register Indirect with Post-Increment ..... 56
Register indirect with pre-decrement ..... 56
Register Information ..... 348
Registers
- ADCR ..... 823
- ADCSR ..... 821
- ADDR ..... 820
- ADMR ..... 825
- BAKCR ..... 112
- BRR ..... 690
- CMPCSR ..... 828
- CMPR ..... 826
- CMPVAL ..... 830
- CRA ..... 342
- CRB ..... 342
- DACR ..... 851
- DADR ..... 850
- DAR ..... 341
- DTCER ..... 343
- DTVECR ..... 345
- ELCSR ..... 88
- GRA ..... 461, 521
- GRB ..... 461, 521
- GRC ..... 461, 521
- GRD ..... 461, 521
- ICCR1 ..... 733
- ICCR2 ..... 735
- ICDRR ..... 745
- ICDRS ..... 745
- ICDRT ..... 745
- ICIER ..... 739
- ICMR ..... 737
- ICSR ..... 741
- IER ..... 81
- INCCR ..... 86
- INTCR ..... 78
- IPR ..... 79
- IrCR ..... 696
- ISCR ..... 82
- ISR ..... 85
- LD0CRH ..... 866
- LD0CRL ..... 867
- LD1CRH ..... 863
- LD1CRL ..... 865
- LD2CRH ..... 860
- LD2CRL ..... 862
- LPCR1 ..... 116
- LPCR2 ..... 118
- LPCR3 ..... 119
- MRA ..... 338
- MRB ..... 340

chanhvinh.com

OSCCSR ..... 121
PCR .....273, 279, 285, 291, 297, 303,
.....309, 315, 319, 323, 327, 332
PDR .....274, 280, 286, 292, 298,
.....304, 310, 316, 324, 333
PMR .....272, 278, 284, 290, 296,
.....302, 308, 314, 318, 322, 331
POCR ..... 533
RDR ..... 683
RSR ..... 683
SAR ..... 341, 744
SCR3 ..... 686
SMR ..... 684
SPMR ..... 695
SSCRH ..... 770
SSCRL ..... 771
SSER ..... 777
SSMR ..... 773
SSMR2 ..... 774
SSR ..... 688
SSRDR ..... 780
SSSR ..... 778
SSTDR ..... 780
SYSCCR ..... 114
TDR ..... 684
TRCCNT ..... 460
TRCCR2 ..... 448
TRCDF ..... 458
TRCIER ..... 449
TRCIOR0 ..... 453
TRCIOR1 ..... 455
TRCMR ..... 445
TRCOER ..... 457
TRCSR ..... 450
TRDCNT ..... 520
TRDCR ..... 523
TRDDF ..... 534
TRDFCR ..... 513
TRDIER ..... 532
TRDIORA ..... 525
TRDIORC ..... 525
TRDMDR ..... 511
TRDOCR ..... 517
TRDOER1 ..... 515
TRDOER2 ..... 517
TRDPMR ..... 512
TRDSR ..... 529
TRDSTR ..... 509
VOFR ..... 87
Repeat Mode ..... 356
Reset ..... 66
Reset exception handling ..... 69
Reset synchronous PWM mode ..... 561

## S

Scan Mode ..... 835, 838
Shift Instructions ..... 48
Single Mode ..... 833, 837
Slave address ..... 747
Software Activation ..... 367
Stack Pointer (SP) ..... 34
Stack Status after Exception Handling ..... 73
Start condition ..... 747
Stop condition ..... 747
SWDTEND ..... 359
Synchronous operation ..... 554
Synchronous serial communication unit (SSU) ..... 767
System Control Instruction ..... 52

## T

Timer RC ..... 441
Timer RD ..... 499
Trace Bit ..... 35
Trace Exception Handling ..... 71
Transfer clock ..... 781
Transfer rate ..... 734
Trap Instruction Exception Handling ..... 72

chanhvinh.com

TRAPA ..... 57
TRAPA instruction ..... 72

**V**

Vector number for the software activation interrupt ..... 345

**W**

Waveform output by compare match ..... 548

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

chanhvinh.com

H8S/20103, H8S/20203, H8S/20223, H8S/20115, H8S/20215, H8S/20235 Group User's Manual: Hardware

Publication Date: Rev.1.00 Oct 03, 2008
Rev.3.00 Sep 17, 2010

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

© 2010 Renesas Electronics Corporation. All rights reserved.
Colophon 1.0

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

chanhvinh.com

# H8S/20103, H8S/20203, H8S/20223, H8S/20115, H8S/20215, H8S/20235 Group

chanhvinh.com
chanhvinh.com
chanhvinh.com

Renesas Electronics Corporation

REJ09B0465-0300