

To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: <http://www.renesas.com>

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (<http://www.renesas.com>)

Send any inquiries to <http://www.renesas.com/inquiry>.

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.

R8C/2A Group, R8C/2B Group

Hardware Manual

RENESAS MCU

R8C FAMILY / R8C/2x SERIES

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

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, 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 products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (<http://www.renesas.com>)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
 - (1) artificial life support devices or systems
 - (2) surgical implantations
 - (3) healthcare intervention (e.g., excision, administration of medication, etc.)
 - (4) any other purposes that pose a direct threat to human lifeRenesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas 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 applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

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 occur due to the false recognition of the pin state as an input signal become possible. 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.

How to Use This Manual

1. Purpose and Target Readers

This manual is designed to provide the user with an understanding of the hardware functions and electrical characteristics of the MCU. It is intended for users designing application systems incorporating the MCU. A basic knowledge of electric circuits, logical circuits, and MCUs is necessary in order to use this manual.

The manual comprises an overview of the product; descriptions of the CPU, system control functions, peripheral functions, and electrical characteristics; and usage notes.

Particular attention should be paid to the precautionary notes when using the manual. These notes occur within the body of the text, at the end of each section, and in the Usage Notes section.

The revision history summarizes the locations of revisions and additions. It does not list all revisions. Refer to the text of the manual for details.

The following documents apply to the R8C/2A Group, R8C/2B Group. Make sure to refer to the latest versions of these documents. The newest versions of the documents listed may be obtained from the Renesas Technology Web site.

| Document Type | Description | Document Title | Document No. |
|--------------------------|--|--|----------------------|
| Datasheet | Hardware overview and electrical characteristics | R8C/2A Group, R8C/2B Group Group Datasheet | REJ03B0182 |
| Hardware manual | Hardware specifications (pin assignments, memory maps, peripheral function specifications, electrical characteristics, timing charts) and operation description Note: Refer to the application notes for details on using peripheral functions. | R8C/2A Group, R8C/2B Group Hardware Manual | This hardware manual |
| Software manual | Description of CPU instruction set | R8C/Tiny Series Software Manual | REJ09B0001 |
| Application note | Information on using peripheral functions and application examples Sample programs Information on writing programs in assembly language and C | Available from Renesas Technology Web site. | |
| Renesas technical update | Product specifications, updates on documents, etc. | | |

2. Notation of Numbers and Symbols

The notation conventions for register names, bit names, numbers, and symbols used in this manual are described below.

(1) Register Names, Bit Names, and Pin Names

Registers, bits, and pins are referred to in the text by symbols. The symbol is accompanied by the word “register,” “bit,” or “pin” to distinguish the three categories.

Examples the PM03 bit in the PM0 register
P3_5 pin, VCC pin

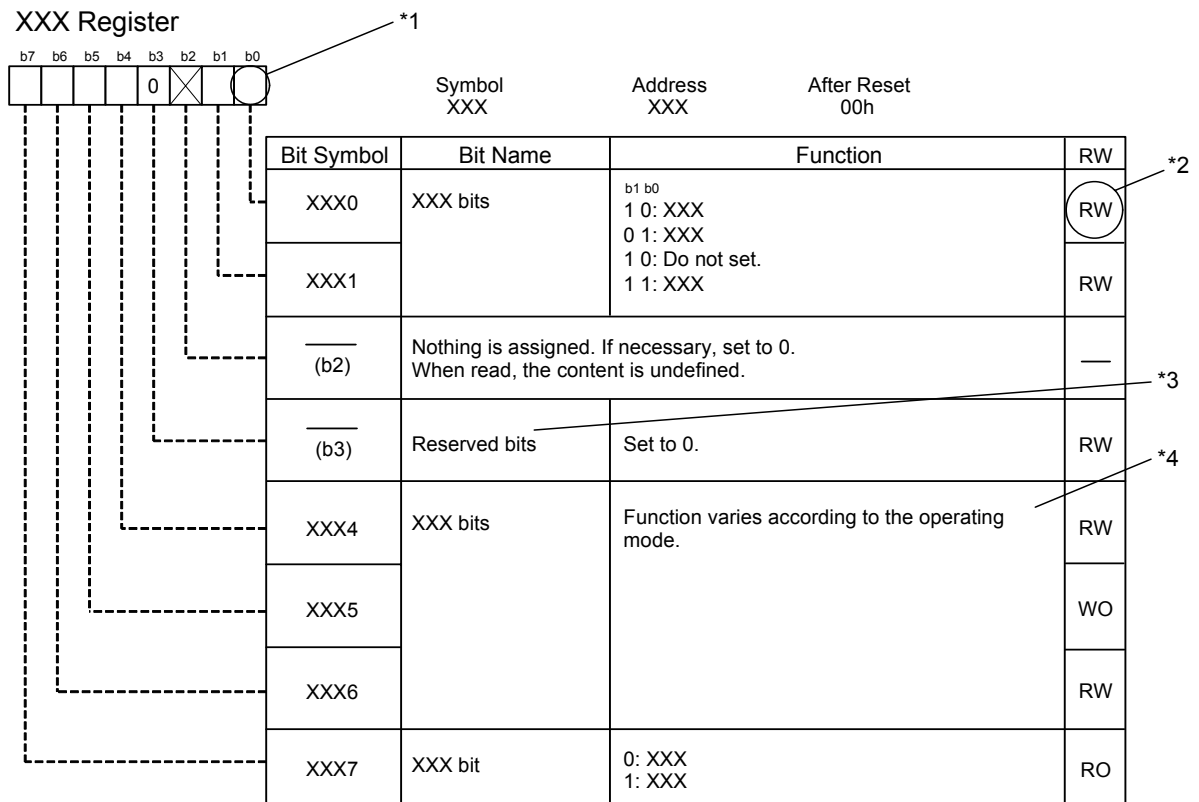
(2) Notation of Numbers

The indication “b” is appended to numeric values given in binary format. However, nothing is appended to the values of single bits. The indication “h” is appended to numeric values given in hexadecimal format. Nothing is appended to numeric values given in decimal format.

Examples Binary: 11b
Hexadecimal: EFA0h
Decimal: 1234

3. Register Notation

The symbols and terms used in register diagrams are described below.



*1
Blank: Set to 0 or 1 according to the application.
0: Set to 0.
1: Set to 1.
X: Nothing is assigned.

*2
RW: Read and write.
RO: Read only.
WO: Write only.
—: Nothing is assigned.

*3
• Reserved bit
Reserved bit. Set to specified value.

*4
• Nothing is assigned
Nothing is assigned to the bit. As the bit may be used for future functions, if necessary, set to 0.
• Do not set to a value
Operation is not guaranteed when a value is set.
• Function varies according to the operating mode.
The function of the bit varies with the peripheral function mode. Refer to the register diagram for information on the individual modes.

4. List of Abbreviations and Acronyms

| Abbreviation | Full Form |
|--------------|--|
| ACIA | Asynchronous Communication Interface Adapter |
| bps | bits per second |
| CRC | Cyclic Redundancy Check |
| DMA | Direct Memory Access |
| DMAC | Direct Memory Access Controller |
| GSM | Global System for Mobile Communications |
| Hi-Z | High Impedance |
| IEBus | Inter Equipment bus |
| I/O | Input/Output |
| IrDA | Infrared Data Association |
| LSB | Least Significant Bit |
| MSB | Most Significant Bit |
| NC | Non-Connection |
| PLL | Phase Locked Loop |
| PWM | Pulse Width Modulation |
| SFR | Special Function Registers |
| SIM | Subscriber Identity Module |
| UART | Universal Asynchronous Receiver/Transmitter |
| VCO | Voltage Controlled Oscillator |

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

Table of Contents

| | |
|--|-----------|
| SFR Page Reference | B - 1 |
| 1. Overview | 1 |
| 1.1 Features | 1 |
| 1.1.1 Applications | 1 |
| 1.1.2 Specifications | 2 |
| 1.2 Product List | 6 |
| 1.3 Block Diagram | 10 |
| 1.4 Pin Assignment | 11 |
| 1.5 Pin Functions | 15 |
| 2. Central Processing Unit (CPU) | 17 |
| 2.1 Data Registers (R0, R1, R2, and R3) | 18 |
| 2.2 Address Registers (A0 and A1) | 18 |
| 2.3 Frame Base Register (FB) | 18 |
| 2.4 Interrupt Table Register (INTB) | 18 |
| 2.5 Program Counter (PC) | 18 |
| 2.6 User Stack Pointer (USP) and Interrupt Stack Pointer (ISP) | 18 |
| 2.7 Static Base Register (SB) | 18 |
| 2.8 Flag Register (FLG) | 18 |
| 2.8.1 Carry Flag (C) | 18 |
| 2.8.2 Debug Flag (D) | 18 |
| 2.8.3 Zero Flag (Z) | 18 |
| 2.8.4 Sign Flag (S) | 18 |
| 2.8.5 Register Bank Select Flag (B) | 18 |
| 2.8.6 Overflow Flag (O) | 18 |
| 2.8.7 Interrupt Enable Flag (I) | 19 |
| 2.8.8 Stack Pointer Select Flag (U) | 19 |
| 2.8.9 Processor Interrupt Priority Level (IPL) | 19 |
| 2.8.10 Reserved Bit | 19 |
| 3. Memory | 20 |
| 3.1 R8C/2A Group | 20 |
| 3.2 R8C/2B Group | 21 |
| 4. Special Function Registers (SFRs) | 22 |
| 5. Resets | 34 |
| 5.1 Hardware Reset | 37 |
| 5.1.1 When Power Supply is Stable | 37 |
| 5.1.2 Power On | 37 |
| 5.2 Power-On Reset Function | 39 |
| 5.3 Voltage Monitor 0 Reset | 40 |
| 5.4 Voltage Monitor 1 Reset | 40 |
| 5.5 Voltage Monitor 2 Reset | 40 |
| 5.6 Watchdog Timer Reset | 41 |
| 5.7 Software Reset | 41 |

| | | |
|--------|---|-----|
| 6. | Voltage Detection Circuit | 42 |
| 6.1 | VCC Input Voltage | 49 |
| 6.1.1 | Monitoring Vdet0 | 49 |
| 6.1.2 | Monitoring Vdet1 | 49 |
| 6.1.3 | Monitoring Vdet2 | 49 |
| 6.2 | Voltage Monitor 0 Reset | 50 |
| 6.3 | Voltage Monitor 1 Interrupt and Voltage Monitor 1 Reset | 51 |
| 6.4 | Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset | 53 |
| 7. | Programmable I/O Ports | 55 |
| 7.1 | Functions of Programmable I/O Ports | 55 |
| 7.2 | Effect on Peripheral Functions | 56 |
| 7.3 | Pins Other than Programmable I/O Ports | 56 |
| 7.4 | Port settings | 72 |
| 7.5 | Unassigned Pin Handling | 89 |
| 8. | Processor Mode | 90 |
| 8.1 | Processor Modes | 90 |
| 9. | Bus | 91 |
| 10. | Clock Generation Circuit | 93 |
| 10.1 | XIN Clock | 103 |
| 10.2 | On-Chip Oscillator Clocks | 104 |
| 10.2.1 | Low-Speed On-Chip Oscillator Clock | 104 |
| 10.2.2 | High-Speed On-Chip Oscillator Clock | 104 |
| 10.3 | XCIN Clock | 105 |
| 10.4 | CPU Clock and Peripheral Function Clock | 106 |
| 10.4.1 | System Clock | 106 |
| 10.4.2 | CPU Clock | 106 |
| 10.4.3 | Peripheral Function Clock (f1, f2, f4, f8, and f32) | 106 |
| 10.4.4 | fOCO | 106 |
| 10.4.5 | fOCO40M | 106 |
| 10.4.6 | fOCO-F | 106 |
| 10.4.7 | fOCO-S | 106 |
| 10.4.8 | fOCO128 | 106 |
| 10.4.9 | fC4 and fC32 | 107 |
| 10.5 | Power Control | 108 |
| 10.5.1 | Standard Operating Mode | 108 |
| 10.5.2 | Wait Mode | 110 |
| 10.5.3 | Stop Mode | 114 |
| 10.6 | Oscillation Stop Detection Function | 117 |
| 10.6.1 | How to Use Oscillation Stop Detection Function | 117 |
| 10.7 | Notes on Clock Generation Circuit | 120 |
| 10.7.1 | Stop Mode | 120 |
| 10.7.2 | Wait Mode | 120 |
| 10.7.3 | Oscillation Stop Detection Function | 120 |
| 10.7.4 | Oscillation Circuit Constants | 120 |

| | | |
|--------|---|-----|
| 11. | Protection | 121 |
| 12. | Interrupts | 122 |
| 12.1 | Interrupt Overview | 122 |
| 12.1.1 | Types of Interrupts | 122 |
| 12.1.2 | Software Interrupts | 123 |
| 12.1.3 | Special Interrupts | 124 |
| 12.1.4 | Peripheral Function Interrupt | 124 |
| 12.1.5 | Interrupts and Interrupt Vectors | 125 |
| 12.1.6 | Interrupt Control | 127 |
| 12.2 | $\overline{\text{INT}}$ Interrupt | 136 |
| 12.2.1 | $\overline{\text{INT}}_i$ Interrupt (i = 0 to 3) | 136 |
| 12.2.2 | $\overline{\text{INT}}_i$ Input Filter (i = 0 to 3) | 139 |
| 12.3 | Key Input Interrupt | 140 |
| 12.4 | Address Match Interrupt | 142 |
| 12.5 | Timer RC Interrupt, Timer RD Interrupt, Clock Synchronous Serial I/O with Chip Select Interrupts, and I ² C bus Interface Interrupt (Interrupts with Multiple Interrupt Request Sources) | 144 |
| 12.6 | Notes on Interrupts | 146 |
| 12.6.1 | Reading Address 00000h | 146 |
| 12.6.2 | SP Setting | 146 |
| 12.6.3 | External Interrupt and Key Input Interrupt | 146 |
| 12.6.4 | Changing Interrupt Sources | 147 |
| 12.6.5 | Changing Interrupt Control Register Contents | 148 |
| 13. | Watchdog Timer | 149 |
| 13.1 | Count Source Protection Mode Disabled | 153 |
| 13.2 | Count Source Protection Mode Enabled | 154 |
| 14. | Timers | 155 |
| 14.1 | Timer RA | 158 |
| 14.1.1 | Timer Mode | 161 |
| 14.1.2 | Pulse Output Mode | 163 |
| 14.1.3 | Event Counter Mode | 165 |
| 14.1.4 | Pulse Width Measurement Mode | 167 |
| 14.1.5 | Pulse Period Measurement Mode | 170 |
| 14.1.6 | Notes on Timer RA | 173 |
| 14.2 | Timer RB | 174 |
| 14.2.1 | Timer Mode | 178 |
| 14.2.2 | Programmable Waveform Generation Mode | 181 |
| 14.2.3 | Programmable One-shot Generation Mode | 183 |
| 14.2.4 | Programmable Wait One-Shot Generation Mode | 187 |
| 14.2.5 | Notes on Timer RB | 191 |
| 14.3 | Timer RC | 195 |
| 14.3.1 | Overview | 195 |
| 14.3.2 | Registers Associated with Timer RC | 197 |
| 14.3.3 | Common Items for Multiple Modes | 207 |
| 14.3.4 | Timer Mode (Input Capture Function) | 213 |
| 14.3.5 | Timer Mode (Output Compare Function) | 218 |
| 14.3.6 | PWM Mode | 224 |

| | | |
|---------|--|-----|
| 14.3.7 | PWM2 Mode | 229 |
| 14.3.8 | Timer RC Interrupt | 235 |
| 14.3.9 | Notes on Timer RC | 236 |
| 14.4 | Timer RD | 237 |
| 14.4.1 | Count Sources | 242 |
| 14.4.2 | Buffer Operation | 243 |
| 14.4.3 | Synchronous Operation | 245 |
| 14.4.4 | Pulse Output Forced Cutoff | 246 |
| 14.4.5 | Input Capture Function | 248 |
| 14.4.6 | Output Compare Function | 263 |
| 14.4.7 | PWM Mode | 281 |
| 14.4.8 | Reset Synchronous PWM Mode | 295 |
| 14.4.9 | Complementary PWM Mode | 306 |
| 14.4.10 | PWM3 Mode | 321 |
| 14.4.11 | Timer RD Interrupt | 335 |
| 14.4.12 | Notes on Timer RD | 337 |
| 14.5 | Timer RE | 343 |
| 14.5.1 | Real-Time Clock Mode | 344 |
| 14.5.2 | Output Compare Mode | 352 |
| 14.5.3 | Notes on Timer RE | 358 |
| 14.6 | Timer RF | 361 |
| 14.6.1 | Input Capture Mode | 366 |
| 14.6.2 | Output Compare Mode | 369 |
| 14.6.3 | Notes on Timer RF | 373 |
| 15. | Serial Interface | 374 |
| 15.1 | Clock Synchronous Serial I/O Mode | 381 |
| 15.1.1 | Polarity Select Function | 385 |
| 15.1.2 | LSB First/MSB First Select Function | 385 |
| 15.1.3 | Continuous Receive Mode | 386 |
| 15.2 | Clock Asynchronous Serial I/O (UART) Mode | 387 |
| 15.2.1 | Bit Rate | 392 |
| 15.3 | Notes on Serial Interface | 393 |
| 16. | Clock Synchronous Serial Interface | 394 |
| 16.1 | Mode Selection | 394 |
| 16.2 | Clock Synchronous Serial I/O with Chip Select (SSU) | 395 |
| 16.2.1 | Transfer Clock | 404 |
| 16.2.2 | SS Shift Register (SSTRSR) | 406 |
| 16.2.3 | Interrupt Requests | 407 |
| 16.2.4 | Communication Modes and Pin Functions | 408 |
| 16.2.5 | Clock Synchronous Communication Mode | 409 |
| 16.2.6 | Operation in 4-Wire Bus Communication Mode | 416 |
| 16.2.7 | SCS Pin Control and Arbitration | 422 |
| 16.2.8 | Notes on Clock Synchronous Serial I/O with Chip Select | 423 |
| 16.3 | I ² C bus Interface | 424 |
| 16.3.1 | Transfer Clock | 434 |
| 16.3.2 | Interrupt Requests | 435 |
| 16.3.3 | I ² C bus Interface Mode | 436 |

| | | |
|------------|---|------------|
| 16.3.4 | Clock Synchronous Serial Mode | 447 |
| 16.3.5 | Examples of Register Setting | 450 |
| 16.3.6 | Noise Canceller | 454 |
| 16.3.7 | Bit Synchronization Circuit | 455 |
| 16.3.8 | Notes on I ² C bus Interface | 456 |
| 17. | Hardware LIN | 457 |
| 17.1 | Features | 457 |
| 17.2 | Input/Output Pins | 458 |
| 17.3 | Register Configuration | 459 |
| 17.4 | Functional Description | 461 |
| 17.4.1 | Master Mode | 461 |
| 17.4.2 | Slave Mode | 464 |
| 17.4.3 | Bus Collision Detection Function | 468 |
| 17.4.4 | Hardware LIN End Processing | 469 |
| 17.5 | Interrupt Requests | 470 |
| 17.6 | Notes on Hardware LIN | 471 |
| 18. | A/D Converter | 472 |
| 18.1 | One-Shot Mode | 477 |
| 18.2 | Repeat Mode 0 | 480 |
| 18.3 | Sample and Hold | 483 |
| 18.4 | A/D Conversion Cycles | 483 |
| 18.5 | Internal Equivalent Circuit of Analog Input | 484 |
| 18.6 | Output Impedance of Sensor under A/D Conversion | 485 |
| 18.7 | Notes on A/D Converter | 486 |
| 19. | D/A Converter | 487 |
| 20. | Flash Memory | 489 |
| 20.1 | Overview | 489 |
| 20.2 | Memory Map | 490 |
| 20.3 | Functions to Prevent Rewriting of Flash Memory | 492 |
| 20.3.1 | ID Code Check Function | 492 |
| 20.3.2 | ROM Code Protect Function | 493 |
| 20.4 | CPU Rewrite Mode | 494 |
| 20.4.1 | EW0 Mode | 495 |
| 20.4.2 | EW1 Mode | 495 |
| 20.4.3 | Software Commands | 504 |
| 20.4.4 | Status Registers | 509 |
| 20.4.5 | Full Status Check | 510 |
| 20.5 | Standard Serial I/O Mode | 512 |
| 20.5.1 | ID Code Check Function | 512 |
| 20.6 | Parallel I/O Mode | 516 |
| 20.6.1 | ROM Code Protect Function | 516 |
| 20.7 | Notes on Flash Memory | 517 |
| 20.7.1 | CPU Rewrite Mode | 517 |

| | |
|---|-----|
| 21. Electrical Characteristics | 520 |
| 22. Usage Notes | 545 |
| 22.1 Notes on Clock Generation Circuit | 545 |
| 22.1.1 Stop Mode | 545 |
| 22.1.2 Wait Mode | 545 |
| 22.1.3 Oscillation Stop Detection Function | 545 |
| 22.1.4 Oscillation Circuit Constants | 545 |
| 22.2 Notes on Interrupts | 546 |
| 22.2.1 Reading Address 00000h | 546 |
| 22.2.2 SP Setting | 546 |
| 22.2.3 External Interrupt and Key Input Interrupt | 546 |
| 22.2.4 Changing Interrupt Sources | 547 |
| 22.2.5 Changing Interrupt Control Register Contents | 548 |
| 22.3 Notes on Timers | 549 |
| 22.3.1 Notes on Timer RA | 549 |
| 22.3.2 Notes on Timer RB | 550 |
| 22.3.3 Notes on Timer RC | 554 |
| 22.3.4 Notes on Timer RD | 555 |
| 22.3.5 Notes on Timer RE | 561 |
| 22.3.6 Notes on Timer RF | 564 |
| 22.4 Notes on Serial Interface | 565 |
| 22.5 Notes on Clock Synchronous Serial Interface | 566 |
| 22.5.1 Notes on Clock Synchronous Serial I/O with Chip Select | 566 |
| 22.5.2 Notes on I ² C bus Interface | 566 |
| 22.6 Notes on Hardware LIN | 567 |
| 22.7 Notes on A/D Converter | 568 |
| 22.8 Notes on Flash Memory | 569 |
| 22.8.1 CPU Rewrite Mode | 569 |
| 22.9 Notes on Noise | 572 |
| 22.9.1 Inserting a Bypass Capacitor between VCC and VSS Pins as a Countermeasure against Noise and Latch-up | 572 |
| 22.9.2 Countermeasures against Noise Error of Port Control Registers | 572 |
| 23. Notes on On-Chip Debugger | 573 |
| 24. Notes on Emulator Debugger | 574 |
| Appendix 1. Package Dimensions | 575 |
| Appendix 2. Connection Examples between Serial Writer and On-Chip Debugging Emulator | 577 |
| Appendix 3. Example of Oscillation Evaluation Circuit | 578 |
| Index | 579 |

SFR Page Reference

| Address | Register | Symbol | Page |
|---------|--|--------|---|
| 0000h | | | |
| 0001h | | | |
| 0002h | | | |
| 0003h | | | |
| 0004h | Processor Mode Register 0 | PM0 | 90 |
| 0005h | Processor Mode Register 1 | PM1 | 90 |
| 0006h | System Clock Control Register 0 | CM0 | 96 |
| 0007h | System Clock Control Register 1 | CM1 | 97 |
| 0008h | Module Standby Control Register | MSTCR | 198, 250, 265, 283, 297, 308, 323, 396, 426 |
| 0009h | | | |
| 000Ah | Protect Register | PRCR | 121 |
| 000Bh | | | |
| 000Ch | Oscillation Stop Detection Register | OSD | 98 |
| 000Dh | Watchdog Timer Reset Register | WDTR | 151 |
| 000Eh | Watchdog Timer Start Register | WDTS | 151 |
| 000Fh | Watchdog Timer Control Register | WDC | 151 |
| 0010h | Address Match Interrupt Register 0 | RMAD0 | 143 |
| 0011h | | | |
| 0012h | | | |
| 0013h | Address Match Interrupt Enable Register | AIER | 143 |
| 0014h | Address Match Interrupt Register 1 | RMAD1 | 143 |
| 0015h | | | |
| 0016h | | | |
| 0017h | | | |
| 0018h | | | |
| 0019h | | | |
| 001Ah | | | |
| 001Bh | | | |
| 001Ch | Count Source Protection Mode Register | CSPR | 152 |
| 001Dh | | | |
| 001Eh | | | |
| 001Fh | | | |
| 0020h | | | |
| 0021h | | | |
| 0022h | | | |
| 0023h | High-Speed On-Chip Oscillator Control Register 0 | FRA0 | 99 |
| 0024h | High-Speed On-Chip Oscillator Control Register 1 | FRA1 | 99 |
| 0025h | High-Speed On-Chip Oscillator Control Register 2 | FRA2 | 100 |
| 0026h | | | |
| 0027h | | | |
| 0028h | Clock Prescaler Reset Flag | CPSRF | 101 |
| 0029h | | | |
| 002Ah | | | |
| 002Bh | High-Speed On-Chip Oscillator Control Register 6 | FRA6 | 100 |
| 002Ch | High-Speed On-Chip Oscillator Control Register 7 | FRA7 | 100 |
| 002Dh | | | |
| 002Eh | | | |
| 002Fh | | | |
| 0030h | | | |
| 0031h | Voltage Detection Register 1 | VCA1 | 45 |
| 0032h | Voltage Detection Register 2 | VCA2 | 45, 101 |
| 0033h | | | |
| 0034h | | | |
| 0035h | | | |
| 0036h | Voltage Monitor 1 Circuit Control Register | VW1C | 47 |
| 0037h | Voltage Monitor 2 Circuit Control Register | VW2C | 48 |
| 0038h | Voltage Monitor 0 Circuit Control Register | VW0C | 46 |
| 0039h | | | |
| 003Ah | | | |
| 003Bh | | | |
| 003Ch | | | |
| 003Dh | | | |
| 003Eh | | | |
| 003Fh | | | |

| Address | Register | Symbol | Page |
|---------|---|-------------|------|
| 0040h | | | |
| 0041h | | | |
| 0042h | | | |
| 0043h | | | |
| 0044h | | | |
| 0045h | | | |
| 0046h | | | |
| 0047h | Timer RC Interrupt Control Register | TRCIC | 128 |
| 0048h | Timer RD0 Interrupt Control Register | TRD0IC | 128 |
| 0049h | Timer RD1 Interrupt Control Register | TRD1IC | 128 |
| 004Ah | Timer RE Interrupt Control Register | TREIC | 127 |
| 004Bh | UART2 Transmit Interrupt Control Register | S2TIC | 127 |
| 004Ch | UART2 Receive Interrupt Control Register | S2RIC | 127 |
| 004Dh | Key Input Interrupt Control Register | KUPIC | 127 |
| 004Eh | | | |
| 004Fh | SSU/IIC Interrupt Control Register | SSUIC/IICIC | 128 |
| 0050h | Compare 1 Interrupt Control Register | CMP1IC | 127 |
| 0051h | UART0 Transmit Interrupt Control Register | S0TIC | 127 |
| 0052h | UART0 Receive Interrupt Control Register | S0RIC | 127 |
| 0053h | UART1 Transmit Interrupt Control Register | S1TIC | 127 |
| 0054h | UART1 Receive Interrupt Control Register | S1RIC | 127 |
| 0055h | INT2 Interrupt Control Register | INT2IC | 129 |
| 0056h | Timer RA Interrupt Control Register | TRAIC | 127 |
| 0057h | | | |
| 0058h | Timer RB Interrupt Control Register | TRBIC | 127 |
| 0059h | INT1 Interrupt Control Register | INT1IC | 129 |
| 005Ah | INT3 Interrupt Control Register | INT3IC | 129 |
| 005Bh | Timer RF Interrupt Control Register | TRFIC | 127 |
| 005Ch | Compare 0 Interrupt Control Register | CMP0IC | 127 |
| 005Dh | INT0 Interrupt Control Register | INT0IC | 129 |
| 005Eh | A/D Conversion Interrupt Control Register | ADIC | 127 |
| 005Fh | Capture Interrupt Control Register | CAPIC | 127 |
| 0060h | | | |
| 0061h | | | |
| 0062h | | | |
| 0063h | | | |
| 0064h | | | |
| 0065h | | | |
| 0066h | | | |
| 0067h | | | |
| 0068h | | | |
| 0069h | | | |
| 006Ah | | | |
| 006Bh | | | |
| 006Ch | | | |
| 006Dh | | | |
| 006Eh | | | |
| 006Fh | | | |
| 0070h | | | |
| 0071h | | | |
| 0072h | | | |
| 0073h | | | |
| 0074h | | | |
| 0075h | | | |
| 0076h | | | |
| 0077h | | | |
| 0078h | | | |
| 0079h | | | |
| 007Ah | | | |
| 007Bh | | | |
| 007Ch | | | |
| 007Dh | | | |
| 007Eh | | | |
| 007Fh | | | |

NOTE:

1. The blank regions are reserved. Do not access locations in these regions.

| Address | Register | Symbol | Page |
|---------|--|-------------|----------|
| 0080h | | | |
| 0081h | | | |
| 0082h | | | |
| 0083h | | | |
| 0084h | | | |
| 0085h | | | |
| 0086h | | | |
| 0087h | | | |
| 0088h | | | |
| 0089h | | | |
| 008Ah | | | |
| 008Bh | | | |
| 008Ch | | | |
| 008Dh | | | |
| 008Eh | | | |
| 008Fh | | | |
| 0090h | | | |
| 0091h | | | |
| 0092h | | | |
| 0093h | | | |
| 0094h | | | |
| 0095h | | | |
| 0096h | | | |
| 0097h | | | |
| 0098h | | | |
| 0099h | | | |
| 009Ah | | | |
| 009Bh | | | |
| 009Ch | | | |
| 009Dh | | | |
| 009Eh | | | |
| 009Fh | | | |
| 00A0h | UART0 Transmit/Receive Mode Register | U0MR | 377 |
| 00A1h | UART0 Bit Rate Register | U0BRG | 377 |
| 00A2h | UART0 Transmit Buffer Register | U0TB | 378 |
| 00A3h | | | |
| 00A4h | UART0 Transmit / Receive Control Register 0 | U0C0 | 378 |
| 00A5h | UART0 Transmit / Receive Control Register 1 | U0C1 | 379 |
| 00A6h | UART0 Receive Buffer Register | U0RB | 379 |
| 00A7h | | | |
| 00A8h | UART1 Transmit/Receive Mode Register | U1MR | 377 |
| 00A9h | UART1 Bit Rate Register | U1BRG | 377 |
| 00AAh | UART1 Transmit Buffer Register | U1TB | 378 |
| 00ABh | | | |
| 00ACh | UART1 Transmit/Receive Control Register 0 | U1C0 | 378 |
| 00ADh | UART1 Transmit/Receive Control Register 1 | U1C1 | 379 |
| 00AEh | UART1 Receive Buffer Register | U1RB | 379 |
| 00AFh | | | |
| 00B0h | | | |
| 00B1h | | | |
| 00B2h | | | |
| 00B3h | | | |
| 00B4h | | | |
| 00B5h | | | |
| 00B6h | | | |
| 00B7h | | | |
| 00B8h | SS Control Register H / IIC bus Control Register 1 | SSCRH/ICCR1 | 397, 427 |
| 00B9h | SS Control Register L / IIC bus Control Register 2 | SSCRL/ICCR2 | 398, 428 |
| 00BAh | SS Mode Register / IIC bus Mode Register | SSMR/ICMR | 399, 429 |
| 00BBh | SS Enable Register / IIC bus Interrupt Enable Register | SSER/ICIER | 400, 430 |
| 00BCh | SS Status Register / IIC bus Status Register | SSSR/ICSR | 401, 431 |
| 00BDh | SS Mode Register 2 / Slave Address Register | SSMR2/SAR | 402, 432 |
| 00BEh | SS Transmit Data Register/IIC Bus Transmit Data Register | SSTDR/ICDRT | 403, 432 |
| 00BFh | SS Receive Data Register/IIC Bus Receive Data Register | SSRDR/ICDRR | 403, 433 |

| Address | Register | Symbol | Page |
|---------|---|--------|------------------------|
| 00C0h | | | |
| 00C1h | | | |
| 00C2h | | | |
| 00C3h | | | |
| 00C4h | | | |
| 00C5h | | | |
| 00C6h | | | |
| 00C7h | | | |
| 00C8h | | | |
| 00C9h | | | |
| 00CAh | | | |
| 00CBh | | | |
| 00CCh | | | |
| 00CDh | | | |
| 00CEh | | | |
| 00CFh | | | |
| 00D0h | | | |
| 00D1h | | | |
| 00D2h | | | |
| 00D3h | | | |
| 00D4h | | | |
| 00D5h | | | |
| 00D6h | | | |
| 00D7h | | | |
| 00D8h | D/A Register 0 | DA0 | 488 |
| 00D9h | | | |
| 00DAh | D/A Register 1 | DA1 | 488 |
| 00DBh | | | |
| 00DCh | D/A Control Register | DACON | 488 |
| 00DDh | | | |
| 00DEh | | | |
| 00DFh | | | |
| 00E0h | Port P0 Register | P0 | 69 |
| 00E1h | Port P1 Register | P1 | 69 |
| 00E2h | Port P0 Direction Register | PD0 | 68 |
| 00E3h | Port P1 Direction Register | PD1 | 68 |
| 00E4h | Port P2 Register | P2 | 69 |
| 00E5h | Port P3 Register | P3 | 69 |
| 00E6h | Port P2 Direction Register | PD2 | 68 |
| 00E7h | Port P3 Direction Register | PD3 | 68 |
| 00E8h | Port P4 Register | P4 | 69 |
| 00E9h | Port P5 Register | P5 | 69 |
| 00EAh | Port P4 Direction Register | PD4 | 68 |
| 00EBh | Port P5 Direction Register | PD5 | 68 |
| 00ECh | Port P6 Register | P6 | 69 |
| 00EDh | | | |
| 00EEh | Port P6 Direction Register | PD6 | 68 |
| 00EFh | | | |
| 00F0h | | | |
| 00F1h | | | |
| 00F2h | | | |
| 00F3h | | | |
| 00F4h | Port P2 Drive Capacity Control Register | P2DRR | 70 |
| 00F5h | UART1 Function Select Register | U1SR | 380 |
| 00F6h | | | |
| 00F7h | | | |
| 00F8h | Port Mode Register | PMR | 70, 136, 380, 403, 433 |
| 00F9h | External Input Enable Register | INTEN | 137 |
| 00FAh | INT Input Filter Select Register | INTF | 138 |
| 00FBh | Key Input Enable Register | KIEN | 141 |
| 00FCh | Pull-Up Control Register 0 | PUR0 | 71 |
| 00FDh | Pull-Up Control Register 1 | PUR1 | 71 |
| 00FEh | | | |
| 00FFh | | | |

NOTE:

1. The blank regions are reserved. Do not access locations in these regions.

| Address | Register | Symbol | Page |
|---------|---|---------|------------------------------|
| 0100h | Timer RA Control Register | TRACR | 159 |
| 0101h | Timer RA I/O Control Register | TRAIOC | 159, 161, 164, 166, 168, 171 |
| 0102h | Timer RA Mode Register | TRAMR | 160 |
| 0103h | Timer RA Prescaler Register | TRAPRE | 160 |
| 0104h | Timer RA Register | TRA | 160 |
| 0105h | LIN Control Register 2 | LINCR2 | 459 |
| 0106h | LIN Control Register | LINCR | 459 |
| 0107h | LIN Status Register | LINST | 460 |
| 0108h | Timer RB Control Register | TRBCR | 175 |
| 0109h | Timer RB One-Shot Control Register | TRBOCR | 175 |
| 010Ah | Timer RB I/O Control Register | TRBIOC | 176, 178, 182, 184, 189 |
| 010Bh | Timer RB Mode Register | TRBMR | 176 |
| 010Ch | Timer RB Prescaler Register | TRBPRE | 177 |
| 010Dh | Timer RB Secondary Register | TRBSC | 177 |
| 010Eh | Timer RB Primary Register | TRBPR | 177 |
| 010Fh | | | |
| 0110h | | | |
| 0111h | | | |
| 0112h | | | |
| 0113h | | | |
| 0114h | | | |
| 0115h | | | |
| 0116h | | | |
| 0117h | | | |
| 0118h | Timer RE Second Data Register / Counter Data Register | TRESEC | 346, 354 |
| 0119h | Timer RE Minute Data Register / Compare Data Register | TREMIN | 346, 354 |
| 011Ah | Timer RE Hour Data Register | TREHR | 347 |
| 011Bh | Timer RE Day of Week Data Register | TREWK | 347 |
| 011Ch | Timer RE Control Register 1 | TRECR1 | 348, 355 |
| 011Dh | Timer RE Control Register 2 | TRECR2 | 349, 355 |
| 011Eh | Timer RE Clock Source Select Register | TRECSR | 350, 356 |
| 011Fh | | | |
| 0120h | Timer RC Mode Register | TRCMR | 198 |
| 0121h | Timer RC Control Register 1 | TRCCR1 | 199, 222, 226, 231 |
| 0122h | Timer RC Interrupt Enable Register | TRCIER | 200 |
| 0123h | Timer RC Status Register | TRCSR | 201 |
| 0124h | Timer RC I/O Control Register 0 | TRCIOR0 | 206, 215, 220 |
| 0125h | Timer RC I/O Control Register 1 | TRCIOR1 | 206, 216, 221 |
| 0126h | Timer RC Counter | TRC | 202 |
| 0127h | | | |
| 0128h | Timer RC General Register A | TRCGRA | 202 |
| 0129h | | | |
| 012Ah | Timer RC General Register B | TRCGRB | 202 |
| 012Bh | | | |
| 012Ch | Timer RC General Register C | TRCGRC | 202 |
| 012Dh | | | |
| 012Eh | Timer RC General Register D | TRCGRD | 202 |
| 012Fh | | | |

NOTE:

- The blank regions are reserved. Do not access locations in these regions.

| Address | Register | Symbol | Page |
|---------|--|----------|------------------------------|
| 0130h | Timer RC Control Register 2 | TRCCR2 | 203 |
| 0131h | Timer RC Digital Filter Function Select Register | TRCDF | 204 |
| 0132h | Timer RC Output Master Enable Register | TRCOER | 205 |
| 0133h | | | |
| 0134h | | | |
| 0135h | | | |
| 0136h | | | |
| 0137h | Timer RD Start Register | TRDSTR | 251, 266, 284, 298, 309, 324 |
| 0138h | Timer RD Mode Register | TRDMR | 251, 266, 284, 298, 310, 325 |
| 0139h | Timer RD PWM Mode Register | TRDPMR | 252, 267, 285 |
| 013Ah | Timer RD Function Control Register | TRDFCR | 253, 268, 286, 299, 311, 326 |
| 013Bh | Timer RD Output Master Enable Register 1 | TRDOER1 | 269, 287, 300, 312, 327 |
| 013Ch | Timer RD Output Master Enable Register 2 | TRDOER2 | 269, 287, 300, 312, 327 |
| 013Dh | Timer RD Output Control Register | TRDOCR | 270, 288, 328 |
| 013Eh | Timer RD Digital Filter Function Select Register 0 | TRDDF0 | 254 |
| 013Fh | Timer RD Digital Filter Function Select Register 1 | TRDDF1 | 254 |
| 0140h | Timer RD Control Register 0 | TRDCR0 | 255, 271, 288, 301, 313, 329 |
| 0141h | Timer RD I/O Control Register A0 | TRDIORA0 | 256, 272 |
| 0142h | Timer RD I/O Control Register C0 | TRDIORC0 | 257, 273 |
| 0143h | Timer RD Status Register 0 | TRDSR0 | 258, 274, 289, 302, 314, 330 |
| 0144h | Timer RD Interrupt Enable Register 0 | TRDIER0 | 259, 275, 290, 303, 315, 331 |
| 0145h | Timer RD PWM Mode Output Level Control Register 0 | TRDPOCR0 | 291 |
| 0146h | Timer RD Counter 0 | TRD0 | 259, 276, 291, 303, 316, 331 |
| 0147h | | | |
| 0148h | Timer RD General Register A0 | TRDGRA0 | 260, 276, 292, 304, 316, 332 |
| 0149h | | | |
| 014Ah | Timer RD General Register B0 | TRDGRB0 | 260, 276, 292, 304, 316, 332 |
| 014Bh | | | |
| 014Ch | Timer RD General Register C0 | TRDGRC0 | 260, 276, 292, 304, 332 |
| 014Dh | | | |
| 014Eh | Timer RD General Register D0 | TRDGRD0 | 260, 276, 292, 304, 316, 332 |
| 014Fh | | | |
| 0150h | Timer RD Control Register 1 | TRDCR1 | 255, 271, 288, 313 |
| 0151h | Timer RD I/O Control Register A1 | TRDIORA1 | 256, 272 |
| 0152h | Timer RD I/O Control Register C1 | TRDIORC1 | 257, 273 |
| 0153h | Timer RD Status Register 1 | TRDSR1 | 258, 274, 289, 302, 314, 330 |
| 0154h | Timer RD Interrupt Enable Register 1 | TRDIER1 | 259, 275, 290, 303, 315, 331 |
| 0155h | Timer RD PWM Mode Output Level Control Register 1 | TRDPOCR1 | 291 |
| 0156h | Timer RD Counter 1 | TRD1 | 259, 276, 291, 316 |
| 0157h | | | |
| 0158h | Timer RD General Register A1 | TRDGRA1 | 260, 276, 292, 304, 316, 332 |
| 0159h | | | |
| 015Ah | Timer RD General Register B1 | TRDGRB1 | 260, 276, 292, 304, 316, 332 |
| 015Bh | | | |
| 015Ch | Timer RD General Register C1 | TRDGRC1 | 260, 276, 292, 304, 316, 332 |
| 015Dh | | | |
| 015Eh | Timer RD General Register D1 | TRDGRD1 | 260, 276, 292, 304, 316, 332 |
| 015Fh | | | |

| Address | Register | Symbol | Page |
|---------|---|--------|------|
| 0160h | UART2 Transmit/Receive Mode Register | U2MR | 377 |
| 0161h | UART2 Bit Rate Register | U2BRG | 377 |
| 0162h | UART2 Transmit Buffer Register | U2TB | 378 |
| 0163h | | | |
| 0164h | UART2 Transmit/Receive Control Register 0 | U2C0 | 378 |
| 0165h | UART2 Transmit/Receive Control Register 1 | U2C1 | 379 |
| 0166h | UART2 Receive Buffer Register | U2RB | 379 |
| 0167h | | | |
| 0168h | | | |
| 0169h | | | |
| 016Ah | | | |
| 016Bh | | | |
| 016Ch | | | |
| 016Dh | | | |
| 016Eh | | | |
| 016Fh | | | |
| 0170h | | | |
| 0171h | | | |
| 0172h | | | |
| 0173h | | | |
| 0174h | | | |
| 0175h | | | |
| 0176h | | | |
| 0177h | | | |
| 0178h | | | |
| 0179h | | | |
| 017Ah | | | |
| 017Bh | | | |
| 017Ch | | | |
| 017Dh | | | |
| 017Eh | | | |
| 017Fh | | | |
| 0180h | | | |
| 0181h | | | |
| 0182h | | | |
| 0183h | | | |
| 0184h | | | |
| 0185h | | | |
| 0186h | | | |
| 0187h | | | |
| 0188h | | | |
| 0189h | | | |
| 018Ah | | | |
| 018Bh | | | |
| 018Ch | | | |
| 018Dh | | | |
| 018Eh | | | |
| 018Fh | | | |
| 0190h | | | |
| 0191h | | | |
| 0192h | | | |
| 0193h | | | |
| 0194h | | | |
| 0195h | | | |
| 0196h | | | |
| 0197h | | | |
| 0198h | | | |
| 0199h | | | |
| 019Ah | | | |
| 019Bh | | | |
| 019Ch | | | |
| 019Dh | | | |
| 019Eh | | | |
| 019Fh | | | |

| Address | Register | Symbol | Page |
|---------|---------------------------------|--------|------|
| 01A0h | | | |
| 01A1h | | | |
| 01A2h | | | |
| 01A3h | | | |
| 01A4h | | | |
| 01A5h | | | |
| 01A6h | | | |
| 01A7h | | | |
| 01A8h | | | |
| 01A9h | | | |
| 01AAh | | | |
| 01ABh | | | |
| 01ACh | | | |
| 01ADh | | | |
| 01AEh | | | |
| 01AFh | | | |
| 01B0h | | | |
| 01B1h | | | |
| 01B2h | | | |
| 01B3h | Flash Memory Control Register 4 | FMR4 | 500 |
| 01B4h | | | |
| 01B5h | Flash Memory Control Register 1 | FMR1 | 499 |
| 01B6h | | | |
| 01B7h | Flash Memory Control Register 0 | FMR0 | 498 |
| 01B8h | | | |
| 01B9h | | | |
| 01BAh | | | |
| 01BBh | | | |
| 01BCh | | | |
| 01BDh | | | |
| 01BEh | | | |
| 01C0h | | | |
| 01C1h | | | |
| 01C2h | | | |
| 01C3h | | | |
| 01C4h | | | |
| 01C5h | | | |
| 01C6h | | | |
| 01C7h | | | |
| 01C8h | | | |
| 01C9h | | | |
| 01CAh | | | |
| 01CBh | | | |
| 01CCh | | | |
| 01CDh | | | |
| 01CEh | | | |
| 01CFh | | | |
| 01D0h | | | |
| 01D1h | | | |
| 01D2h | | | |
| 01D3h | | | |
| 01D4h | | | |
| 01D5h | | | |
| 01D6h | | | |
| 01D7h | | | |
| 01D8h | | | |
| 01D9h | | | |
| 01DAh | | | |
| 01DBh | | | |
| 01DCh | | | |
| 01DDh | | | |
| 01DEh | | | |
| 01DFh | | | |

NOTE:
1. The blank regions are reserved. Do not access locations in these regions.

| Address | Register | Symbol | Page |
|---------|----------|--------|------|
| 01E0h | | | |
| 01E1h | | | |
| 01E2h | | | |
| 01E3h | | | |
| 01E4h | | | |
| 01E5h | | | |
| 01E6h | | | |
| 01E7h | | | |
| 01E8h | | | |
| 01E9h | | | |
| 01EAh | | | |
| 01EBh | | | |
| 01ECh | | | |
| 01EDh | | | |
| 01EEh | | | |
| 01EFh | | | |
| 01F0h | | | |
| 01F1h | | | |
| 01F2h | | | |
| 01F3h | | | |
| 01F4h | | | |
| 01F5h | | | |
| 01F6h | | | |
| 01F7h | | | |
| 01F8h | | | |
| 01F9h | | | |
| 01FAh | | | |
| 01FBh | | | |
| 01FCh | | | |
| 01FDh | | | |
| 01FEh | | | |
| 01FFh | | | |
| 0200h | | | |
| 0201h | | | |
| 0202h | | | |
| 0203h | | | |
| 0204h | | | |
| 0205h | | | |
| 0206h | | | |
| 0207h | | | |
| 0208h | | | |
| 0209h | | | |
| 020Ah | | | |
| 020Bh | | | |
| 020Ch | | | |
| 020Dh | | | |
| 020Eh | | | |
| 020Fh | | | |
| 0210h | | | |
| 0211h | | | |
| 0212h | | | |
| 0213h | | | |
| 0214h | | | |
| 0215h | | | |
| 0216h | | | |
| 0217h | | | |
| 0218h | | | |
| 0219h | | | |
| 021Ah | | | |
| 021Bh | | | |
| 021Ch | | | |
| 021Dh | | | |
| 021Eh | | | |
| 021Fh | | | |

| Address | Register | Symbol | Page |
|---------|----------|--------|------|
| 0220h | | | |
| 0221h | | | |
| 0222h | | | |
| 0223h | | | |
| 0224h | | | |
| 0225h | | | |
| 0226h | | | |
| 0227h | | | |
| 0228h | | | |
| 0229h | | | |
| 022Ah | | | |
| 022Bh | | | |
| 022Ch | | | |
| 022Dh | | | |
| 022Eh | | | |
| 022Fh | | | |
| 0230h | | | |
| 0231h | | | |
| 0232h | | | |
| 0233h | | | |
| 0234h | | | |
| 0235h | | | |
| 0236h | | | |
| 0237h | | | |
| 0238h | | | |
| 0239h | | | |
| 023Ah | | | |
| 023Bh | | | |
| 023Ch | | | |
| 023Dh | | | |
| 023Eh | | | |
| 023Fh | | | |
| 0240h | | | |
| 0241h | | | |
| 0242h | | | |
| 0243h | | | |
| 0244h | | | |
| 0245h | | | |
| 0246h | | | |
| 0247h | | | |
| 0248h | | | |
| 0249h | | | |
| 024Ah | | | |
| 024Bh | | | |
| 024Ch | | | |
| 024Dh | | | |
| 024Eh | | | |
| 024Fh | | | |
| 0250h | | | |
| 0251h | | | |
| 0252h | | | |
| 0253h | | | |
| 0254h | | | |
| 0255h | | | |
| 0256h | | | |
| 0257h | | | |
| 0258h | | | |
| 0259h | | | |
| 025Ah | | | |
| 025Bh | | | |
| 025Ch | | | |
| 025Dh | | | |
| 025Eh | | | |
| 025Fh | | | |

NOTE:
1. The blank regions are reserved. Do not access locations in these regions.

| Address | Register | Symbol | Page |
|---------|------------------------------|--------|------|
| 0260h | | | |
| 0261h | | | |
| 0262h | | | |
| 0263h | | | |
| 0264h | | | |
| 0265h | | | |
| 0266h | | | |
| 0267h | | | |
| 0268h | | | |
| 0269h | | | |
| 026Ah | | | |
| 026Bh | | | |
| 026Ch | | | |
| 026Dh | | | |
| 026Eh | | | |
| 026Fh | | | |
| 0270h | | | |
| 0271h | | | |
| 0272h | | | |
| 0273h | | | |
| 0274h | | | |
| 0275h | | | |
| 0276h | | | |
| 0277h | | | |
| 0278h | | | |
| 0279h | | | |
| 027Ah | | | |
| 027Bh | | | |
| 027Ch | | | |
| 027Dh | | | |
| 027Eh | | | |
| 027Fh | | | |
| 0280h | | | |
| 0281h | | | |
| 0282h | | | |
| 0283h | | | |
| 0284h | | | |
| 0285h | | | |
| 0286h | | | |
| 0287h | | | |
| 0288h | | | |
| 0289h | | | |
| 028Ah | | | |
| 028Bh | | | |
| 028Ch | | | |
| 028Dh | | | |
| 028Eh | | | |
| 028Fh | | | |
| 0290h | Timer RF Register | TRF | 363 |
| 0291h | | | |
| 0292h | | | |
| 0293h | | | |
| 0294h | | | |
| 0295h | | | |
| 0296h | | | |
| 0297h | | | |
| 0298h | | | |
| 0299h | | | |
| 029Ah | Timer RF Control Register 0 | TRFCR0 | 364 |
| 029Bh | Timer RF Control Register 1 | TRFCR1 | 365 |
| 029Ch | Capture / Compare 0 Register | TRFM0 | 363 |
| 029Dh | | | |
| 029Eh | Compare 1 Register | TRFM1 | 363 |
| 029Fh | | | |

| Address | Register | Symbol | Page |
|---------|------------------------|--------|------|
| 02A0h | | | |
| 02A1h | | | |
| 02A2h | | | |
| 02A3h | | | |
| 02A4h | | | |
| 02A5h | | | |
| 02A6h | | | |
| 02A7h | | | |
| 02A8h | | | |
| 02A9h | | | |
| 02AAh | | | |
| 02ABh | | | |
| 02ACh | | | |
| 02ADh | | | |
| 02AEh | | | |
| 02AFh | | | |
| 02B0h | | | |
| 02B1h | | | |
| 02B2h | | | |
| 02B3h | | | |
| 02B4h | | | |
| 02B5h | | | |
| 02B6h | | | |
| 02B7h | | | |
| 02B8h | | | |
| 02B9h | | | |
| 02BAh | | | |
| 02BBh | | | |
| 02BCh | | | |
| 02BDh | | | |
| 02BEh | | | |
| 02BFh | | | |
| 02C0h | A/D Register 0 | AD0 | 474 |
| 02C1h | | | |
| 02C2h | | | |
| 02C3h | | | |
| 02C4h | | | |
| 02C5h | | | |
| 02C6h | | | |
| 02C7h | | | |
| 02C8h | | | |
| 02C9h | | | |
| 02CAh | | | |
| 02CBh | | | |
| 02CCh | | | |
| 02CDh | | | |
| 02CEh | | | |
| 02CFh | | | |
| 02D0h | | | |
| 02D1h | | | |
| 02D2h | | | |
| 02D3h | | | |
| 02D4h | A/D Control Register 2 | ADCON2 | 474 |
| 02D5h | | | |
| 02D6h | A/D Control Register 0 | ADCON0 | 475 |
| 02D7h | A/D Control Register 1 | ADCON1 | 476 |
| 02D8h | | | |
| 02D9h | | | |
| 02DAh | | | |
| 02DBh | | | |
| 02DCh | | | |
| 02DDh | | | |
| 02DEh | | | |
| 02DFh | | | |

NOTE:

1. The blank regions are reserved. Do not access locations in these regions.

| Address | Register | Symbol | Page |
|---------|----------------------------------|--------|--------------|
| 02E0h | | | |
| 02E1h | | | |
| 02E2h | | | |
| 02E3h | | | |
| 02E4h | Port P8 Direction Register | PD8 | 68 |
| 02E5h | | | |
| 02E6h | Port P8 Register | P8 | 69 |
| 02E7h | | | |
| 02E8h | | | |
| 02E9h | | | |
| 02EAh | | | |
| 02EBh | | | |
| 02ECh | | | |
| 02EDh | | | |
| 02EEh | | | |
| 02EFh | | | |
| 02F0h | | | |
| 02F1h | | | |
| 02F2h | | | |
| 02F3h | | | |
| 02F4h | | | |
| 02F5h | | | |
| 02F6h | | | |
| 02F7h | | | |
| 02F8h | | | |
| 02F9h | | | |
| 02FAh | | | |
| 02FBh | | | |
| 02FCh | Pull-Up Control Register 2 | PUR2 | 71 |
| 02FDh | | | |
| 02FEh | | | |
| 02FFh | Timer RF Output Control Register | TRFOUT | 365 |
| FFFh | Option Function Select Register | OFS | 36, 152, 493 |

NOTE:

1. The blank regions are reserved. Do not access locations in these regions.

1. Overview

1.1 Features

The R8C/2A Group and R8C/2B Group of single-chip MCUs incorporates the R8C/Tiny Series CPU core, employing sophisticated instructions for a high level of efficiency. With 1 Mbyte of address space, and it is capable of executing instructions at high speed. In addition, the CPU core boasts a multiplier for high-speed operation processing.

Power consumption is low, and the supported operating modes allow additional power control. These MCUs also use an anti-noise configuration to reduce emissions of electromagnetic noise and are designed to withstand EMI.

Integration of many peripheral functions, including multifunction timer and serial interface, reduces the number of system components.

Furthermore, the R8C/2B Group has on-chip data flash (1 KB × 2 blocks).

The difference between the R8C/2A Group and R8C/2B Group is only the presence or absence of data flash. Their peripheral functions are the same.

1.1.1 Applications

Electronic household appliances, office equipment, audio equipment, consumer equipment, etc.

1.1.2 Specifications

Tables 1.1 and 1.2 outlines the Specifications for R8C/2A Group and Tables 1.3 and 1.4 outlines the Specifications for R8C/2B Group.

Table 1.1 Specifications for R8C/2A Group (1)

| Item | Function | Specification |
|--------------------------------|---------------------------|---|
| CPU | Central processing unit | R8C/Tiny series core <ul style="list-style-type: none"> Number of fundamental instructions: 89 Minimum instruction execution time: <ul style="list-style-type: none"> 50 ns ($f(XIN) = 20$ MHz, $VCC = 3.0$ to 5.5 V) 100 ns ($f(XIN) = 10$ MHz, $VCC = 2.7$ to 5.5 V) 200 ns ($f(XIN) = 5$ MHz, $VCC = 2.2$ to 5.5 V) Multiplier: 16 bits \times 16 bits \rightarrow 32 bits Multiply-accumulate instruction: 16 bits \times 16 bits + 32 bits \rightarrow 32 bits Operation mode: Single-chip mode (address space: 1 Mbyte) |
| Memory | ROM, RAM | Refer to Table 1.5 Product List for R8C/2A Group . |
| Power Supply Voltage Detection | Voltage detection circuit | <ul style="list-style-type: none"> Power-on reset Voltage detection 2 |
| I/O Ports | Programmable I/O ports | <ul style="list-style-type: none"> Input-only: 2 pins CMOS I/O ports: 55, selectable pull-up resistor High current drive ports: 8 |
| Clock | Clock generation circuits | 3 circuits: XIN clock oscillation circuit (with on-chip feedback resistor), On-chip oscillator (high-speed, low-speed) (high-speed on-chip oscillator has a frequency adjustment function), XCIN clock oscillation circuit (32 kHz) <ul style="list-style-type: none"> Oscillation stop detection: XIN clock oscillation stop detection function Frequency divider circuit: Dividing selectable 1, 2, 4, 8, and 16 Low power consumption modes: <ul style="list-style-type: none"> Standard operating mode (high-speed clock, low-speed clock, high-speed on-chip oscillator, low-speed on-chip oscillator), wait mode, stop mode Real-time clock (timer RE) |
| Interrupts | | <ul style="list-style-type: none"> External: 5 sources, Internal: 23 sources, Software: 4 sources Priority levels: 7 levels |
| Watchdog Timer | | 15 bits \times 1 (with prescaler), reset start selectable |
| Timer | Timer RA | 8 bits \times 1 (with 8-bit prescaler) Timer mode (period timer), pulse output mode (output level inverted every period), event counter mode, pulse width measurement mode, pulse period measurement mode |
| | Timer RB | 8 bits \times 1 (with 8-bit prescaler) Timer mode (period timer), programmable waveform generation mode (PWM output), programmable one-shot generation mode, programmable wait one-shot generation mode |
| | Timer RC | 16 bits \times 1 (with 4 capture/compare registers) Timer mode (input capture function, output compare function), PWM mode (output 3 pins), PWM2 mode (PWM output pin) |
| | Timer RD | 16 bits \times 2 (with 4 capture/compare registers) Timer mode (input capture function, output compare function), PWM mode (output 6 pins), reset synchronous PWM mode (output three-phase waveforms (6 pins), sawtooth wave modulation), complementary PWM mode (output three-phase waveforms (6 pins), triangular wave modulation), PWM3 mode (PWM output 2 pins with fixed period) |
| | Timer RE | 8 bits \times 1 Real-time clock mode (count seconds, minutes, hours, days of week), output compare mode |
| | Timer RF | 16 bits \times 1 (with capture/compare register pin and compare register pin) Input capture mode, output compare mode |

Table 1.2 Specifications for R8C/2A Group (2)

| Item | Function | Specification |
|---|---------------------|---|
| Serial Interface | UART0, UART1, UART2 | Clock synchronous serial I/O/UART × 3 |
| Clock Synchronous Serial I/O with Chip Select (SSU) | | 1 (shared with I ² C-bus) |
| I ² C bus ⁽¹⁾ | | 1 (shared with SSU) |
| LIN Module | | Hardware LIN: 1 (timer RA, UART0) |
| A/D Converter | | 10-bit resolution × 12 channels, includes sample and hold function |
| D/A Converter | | 8-bit resolution × 2 circuits |
| Flash Memory | | <ul style="list-style-type: none"> • Programming and erasure voltage: VCC = 2.7 to 5.5 V • Programming and erasure endurance: 100 times • Program security: ROM code protect, ID code check • Debug functions: On-chip debug, on-board flash rewrite function |
| Operating Frequency/Supply Voltage | | f(XIN) = 20 MHz (VCC = 3.0 to 5.5 V) f(XIN) = 10 MHz (VCC = 2.7 to 5.5 V) f(XIN) = 5 MHz (VCC = 2.2 to 5.5 V) |
| Current consumption | | 12 mA (VCC = 5.0 V, f(XIN) = 20 MHz) 5.5 mA (VCC = 3.0 V, f(XIN) = 10 MHz) 2.1 μA (VCC = 3.0 V, wait mode (f(XCIN) = 32 kHz)) 0.65 μA (VCC = 3.0 V, stop mode) |
| Operating Ambient Temperature | | -20 to 85°C (N version) -40 to 85°C (D version) ⁽²⁾ -20 to 105°C (Y version) ⁽³⁾ |
| Package | | 64-pin LQFP <ul style="list-style-type: none"> • Package code: PLQP0064KB-A (previous code: 64P6Q-A) • Package code: PLQP0064GA-A (previous code: 64P6U-A) 64-pin FLGA <ul style="list-style-type: none"> • Package code: PTLG0064JA-A (previous code: 64F0G) |

NOTES:

1. I²C bus is a trademark of Koninklijke Philips Electronics N. V.
2. Specify the D version if D version functions are to be used.
3. Please contact Renesas Technology sales offices for the Y version.

Table 1.3 Specifications for R8C/2B Group (1)

| Item | Function | Specification |
|--------------------------------|---------------------------|---|
| CPU | Central processing unit | R8C/Tiny series core <ul style="list-style-type: none"> • Number of fundamental instructions: 89 • Minimum instruction execution time: <ul style="list-style-type: none"> 50 ns ($f(XIN) = 20$ MHz, $VCC = 3.0$ to 5.5 V) 100 ns ($f(XIN) = 10$ MHz, $VCC = 2.7$ to 5.5 V) 200 ns ($f(XIN) = 5$ MHz, $VCC = 2.2$ to 5.5 V) • Multiplier: 16 bits \times 16 bits \rightarrow 32 bits • Multiply-accumulate instruction: 16 bits \times 16 bits + 32 bits \rightarrow 32 bits • Operation mode: Single-chip mode (address space: 1 Mbyte) |
| Memory | ROM, RAM | Refer to Table 1.6 Product List for R8C/2B Group . |
| Power Supply Voltage Detection | Voltage detection circuit | <ul style="list-style-type: none"> • Power-on reset • Voltage detection 2 |
| I/O Ports | Programmable I/O ports | <ul style="list-style-type: none"> • Input-only: 2 pins • CMOS I/O ports: 55, selectable pull-up resistor • High current drive ports: 8 |
| Clock | Clock generation circuits | <p>3 circuits: XIN clock oscillation circuit (with on-chip feedback resistor), On-chip oscillator (high-speed, low-speed) (high-speed on-chip oscillator has a frequency adjustment function), XCIN clock oscillation circuit (32 kHz)</p> <ul style="list-style-type: none"> • Oscillation stop detection: XIN clock oscillation stop detection function • Frequency divider circuit: Dividing selectable 1, 2, 4, 8, and 16 • Low power consumption modes: <ul style="list-style-type: none"> Standard operating mode (high-speed clock, low-speed clock, high-speed on-chip oscillator, low-speed on-chip oscillator), wait mode, stop mode <p>Real-time clock (timer RE)</p> |
| Interrupts | | <ul style="list-style-type: none"> • External: 5 sources, Internal: 23 sources, Software: 4 sources • Priority levels: 7 levels |
| Watchdog Timer | | 15 bits \times 1 (with prescaler), reset start selectable |
| Timer | Timer RA | 8 bits \times 1 (with 8-bit prescaler) Timer mode (period timer), pulse output mode (output level inverted every period), event counter mode, pulse width measurement mode, pulse period measurement mode |
| | Timer RB | 8 bits \times 1 (with 8-bit prescaler) Timer mode (period timer), programmable waveform generation mode (PWM output), programmable one-shot generation mode, programmable wait one-shot generation mode |
| | Timer RC | 16 bits \times 1 (with 4 capture/compare registers) Timer mode (input capture function, output compare function), PWM mode (output 3 pins), PWM2 mode (PWM output pin) |
| | Timer RD | 16 bits \times 2 (with 4 capture/compare registers) Timer mode (input capture function, output compare function), PWM mode (output 6 pins), reset synchronous PWM mode (output three-phase waveforms (6 pins), sawtooth wave modulation), complementary PWM mode (output three-phase waveforms (6 pins), triangular wave modulation), PWM3 mode (PWM output 2 pins with fixed period) |
| | Timer RE | 8 bits \times 1 Real-time clock mode (count seconds, minutes, hours, days of week), output compare mode |
| | Timer RF | 16 bits \times 1 (with capture/compare register pin and compare register pin) Input capture mode, output compare mode |

Table 1.4 Specifications for R8C/2B Group (2)

| Item | Function | Specification |
|---|---------------------|---|
| Serial Interface | UART0, UART1, UART2 | Clock synchronous serial I/O/UART × 3 |
| Clock Synchronous Serial I/O with Chip Select (SSU) | | 1 (shared with I ² C-bus) |
| I ² C bus ⁽¹⁾ | | 1 (shared with SSU) |
| LIN Module | | Hardware LIN: 1 (timer RA, UART0) |
| A/D Converter | | 10-bit resolution × 12 channels, includes sample and hold function |
| D/A Converter | | 8-bit resolution × 2 circuits |
| Flash Memory | | <ul style="list-style-type: none"> • Programming and erasure voltage: VCC = 2.7 to 5.5 V • Programming and erasure endurance: 10,000 times (data flash) 1,000 times (program ROM) • Program security: ROM code protect, ID code check • Debug functions: On-chip debug, on-board flash rewrite function |
| Operating Frequency/Supply Voltage | | f(XIN) = 20 MHz (VCC = 3.0 to 5.5 V) f(XIN) = 10 MHz (VCC = 2.7 to 5.5 V) f(XIN) = 5 MHz (VCC = 2.2 to 5.5 V) |
| Current consumption | | 12 mA (VCC = 5.0 V, f(XIN) = 20 MHz) 5.5 mA (VCC = 3.0 V, f(XIN) = 10 MHz) 2.1 μA (VCC = 3.0 V, wait mode (f(XCIN) = 32 kHz)) 0.65 μA (VCC = 3.0 V, stop mode) |
| Operating Ambient Temperature | | -20 to 85°C (N version) -40 to 85°C (D version) ⁽²⁾ -20 to 105°C (Y version) ⁽³⁾ |
| Package | | 64-pin LQFP <ul style="list-style-type: none"> • Package code: PLQP0064KB-A (previous code: 64P6Q-A) • Package code: PLQP0064GA-A (previous code: 64P6U-A) 64-pin FLGA <ul style="list-style-type: none"> • Package code: PTLG0064JA-A (previous code: 64F0G) |

NOTES:

1. I²C bus is a trademark of Koninklijke Philips Electronics N. V.
2. Specify the D version if D version functions are to be used.
3. Please contact Renesas Technology sales offices for the Y version.

1.2 Product List

Table 1.5 lists Product List for R8C/2A Group, Figure 1.1 shows a Part Number, Memory Size, and Package of R8C/2A Group, Table 1.6 lists Product List for R8C/2B Group, and Figure 1.2 shows a Part Number, Memory Size, and Package of R8C/2B Group.

Table 1.5 Product List for R8C/2A Group

Current of Nov. 2007

| Part No. | ROM Capacity | RAM Capacity | Package Type | Remarks | | | |
|------------------|--------------|--------------|--------------|-----------|--|-----------|--|
| R5F212A7SNFP | 48 Kbytes | 2.5 Kbytes | PLQP0064KB-A | N version | | | |
| R5F212A7SNFA | 48 Kbytes | 2.5 Kbytes | PLQP0064GA-A | | | | |
| R5F212A7SNLNG | 48 Kbytes | 2.5 Kbytes | PTLG0064JA-A | | | | |
| R5F212A8SNFP | 64 Kbytes | 3 Kbytes | PLQP0064KB-A | | | | |
| R5F212A8SNFA | 64 Kbytes | 3 Kbytes | PLQP0064GA-A | | | | |
| R5F212A8SNLNG | 64 Kbytes | 3 Kbytes | PLTG0064JA-A | | | | |
| R5F212AASNFP | 96 Kbytes | 7 Kbytes | PLQP0064KB-A | | | | |
| R5F212AASNFA | 96 Kbytes | 7 Kbytes | PLQP0064GA-A | | | | |
| R5F212AASNLNG | 96 Kbytes | 7 Kbytes | PLTG0064JA-A | | | | |
| R5F212ACSNFP | 128 Kbytes | 7.5 Kbytes | PLQP0064KB-A | | | | |
| R5F212ACSNFA | 128 Kbytes | 7.5 Kbytes | PLQP0064GA-A | | | | |
| R5F212ACSNLNG | 128 Kbytes | 7.5 Kbytes | PLTG0064JA-A | | | | |
| R5F212A7SDFP | 48 Kbytes | 2.5 Kbytes | PLQP0064KB-A | D version | | | |
| R5F212A7SDFA | 48 Kbytes | 2.5 Kbytes | PLQP0064GA-A | | | | |
| R5F212A8SDFP | 64 Kbytes | 3 Kbytes | PLQP0064KB-A | | | | |
| R5F212A8SDFA | 64 Kbytes | 3 Kbytes | PLQP0064GA-A | | | | |
| R5F212AASDFP | 96 Kbytes | 7 Kbytes | PLQP0064KB-A | | | | |
| R5F212AASDFA | 96 Kbytes | 7 Kbytes | PLQP0064GA-A | | | | |
| R5F212ACSDFP | 128 Kbytes | 7.5 Kbytes | PLQP0064KB-A | | | | |
| R5F212ACSDFA | 128 Kbytes | 7.5 Kbytes | PLQP0064GA-A | | | | |
| R5F212A7SNXXXFP | 48 Kbytes | 2.5 Kbytes | PLQP0064KB-A | | | N version | Factory programming product ⁽¹⁾ |
| R5F212A7SNXXXFA | 48 Kbytes | 2.5 Kbytes | PLQP0064GA-A | | | | |
| R5F212A7SNXXXLNG | 48 Kbytes | 2.5 Kbytes | PTLG0064JA-A | | | | |
| R5F212A8SNXXXFP | 64 Kbytes | 3 Kbytes | PLQP0064KB-A | | | | |
| R5F212A8SNXXXFA | 64 Kbytes | 3 Kbytes | PLQP0064GA-A | | | | |
| R5F212A8SNXXXLNG | 64 Kbytes | 3 Kbytes | PLTG0064JA-A | | | | |
| R5F212AASNXXXFP | 96 Kbytes | 7 Kbytes | PLQP0064KB-A | | | | |
| R5F212AASNXXXFA | 96 Kbytes | 7 Kbytes | PLQP0064GA-A | | | | |
| R5F212AASNXXXLNG | 96 Kbytes | 7 Kbytes | PLTG0064JA-A | | | | |
| R5F212ACSNXXXFP | 128 Kbytes | 7.5 Kbytes | PLQP0064KB-A | | | | |
| R5F212ACSNXXXFA | 128 Kbytes | 7.5 Kbytes | PLQP0064GA-A | | | | |
| R5F212ACSNXXXLNG | 128 Kbytes | 7.5 Kbytes | PLTG0064JA-A | | | | |
| R5F212A7SDXXXFP | 48 Kbytes | 2.5 Kbytes | PLQP0064KB-A | D version | | | |
| R5F212A7SDXXXFA | 48 Kbytes | 2.5 Kbytes | PLQP0064GA-A | | | | |
| R5F212A8SDXXXFP | 64 Kbytes | 3 Kbytes | PLQP0064KB-A | | | | |
| R5F212A8SDXXXFA | 64 Kbytes | 3 Kbytes | PLQP0064GA-A | | | | |
| R5F212AASDXXXFP | 96 Kbytes | 7 Kbytes | PLQP0064KB-A | | | | |
| R5F212AASDXXXFA | 96 Kbytes | 7 Kbytes | PLQP0064GA-A | | | | |
| R5F212ACSDXXXFP | 128 Kbytes | 7.5 Kbytes | PLQP0064KB-A | | | | |
| R5F212ACSDXXXFA | 128 Kbytes | 7.5 Kbytes | PLQP0064GA-A | | | | |

NOTE:

1. The user ROM is programmed before shipment.

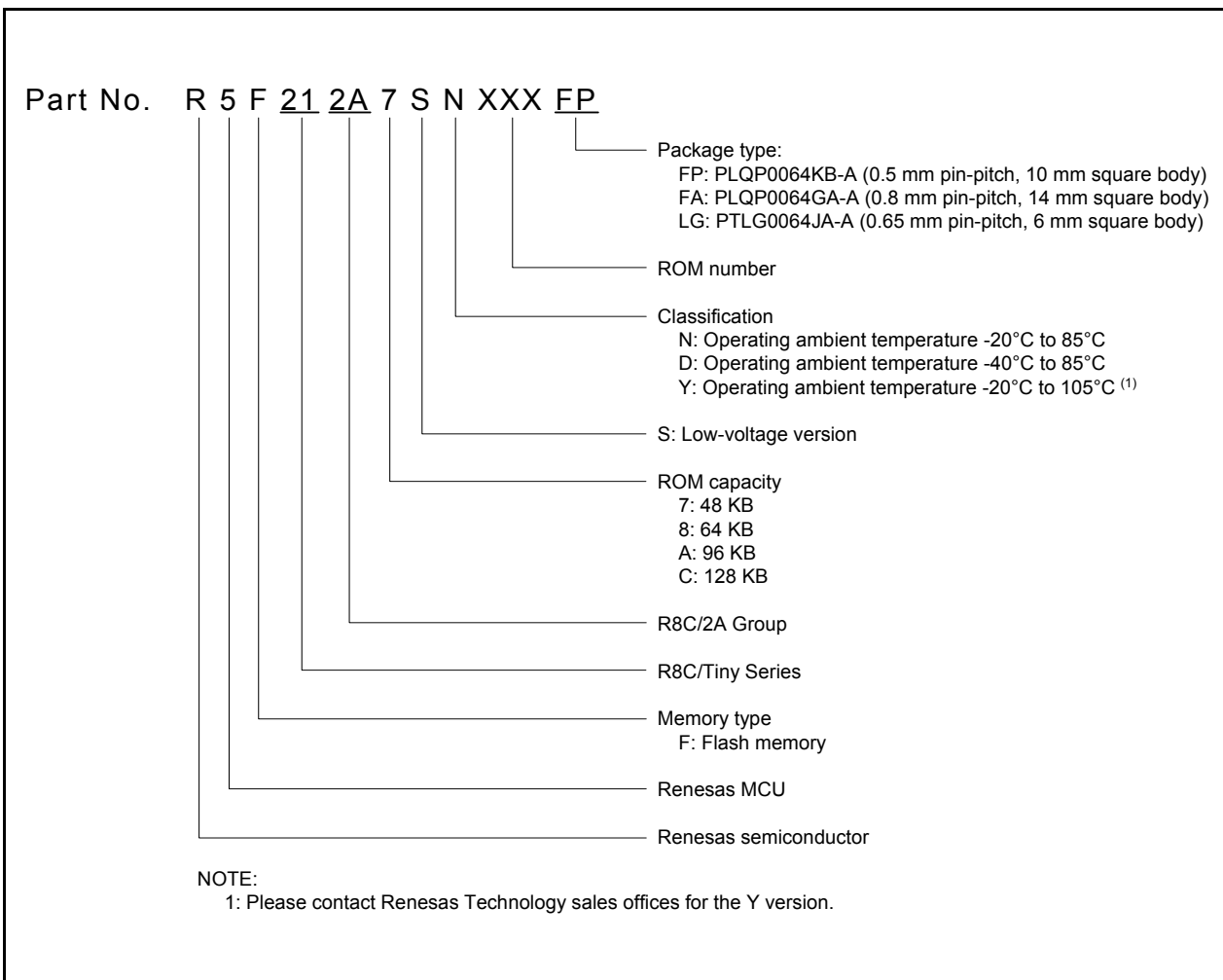


Figure 1.1 Part Number, Memory Size, and Package of R8C/2A Group

Table 1.6 Product List for R8C/2B Group

Current of Nov. 2007

| Part No. | ROM Capacity | | RAM Capacity | Package Type | Remarks | | | |
|-----------------|--------------|-------------|--------------|--------------|-----------|--|-----------|--|
| | Program ROM | Data flash | | | | | | |
| R5F212B7SNFP | 48 Kbytes | 1 Kbyte × 2 | 2.5 Kbytes | PLQP0064KB-A | N version | | | |
| R5F212B7SNFA | 48 Kbytes | 1 Kbyte × 2 | 2.5 Kbytes | PLQP0064GA-A | | | | |
| R5F212B7SNLG | 48 Kbytes | 1 Kbyte × 2 | 2.5 Kbytes | PTLG0064JA-A | | | | |
| R5F212B8SNFP | 64 Kbytes | 1 Kbyte × 2 | 3 Kbytes | PLQP0064KB-A | | | | |
| R5F212B8SNFA | 64 Kbytes | 1 Kbyte × 2 | 3 Kbytes | PLQP0064GA-A | | | | |
| R5F212B8SNLG | 64 Kbytes | 1 Kbyte × 2 | 3 Kbytes | PTLG0064JA-A | | | | |
| R5F212BASNFP | 96 Kbytes | 1 Kbyte × 2 | 7 Kbytes | PLQP0064KB-A | | | | |
| R5F212BASNFA | 96 Kbytes | 1 Kbyte × 2 | 7 Kbytes | PLQP0064GA-A | | | | |
| R5F212BASNLG | 96 Kbytes | 1 Kbyte × 2 | 7 Kbytes | PTLG0064JA-A | | | | |
| R5F212BCSNFP | 128 Kbytes | 1 Kbyte × 2 | 7.5 Kbytes | PLQP0064KB-A | | | | |
| R5F212BCSNFA | 128 Kbytes | 1 Kbyte × 2 | 7.5 Kbytes | PLQP0064GA-A | | | | |
| R5F212BCSNLG | 128 Kbytes | 1 Kbyte × 2 | 7.5 Kbytes | PTLG0064JA-A | | | | |
| R5F212B7SDFP | 48 Kbytes | 1 Kbyte × 2 | 2.5 Kbytes | PLQP0064KB-A | | | D version | |
| R5F212B7SDFA | 48 Kbytes | 1 Kbyte × 2 | 2.5 Kbytes | PLQP0064GA-A | | | | |
| R5F212B8SDFP | 64 Kbytes | 1 Kbyte × 2 | 3 Kbytes | PLQP0064KB-A | | | | |
| R5F212B8SDFA | 64 Kbytes | 1 Kbyte × 2 | 3 Kbytes | PLQP0064GA-A | | | | |
| R5F212BASDFP | 96 Kbytes | 1 Kbyte × 2 | 7 Kbytes | PLQP0064KB-A | | | | |
| R5F212BASDFA | 96 Kbytes | 1 Kbyte × 2 | 7 Kbytes | PLQP0064GA-A | | | | |
| R5F212BCSDFP | 128 Kbytes | 1 Kbyte × 2 | 7.5 Kbytes | PLQP0064KB-A | | | | |
| R5F212BCSDFA | 128 Kbytes | 1 Kbyte × 2 | 7.5 Kbytes | PLQP0064GA-A | | | | |
| R5F212B7SNXXXFP | 48 Kbytes | 1 Kbyte × 2 | 2.5 Kbytes | PLQP0064KB-A | N version | Factory programming product ⁽¹⁾ | | |
| R5F212B7SNXXXFA | 48 Kbytes | 1 Kbyte × 2 | 2.5 Kbytes | PLQP0064GA-A | | | | |
| R5F212B7SNXXXLG | 48 Kbytes | 1 Kbyte × 2 | 2.5 Kbytes | PTLG0064JA-A | | | | |
| R5F212B8SNXXXFP | 64 Kbytes | 1 Kbyte × 2 | 3 Kbytes | PLQP0064KB-A | | | | |
| R5F212B8SNXXXFA | 64 Kbytes | 1 Kbyte × 2 | 3 Kbytes | PLQP0064GA-A | | | | |
| R5F212B8SNXXXLG | 64 Kbytes | 1 Kbyte × 2 | 3 Kbytes | PTLG0064JA-A | | | | |
| R5F212BASNXXXFP | 96 Kbytes | 1 Kbyte × 2 | 7 Kbytes | PLQP0064KB-A | | | | |
| R5F212BASNXXXFA | 96 Kbytes | 1 Kbyte × 2 | 7 Kbytes | PLQP0064GA-A | | | | |
| R5F212BASNXXXLG | 96 Kbytes | 1 Kbyte × 2 | 7 Kbytes | PTLG0064JA-A | | | | |
| R5F212BCSNXXXFP | 128 Kbytes | 1 Kbyte × 2 | 7.5 Kbytes | PLQP0064KB-A | | | | |
| R5F212BCSNXXXFA | 128 Kbytes | 1 Kbyte × 2 | 7.5 Kbytes | PLQP0064GA-A | | | | |
| R5F212BCSNXXXLG | 128 Kbytes | 1 Kbyte × 2 | 7.5 Kbytes | PTLG0064JA-A | | | | |
| R5F212B7SDXXXFP | 48 Kbytes | 1 Kbyte × 2 | 2.5 Kbytes | PLQP0064KB-A | | | D version | |
| R5F212B7SDXXXFA | 48 Kbytes | 1 Kbyte × 2 | 2.5 Kbytes | PLQP0064GA-A | | | | |
| R5F212B8SDXXXFP | 64 Kbytes | 1 Kbyte × 2 | 3 Kbytes | PLQP0064KB-A | | | | |
| R5F212B8SDXXXFA | 64 Kbytes | 1 Kbyte × 2 | 3 Kbytes | PLQP0064GA-A | | | | |
| R5F212BASDXXXFP | 96 Kbytes | 1 Kbyte × 2 | 7 Kbytes | PLQP0064KB-A | | | | |
| R5F212BASDXXXFA | 96 Kbytes | 1 Kbyte × 2 | 7 Kbytes | PLQP0064GA-A | | | | |
| R5F212BCSDXXXFP | 128 Kbytes | 1 Kbyte × 2 | 7.5 Kbytes | PLQP0064KB-A | | | | |
| R5F212BCSDXXXFA | 128 Kbytes | 1 Kbyte × 2 | 7.5 Kbytes | PLQP0064GA-A | | | | |

NOTE:

1. The user ROM is programmed before shipment.

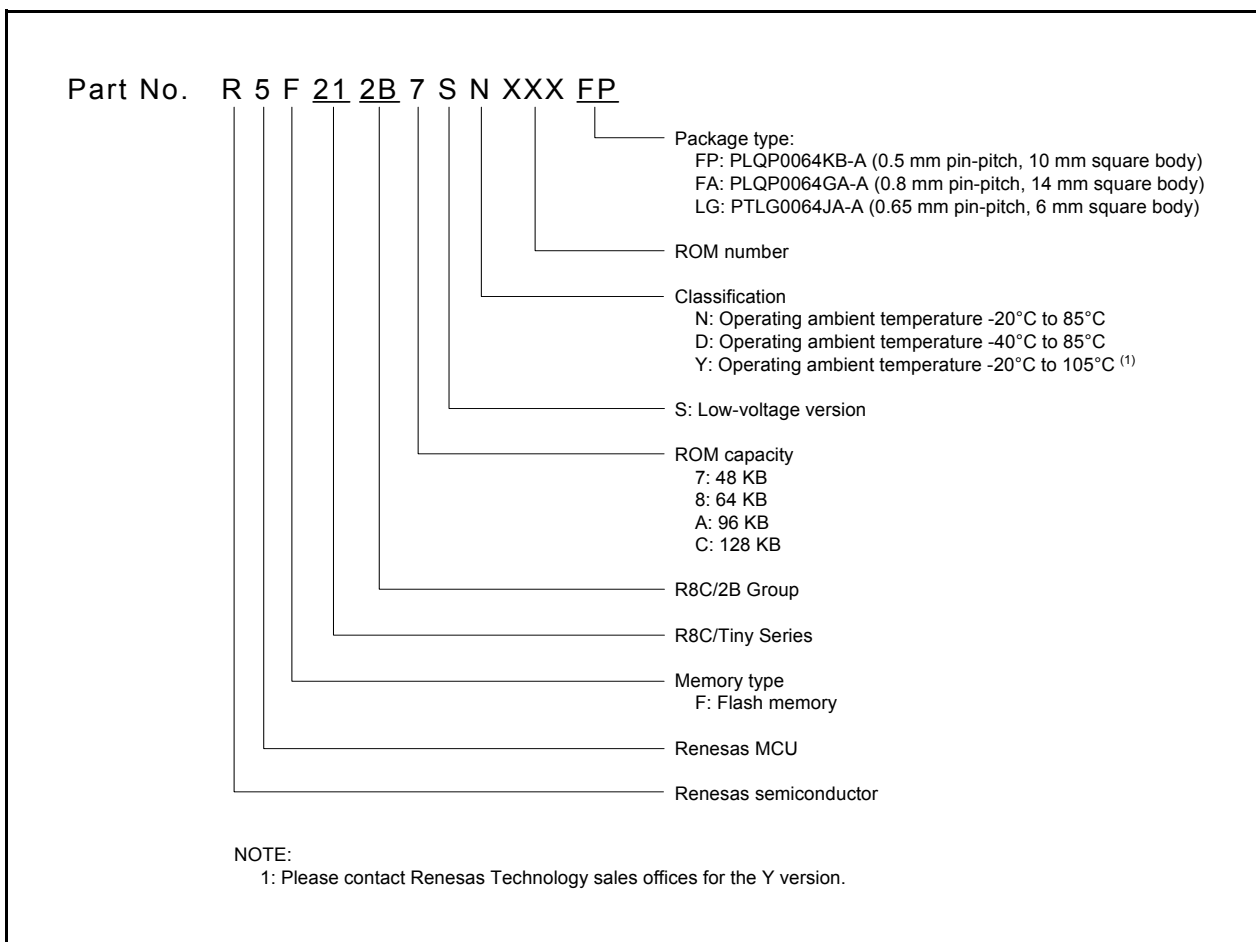


Figure 1.2 Part Number, Memory Size, and Package of R8C/2B Group

1.3 Block Diagram

Figure 1.3 shows a Block Diagram.

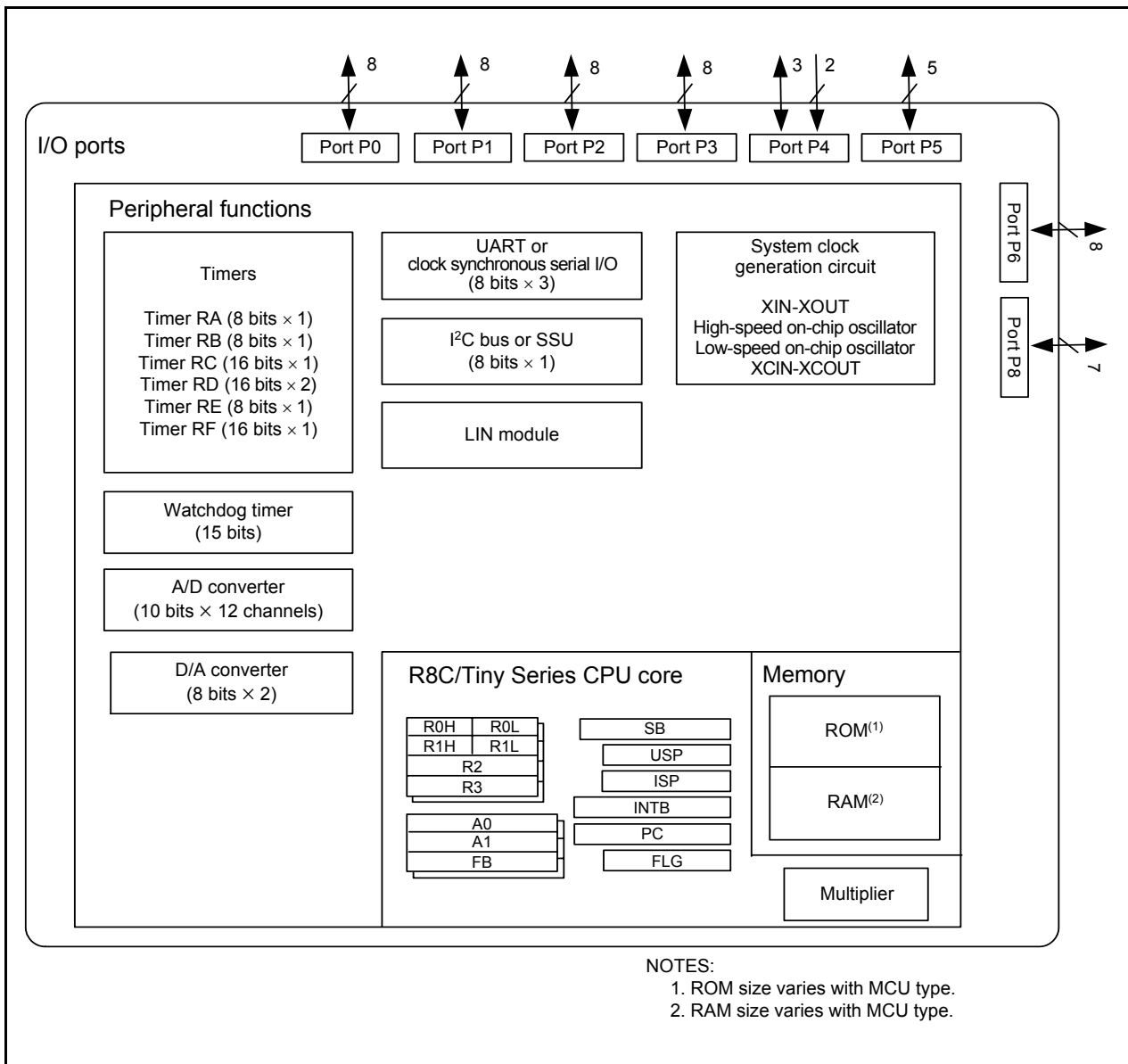


Figure 1.3 Block Diagram

1.4 Pin Assignment

Figure 1.4 shows 64-pin LQFP Package Pin Assignment (Top View). Figure 1.5 shows 64-pin FLGA Package Pin Assignment (Top Perspective View). Tables 1.7 and 1.8 outlines the Pin Name Information by Pin Number.

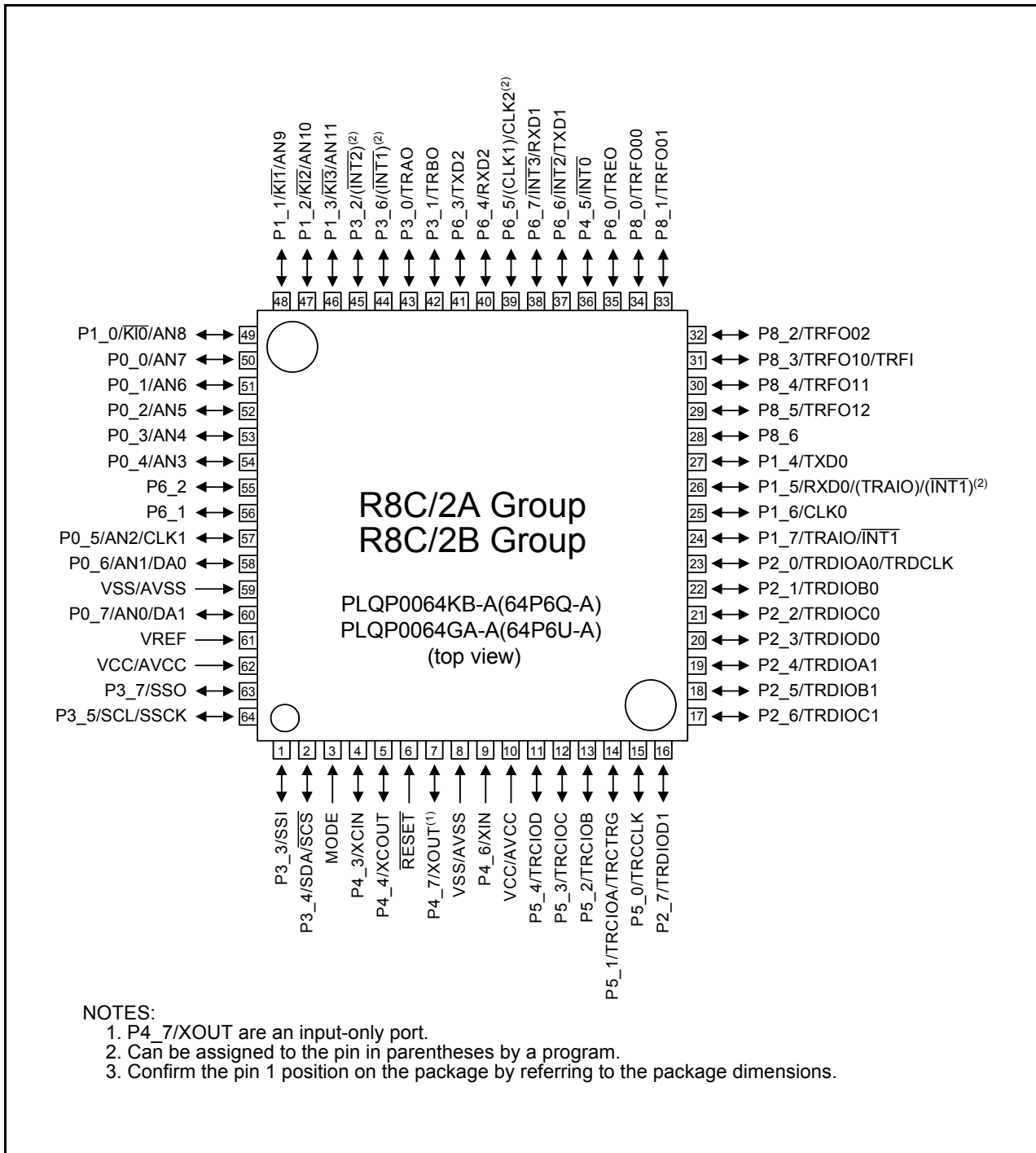


Figure 1.4 64-pin LQFP Package Pin Assignment (Top View)

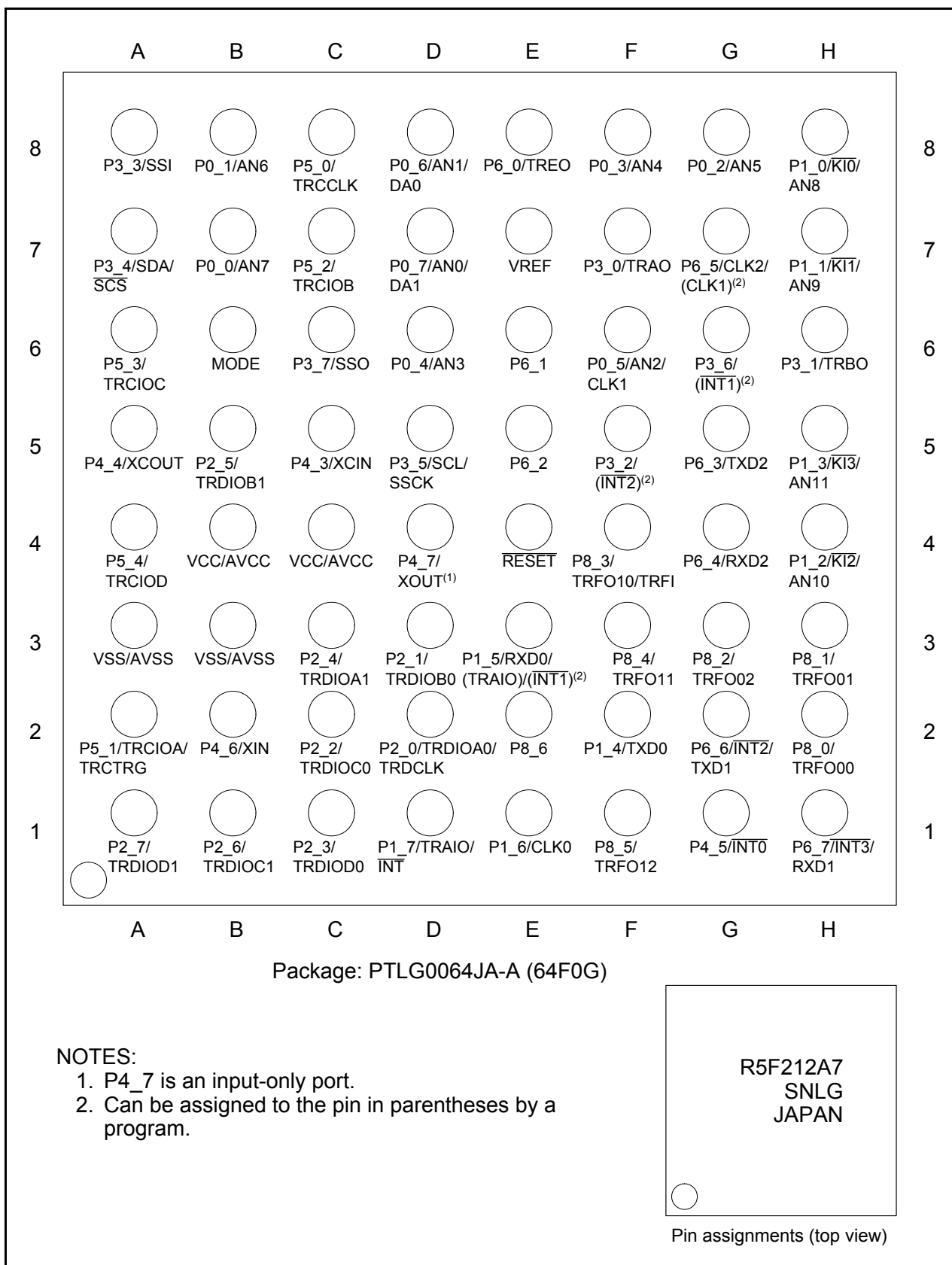


Figure 1.5 64-pin FLGA Package Pin Assignment (Top Perspective View)

Table 1.7 Pin Name Information by Pin Number (1)

| Pin Number | Control Pin | Port | I/O Pin Functions for of Peripheral Modules | | | | | |
|------------|-------------|------|---|------------------------|---------------------------------|-----|----------------------|------------------------------|
| | | | Interrupt | Timer | Serial Interface | SSU | I ² C bus | A/D Converter, D/A Converter |
| 1 | | P3_3 | | | | SSI | | |
| 2 | | P3_4 | | | | SCS | SDA | |
| 3 | MODE | | | | | | | |
| 4 | XCIN | P4_3 | | | | | | |
| 5 | XCOU | P4_4 | | | | | | |
| 6 | RESET | | | | | | | |
| 7 | XOUT | P4_7 | | | | | | |
| 8 | VSS/AVSS | | | | | | | |
| 9 | XIN | P4_6 | | | | | | |
| 10 | VCC/AVCC | | | | | | | |
| 11 | | P5_4 | | TRCIOD | | | | |
| 12 | | P5_3 | | TRCIOC | | | | |
| 13 | | P5_2 | | TRCIOB | | | | |
| 14 | | P5_1 | | TRCIOA/TRCTR | | | | |
| 15 | | P5_0 | | TRCCLK | | | | |
| 16 | | P2_7 | | TRDIOD1 | | | | |
| 17 | | P2_6 | | TRDIOC1 | | | | |
| 18 | | P2_5 | | TRDIQB1 | | | | |
| 19 | | P2_4 | | TRDIOA1 | | | | |
| 20 | | P2_3 | | TRDIOD0 | | | | |
| 21 | | P2_2 | | TRDIOC0 | | | | |
| 22 | | P2_1 | | TRDIQB0 | | | | |
| 23 | | P2_0 | | TRDIOA0/TRDCLK | | | | |
| 24 | | P1_7 | INT1 | TRAIO | | | | |
| 25 | | P1_6 | | | CLK0 | | | |
| 26 | | P1_5 | (INT1) ⁽¹⁾ | (TRAIO) ⁽¹⁾ | RXD0 | | | |
| 27 | | P1_4 | | | TXD0 | | | |
| 28 | | P8_6 | | | | | | |
| 29 | | P8_5 | | TRFO12 | | | | |
| 30 | | P8_4 | | TRFO11 | | | | |
| 31 | | P8_3 | | TRFO10/TRFI | | | | |
| 32 | | P8_2 | | TRFO02 | | | | |
| 33 | | P8_1 | | TRFO01 | | | | |
| 34 | | P8_0 | | TRFO00 | | | | |
| 35 | | P6_0 | | TREO | | | | |
| 36 | | P4_5 | INT0 | INT0 | | | | |
| 37 | | P6_6 | INT2 | | TXD1 | | | |
| 38 | | P6_7 | INT3 | | RXD1 | | | |
| 39 | | P6_5 | | | (CLK1) ⁽¹⁾ / CLK2 | | | |
| 40 | | P6_4 | | | RXD2 | | | |
| 41 | | P6_3 | | | TXD2 | | | |
| 42 | | P3_1 | | TRBO | | | | |
| 43 | | P3_0 | | TRA0 | | | | |
| 44 | | P3_6 | (INT1) ⁽¹⁾ | | | | | |
| 45 | | P3_2 | (INT2) ⁽¹⁾ | | | | | |

NOTE:

1. Can be assigned to the pin in parentheses by a program.

Table 1.8 Pin Name Information by Pin Number (2)

| Pin Number | Control Pin | Port | I/O Pin Functions for of Peripheral Modules | | | | | A/D Converter, D/A Converter |
|------------|-------------|------|---|-------|------------------|------|----------------------|------------------------------|
| | | | Interrupt | Timer | Serial Interface | SSU | I ² C bus | |
| 46 | | P1_3 | $\overline{KI3}$ | | | | | AN11 |
| 47 | | P1_2 | $\overline{KI2}$ | | | | | AN10 |
| 48 | | P1_1 | $\overline{KI1}$ | | | | | AN9 |
| 49 | | P1_0 | $\overline{KI0}$ | | | | | AN8 |
| 50 | | P0_0 | | | | | | AN7 |
| 51 | | P0_1 | | | | | | AN6 |
| 52 | | P0_2 | | | | | | AN5 |
| 53 | | P0_3 | | | | | | AN4 |
| 54 | | P0_4 | | | | | | AN3 |
| 55 | | P6_2 | | | | | | |
| 56 | | P6_1 | | | | | | |
| 57 | | P0_5 | | | CLK1 | | | AN2 |
| 58 | | P0_6 | | | | | | AN1/DA0 |
| 59 | VSS/AVSS | | | | | | | |
| 60 | | P0_7 | | | | | | AN0/DA1 |
| 61 | VREF | | | | | | | |
| 62 | VCC/AVCC | | | | | | | |
| 63 | | P3_7 | | | | SSO | | |
| 64 | | P3_5 | | | | SSCK | SCL | |

1.5 Pin Functions

Tables 1.9 and 1.10 list Pin Functions.

Table 1.9 Pin Functions (1)

| Item | Pin Name | I/O Type | Description |
|---|--|----------|---|
| Power supply input | VCC, VSS | – | Apply 2.2 V to 5.5 V to the VCC pin. Apply 0 V to the VSS pin. |
| Analog power supply input | AVCC, AVSS | – | Power supply for the A/D converter. Connect a capacitor between AVCC and AVSS. |
| Reset input | $\overline{\text{RESET}}$ | I | Input “L” on this pin resets the MCU. |
| MODE | MODE | I | Connect this pin to VCC via a resistor. |
| XIN clock input | XIN | I | These pins are provided for XIN clock generation circuit I/O. Connect a ceramic resonator or a crystal oscillator between the XIN and XOUT pins ⁽¹⁾ . To use an external clock, input it to the XIN pin and leave the XOUT pin open. |
| XIN clock output | XOUT | O | |
| XCIN clock input | XCIN | I | These pins are provided for XCIN clock generation circuit I/O. Connect a crystal oscillator between the XCIN and XCOU pins ⁽¹⁾ . To use an external clock, input it to the XCIN pin and leave the XCOU pin open. |
| XCIN clock output | XCOU | O | |
| $\overline{\text{INT}}$ interrupt input | $\overline{\text{INT0}}$ to $\overline{\text{INT3}}$ | I | $\overline{\text{INT}}$ interrupt input pins. $\overline{\text{INT0}}$ is timer RD input pin. $\overline{\text{INT1}}$ is timer RA input pin. |
| Key input interrupt | $\overline{\text{KI0}}$ to $\overline{\text{KI3}}$ | I | Key input interrupt input pins |
| Timer RA | TRAIO | I/O | Timer RA I/O pin |
| | TRAO | O | Timer RA output pin |
| Timer RB | TRBO | O | Timer RB output pin |
| Timer RC | TRCLK | I | External clock input pin |
| | TRCTR | I | External trigger input pin |
| | TRCIOA, TRCIOB, TRCIO, TRCIOD | I/O | Timer RC I/O pins |
| Timer RD | TRDIOA0, TRDIOA1, TRDIOB0, TRDIOB1, TRDIOC0, TRDIOC1, TRDIOD0, TRDIOD1 | I/O | Timer RD I/O pins |
| | TRDCLK | I | External clock input pin |
| Timer RE | TREO | O | Divided clock output pin |
| Timer RF | TRFI | I | Timer RF input pin |
| | TRFO00 to TRFO02, TRFO10 to TRFO12 | O | Timer RF output pins |
| Serial interface | CLK0, CLK1, CLK2 | I/O | Transfer clock I/O pins |
| | RXD0, RXD1, RXD2 | I | Serial data input pins |
| | TXD0, TXD1, TXD2 | O | Serial data output pins |
| I ² C bus | SCL | I/O | Clock I/O pin |
| | SDA | I/O | Data I/O pin |
| SSU | SSI | I/O | Data I/O pin |
| | $\overline{\text{SCS}}$ | I/O | Chip-select signal I/O pin |
| | SSCK | I/O | Clock I/O pin |
| | SSO | I/O | Data I/O pin |
| Reference voltage input | VREF | I | Reference voltage input pin to A/D converter and D/A converter |

I: Input O: Output I/O: Input and output

NOTE:

1. Refer to the oscillator manufacturer for oscillation characteristics.

Table 1.10 Pin Functions (2)

| Item | Pin Name | I/O Type | Description |
|---------------|---|----------|---|
| A/D converter | AN0 to AN11 | I | Analog input pins to A/D converter |
| D/A converter | DA0 to DA1 | O | D/A converter output pins |
| I/O port | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_3 to P4_5, P5_0 to P5_4, P6_0 to P6_7, P8_0 to P8_6 | I/O | CMOS I/O ports. Each port has an I/O select direction register, allowing each pin in the port to be directed for input or output individually. Any port set to input can be set to use a pull-up resistor or not by a program. P2_0 to P2_7 also function as LED drive ports. |
| Input port | P4_6, P4_7 | I | Input-only ports |

I: Input O: Output I/O: Input and output

2. Central Processing Unit (CPU)

Figure 2.1 shows the CPU Registers. The CPU contains 13 registers. R0, R1, R2, R3, A0, A1, and FB configure a register bank. There are two sets of register bank.

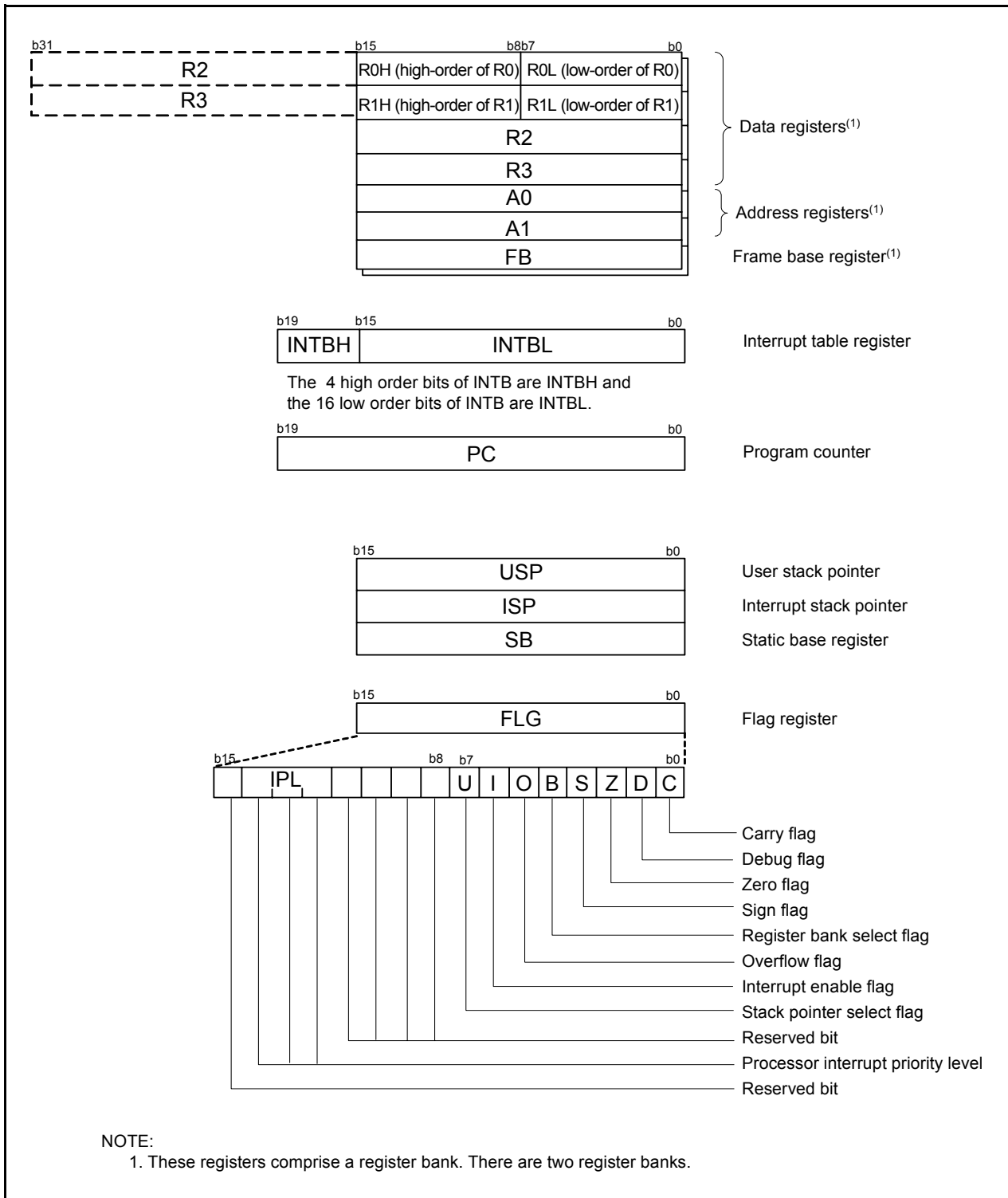


Figure 2.1 CPU Registers

2.1 Data Registers (R0, R1, R2, and R3)

R0 is a 16-bit register for transfer, arithmetic, and logic operations. The same applies to R1 to R3. R0 can be split into high-order bits (R0H) and low-order bits (R0L) to be used separately as 8-bit data registers. R1H and R1L are analogous to R0H and R0L. R2 can be combined with R0 and used as a 32-bit data register (R2R0). R3R1 is analogous to R2R0.

2.2 Address Registers (A0 and A1)

A0 is a 16-bit register for address register indirect addressing and address register relative addressing. It is also used for transfer, arithmetic, and logic operations. A1 is analogous to A0. A1 can be combined with A0 and as a 32-bit address register (A1A0).

2.3 Frame Base Register (FB)

FB is a 16-bit register for FB relative addressing.

2.4 Interrupt Table Register (INTB)

INTB is a 20-bit register that indicates the start address of an interrupt vector table.

2.5 Program Counter (PC)

PC is 20 bits wide and indicates the address of the next instruction to be executed.

2.6 User Stack Pointer (USP) and Interrupt Stack Pointer (ISP)

The stack pointers (SP), USP, and ISP, are each 16 bits wide. The U flag of FLG is used to switch between USP and ISP.

2.7 Static Base Register (SB)

SB is a 16-bit register for SB relative addressing.

2.8 Flag Register (FLG)

FLG is an 11-bit register indicating the CPU state.

2.8.1 Carry Flag (C)

The C flag retains carry, borrow, or shift-out bits that have been generated by the arithmetic and logic unit.

2.8.2 Debug Flag (D)

The D flag is for debugging only. Set it to 0.

2.8.3 Zero Flag (Z)

The Z flag is set to 1 when an arithmetic operation results in 0; otherwise to 0.

2.8.4 Sign Flag (S)

The S flag is set to 1 when an arithmetic operation results in a negative value; otherwise to 0.

2.8.5 Register Bank Select Flag (B)

Register bank 0 is selected when the B flag is 0. Register bank 1 is selected when this flag is set to 1.

2.8.6 Overflow Flag (O)

The O flag is set to 1 when an operation results in an overflow; otherwise to 0.

2.8.7 Interrupt Enable Flag (I)

The I flag enables maskable interrupts.

Interrupt are disabled when the I flag is set to 0, and are enabled when the I flag is set to 1. The I flag is set to 0 when an interrupt request is acknowledged.

2.8.8 Stack Pointer Select Flag (U)

ISP is selected when the U flag is set to 0; USP is selected when the U flag is set to 1.

The U flag is set to 0 when a hardware interrupt request is acknowledged or the INT instruction of software interrupt numbers 0 to 31 is executed.

2.8.9 Processor Interrupt Priority Level (IPL)

IPL is 3 bits wide and assigns processor interrupt priority levels from level 0 to level 7.

If a requested interrupt has higher priority than IPL, the interrupt is enabled.

2.8.10 Reserved Bit

If necessary, set to 0. When read, the content is undefined.

3. Memory

3.1 R8C/2A Group

Figure 3.1 is a Memory Map of R8C/2A Group. The R8C/2A group has 1 Mbyte of address space from addresses 00000h to FFFFFh.

The internal ROM is allocated lower addresses, beginning with address 0FFFFh. For example, a 48-Kbyte internal ROM area is allocated addresses 04000h to 0FFFFh.

The fixed interrupt vector table is allocated addresses 0FFDCh to 0FFFFh. They store the starting address of each interrupt routine.

The internal RAM is allocated higher addresses, beginning with address 00400h. For example, a 2.5-Kbyte internal RAM area is allocated addresses 00400h to 00DFFh. The internal RAM is used not only for storing data but also for calling subroutines and as stacks when interrupt requests are acknowledged.

Special function registers (SFRs) are allocated addresses 00000h to 002FFh. The peripheral function control registers are allocated here. All addresses within the SFR, which have nothing allocated are reserved for future use and cannot be accessed by users.

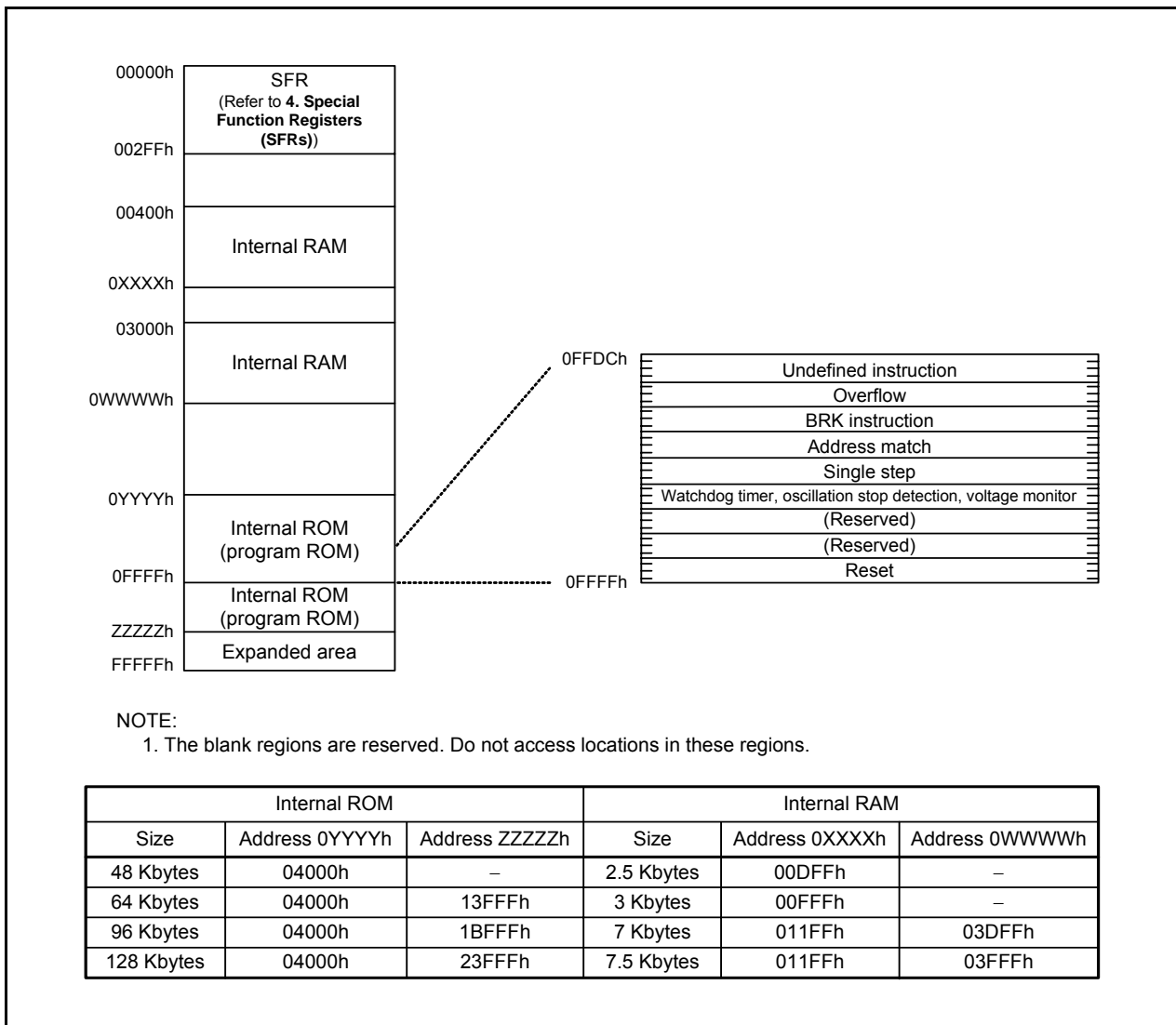


Figure 3.1 Memory Map of R8C/2A Group

3.2 R8C/2B Group

Figure 3.2 is a Memory Map of R8C/2B Group. The R8C/2B group has 1 Mbyte of address space from addresses 00000h to FFFFFh.

The internal ROM (program ROM) is allocated lower addresses, beginning with address 0FFFFh. For example, a 48-Kbyte internal ROM area is allocated addresses 04000h to 0FFFFh.

The fixed interrupt vector table is allocated addresses 0FFDCh to 0FFFFh. They store the starting address of each interrupt routine.

The internal ROM (data flash) is allocated addresses 02400h to 02BFFh.

The internal RAM area is allocated higher addresses, beginning with address 00400h. For example, a 2.5-Kbyte internal RAM is allocated addresses 00400h to 00DFFh. The internal RAM is used not only for storing data but also for calling subroutines and as stacks when interrupt requests are acknowledged.

Special function registers (SFRs) are allocated addresses 00000h to 002FFh. The peripheral function control registers are allocated here. All addresses within the SFR, which have nothing allocated are reserved for future use and cannot be accessed by users.

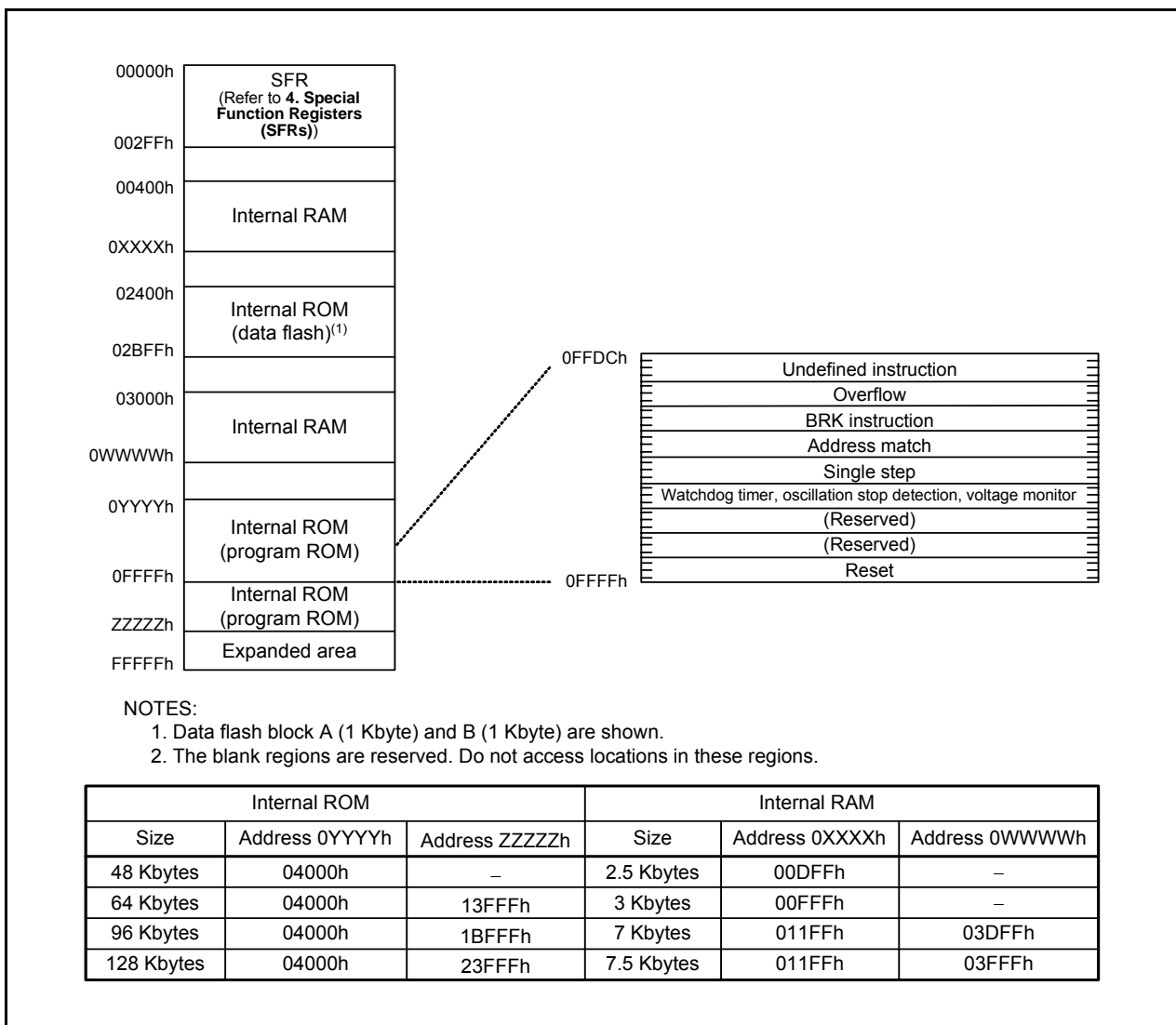


Figure 3.2 Memory Map of R8C/2B Group

4. Special Function Registers (SFRs)

An SFR (special function register) is a control register for a peripheral function. Tables 4.1 to 4.12 list the special function registers.

Table 4.1 SFR Information (1)(1)

| Address | Register | Symbol | After reset |
|---------|---|--------|--|
| 0000h | | | |
| 0001h | | | |
| 0002h | | | |
| 0003h | | | |
| 0004h | Processor Mode Register 0 | PM0 | 00h |
| 0005h | Processor Mode Register 1 | PM1 | 00h |
| 0006h | System Clock Control Register 0 | CM0 | 01101000b |
| 0007h | System Clock Control Register 1 | CM1 | 00100000b |
| 0008h | Module Operation Enable Register | MSTCR | 00h |
| 0009h | | | |
| 000Ah | Protect Register | PRCR | 00h |
| 000Bh | | | |
| 000Ch | Oscillation Stop Detection Register | OCD | 00000100b |
| 000Dh | Watchdog Timer Reset Register | WDTR | XXh |
| 000Eh | Watchdog Timer Start Register | WDTS | XXh |
| 000Fh | Watchdog Timer Control Register | WDC | 00X11111b |
| 0010h | Address Match Interrupt Register 0 | RMAD0 | 00h |
| 0011h | | | 00h |
| 0012h | | | 00h |
| 0013h | Address Match Interrupt Enable Register | AIER | 00h |
| 0014h | Address Match Interrupt Register 1 | RMAD1 | 00h |
| 0015h | | | 00h |
| 0016h | | | 00h |
| 0017h | | | |
| 0018h | | | |
| 0019h | | | |
| 001Ah | | | |
| 001Bh | | | |
| 001Ch | Count Source Protection Mode Register | CSPR | 00h 10000000b ⁽⁶⁾ |
| 001Dh | | | |
| 001Eh | | | |
| 001Fh | | | |
| 0020h | | | |
| 0021h | | | |
| 0022h | | | |
| 0023h | High-Speed On-Chip Oscillator Control Register 0 | FRA0 | 00h |
| 0024h | High-Speed On-Chip Oscillator Control Register 1 | FRA1 | When shipping |
| 0025h | High-Speed On-Chip Oscillator Control Register 2 | FRA2 | 00h |
| 0026h | | | |
| 0027h | | | |
| 0028h | Clock Prescaler Reset Flag | CPSRF | 00h |
| 0029h | | | |
| 002Ah | | | |
| 002Bh | High-Speed On-Chip Oscillator Control Register 6 | FRA6 | When Shipping |
| 002Ch | High-Speed On-Chip Oscillator Control Register 7 | FRA7 | When Shipping |
| 0030h | | | |
| 0031h | Voltage Detection Register 1 ⁽²⁾ | VCA1 | 00001000b |
| 0032h | Voltage Detection Register 2 ⁽²⁾ | VCA2 | 00h ⁽³⁾ 00100000b ⁽⁴⁾ |
| 0033h | | | |
| 0034h | | | |
| 0035h | | | |
| 0036h | Voltage Monitor 1 Circuit Control Register ⁽⁵⁾ | VW1C | 00001000b |
| 0037h | Voltage Monitor 2 Circuit Control Register ⁽⁵⁾ | VW2C | 00h |
| 0038h | Voltage Monitor 0 Circuit Control Register ⁽²⁾ | VW0C | 0000X000b ⁽³⁾ 0100X001b ⁽⁴⁾ |
| 0039h | | | |
| 003Ah | | | |
| 003Eh | | | |
| 003Fh | | | |

X: Undefined

NOTES:

1. The blank regions are reserved. Do not access locations in these regions.
2. Software reset, watchdog timer reset, voltage monitor 1 reset, or voltage monitor 2 reset do not affect this register.
3. The LVD0ON bit in the OFS register is set to 1 and hardware reset.
4. Power-on reset, voltage monitor 0 reset, or the LVD0ON bit in the OFS register is set to 0 and hardware reset.
5. Software reset, watchdog timer reset, voltage monitor 1 reset, or voltage monitor 2 reset do not affect b2 and b3.
6. The CSPROINI bit in the OFS register is set to 0.

Table 4.2 SFR Information (2)(1)

| Address | Register | Symbol | After reset |
|---------|---|---------------|-------------|
| 0040h | | | |
| 0041h | | | |
| 0042h | | | |
| 0043h | | | |
| 0044h | | | |
| 0045h | | | |
| 0046h | | | |
| 0047h | Timer RC Interrupt Control Register | TRCIC | XXXXX000b |
| 0048h | Timer RD0 Interrupt Control Register | TRD0IC | XXXXX000b |
| 0049h | Timer RD1 Interrupt Control Register | TRD1IC | XXXXX000b |
| 004Ah | Timer RE Interrupt Control Register | TREIC | XXXXX000b |
| 004Bh | UART2 Transmit Interrupt Control Register | S2TIC | XXXXX000b |
| 004Ch | UART2 Receive Interrupt Control Register | S2RIC | XXXXX000b |
| 004Dh | Key Input Interrupt Control Register | KUPIC | XXXXX000b |
| 004Eh | | | |
| 004Fh | SSU/IIC Interrupt Control Register ⁽²⁾ | SSUIC / IICIC | XXXXX000b |
| 0050h | Compare 1 Interrupt Control Register | CMP1IC | XXXXX000b |
| 0051h | UART0 Transmit Interrupt Control Register | S0TIC | XXXXX000b |
| 0052h | UART0 Receive Interrupt Control Register | S0RIC | XXXXX000b |
| 0053h | UART1 Transmit Interrupt Control Register | S1TIC | XXXXX000b |
| 0054h | UART1 Receive Interrupt Control Register | S1RIC | XXXXX000b |
| 0055h | INT2 Interrupt Control Register | INT2IC | XX00X000b |
| 0056h | Timer RA Interrupt Control Register | TRAIC | XXXXX000b |
| 0057h | | | |
| 0058h | Timer RB Interrupt Control Register | TRBIC | XXXXX000b |
| 0059h | INT1 Interrupt Control Register | INT1IC | XX00X000b |
| 005Ah | INT3 Interrupt Control Register | INT3IC | XX00X000b |
| 005Bh | Timer RF Interrupt Control Register | TRFIC | XXXXX000b |
| 005Ch | Compare 0 Interrupt Control Register | CMP0IC | XXXXX000b |
| 005Dh | INT0 Interrupt Control Register | INT0IC | XX00X000b |
| 005Eh | A/D Conversion Interrupt Control Register | ADIC | XXXXX000b |
| 005Fh | Capture Interrupt Control Register | CAPIC | XXXXX000b |
| 0060h | | | |
| 0061h | | | |
| 0062h | | | |
| 0063h | | | |
| 0064h | | | |
| 0065h | | | |
| 0066h | | | |
| 0067h | | | |
| 0068h | | | |
| 0069h | | | |
| 006Ah | | | |
| 006Bh | | | |
| 006Ch | | | |
| 006Dh | | | |
| 006Eh | | | |
| 006Fh | | | |
| 0070h | | | |
| 0071h | | | |
| 0072h | | | |
| 0073h | | | |
| 0074h | | | |
| 0075h | | | |
| 0076h | | | |
| 0077h | | | |
| 0078h | | | |
| 0079h | | | |
| 007Ah | | | |
| 007Bh | | | |
| 007Ch | | | |
| 007Dh | | | |
| 007Eh | | | |
| 007Fh | | | |

X: Undefined

NOTES:

1. The blank regions are reserved. Do not access locations in these regions.
2. Selected by the IICSEL bit in the PMR register.

Table 4.3 SFR Information (3)⁽¹⁾

| Address | Register | Symbol | After reset |
|---------|---|---------------|-----------------|
| 0080h | | | |
| 0081h | | | |
| 0082h | | | |
| 0083h | | | |
| 0084h | | | |
| 0085h | | | |
| 0086h | | | |
| 0087h | | | |
| 0088h | | | |
| 0089h | | | |
| 008Ah | | | |
| 008Bh | | | |
| 008Ch | | | |
| 008Dh | | | |
| 008Eh | | | |
| 008Fh | | | |
| 0090h | | | |
| 0091h | | | |
| 0092h | | | |
| 0093h | | | |
| 0094h | | | |
| 0095h | | | |
| 0096h | | | |
| 0097h | | | |
| 0098h | | | |
| 0099h | | | |
| 009Ah | | | |
| 009Bh | | | |
| 009Ch | | | |
| 009Dh | | | |
| 009Eh | | | |
| 009Fh | | | |
| 00A0h | UART0 Transmit/Receive Mode Register | U0MR | 00h |
| 00A1h | UART0 Bit Rate Register | U0BRG | XXh |
| 00A2h | UART0 Transmit Buffer Register | U0TB | XXh |
| 00A3h | | | XXh |
| 00A4h | UART0 Transmit/Receive Control Register 0 | U0C0 | 00001000b |
| 00A5h | UART0 Transmit/Receive Control Register 1 | U0C1 | 00000010b |
| 00A6h | UART0 Receive Buffer Register | U0RB | XXh |
| 00A7h | | | XXh |
| 00A8h | UART1 Transmit/Receive Mode Register | U1MR | 00h |
| 00A9h | UART1 Bit Rate Register | U1BRG | XXh |
| 00AAh | UART1 Transmit Buffer Register | U1TB | XXh |
| 00ABh | | | XXh |
| 00ACh | UART1 Transmit/Receive Control Register 0 | U1C0 | 00001000b |
| 00ADh | UART1 Transmit/Receive Control Register 1 | U1C1 | 00000010b |
| 00AEh | UART1 Receive Buffer Register | U1RB | XXh |
| 00AFh | | | XXh |
| 00B0h | | | |
| 00B1h | | | |
| 00B2h | | | |
| 00B3h | | | |
| 00B4h | | | |
| 00B5h | | | |
| 00B6h | | | |
| 00B7h | | | |
| 00B8h | SS Control Register H / IIC bus Control Register 1 ⁽²⁾ | SSCRH / ICCR1 | 00h |
| 00B9h | SS Control Register L / IIC bus Control Register 2 ⁽²⁾ | SSCRL / ICCR2 | 01111101b |
| 00BAh | SS Mode Register / IIC bus Mode Register ⁽²⁾ | SSMR / ICMR | 00011000b |
| 00BBh | SS Enable Register / IIC bus Interrupt Enable Register ⁽²⁾ | SSER / ICIER | 00h |
| 00BCh | SS Status Register / IIC bus Status Register ⁽²⁾ | SSSR / ICSR | 00h / 0000X000b |
| 00BDh | SS Mode Register 2 / Slave Address Register ⁽²⁾ | SSMR2 / SAR | 00h |
| 00BEh | SS Transmit Data Register / IIC bus Transmit Data Register ⁽²⁾ | SSTDR / ICDRT | FFh |
| 00BFh | SS Receive Data Register / IIC bus Receive Data Register ⁽²⁾ | SSRDR / ICDRR | FFh |

X: Undefined

NOTES:

1. The blank regions are reserved. Do not access locations in these regions.
2. Selected by the IICSEL bit in the PMR register.

Table 4.4 SFR Information (4)⁽¹⁾

| Address | Register | Symbol | After reset |
|---------|---|--------|-------------|
| 00C0h | | | |
| 00C1h | | | |
| 00C2h | | | |
| 00C3h | | | |
| 00C4h | | | |
| 00C5h | | | |
| 00C6h | | | |
| 00C7h | | | |
| 00C8h | | | |
| 00C9h | | | |
| 00CAh | | | |
| 00CBh | | | |
| 00CCh | | | |
| 00CDh | | | |
| 00CEh | | | |
| 00CFh | | | |
| 00D0h | | | |
| 00D1h | | | |
| 00D2h | | | |
| 00D3h | | | |
| 00D4h | | | |
| 00D5h | | | |
| 00D6h | | | |
| 00D7h | | | |
| 00D8h | D/A Register 0 | DA0 | 00h |
| 00D9h | | | |
| 00DAh | D/A Register 1 | DA1 | 00h |
| 00DBh | | | |
| 00DCh | D/A Control Register | DACON | 00h |
| 00DDh | | | |
| 00DEh | | | |
| 00DFh | | | |
| 00E0h | Port P0 Register | P0 | XXh |
| 00E1h | Port P1 Register | P1 | XXh |
| 00E2h | Port P0 Direction Register | PD0 | 00h |
| 00E3h | Port P1 Direction Register | PD1 | 00h |
| 00E4h | Port P2 Register | P2 | XXh |
| 00E5h | Port P3 Register | P3 | XXh |
| 00E6h | Port P2 Direction Register | PD2 | 00h |
| 00E7h | Port P3 Direction Register | PD3 | 00h |
| 00E8h | Port P4 Register | P4 | XXh |
| 00E9h | Port P5 Register | P5 | XXh |
| 00EAh | Port P4 Direction Register | PD4 | 00h |
| 00EBh | Port P5 Direction Register | PD5 | 00h |
| 00ECh | Port P6 Register | P6 | XXh |
| 00EDh | | | |
| 00EEh | Port P6 Direction Register | PD6 | 00h |
| 00EFh | | | |
| 00F0h | | | |
| 00F1h | | | |
| 00F2h | | | |
| 00F3h | | | |
| 00F4h | Port P2 Drive Capacity Control Register | P2DRR | 00h |
| 00F5h | UART1 Function Select Register | U1SR | 000000XXb |
| 00F6h | | | |
| 00F7h | | | |
| 00F8h | Port Mode Register | PMR | 00h |
| 00F9h | External Input Enable Register | INTEN | 00h |
| 00FAh | INT Input Filter Select Register | INTF | 00h |
| 00FBh | Key Input Enable Register | KIEN | 00h |
| 00FCh | Pull-Up Control Register 0 | PUR0 | 00h |
| 00FDh | Pull-Up Control Register 1 | PUR1 | XX000000b |
| 00FEh | | | |
| 00FFh | | | |

X: Undefined

NOTE:

1. The blank regions are reserved. Do not access locations in these regions.

Table 4.5 SFR Information (5)(1)

| Address | Register | Symbol | After reset |
|---------|---|---------|-------------|
| 0100h | Timer RA Control Register | TRACR | 00h |
| 0101h | Timer RA I/O Control Register | TRAIOC | 00h |
| 0102h | Timer RA Mode Register | TRAMR | 00h |
| 0103h | Timer RA Prescaler Register | TRAPRE | FFh |
| 0104h | Timer RA Register | TRA | FFh |
| 0105h | LIN Control Register 2 | LINCR2 | 00h |
| 0106h | LIN Control Register | LINCR | 00h |
| 0107h | LIN Status Register | LINST | 00h |
| 0108h | Timer RB Control Register | TRBCR | 00h |
| 0109h | Timer RB One-Shot Control Register | TRBOCR | 00h |
| 010Ah | Timer RB I/O Control Register | TRBIOC | 00h |
| 010Bh | Timer RB Mode Register | TRBMR | 00h |
| 010Ch | Timer RB Prescaler Register | TRBPRE | FFh |
| 010Dh | Timer RB Secondary Register | TRBSC | FFh |
| 010Eh | Timer RB Primary Register | TRBPR | FFh |
| 010Fh | | | |
| 0110h | | | |
| 0111h | | | |
| 0112h | | | |
| 0113h | | | |
| 0114h | | | |
| 0115h | | | |
| 0116h | | | |
| 0117h | | | |
| 0118h | Timer RE Second Data Register / Counter Data Register | TRESEC | 00h |
| 0119h | Timer RE Minute Data Register / Compare Data Register | TREMIN | 00h |
| 011Ah | Timer RE Hour Data Register | TREHR | 00h |
| 011Bh | Timer RE Day of Week Data Register | TREWK | 00h |
| 011Ch | Timer RE Control Register 1 | TRECR1 | 00h |
| 011Dh | Timer RE Control Register 2 | TRECR2 | 00h |
| 011Eh | Timer RE Clock Source Select Register | TRECSR | 00001000b |
| 011Fh | | | |
| 0120h | Timer RC Mode Register | TRCMR | 01001000b |
| 0121h | Timer RC Control Register 1 | TRCCR1 | 00h |
| 0122h | Timer RC Interrupt Enable Register | TRCIER | 01110000b |
| 0123h | Timer RC Status Register | TRCSR | 01110000b |
| 0124h | Timer RC I/O Control Register 0 | TRCIOR0 | 10001000b |
| 0125h | Timer RC I/O Control Register 1 | TRCIOR1 | 10001000b |
| 0126h | Timer RC Counter | TRC | 00h |
| 0127h | | | 00h |
| 0128h | Timer RC General Register A | TRCGRA | FFh |
| 0129h | | | FFh |
| 012Ah | Timer RC General Register B | TRCGRB | FFh |
| 012Bh | | | FFh |
| 012Ch | Timer RC General Register C | TRCGRC | FFh |
| 012Dh | | | FFh |
| 012Eh | Timer RC General Register D | TRCGRD | FFh |
| 012Fh | | | FFh |
| 0130h | Timer RC Control Register 2 | TRCCR2 | 00011111b |
| 0131h | Timer RC Digital Filter Function Select Register | TRCDF | 00h |
| 0132h | Timer RC Output Master Enable Register | TRCOER | 01111111b |
| 0133h | | | |
| 0134h | | | |
| 0135h | | | |
| 0136h | | | |
| 0137h | Timer RD Start Register | TRDSTR | 1111100b |
| 0138h | Timer RD Mode Register | TRDMR | 00001110b |
| 0139h | Timer RD PWM Mode Register | TRDPMR | 10001000b |
| 013Ah | Timer RD Function Control Register | TRDFCR | 10000000b |
| 013Bh | Timer RD Output Master Enable Register 1 | TRDOER1 | FFh |
| 013Ch | Timer RD Output Master Enable Register 2 | TRDOER2 | 01111111b |
| 013Dh | Timer RD Output Control Register | TRDOCR | 00h |
| 013Eh | Timer RD Digital Filter Function Select Register 0 | TRDDF0 | 00h |
| 013Fh | Timer RD Digital Filter Function Select Register 1 | TRDDF1 | 00h |

NOTE:

1. The blank regions are reserved. Do not access locations in these regions

Table 4.6 SFR Information (6)(1)

| Address | Register | Symbol | After reset |
|---------|---|----------|-------------|
| 0140h | Timer RD Control Register 0 | TRDCR0 | 00h |
| 0141h | Timer RD I/O Control Register A0 | TRDIORA0 | 10001000b |
| 0142h | Timer RD I/O Control Register C0 | TRDIORC0 | 10001000b |
| 0143h | Timer RD Status Register 0 | TRDSR0 | 11000000b |
| 0144h | Timer RD Interrupt Enable Register 0 | TRDIER0 | 11100000b |
| 0145h | Timer RD PWM Mode Output Level Control Register 0 | TRDPOCR0 | 11111000b |
| 0146h | Timer RD Counter 0 | TRD0 | 00h |
| 0147h | | | 00h |
| 0148h | Timer RD General Register A0 | TRDGRA0 | FFh |
| 0149h | | | FFh |
| 014Ah | Timer RD General Register B0 | TRDGRB0 | FFh |
| 014Bh | | | FFh |
| 014Ch | Timer RD General Register C0 | TRDGRC0 | FFh |
| 014Dh | | | FFh |
| 014Eh | Timer RD General Register D0 | TRDGRD0 | FFh |
| 014Fh | | | FFh |
| 0150h | Timer RD Control Register 1 | TRDCR1 | 00h |
| 0151h | Timer RD I/O Control Register A1 | TRDIORA1 | 10001000b |
| 0152h | Timer RD I/O Control Register C1 | TRDIORC1 | 10001000b |
| 0153h | Timer RD Status Register 1 | TRDSR1 | 11000000b |
| 0154h | Timer RD Interrupt Enable Register 1 | TRDIER1 | 11100000b |
| 0155h | Timer RD PWM Mode Output Level Control Register 1 | TRDPOCR1 | 11111000b |
| 0156h | Timer RD Counter 1 | TRD1 | 00h |
| 0157h | | | 00h |
| 0158h | Timer RD General Register A1 | TRDGRA1 | FFh |
| 0159h | | | FFh |
| 015Ah | Timer RD General Register B1 | TRDGRB1 | FFh |
| 015Bh | | | FFh |
| 015Ch | Timer RD General Register C1 | TRDGRC1 | FFh |
| 015Dh | | | FFh |
| 015Eh | Timer RD General Register D1 | TRDGRD1 | FFh |
| 015Fh | | | FFh |
| 0160h | UART2 Transmit/Receive Mode Register | U2MR | 00h |
| 0161h | UART2 Bit Rate Register | U2BRG | XXh |
| 0162h | UART2 Transmit Buffer Register | U2TB | XXh |
| 0163h | | | XXh |
| 0164h | UART2 Transmit/Receive Control Register 0 | U2C0 | 00001000b |
| 0165h | UART2 Transmit/Receive Control Register 1 | U2C1 | 0000010b |
| 0166h | UART2 Receive Buffer Register | U2RB | XXh |
| 0167h | | | XXh |
| 0168h | | | |
| 0169h | | | |
| 016Ah | | | |
| 016Bh | | | |
| 016Ch | | | |
| 016Dh | | | |
| 016Eh | | | |
| 016Fh | | | |
| 0170h | | | |
| 0171h | | | |
| 0172h | | | |
| 0173h | | | |
| 0174h | | | |
| 0175h | | | |
| 0176h | | | |
| 0177h | | | |
| 0178h | | | |
| 0179h | | | |
| 017Ah | | | |
| 017Bh | | | |
| 017Ch | | | |
| 017Dh | | | |
| 017Eh | | | |
| 017Fh | | | |

X: Undefined

NOTE:

1. The blank regions are reserved. Do not access locations in these regions.

Table 4.7 SFR Information (7)(1)

| Address | Register | Symbol | After reset |
|---------|---------------------------------|--------|-------------|
| 0180h | | | |
| 0181h | | | |
| 0182h | | | |
| 0183h | | | |
| 0184h | | | |
| 0185h | | | |
| 0186h | | | |
| 0187h | | | |
| 0188h | | | |
| 0189h | | | |
| 018Ah | | | |
| 018Bh | | | |
| 018Ch | | | |
| 018Dh | | | |
| 018Eh | | | |
| 018Fh | | | |
| 0190h | | | |
| 0191h | | | |
| 0192h | | | |
| 0193h | | | |
| 0194h | | | |
| 0195h | | | |
| 0196h | | | |
| 0197h | | | |
| 0198h | | | |
| 0199h | | | |
| 019Ah | | | |
| 019Bh | | | |
| 019Ch | | | |
| 019Dh | | | |
| 019Eh | | | |
| 019Fh | | | |
| 01A0h | | | |
| 01A1h | | | |
| 01A2h | | | |
| 01A3h | | | |
| 01A4h | | | |
| 01A5h | | | |
| 01A6h | | | |
| 01A7h | | | |
| 01A8h | | | |
| 01A9h | | | |
| 01AAh | | | |
| 01ABh | | | |
| 01ACh | | | |
| 01ADh | | | |
| 01AEh | | | |
| 01AFh | | | |
| 01B0h | | | |
| 01B1h | | | |
| 01B2h | | | |
| 01B3h | Flash Memory Control Register 4 | FMR4 | 01000000b |
| 01B4h | | | |
| 01B5h | Flash Memory Control Register 1 | FMR1 | 1000000Xb |
| 01B6h | | | |
| 01B7h | Flash Memory Control Register 0 | FMR0 | 00000001b |
| 01B8h | | | |
| 01B9h | | | |
| 01BAh | | | |
| 01BBh | | | |
| 01BCh | | | |
| 01BDh | | | |
| 01BEh | | | |
| 01BFh | | | |

X: Undefined

NOTE:

1. The blank regions are reserved. Do not access locations in these regions.

Table 4.8 SFR Information (8)(1)

| Address | Register | Symbol | After reset |
|---------|----------|--------|-------------|
| 01C0h | | | |
| 01C1h | | | |
| 01C2h | | | |
| 01C3h | | | |
| 01C4h | | | |
| 01C5h | | | |
| 01C6h | | | |
| 01C7h | | | |
| 01C8h | | | |
| 01C9h | | | |
| 01CAh | | | |
| 01CBh | | | |
| 01CCh | | | |
| 01CDh | | | |
| 01CEh | | | |
| 01CFh | | | |
| 01D0h | | | |
| 01D1h | | | |
| 01D2h | | | |
| 01D3h | | | |
| 01D4h | | | |
| 01D5h | | | |
| 01D6h | | | |
| 01D7h | | | |
| 01D8h | | | |
| 01D9h | | | |
| 01DAh | | | |
| 01DBh | | | |
| 01DCh | | | |
| 01DDh | | | |
| 01DEh | | | |
| 01DFh | | | |
| 01E0h | | | |
| 01E1h | | | |
| 01E2h | | | |
| 01E3h | | | |
| 01E4h | | | |
| 01E5h | | | |
| 01E6h | | | |
| 01E7h | | | |
| 01E8h | | | |
| 01E9h | | | |
| 01EAh | | | |
| 01EBh | | | |
| 01ECh | | | |
| 01EDh | | | |
| 01EEh | | | |
| 01EFh | | | |
| 01F0h | | | |
| 01F1h | | | |
| 01F2h | | | |
| 01F3h | | | |
| 01F4h | | | |
| 01F5h | | | |
| 01F6h | | | |
| 01F7h | | | |
| 01F8h | | | |
| 01F9h | | | |
| 01FAh | | | |
| 01FBh | | | |
| 01FCh | | | |
| 01FDh | | | |
| 01FEh | | | |
| 01FFh | | | |

NOTE:
 1. The blank regions are reserved. Do not access locations in these regions.

Table 4.9 SFR Information (9)⁽¹⁾

| Address | Register | Symbol | After reset |
|---------|----------|--------|-------------|
| 0200h | | | |
| 0201h | | | |
| 0202h | | | |
| 0203h | | | |
| 0204h | | | |
| 0205h | | | |
| 0206h | | | |
| 0207h | | | |
| 0208h | | | |
| 0209h | | | |
| 020Ah | | | |
| 020Bh | | | |
| 020Ch | | | |
| 020Dh | | | |
| 020Eh | | | |
| 020Fh | | | |
| 0210h | | | |
| 0211h | | | |
| 0212h | | | |
| 0213h | | | |
| 0214h | | | |
| 0215h | | | |
| 0216h | | | |
| 0217h | | | |
| 0218h | | | |
| 0219h | | | |
| 021Ah | | | |
| 021Bh | | | |
| 021Ch | | | |
| 021Dh | | | |
| 021Eh | | | |
| 021Fh | | | |
| 0220h | | | |
| 0221h | | | |
| 0222h | | | |
| 0223h | | | |
| 0224h | | | |
| 0225h | | | |
| 0226h | | | |
| 0227h | | | |
| 0228h | | | |
| 0229h | | | |
| 022Ah | | | |
| 022Bh | | | |
| 022Ch | | | |
| 022Dh | | | |
| 022Eh | | | |
| 022Fh | | | |
| 0230h | | | |
| 0231h | | | |
| 0232h | | | |
| 0233h | | | |
| 0234h | | | |
| 0235h | | | |
| 0236h | | | |
| 0237h | | | |
| 0238h | | | |
| 0239h | | | |
| 023Ah | | | |
| 023Bh | | | |
| 023Ch | | | |
| 023Dh | | | |
| 023Eh | | | |
| 023Fh | | | |

NOTE:
 1. The blank regions are reserved. Do not access locations in these regions.

Table 4.10 SFR Information (10)(1)

| Address | Register | Symbol | After reset |
|---------|----------|--------|-------------|
| 0240h | | | |
| 0241h | | | |
| 0242h | | | |
| 0243h | | | |
| 0244h | | | |
| 0245h | | | |
| 0246h | | | |
| 0247h | | | |
| 0248h | | | |
| 0249h | | | |
| 024Ah | | | |
| 024Bh | | | |
| 024Ch | | | |
| 024Dh | | | |
| 024Eh | | | |
| 024Fh | | | |
| 0250h | | | |
| 0251h | | | |
| 0252h | | | |
| 0253h | | | |
| 0254h | | | |
| 0255h | | | |
| 0256h | | | |
| 0257h | | | |
| 0258h | | | |
| 0259h | | | |
| 025Ah | | | |
| 025Bh | | | |
| 025Ch | | | |
| 025Dh | | | |
| 025Eh | | | |
| 025Fh | | | |
| 0260h | | | |
| 0261h | | | |
| 0262h | | | |
| 0263h | | | |
| 0264h | | | |
| 0265h | | | |
| 0266h | | | |
| 0267h | | | |
| 0268h | | | |
| 0269h | | | |
| 026Ah | | | |
| 026Bh | | | |
| 026Ch | | | |
| 026Dh | | | |
| 026Eh | | | |
| 026Fh | | | |
| 0270h | | | |
| 0271h | | | |
| 0272h | | | |
| 0273h | | | |
| 0274h | | | |
| 0275h | | | |
| 0276h | | | |
| 0277h | | | |
| 0278h | | | |
| 0279h | | | |
| 027Ah | | | |
| 027Bh | | | |
| 027Ch | | | |
| 027Dh | | | |
| 027Eh | | | |
| 027Fh | | | |

NOTE:
 1. The blank regions are reserved. Do not access locations in these regions.

Table 4.11 SFR Information (11)(1)

| Address | Register | Symbol | After reset |
|---------|------------------------------|--------|----------------------|
| 0280h | | | |
| 0281h | | | |
| 0282h | | | |
| 0283h | | | |
| 0284h | | | |
| 0285h | | | |
| 0286h | | | |
| 0287h | | | |
| 0288h | | | |
| 0289h | | | |
| 028Ah | | | |
| 028Bh | | | |
| 028Ch | | | |
| 028Dh | | | |
| 028Eh | | | |
| 028Fh | | | |
| 0290h | Timer RF Register | TRF | 00h |
| 0291h | | | 00h |
| 0292h | | | |
| 0293h | | | |
| 0294h | | | |
| 0295h | | | |
| 0296h | | | |
| 0297h | | | |
| 0298h | | | |
| 0299h | | | |
| 029Ah | Timer RF Control Register 0 | TRFCR0 | 00h |
| 029Bh | Timer RF Control Register 1 | TRFCR1 | 00h |
| 029Ch | Capture / Compare 0 Register | TRFM0 | 0000h ⁽²⁾ |
| 029Dh | | | FFFFh ⁽³⁾ |
| 029Eh | Compare 1 Register | TRFM1 | FFh |
| 029Fh | | | FFh |
| 02A0h | | | |
| 02A1h | | | |
| 02A2h | | | |
| 02A3h | | | |
| 02A4h | | | |
| 02A5h | | | |
| 02A6h | | | |
| 02A7h | | | |
| 02A8h | | | |
| 02A9h | | | |
| 02AAh | | | |
| 02ABh | | | |
| 02ACh | | | |
| 02ADh | | | |
| 02AEh | | | |
| 02AFh | | | |
| 02B0h | | | |
| 02B1h | | | |
| 02B2h | | | |
| 02B3h | | | |
| 02B4h | | | |
| 02B5h | | | |
| 02B6h | | | |
| 02B7h | | | |
| 02B8h | | | |
| 02B9h | | | |
| 02BAh | | | |
| 02BBh | | | |
| 02BCh | | | |
| 02BDh | | | |
| 02BEh | | | |
| 02BFh | | | |

NOTES:

1. The blank regions are reserved. Do not access locations in these regions.
2. After input capture mode.
3. After output compare mode.

Table 4.12 SFR Information (12)(1)

| Address | Register | Symbol | After reset |
|---------|----------------------------------|--------|-------------|
| 02C0h | A/D Register 0 | AD0 | XXh |
| 02C1h | | | XXh |
| 02C2h | | | |
| 02C3h | | | |
| 02C4h | | | |
| 02C5h | | | |
| 02C6h | | | |
| 02C7h | | | |
| 02C8h | | | |
| 02C9h | | | |
| 02CAh | | | |
| 02CBh | | | |
| 02CCh | | | |
| 02CDh | | | |
| 02CEh | | | |
| 02CFh | | | |
| 02D0h | | | |
| 02D1h | | | |
| 02D2h | | | |
| 02D3h | | | |
| 02D4h | A/D Control Register 2 | ADCON2 | 00001000b |
| 02D5h | | | |
| 02D6h | A/D Control Register 0 | ADCON0 | 00000011b |
| 02D7h | A/D Control Register 1 | ADCON1 | 00h |
| 02D8h | | | |
| 02D9h | | | |
| 02DAh | | | |
| 02DBh | | | |
| 02DCh | | | |
| 02DDh | | | |
| 02DEh | | | |
| 02DFh | | | |
| 02E0h | | | |
| 02E1h | | | |
| 02E2h | | | |
| 02E3h | | | |
| 02E4h | Port P8 Direction Register | PD8 | 00h |
| 02E5h | | | |
| 02E6h | Port P8 Register | P8 | XXh |
| 02E7h | | | |
| 02E8h | | | |
| 02E9h | | | |
| 02EAh | | | |
| 02EBh | | | |
| 02ECh | | | |
| 02EDh | | | |
| 02EEh | | | |
| 02EFh | | | |
| 02F0h | | | |
| 02F1h | | | |
| 02F2h | | | |
| 02F3h | | | |
| 02F4h | | | |
| 02F5h | | | |
| 02F6h | | | |
| 02F7h | | | |
| 02F8h | | | |
| 02F9h | | | |
| 02FAh | | | |
| 02FBh | | | |
| 02FCh | Pull-Up Control Register 2 | PUR2 | XXX00000b |
| 02FDh | | | |
| 02FEh | | | |
| 02FFh | Timer RF Output Control Register | TRFOUT | 00h |
| FFFFh | Option Function Select Register | OFS | (Note 2) |

X: Undefined

NOTES:

1. The blank regions are reserved. Do not access locations in these regions.
2. The OFS register cannot be changed by a program. Use a flash programmer to write to it.

5. Resets

The following resets are implemented: hardware reset, power-on reset, voltage monitor 0 reset, voltage monitor 1 reset, voltage monitor 2 reset, watchdog timer reset, and software reset.

Table 5.1 lists the Reset Names and Sources.

Table 5.1 Reset Names and Sources

| Reset Name | Source |
|-------------------------|--|
| Hardware reset | Input voltage of $\overline{\text{RESET}}$ pin is held "L" |
| Power-on reset | VCC rises |
| Voltage monitor 0 reset | VCC falls (monitor voltage: Vdet0) |
| Voltage monitor 1 reset | VCC falls (monitor voltage: Vdet1) |
| Voltage monitor 2 reset | VCC falls (monitor voltage: Vdet2) |
| Watchdog timer reset | Underflow of watchdog timer |
| Software reset | Write 1 to PM03 bit in PM0 register |

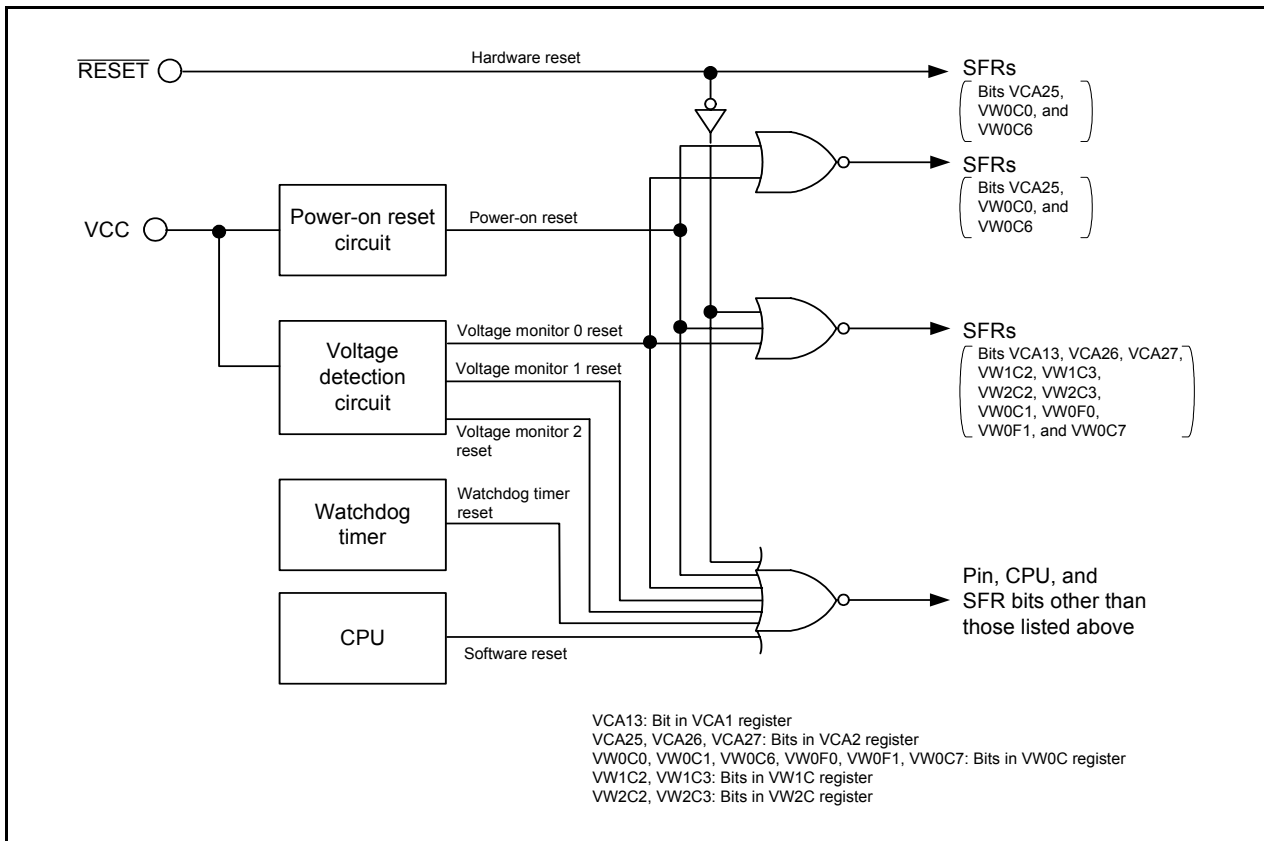


Figure 5.1 Block Diagram of Reset Circuit

Table 5.2 shows the Pin Functions while $\overline{\text{RESET}}$ Pin Level is “L”, Figure 5.2 shows the CPU Register Status after Reset, Figure 5.3 shows the Reset Sequence, and Figure 5.4 shows the OFS Register.

Table 5.2 Pin Functions while $\overline{\text{RESET}}$ Pin Level is “L”

| Pin Name | Pin Functions |
|--------------|---------------|
| P0 to P3 | Input port |
| P4_3 to P4_7 | Input port |
| P5_0 to P5_4 | Input port |
| P6 | Input port |
| P8_0 to P8_6 | Input port |

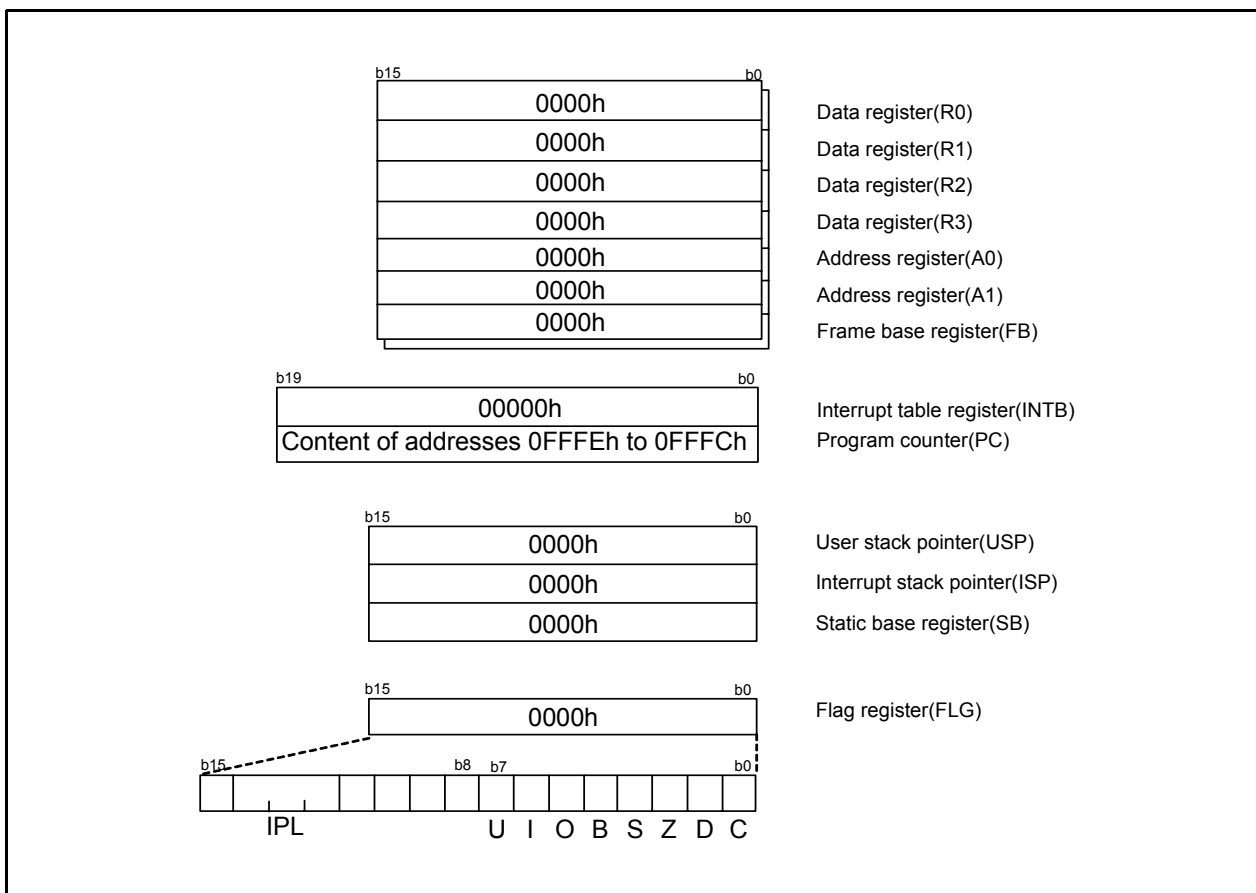


Figure 5.2 CPU Register Status after Reset

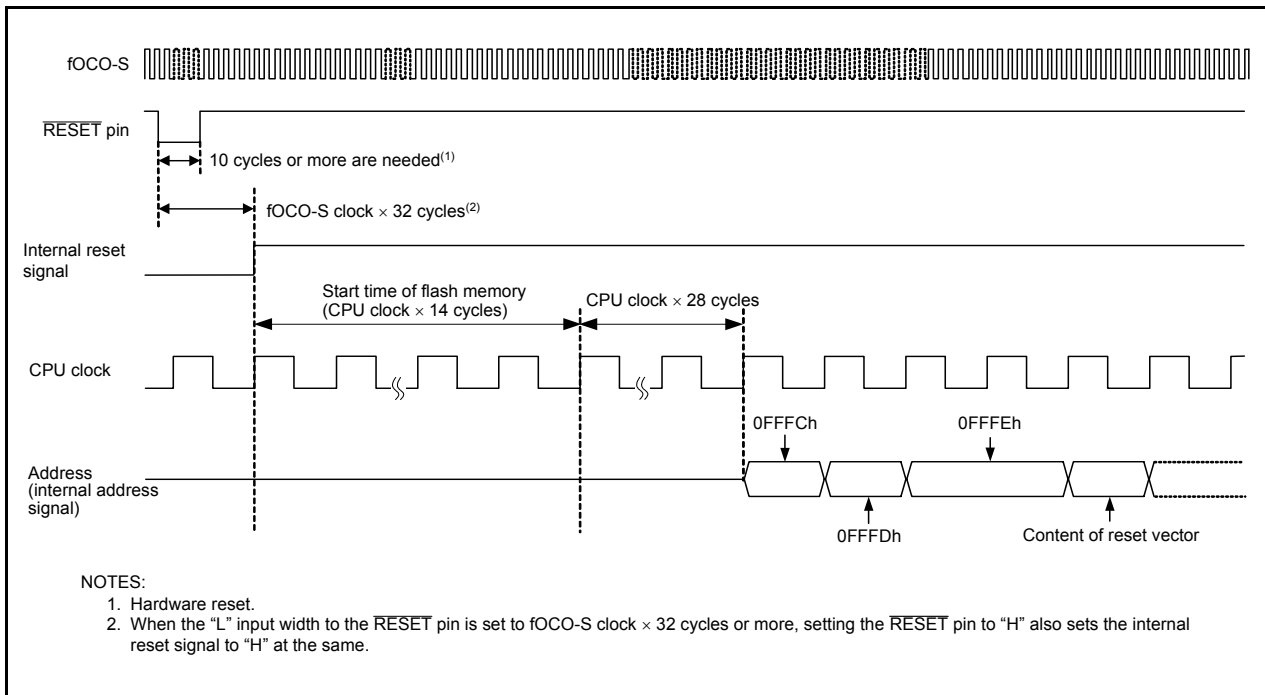


Figure 5.3 Reset Sequence

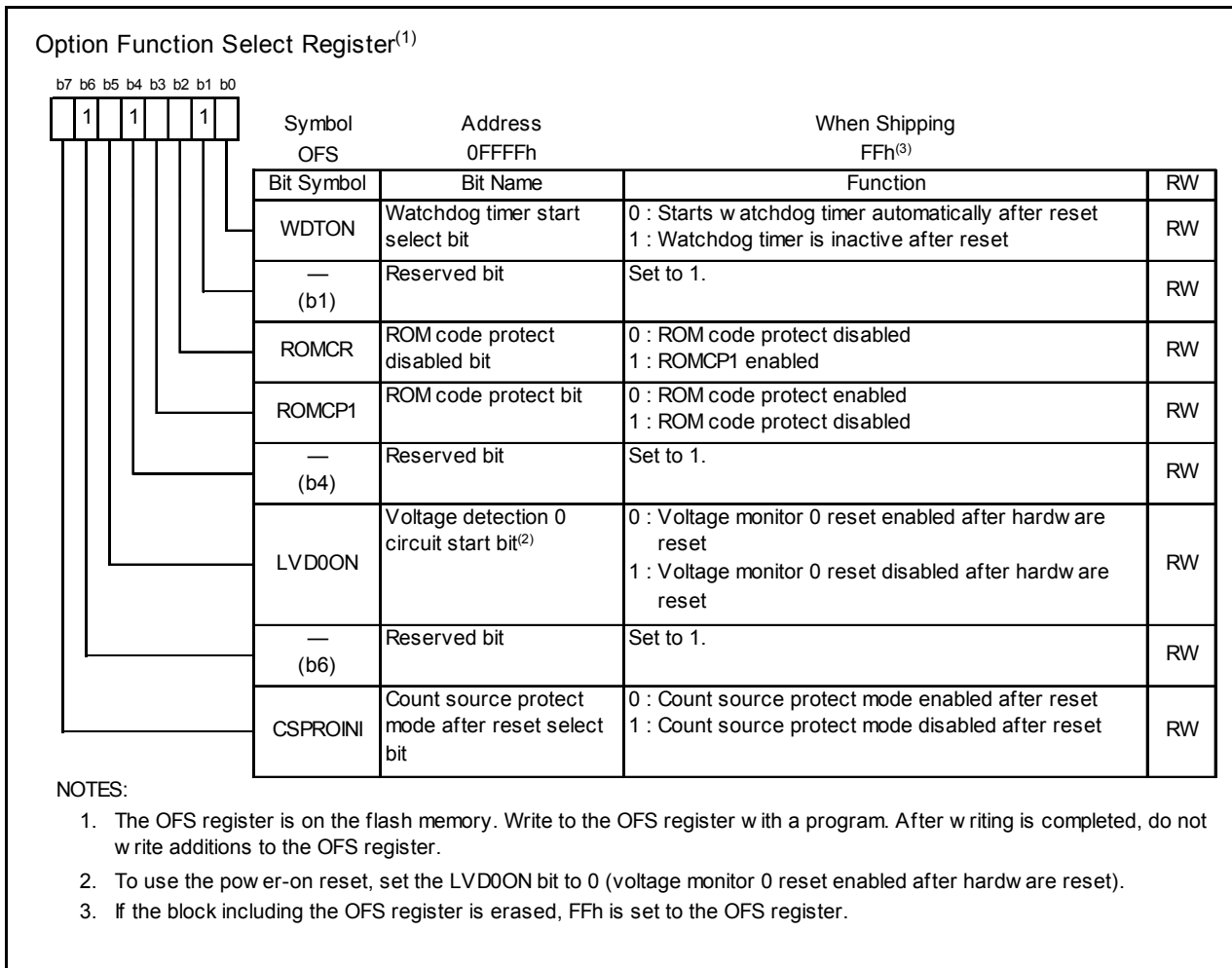


Figure 5.4 OFS Register

5.1 Hardware Reset

A reset is applied using the $\overline{\text{RESET}}$ pin. When an “L” signal is applied to the $\overline{\text{RESET}}$ pin while the supply voltage meets the recommended operating conditions, pins, CPU, and SFRs are all reset (refer to **Table 5.2 Pin Functions while RESET Pin Level is “L”**). When the input level applied to the $\overline{\text{RESET}}$ pin changes from “L” to “H”, a program is executed beginning with the address indicated by the reset vector. After reset, the low-speed on-chip oscillator clock divided by 8 is automatically selected as the CPU clock.

Refer to **4. Special Function Registers (SFRs)** for the state of the SFRs after reset.

The internal RAM is not reset. If the $\overline{\text{RESET}}$ pin is pulled “L” while writing to the internal RAM is in progress, the contents of internal RAM will be undefined.

Figure 5.5 shows an Example of Hardware Reset Circuit and Operation and Figure 5.6 shows an Example of Hardware Reset Circuit (Usage Example of External Supply Voltage Detection Circuit) and Operation.

5.1.1 When Power Supply is Stable

- (1) Apply “L” to the $\overline{\text{RESET}}$ pin.
- (2) Wait for 10 μs or more.
- (3) Apply “H” to the $\overline{\text{RESET}}$ pin.

5.1.2 Power On

- (1) Apply “L” to the $\overline{\text{RESET}}$ pin.
- (2) Let the supply voltage increase until it meets the recommended operating conditions.
- (3) Wait for $t_{d(P-R)}$ or more to allow the internal power supply to stabilize (refer to **21. Electrical Characteristics**).
- (4) Wait for 10 μs or more.
- (5) Apply “H” to the $\overline{\text{RESET}}$ pin.

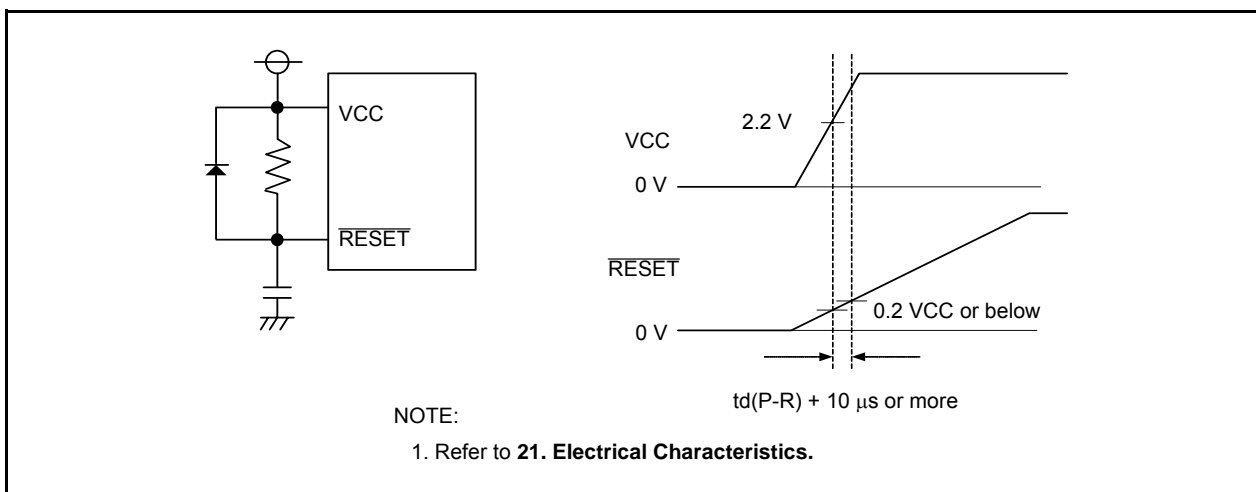


Figure 5.5 Example of Hardware Reset Circuit and Operation

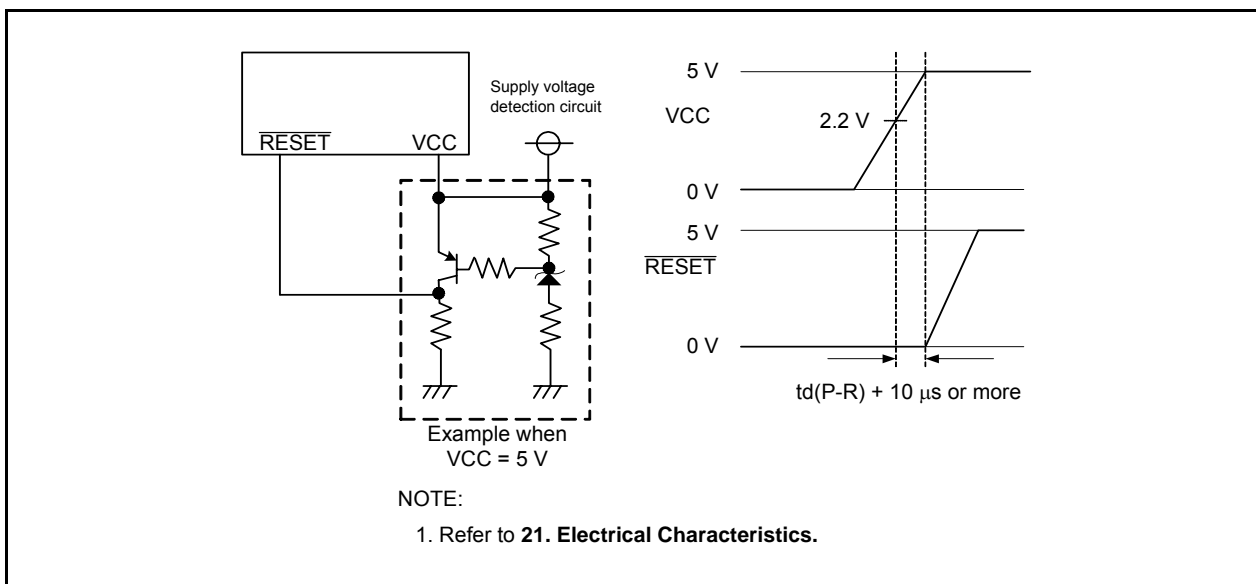


Figure 5.6 Example of Hardware Reset Circuit (Usage Example of External Supply Voltage Detection Circuit) and Operation

5.2 Power-On Reset Function

When the $\overline{\text{RESET}}$ pin is connected to the VCC pin via a pull-up resistor, and the VCC pin voltage level rises while the rise gradient is t_{rth} or more, the power-on reset function is enabled and the MCU resets its pins, CPU, and SFR. When a capacitor is connected to the $\overline{\text{RESET}}$ pin, too, always keep the voltage to the $\overline{\text{RESET}}$ pin $0.8V_{CC}$ or more. When the input voltage to the VCC pin reaches the V_{det0} level or above, the low-speed on-chip oscillator clock starts counting. When the low-speed on-chip oscillator clock count reaches 32, the internal reset signal is held “H” and the MCU enters the reset sequence (refer to Figure 5.3). The low-speed on-chip oscillator clock divided by 8 is automatically selected as the CPU clock after reset.

Refer to **4. Special Function Registers (SFRs)** for the states of the SFR after power-on reset.

The voltage monitor 0 reset is enabled after power-on reset.

Figure 5.7 shows an Example of Power-On Reset Circuit and Operation.

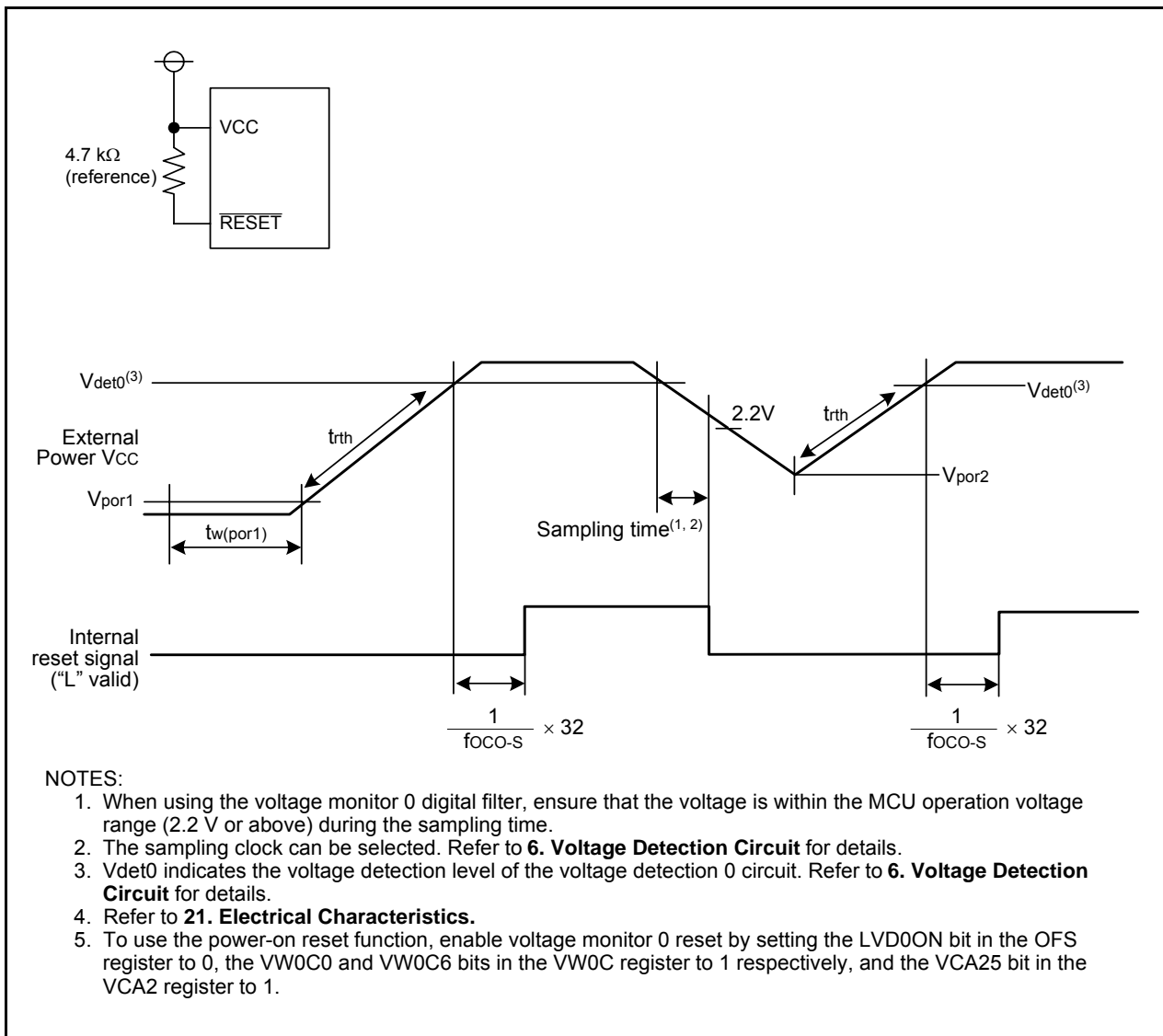


Figure 5.7 Example of Power-On Reset Circuit and Operation

5.3 Voltage Monitor 0 Reset

A reset is applied using the on-chip voltage detection 0 circuit. The voltage detection 0 circuit monitors the input voltage to the VCC pin. The voltage to monitor is Vdet0.

When the input voltage to the VCC pin reaches the Vdet0 level or below, the pins, CPU, and SFR are reset.

When the input voltage to the VCC pin reaches the Vdet0 level or above, the low-speed on-chip oscillator clock start counting. When the low-speed on-chip oscillator clock count reaches 32, the internal reset signal is held “H” and the MCU enters the reset sequence (refer to Figure 5.3). The low-speed on-chip oscillator clock divided by 8 is automatically selected as the CPU clock after reset.

The LVD0ON bit in the OFS register can be used to enable or disable voltage monitor 0 reset after a hardware reset. Setting the LVD0ON bit is only valid after a hardware reset.

To use the power-on reset function, enable voltage monitor 0 reset by setting the LVD0ON bit in the OFS register to 0, the VW0C0 and VW0C6 bits in the VW0C register to 1 respectively, and the VCA25 bit in the VCA2 register to 1.

The LVD0ON bit cannot be changed by a program. To set the LVD0ON bit, write 0 (voltage monitor 0 reset enabled after hardware reset) or 1 (voltage monitor 0 reset disabled after hardware reset) to bit 5 of address 0FFFFh using a flash programmer.

Refer to **Figure 5.4 OFS Register** for details of the OFS register.

Refer to **4. Special Function Registers (SFRs)** for the status of the SFR after voltage monitor 0 reset.

The internal RAM is not reset. When the input voltage to the VCC pin reaches the Vdet0 level or below while writing to the internal RAM is in progress, the contents of internal RAM are undefined.

Refer to **6. Voltage Detection Circuit** for details of voltage monitor 0 reset.

5.4 Voltage Monitor 1 Reset

A reset is applied using the on-chip voltage detection 1 circuit. The voltage detection 1 circuit monitors the input voltage to the VCC pin. The voltage to monitor is Vdet1.

When the input voltage to the VCC pin reaches the Vdet1 level or below, the pins, CPU, and SFR are reset and a program is executed beginning with the address indicated by the reset vector. After reset, the low-speed on-chip oscillator clock divided by 8 is automatically selected as the CPU clock.

The voltage monitor 1 does not reset some portions of the SFR. Refer to **4. Special Function Registers (SFRs)** for details.

The internal RAM is not reset. When the input voltage to the VCC pin reaches the Vdet1 level or below while writing to the internal RAM is in progress, the contents of internal RAM are undefined.

Refer to **6. Voltage Detection Circuit** for details of voltage monitor 1 reset.

5.5 Voltage Monitor 2 Reset

A reset is applied using the on-chip voltage detection 2 circuit. The voltage detection 2 circuit monitors the input voltage to the VCC pin. The voltage to monitor is Vdet2.

When the input voltage to the VCC pin reaches the Vdet2 level or below, the pins, CPU, and SFR are reset and the program beginning with the address indicated by the reset vector is executed. After reset, the low-speed on-chip oscillator clock divided by 8 is automatically selected as the CPU clock.

The voltage monitor 2 does not reset some SFRs. Refer to **4. Special Function Registers (SFRs)** for details.

The internal RAM is not reset. When the input voltage to the VCC pin reaches the Vdet2 level or below while writing to the internal RAM is in progress, the contents of internal RAM are undefined.

Refer to **6. Voltage Detection Circuit** for details of voltage monitor 2 reset.

5.6 Watchdog Timer Reset

When the PM12 bit in the PM1 register is set to 1 (reset when watchdog timer underflows), the MCU resets its pins, CPU, and SFR if the watchdog timer underflows. Then the program beginning with the address indicated by the reset vector is executed. After reset, the low-speed on-chip oscillator clock divided by 8 is automatically selected as the CPU clock.

The watchdog timer reset does not reset some SFRs. Refer to **4. Special Function Registers (SFRs)** for details. The internal RAM is not reset. When the watchdog timer underflows, the contents of internal RAM are undefined. Refer to **13. Watchdog Timer** for details of the watchdog timer.

5.7 Software Reset

When the PM03 bit in the PM0 register is set to 1 (MCU reset), the MCU resets its pins, CPU, and SFR. The program beginning with the address indicated by the reset vector is executed. After reset, the low-speed on-chip oscillator clock divided by 8 is automatically selected for the CPU clock.

The software reset does not reset some SFRs. Refer to **4. Special Function Registers (SFRs)** for details. The internal RAM is not reset.

6. Voltage Detection Circuit

The voltage detection circuit monitors the input voltage to the VCC pin. This circuit can be used to monitor the VCC input voltage by a program. Alternately, voltage monitor 0 reset, voltage monitor 1 interrupt, voltage monitor 1 reset, voltage monitor 2 interrupt, and voltage monitor 2 reset can also be used.

Table 6.1 lists the Specifications of Voltage Detection Circuit and Figures 6.1 to 6.4 show the Block Diagrams. Figures 6.5 to 6.8 show the Associated Registers.

Table 6.1 Specifications of Voltage Detection Circuit

| Item | | Voltage Detection 0 | Voltage Detection 1 | Voltage Detection 2 |
|----------------------------------|-------------------------|--|--|--|
| VCC Monitor | Voltage to monitor | Vdet0 | Vdet1 | Vdet2 |
| | Detection target | Whether passing through Vdet0 by rising or falling | Passing through Vdet1 by rising or falling | Passing through Vdet2 by rising or falling |
| | Monitor | None | VW1C3 bit in VW1C register Whether VCC is higher or lower than Vdet1 | VCA13 bit in VCA1 register Whether VCC is higher or lower than Vdet2 |
| Process When Voltage is Detected | Reset | Voltage monitor 0 reset Reset at Vdet0 > VCC; restart CPU operation at VCC > Vdet0 | Voltage monitor 1 reset Reset at Vdet1 > VCC; restart CPU operation after a specified time | Voltage monitor 2 reset Reset at Vdet2 > VCC; restart CPU operation after a specified time |
| | Interrupt | None | Voltage monitor 1 interrupt Interrupt request at Vdet1 > VCC and VCC > Vdet1 when digital filter is enabled; interrupt request at Vdet1 > VCC or VCC > Vdet1 when digital filter is disabled | Voltage monitor 2 interrupt Interrupt request at Vdet2 > VCC and VCC > Vdet2 when digital filter is enabled; interrupt request at Vdet2 > VCC or VCC > Vdet2 when digital filter is disabled |
| Digital Filter | Switch enabled/disabled | Available | Available | Available |
| | Sampling time | (Divide-by-n of fOCO-S) × 4 n: 1, 2, 4, and 8 | (Divide-by-n of fOCO-S) × 4 n: 1, 2, 4, and 8 | (Divide-by-n of fOCO-S) × 4 n: 1, 2, 4, and 8 |

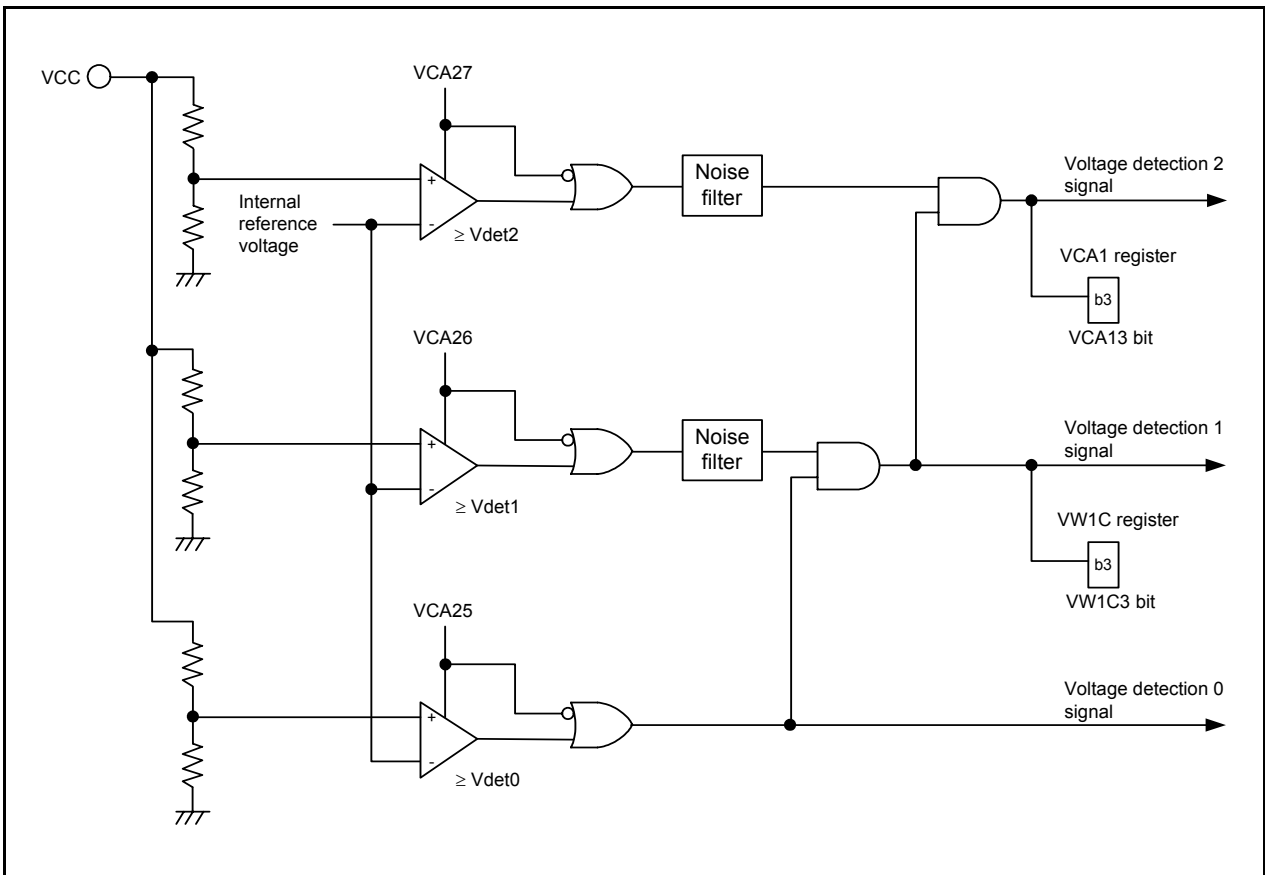


Figure 6.1 Block Diagram of Voltage Detection Circuit

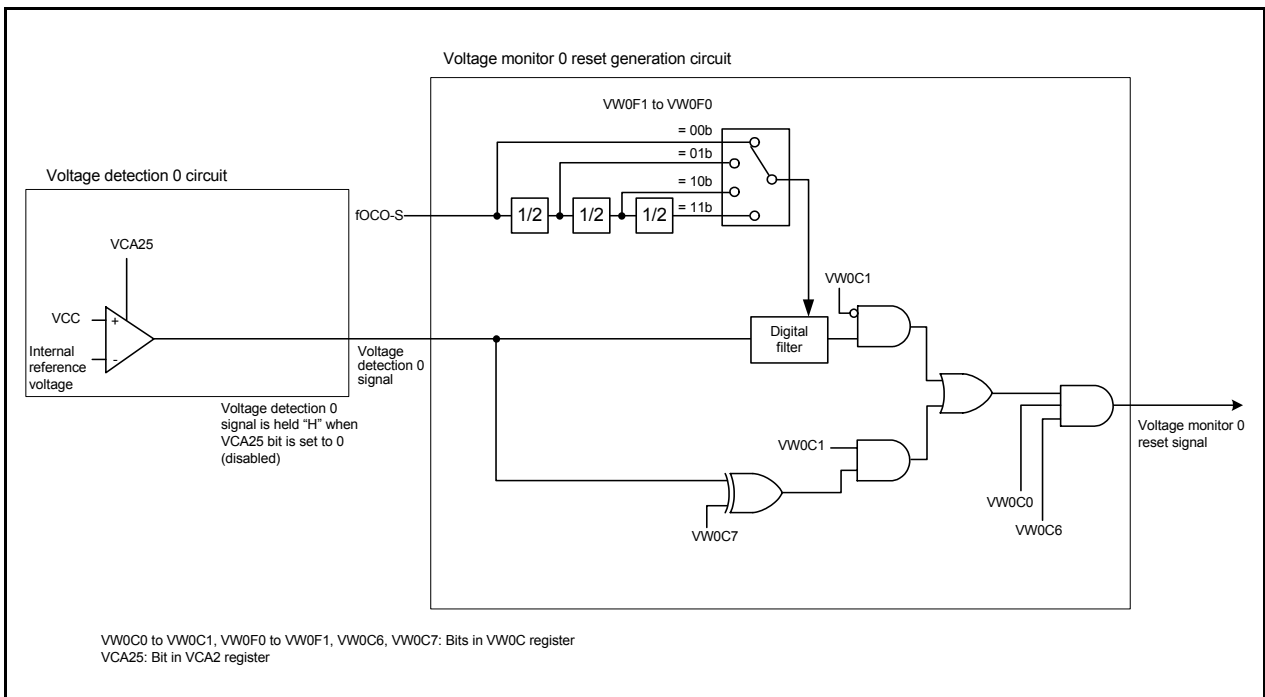


Figure 6.2 Block Diagram of Voltage Monitor 0 Reset Generation Circuit

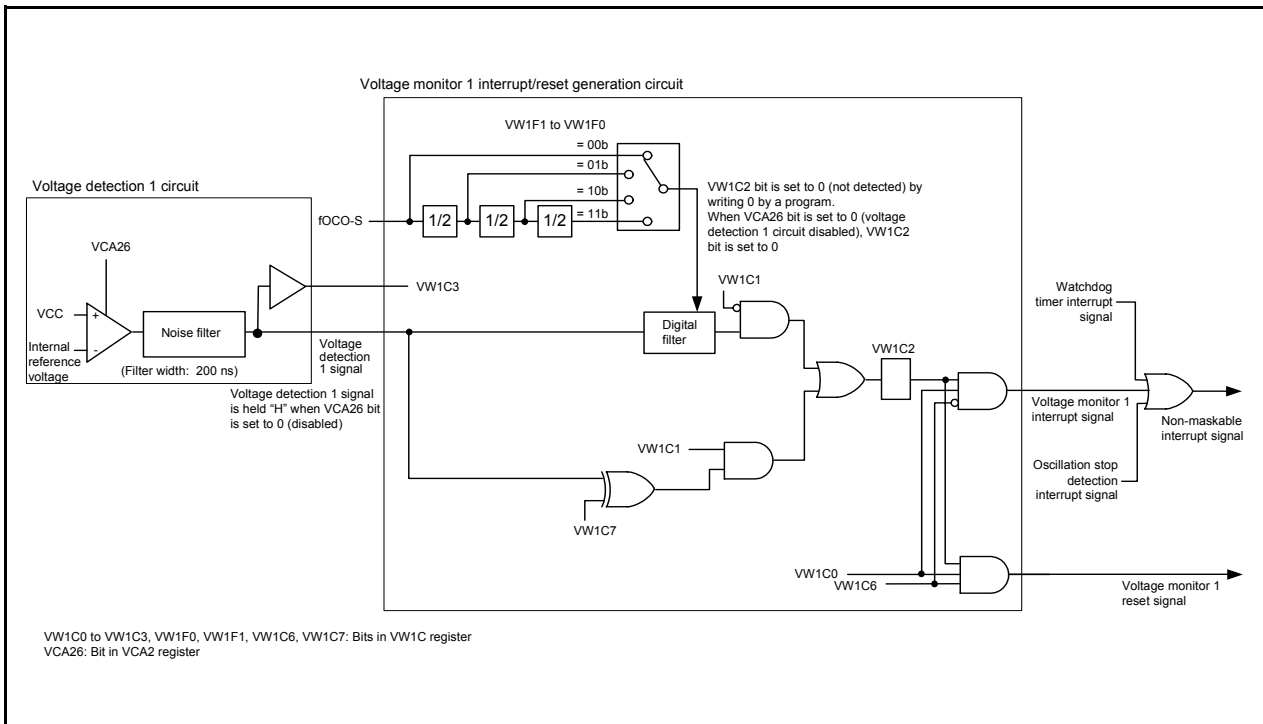


Figure 6.3 Block Diagram of Voltage Monitor 1 Interrupt/Reset Generation Circuit

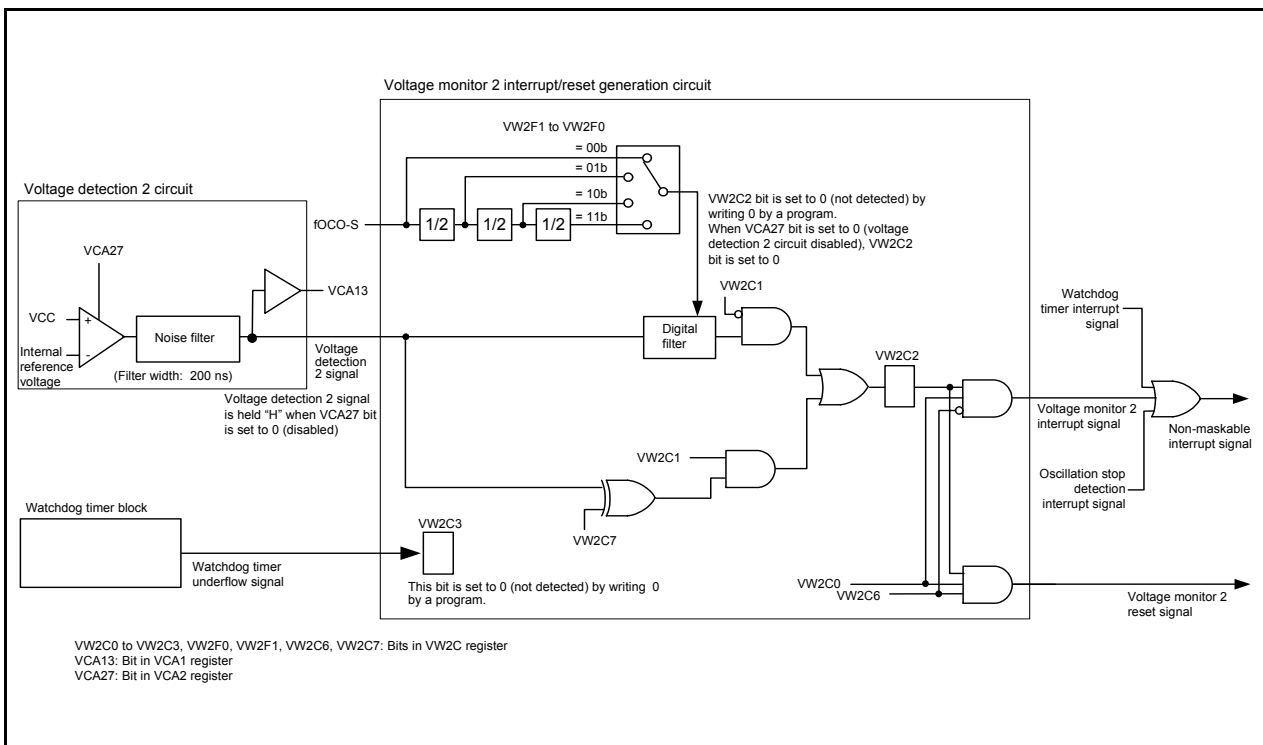


Figure 6.4 Block Diagram of Voltage Monitor 2 Interrupt/Reset Generation Circuit

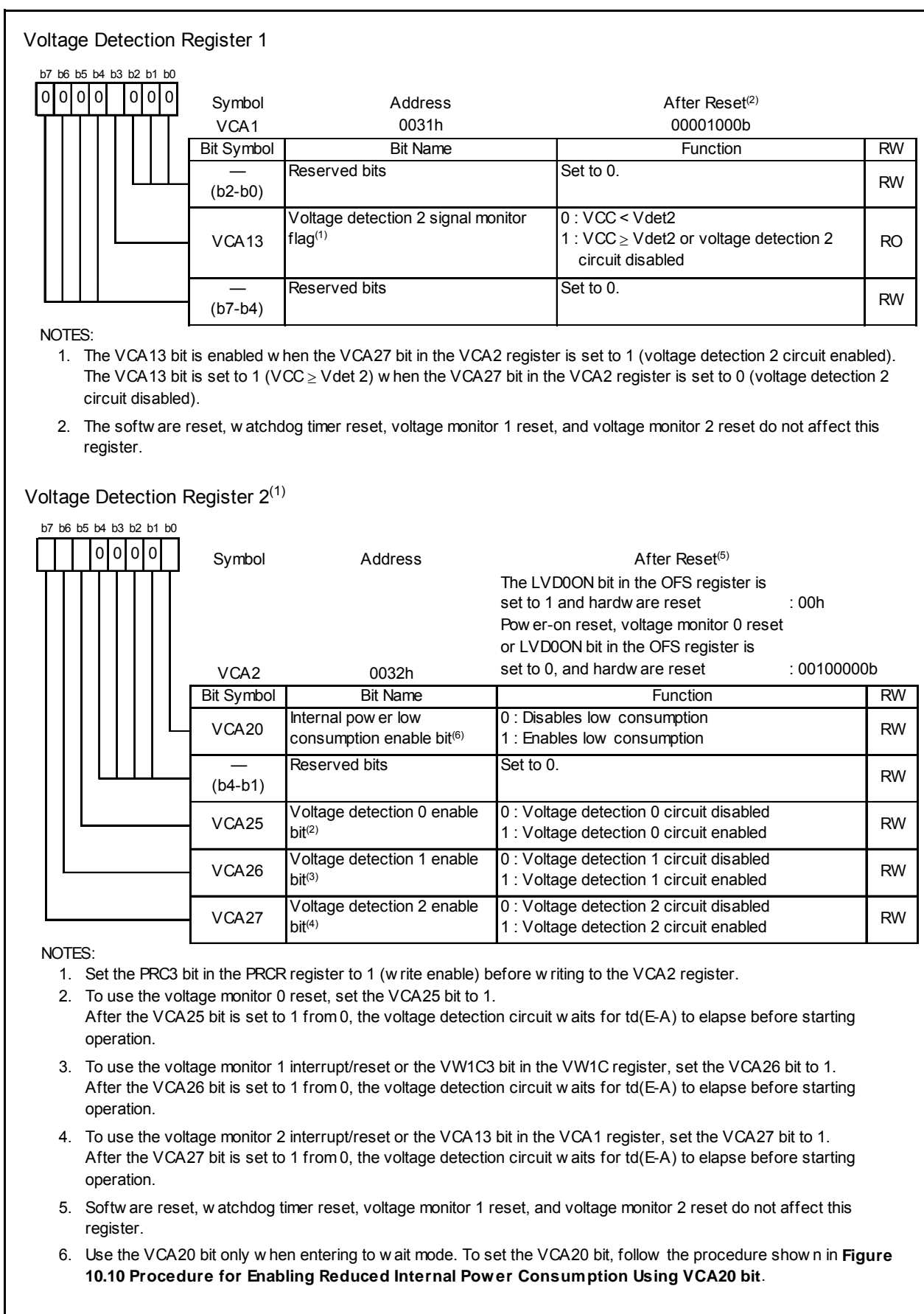


Figure 6.5 Registers VCA1 and VCA2

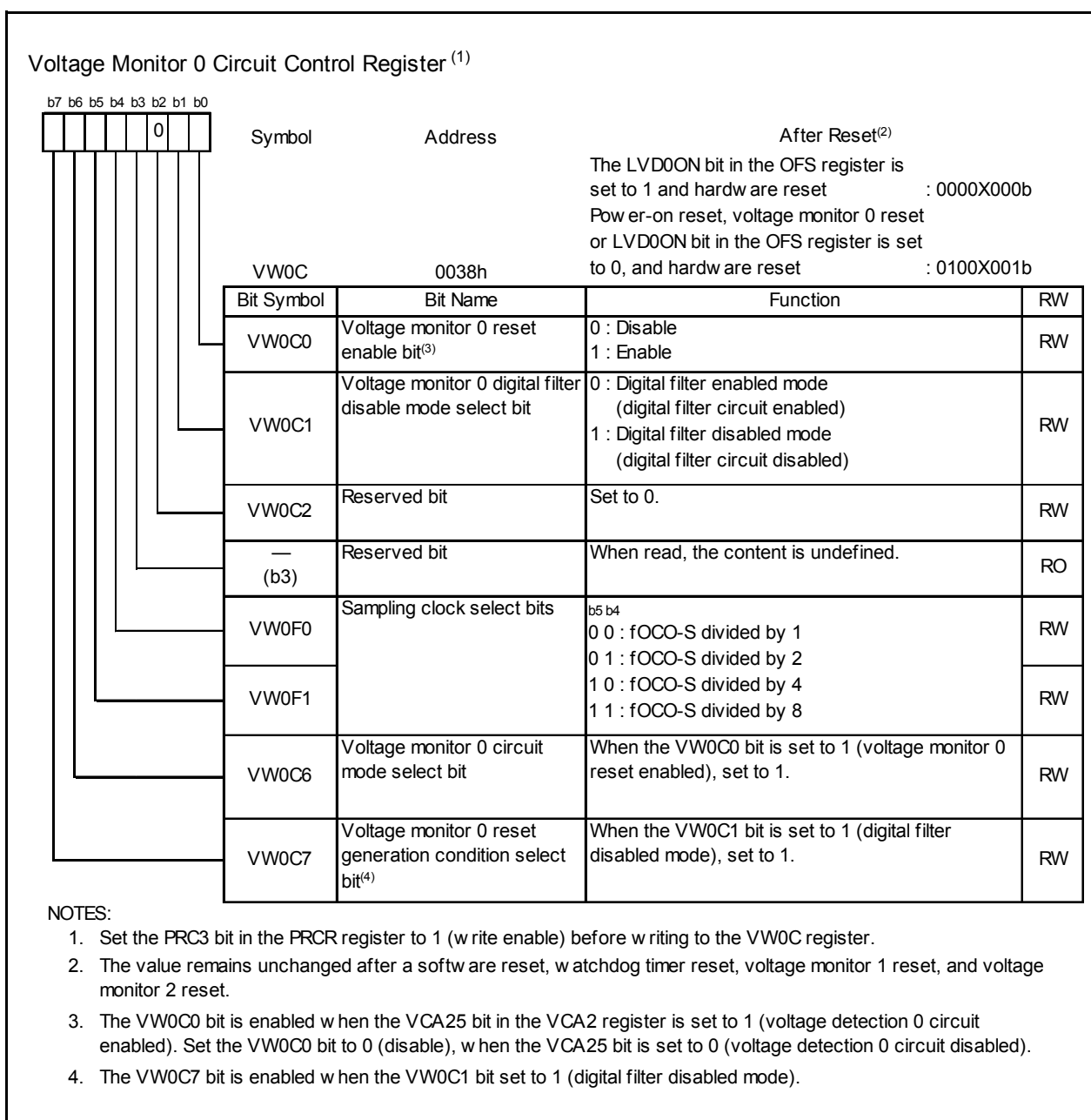


Figure 6.6 VW0C Register

Voltage Monitor 1 Circuit Control Register ⁽¹⁾

| Symbol | Address | After Reset ⁽⁸⁾ |
|--------|---------|----------------------------|
| VW1C | 0036h | 00001000b |

| Bit Symbol | Bit Name | Function | RW |
|------------|---|--|----|
| VW1C0 | Voltage monitor 1 interrupt/reset enable bit ⁽⁶⁾ | 0 : Disable 1 : Enable | RW |
| VW1C1 | Voltage monitor 1 digital filter disable mode select bit ⁽²⁾ | 0 : Digital filter enabled mode (digital filter circuit enabled) 1 : Digital filter disabled mode (digital filter circuit disabled) | RW |
| VW1C2 | Voltage change detection flag ^(3, 4, 8) | 0 : Not detected 1 : Vdet1 crossing detected | RW |
| VW1C3 | Voltage detection 1 signal monitor flag ^(3, 8) | 0 : VCC < Vdet1 1 : VCC ≥ Vdet1 or voltage detection 1 circuit disabled | RO |
| VW1F0 | Sampling clock select bits | b5 b4 0 0 : fOCO-S divided by 1 0 1 : fOCO-S divided by 2 1 0 : fOCO-S divided by 4 1 1 : fOCO-S divided by 8 | RW |
| VW1F1 | | | RW |
| VW1C6 | Voltage monitor 1 circuit mode select bit ⁽⁵⁾ | 0 : Voltage monitor 1 interrupt mode 1 : Voltage monitor 1 reset mode | RW |
| VW1C7 | Voltage monitor 1 interrupt/reset generation condition select bit ^(7, 9) | 0 : When VCC reaches Vdet1 or above 1 : When VCC reaches Vdet1 or below | RW |

NOTES:

- Set the PRC3 bit in the PRCR register to 1 (write enable) before writing to the VW1C register.
- To use the voltage monitor 1 interrupt to exit stop mode and to return again, write 0 to the VW1C1 bit before writing 1.
- Bits VW1C2 and VW1C3 are enabled when the VCA26 bit in the VCA2 register is set to 1 (voltage detection 1 circuit enabled).
- Set this bit to 0 by a program. When 0 is written by a program, it is set to 0 (and remains unchanged even if 1 is written to it).
- The VW1C6 bit is enabled when the VW1C0 bit is set to 1 (voltage monitor 1 interrupt/enabled reset).
- The VW1C0 bit is enabled when the VCA26 bit in the VCA2 register is set to 1 (voltage detection 1 circuit enabled). Set the VW1C0 bit to 0 (disable) when the VCA26 bit is set to 0 (voltage detection 1 circuit disabled).
- The VW1C7 bit is enabled when the VW1C1 bit is set to 1 (digital filter disabled mode).
- Bits VW1C2 and VW1C3 remain unchanged after a software reset, watchdog timer reset, voltage monitor 1 reset, or voltage monitor 2 reset.
- When the VW1C6 bit is set to 1 (voltage monitor 1 reset mode), set the VW1C7 bit to 1 (when VCC reaches Vdet1 or below). (Do not set to 0.)

Figure 6.7 VW1C Register

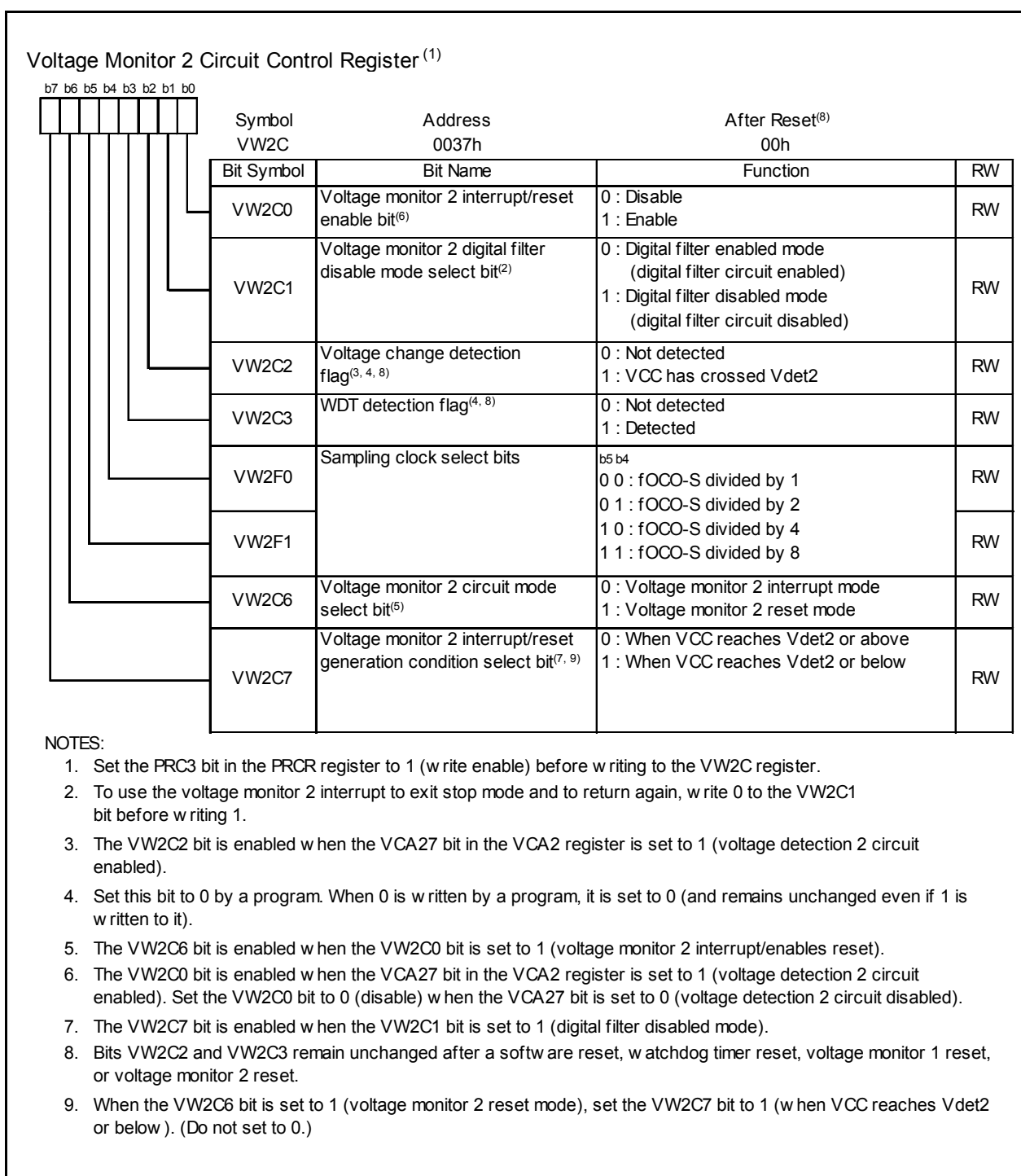


Figure 6.8 VW2C Register

6.1 VCC Input Voltage

6.1.1 Monitoring Vdet0

Vdet0 cannot be monitored.

6.1.2 Monitoring Vdet1

Set the VCA26 bit in the VCA2 register to 1 (voltage detection 1 circuit enabled). After td(E-A) has elapsed (refer to **21. Electrical Characteristics**), Vdet1 can be monitored by the VW1C3 bit in the VW1C register.

6.1.3 Monitoring Vdet2

Set the VCA27 bit in the VCA2 register to 1 (voltage detection 2 circuit enabled). After td(E-A) has elapsed (refer to **21. Electrical Characteristics**), Vdet2 can be monitored by the VCA13 bit in the VCA1 register.

6.2 Voltage Monitor 0 Reset

Table 6.2 lists the Procedure for Setting Bits Associated with Voltage Monitor Reset and Figure 6.9 shows an Example of Voltage Monitor 0 Reset Operation. To use the voltage monitor 0 reset to exit stop mode, set the VW0C1 bit in the VW0C register to 1 (digital filter disabled).

Table 6.2 Procedure for Setting Bits Associated with Voltage Monitor Reset

| Step | When Using Digital Filter | When Not Using Digital Filter |
|------|---|---|
| 1 | Set the VCA25 bit in the VCA2 register to 1 (voltage detection 0 circuit enabled) | |
| 2 | Wait for $t_d(E-A)$ | |
| 3 | Select the sampling clock of the digital filter by the VW0F0 to VW0F1 bits in the VW0C register | Set the VW0C7 bit in the VW0C register to 1 |
| 4(1) | Set the VW0C1 bit in the VW0C register to 0 (digital filter enabled) | Set the VW0C1 bit in the VW0C register to 1 (digital filter disabled) |
| 5(1) | Set the VW0C6 bit in the VW0C register to 1 (voltage monitor 0 reset mode) | |
| 6 | Set the VW0C2 bit in the VW0C register to 0 | |
| 7 | Set the CM14 bit in the CM1 register to 0 (low-speed on-chip oscillator on) | – |
| 8 | Wait for 4 cycles of the sampling clock of the digital filter | – (No wait time required) |
| 9 | Set the VW0C0 bit in the VW0C register to 1 (voltage monitor 0 reset enabled) | |

NOTE:

- When the VW0C0 bit is set to 0, steps 3, 4, and 5 can be executed simultaneously (with 1 instruction).

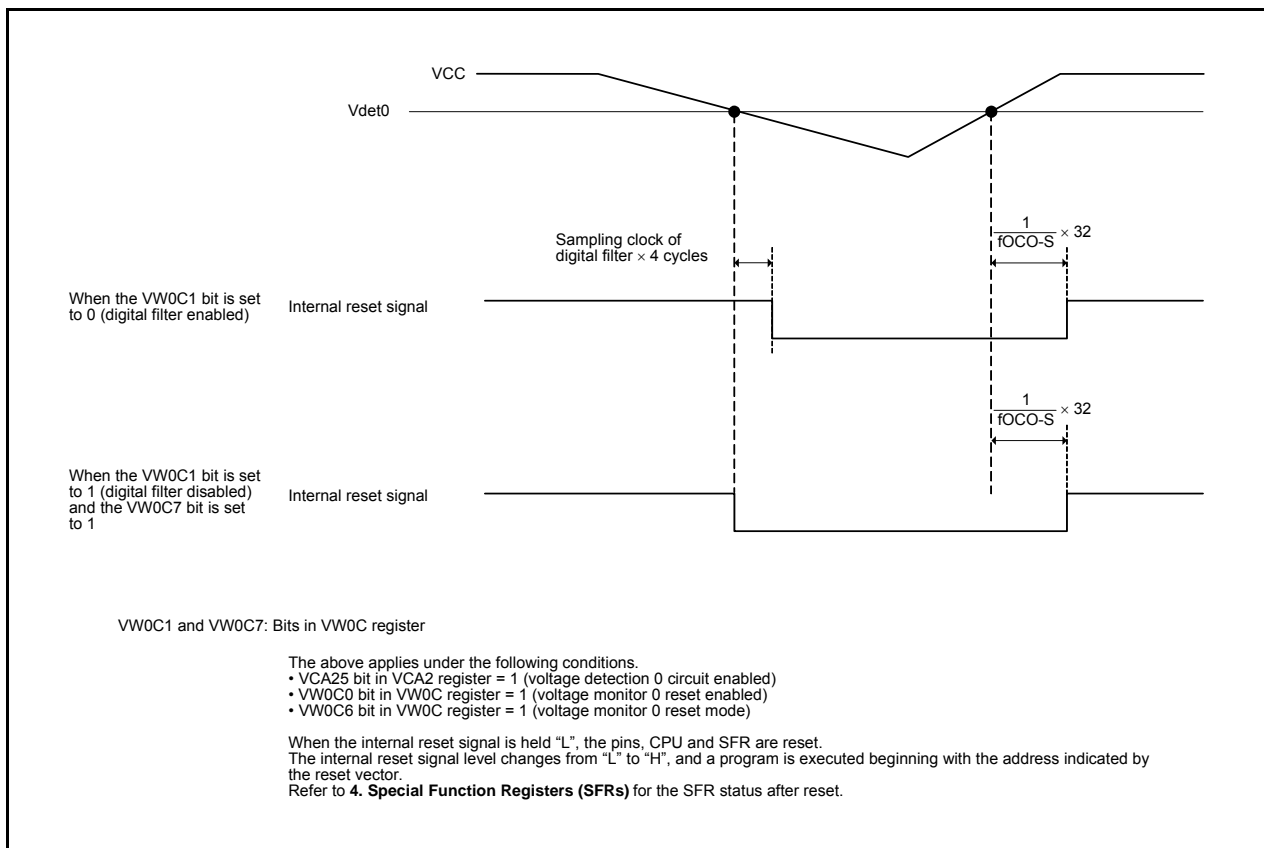


Figure 6.9 Example of Voltage Monitor 0 Reset Operation

6.3 Voltage Monitor 1 Interrupt and Voltage Monitor 1 Reset

Table 6.3 lists the Procedure for Setting Bits Associated with Voltage Monitor 1 Interrupt and Reset. Figure 6.10 shows an Example of Voltage Monitor 1 Interrupt and Voltage Monitor 1 Reset Operation. To use the voltage monitor 1 interrupt or voltage monitor 1 reset to exit stop mode, set the VW1C1 bit in the VW1C register to 1 (digital filter disabled).

Table 6.3 Procedure for Setting Bits Associated with Voltage Monitor 1 Interrupt and Reset

| Step | When Using Digital Filter | | When Not Using Digital Filter | |
|------------------|---|--|---|--|
| | Voltage Monitor 1 Interrupt | Voltage Monitor 1 Reset | Voltage Monitor 1 Interrupt | Voltage Monitor 1 Reset |
| 1 | Set the VCA26 bit in the VCA2 register to 1 (voltage detection 1 circuit enabled) | | | |
| 2 | Wait for $t_d(E-A)$ | | | |
| 3 | Select the sampling clock of the digital filter by the VW1F0 to VW1F1 bits in the VW1C register | | Select the timing of the interrupt and reset request by the VW1C7 bit in the VW1C register ⁽¹⁾ | |
| 4 ⁽²⁾ | Set the VW1C1 bit in the VW1C register to 0 (digital filter enabled) | | Set the VW1C1 bit in the VW1C register to 1 (digital filter disabled) | |
| 5 ⁽²⁾ | Set the VW1C6 bit in the VW1C register to 0 (voltage monitor 1 interrupt mode) | Set the VW1C6 bit in the VW1C register to 1 (voltage monitor 1 reset mode) | Set the VW1C6 bit in the VW1C register to 0 (voltage monitor 1 interrupt mode) | Set the VW1C6 bit in the VW1C register to 1 (voltage monitor 1 reset mode) |
| 6 | Set the VW1C2 bit in the VW1C register to 0 (passing of Vdet1 is not detected) | | | |
| 7 | Set the CM14 bit in the CM1 register to 0 (low-speed on-chip oscillator on) | | - | |
| 8 | Wait for 4 cycles of the sampling clock of the digital filter | | - (No wait time required) | |
| 9 | Set the VW1C0 bit in the VW1C register to 1 (voltage monitor 1 interrupt/reset enabled) | | | |

NOTES:

1. Set the VW1C7 bit to 1 (when VCC reaches Vdet1 or below) for the voltage monitor 1 reset.
2. When the VW1C0 bit is set to 0, steps 3, 4, and 5 can be executed simultaneously (with 1 instruction).

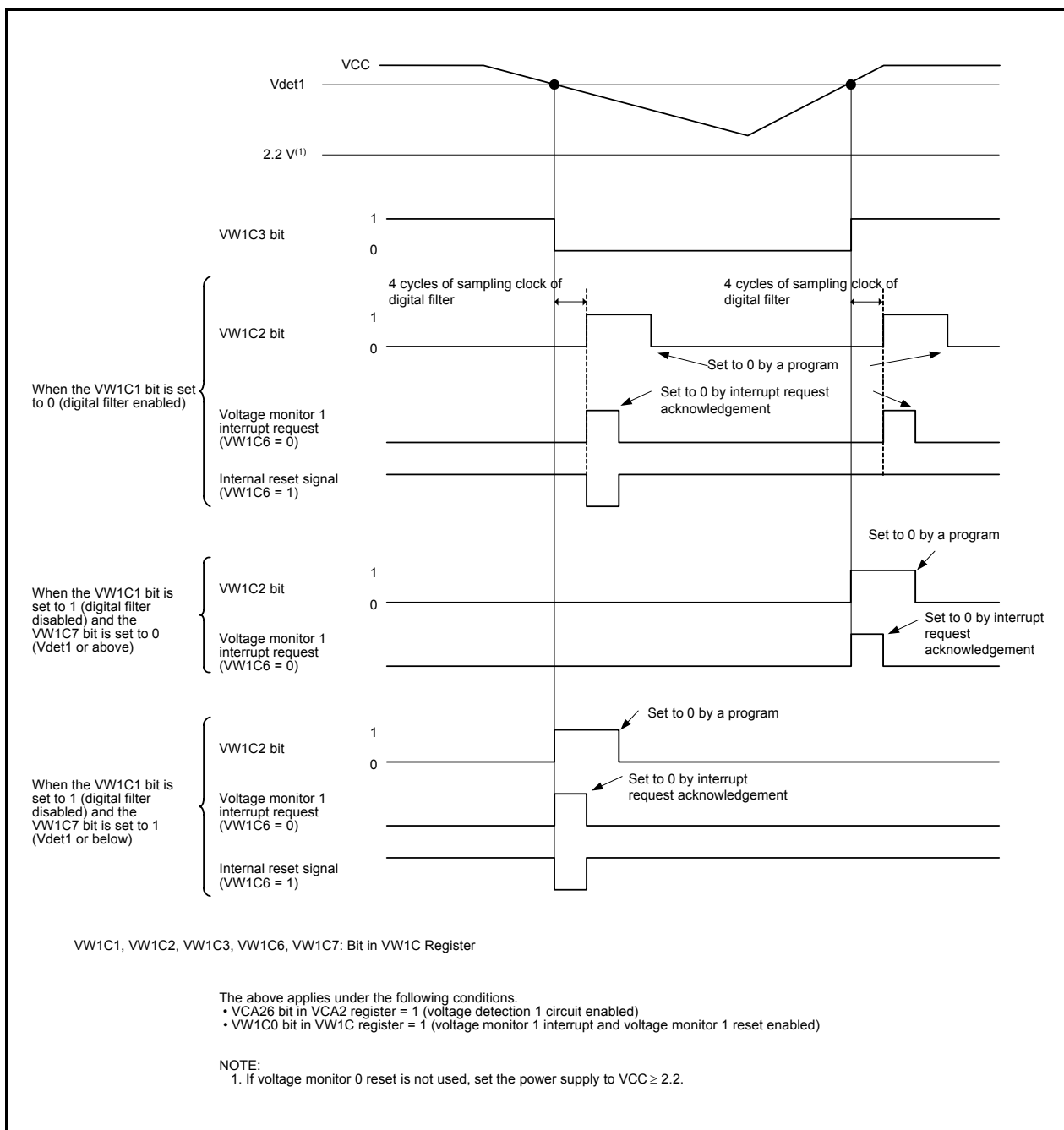


Figure 6.10 Example of Voltage Monitor 1 Interrupt and Voltage Monitor 1 Reset Operation

6.4 Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset

Table 6.4 lists the Procedure for Setting Bits Associated with Voltage Monitor 2 Interrupt and Reset. Figure 6.11 shows an Example of Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset Operation. To use the voltage monitor 2 interrupt or voltage monitor 2 reset to exit stop mode, set the VW2C1 bit in the VW2C register to 1 (digital filter disabled).

Table 6.4 Procedure for Setting Bits Associated with Voltage Monitor 2 Interrupt and Reset

| Step | When Using Digital Filter | | When Not Using Digital Filter | |
|------------------|---|--|---|--|
| | Voltage Monitor 2 Interrupt | Voltage Monitor 2 Reset | Voltage Monitor 2 Interrupt | Voltage Monitor 2 Reset |
| 1 | Set the VCA27 bit in the VCA2 register to 1 (voltage detection 2 circuit enabled) | | | |
| 2 | Wait for $t_d(E-A)$ | | | |
| 3 | Select the sampling clock of the digital filter by the VW2F0 to VW2F1 bits in the VW2C register | | Select the timing of the interrupt and reset request by the VW2C7 bit in the VW2C register ⁽¹⁾ | |
| 4 | Set the VW2C1 bit in the VW2C register to 0 (digital filter enabled) | | Set the VW2C1 bit in the VW2C register to 1 (digital filter disabled) | |
| 5 ⁽²⁾ | Set the VW2C6 bit in the VW2C register to 0 (voltage monitor 2 interrupt mode) | Set the VW2C6 bit in the VW2C register to 1 (voltage monitor 2 reset mode) | Set the VW2C6 bit in the VW2C register to 0 (voltage monitor 2 interrupt mode) | Set the VW2C6 bit in the VW2C register to 1 (voltage monitor 2 reset mode) |
| 6 | Set the VW2C2 bit in the VW2C register to 0 (passing of Vdet2 is not detected) | | | |
| 7 | Set the CM14 bit in the CM1 register to 0 (low-speed on-chip oscillator on) | | - | |
| 8 | Wait for 4 cycles of the sampling clock of the digital filter | | - (No wait time required) | |
| 9 | Set the VW2C0 bit in the VW2C register to 1 (voltage monitor 2 interrupt/reset enabled) | | | |

NOTES:

1. Set the VW2C7 bit to 1 (when VCC reaches Vdet2 or below) for the voltage monitor 2 reset.
2. When the VW2C0 bit is set to 0, steps 3, 4, and 5 can be executed simultaneously (with 1 instruction).

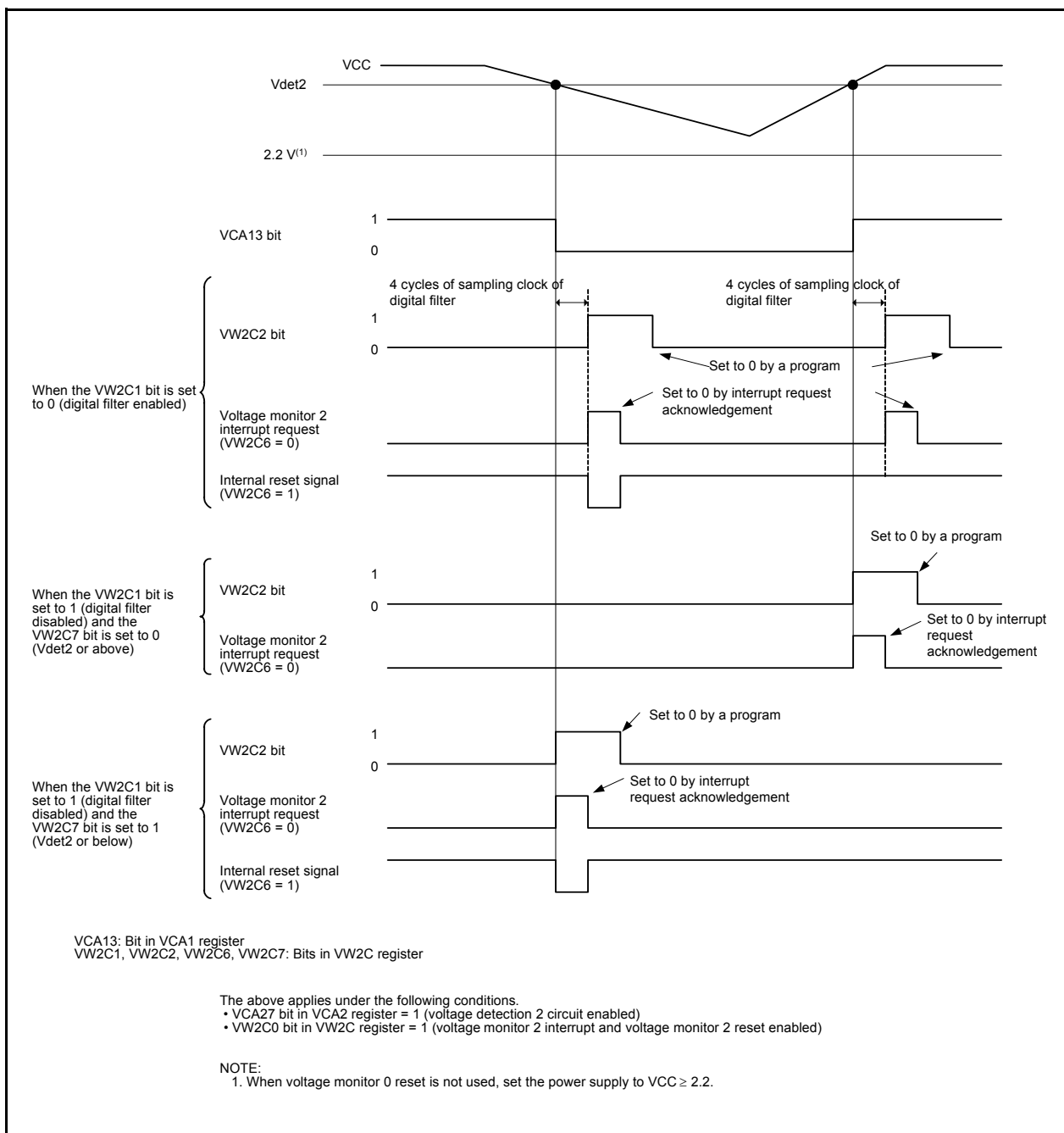


Figure 6.11 Example of Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset Operation

7. Programmable I/O Ports

There are 55 programmable Input/Output ports (I/O ports) P0 to P3, P4_3 to P4_5, P5_0 to P5_4, P6, and P8_0 to P8_6. Also, P4_6 and P4_7 can be used as input-only ports if the XIN clock oscillation circuit is not used.

Table 7.1 lists an Overview of Programmable I/O Ports.

Table 7.1 Overview of Programmable I/O Ports

| Ports | I/O | Type of Output | I/O Setting | Internal Pull-Up Resistor |
|--|-----|----------------------|-------------|---------------------------------|
| P0 to P3, P5_0 to P5_3, P6, P8_0 to P8_3 | I/O | CMOS3 State | Set per bit | Set every 4 bits ⁽¹⁾ |
| P4_3, P5_4 | I/O | CMOS3 State | Set per bit | Set every bit ⁽¹⁾ |
| P4_4, P4_5 | I/O | CMOS3 State | Set per bit | Set every 2 bits ⁽¹⁾ |
| P4_6, P4_7 ⁽²⁾ | I | (No output function) | None | None |
| P8_4 to P8_6 | I/O | CMOS3 State | Set per bit | Set every 3 bits ⁽¹⁾ |

NOTES:

1. In input mode, whether an internal pull-up resistor is connected or not can be selected by registers PUR0, PUR1, and PUR2.
2. When the XIN clock oscillation circuit is not used, these ports can be used as the input-only ports.

7.1 Functions of Programmable I/O Ports

The PDi_j (j = 0 to 7) bit in the PDi (i = 0 to 6, 8) register controls I/O of the ports P0 to P3, P4_3 to P4_5, P5_0 to P5_4, P6, and P8_0 to P8_6. The Pi register consists of a port latch to hold output data and a circuit to read pin states.

Figures 7.1 to 7.10 show the Configurations of Programmable I/O Ports. Table 7.2 lists the Functions of Programmable I/O Ports. Also, Figure 7.12 shows the PDi (i = 0 to 6 and 8) Registers. Figure 7.13 shows the Pi (i = 0 to 6 and 8) Registers, Figure 7.14 shows the P2DRR Register, Figure 7.15 shows the PMR Register, and Figure 7.16 shows Registers PUR0, PUR1, and PUR2.

Table 7.2 Functions of Programmable I/O Ports

| Operation When Accessing Pi Register | Value of PDi_j Bit in PDi Register ⁽¹⁾ | |
|--------------------------------------|---|--|
| | When PDi_j Bit is Set to 0 (Input Mode) | When PDi_j Bit is Set to 1 (Output Mode) |
| Reading | Read pin input level | Read the port latch |
| Writing | Write to the port latch | Write to the port latch. The value written to the port latch is output from the pin. |

i = 0 to 6, 8, j = 0 to 7

NOTE:

1. Nothing is assigned to bits PD4_0 to PD4_2, PD4_6, and PD4_7.

7.2 Effect on Peripheral Functions

Programmable I/O ports function as I/O ports for peripheral functions (Refer to **Table 1.7 Pin Name Information by Pin Number (1)** and **Table 1.8 Pin Name Information by Pin Number (2)**).

Table 7.3 lists the Setting of PDi_j Bit when Functioning as I/O Ports for Peripheral Functions (i = 0 to 6, j = 0 to 7).

Refer to the description of each function for information on how to set peripheral functions.

Table 7.3 Setting of PDi_j Bit when Functioning as I/O Ports for Peripheral Functions (i = 0 to 6, j = 0 to 7)

| I/O of Peripheral Functions | PDi_j Bit Settings for Shared Pin Functions |
|-----------------------------|--|
| Input | Set this bit to 0 (input mode). |
| Output | This bit can be set to either 0 or 1 (output regardless of the port setting) |

7.3 Pins Other than Programmable I/O Ports

Figure 7.11 shows the Configuration of I/O Pins.

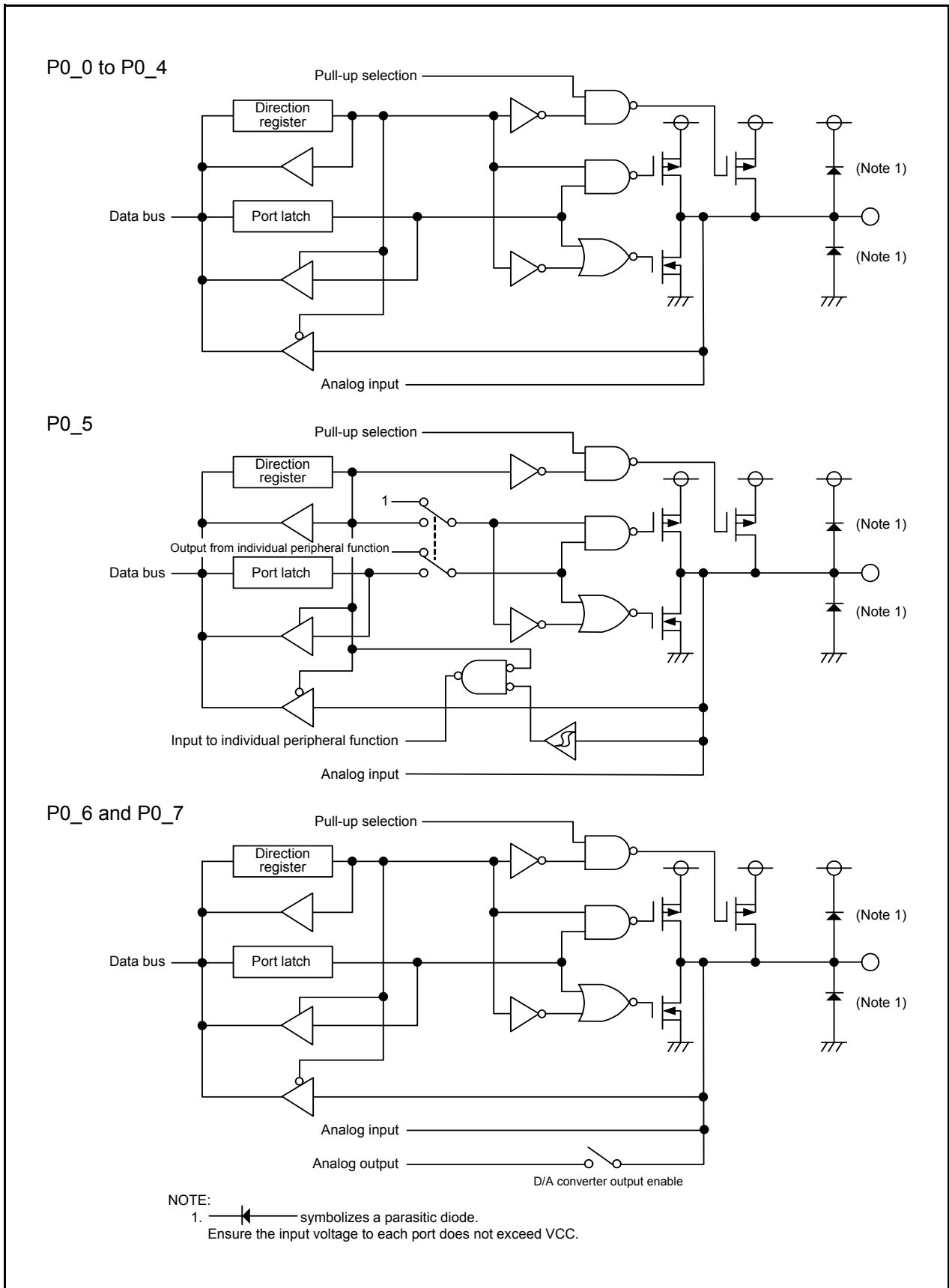


Figure 7.1 Configuration of Programmable I/O Ports (1)

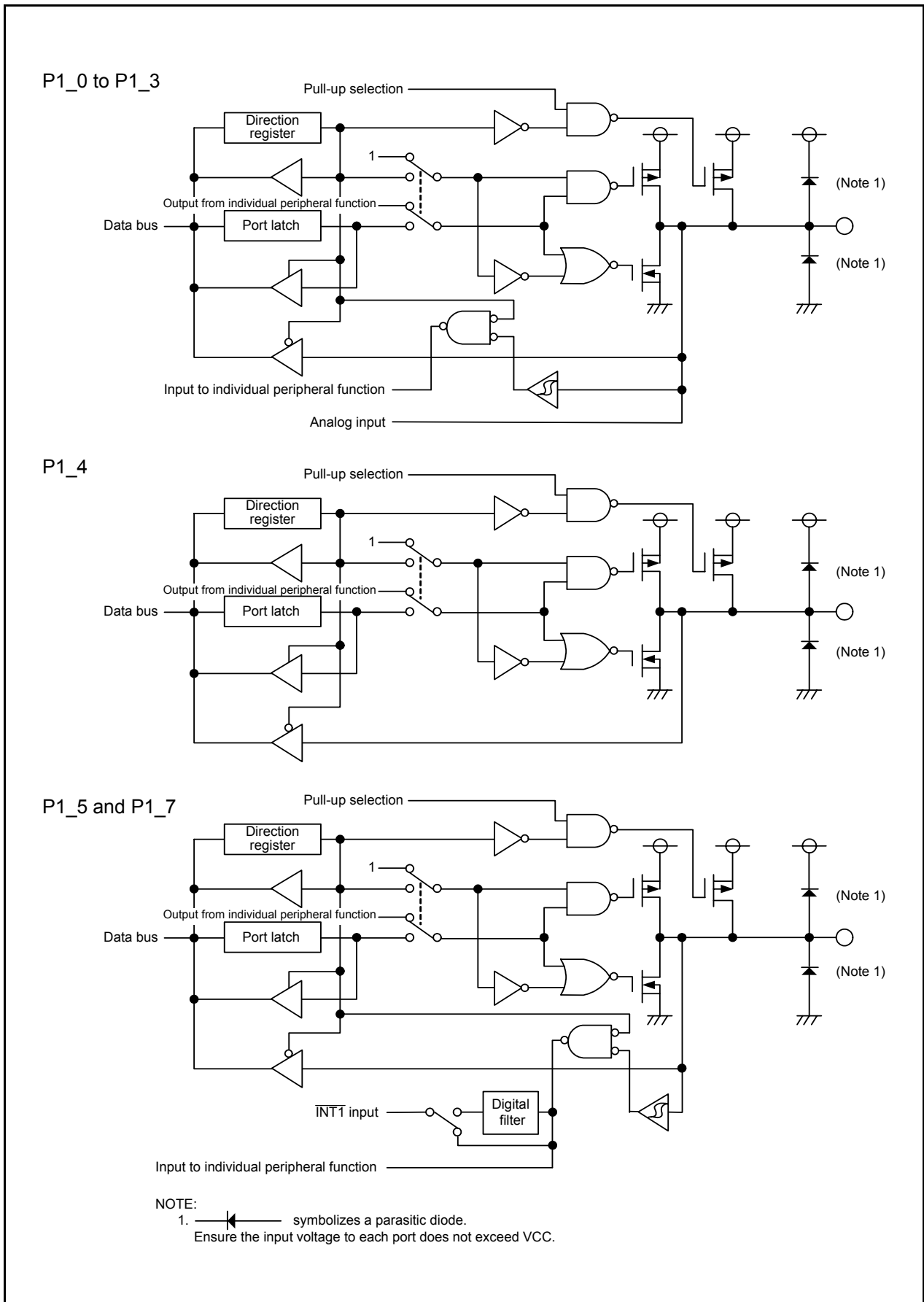


Figure 7.2 Configuration of Programmable I/O Ports (2)

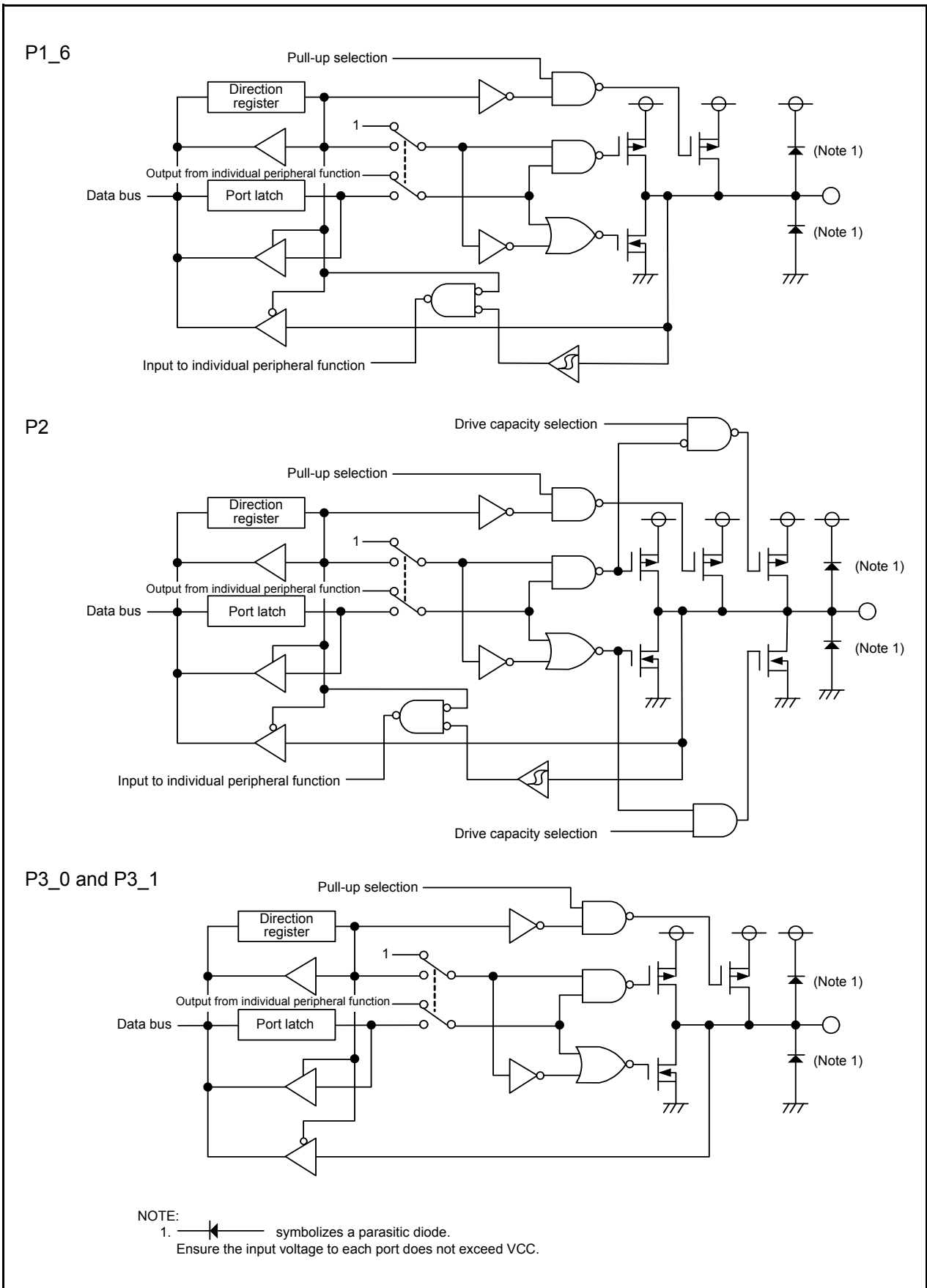


Figure 7.3 Configuration of Programmable I/O Ports (3)

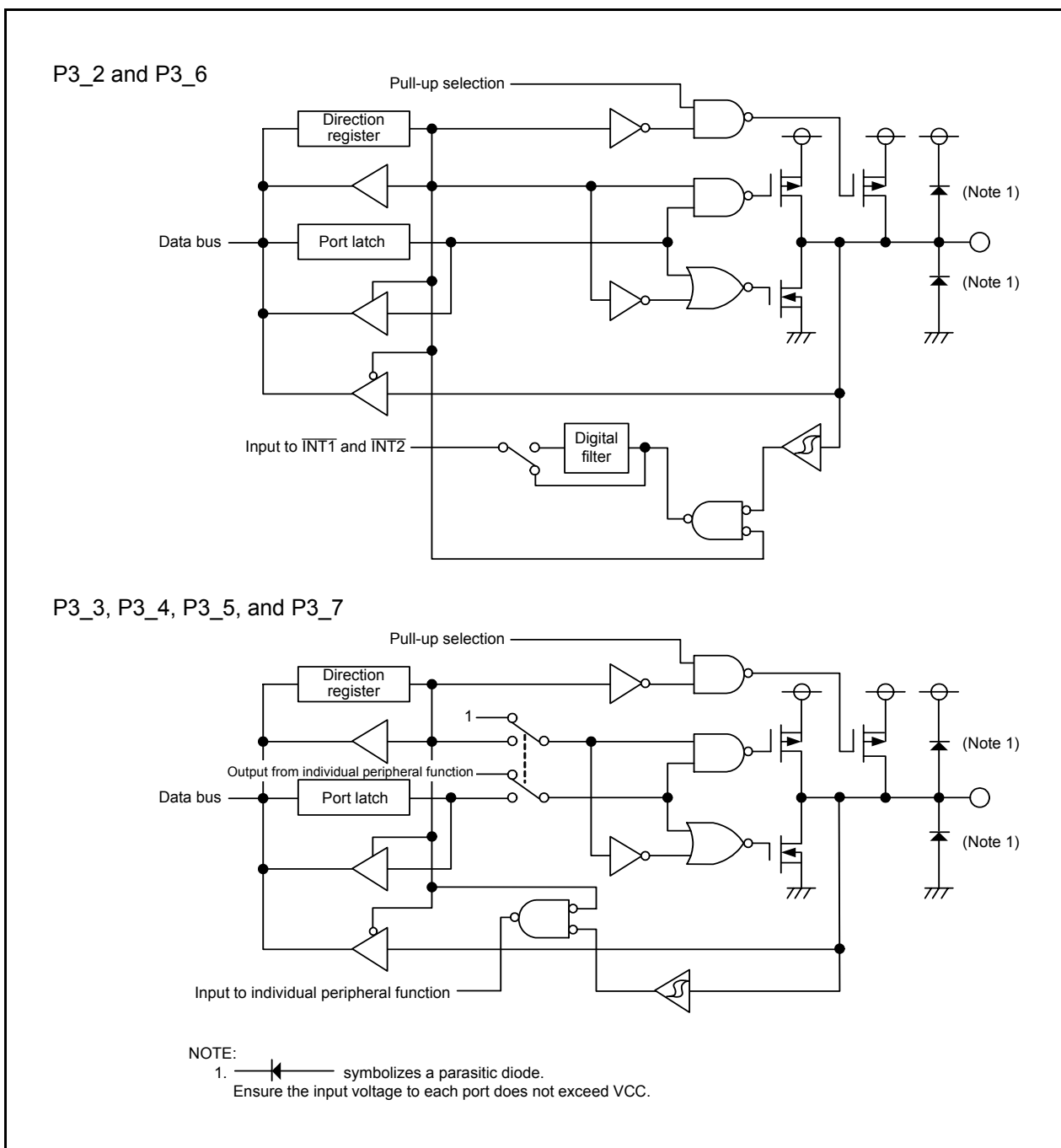


Figure 7.4 Configuration of Programmable I/O Ports (4)

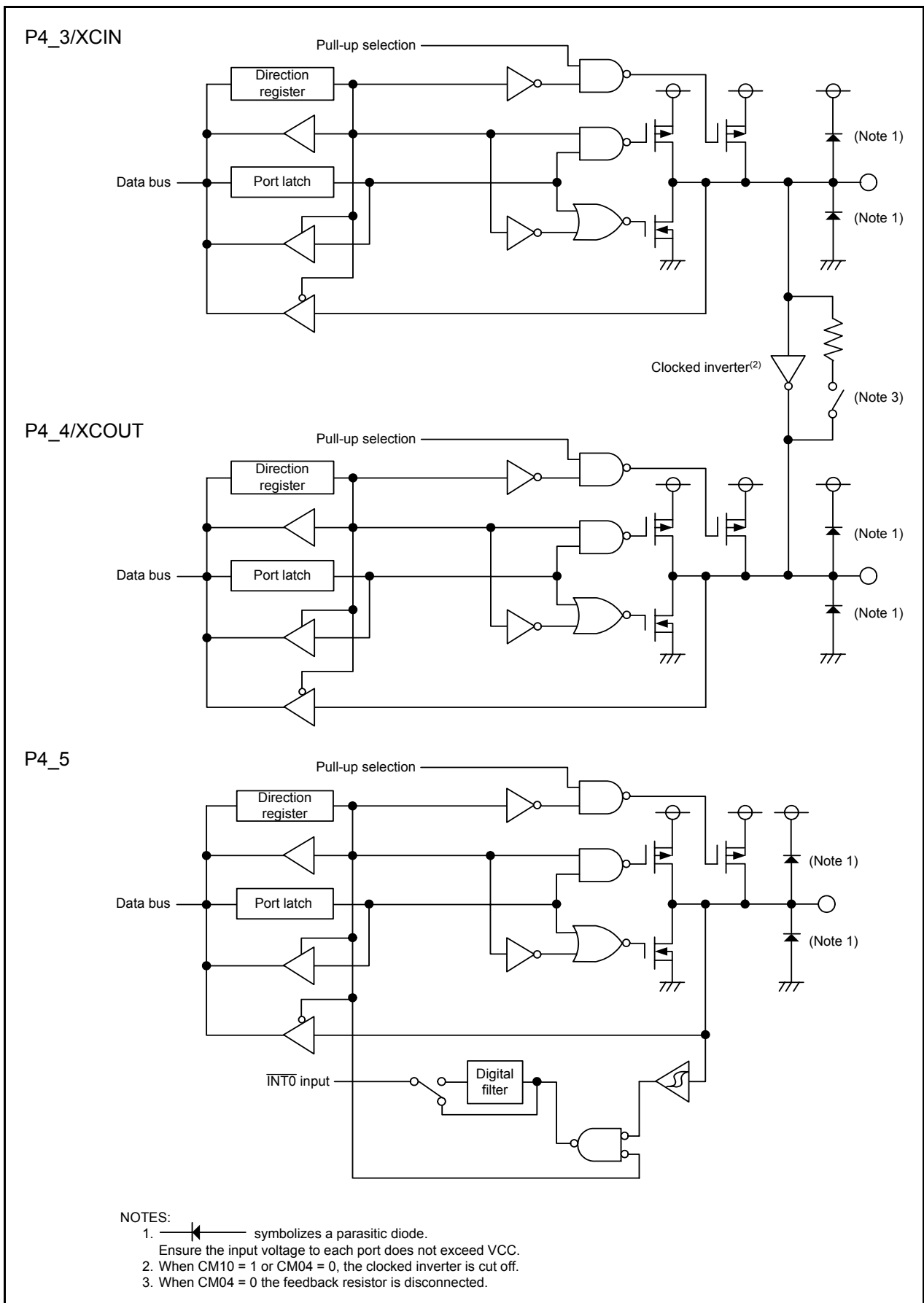


Figure 7.5 Configuration of Programmable I/O Ports (5)

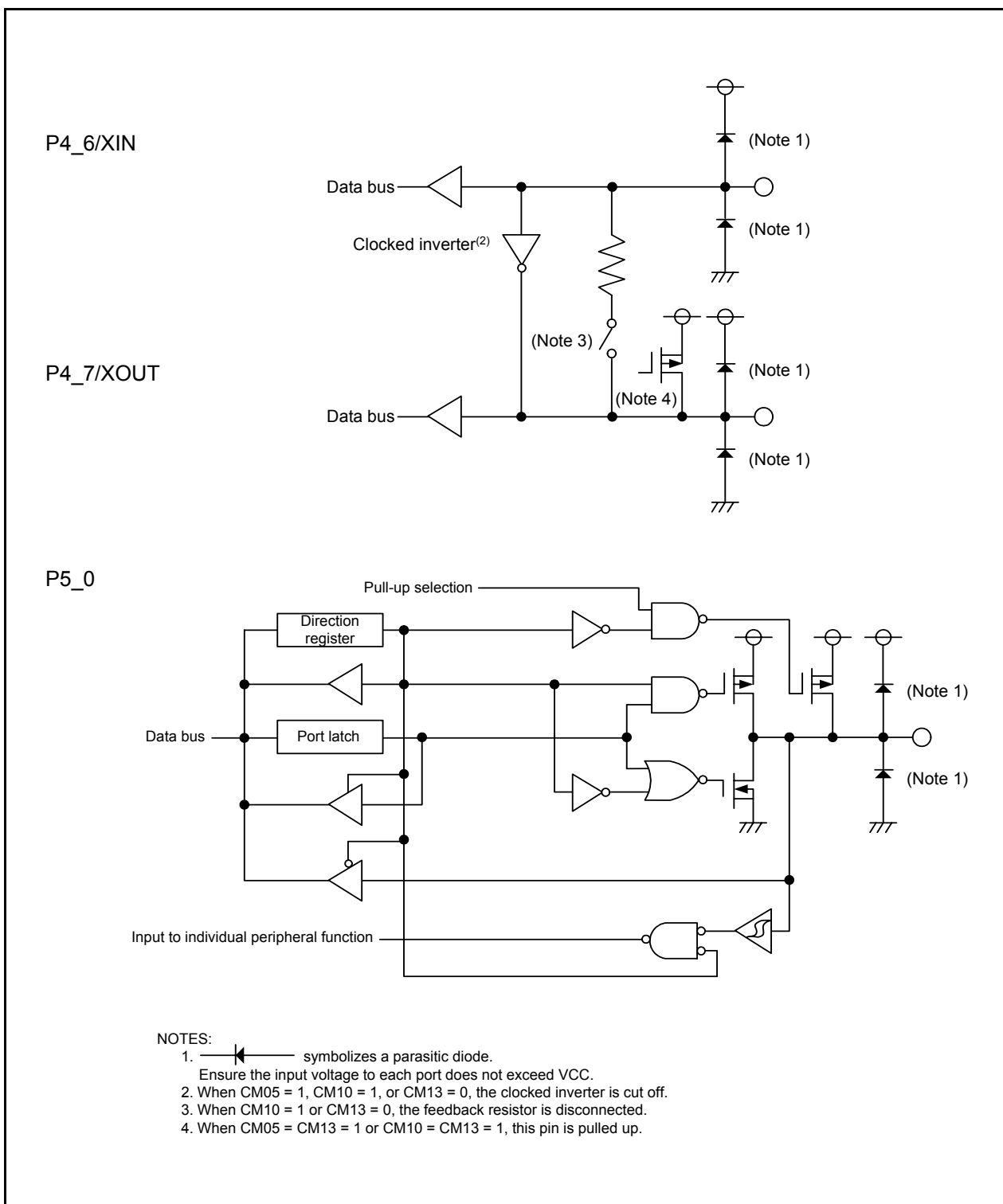


Figure 7.6 Configuration of Programmable I/O Ports (6)

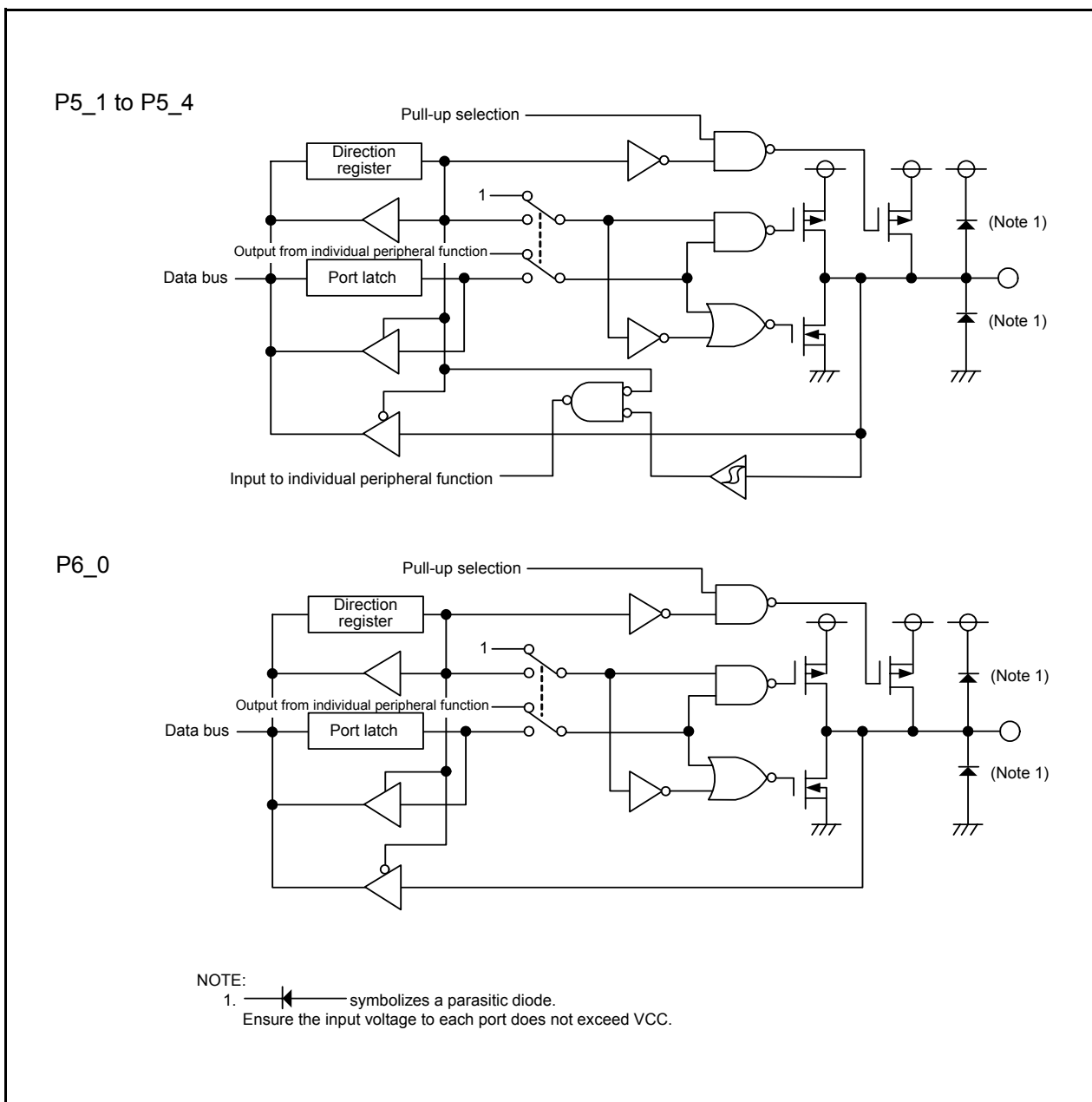


Figure 7.7 Configuration of Programmable I/O Ports (7)

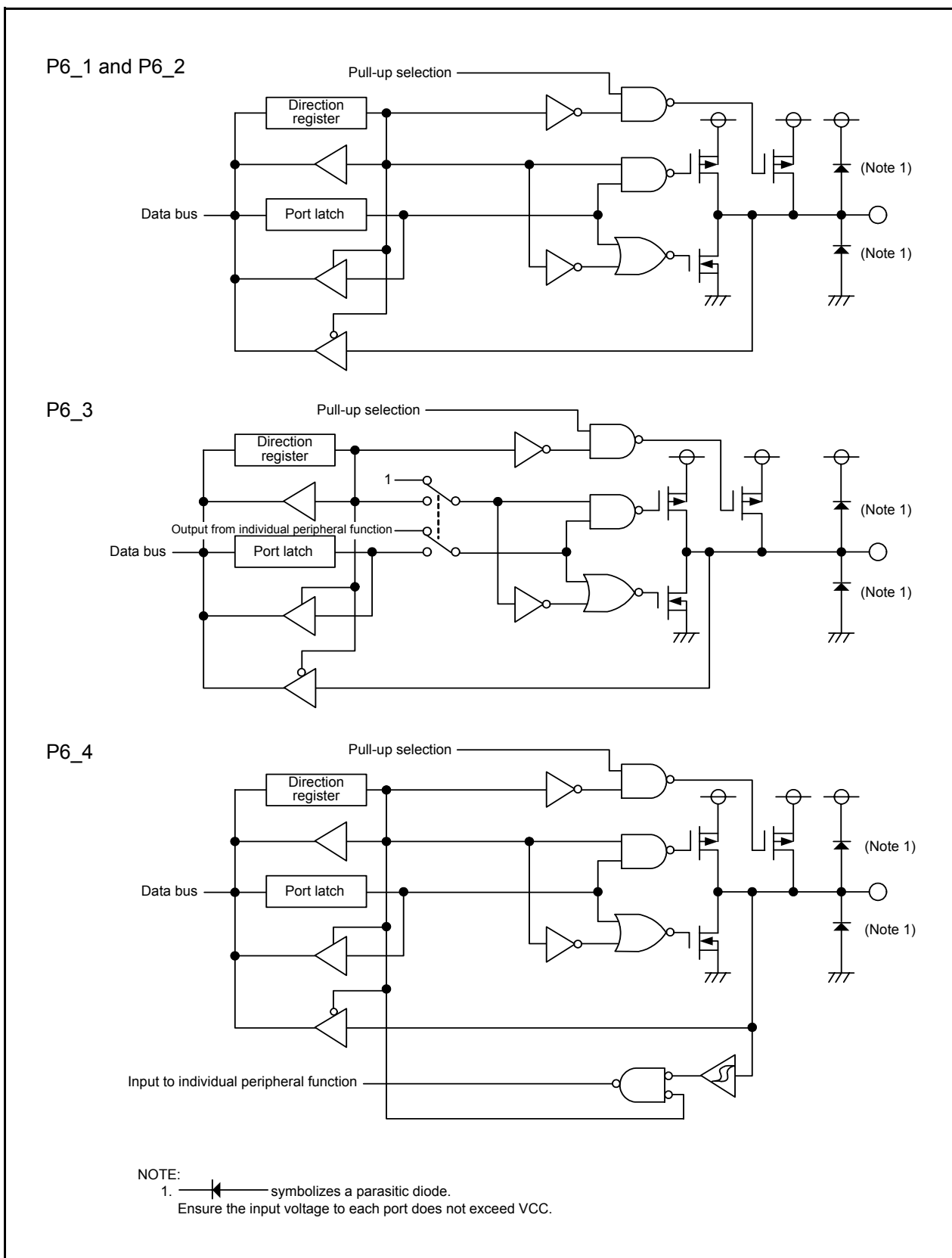


Figure 7.8 Configuration of Programmable I/O Ports (8)

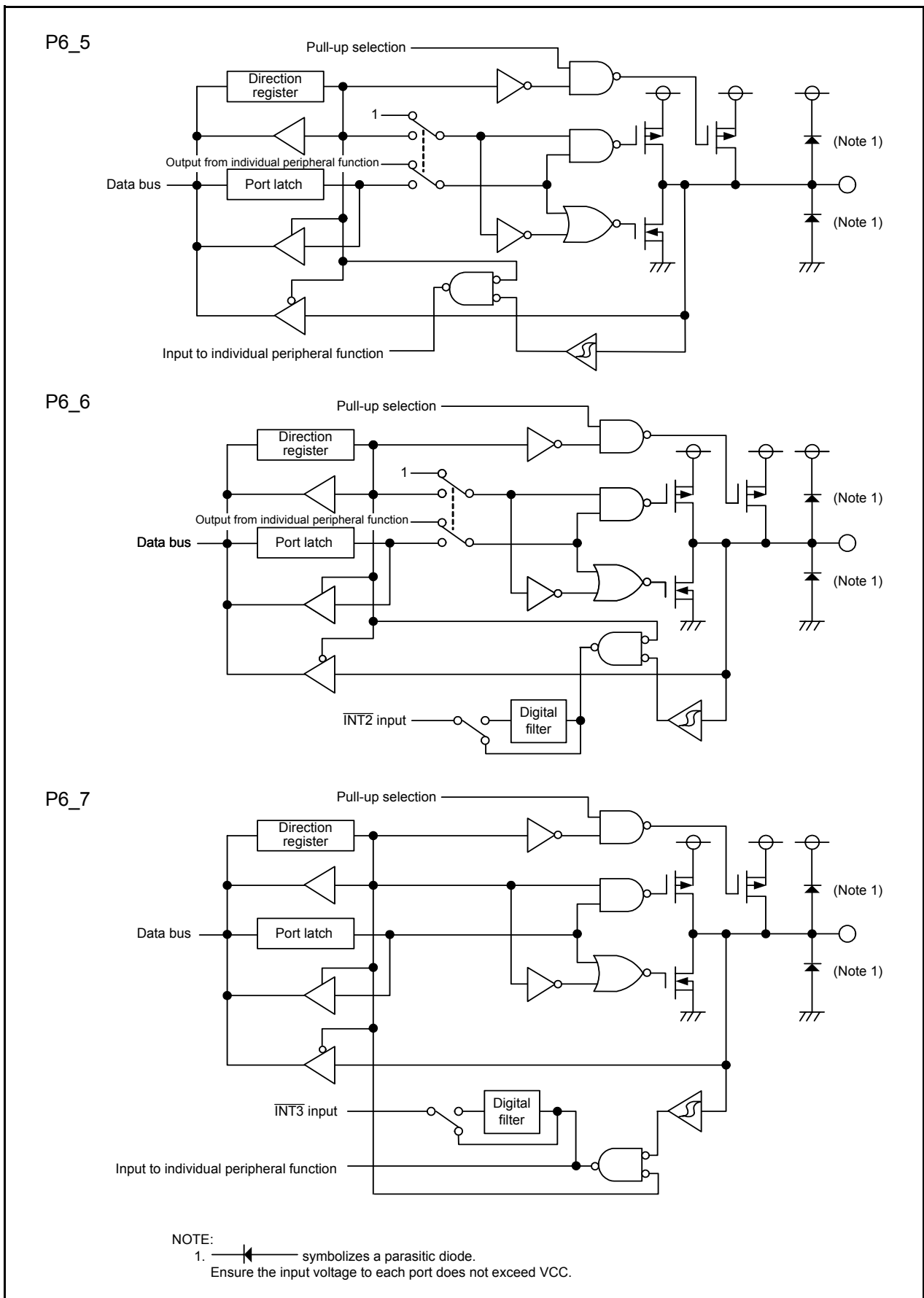


Figure 7.9 Configuration of Programmable I/O Ports (9)

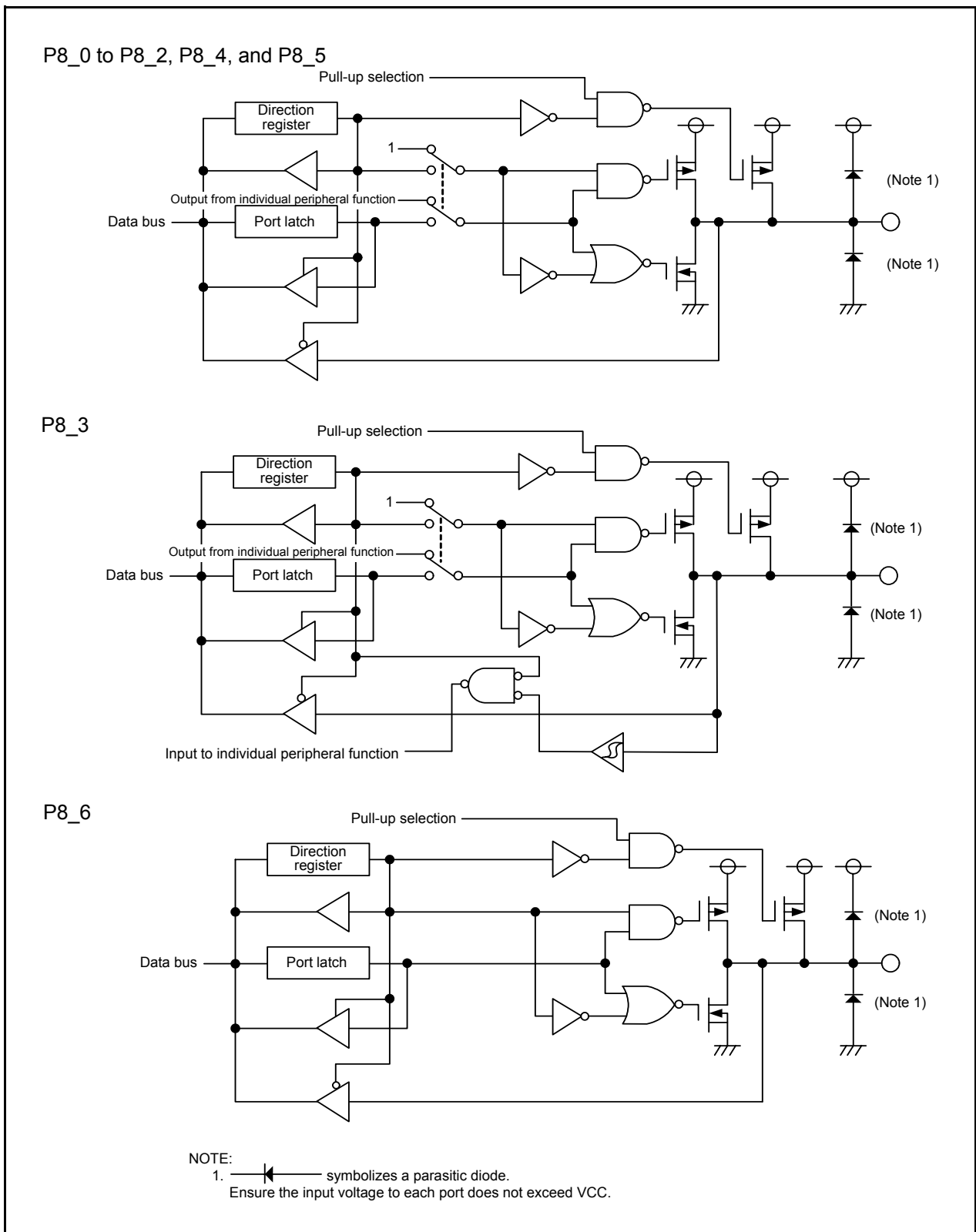
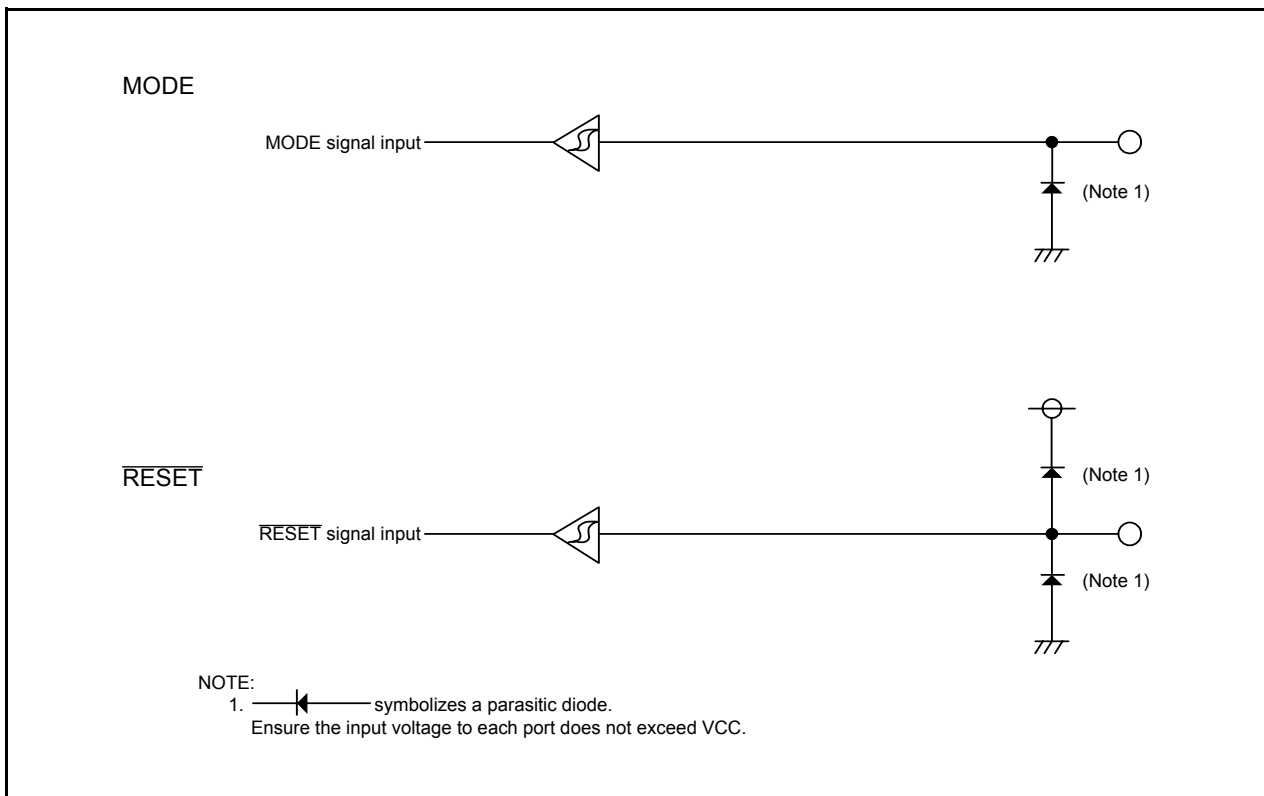


Figure 7.10 Configuration of Programmable I/O Ports (10)

**Figure 7.11 Configuration of I/O Pins**

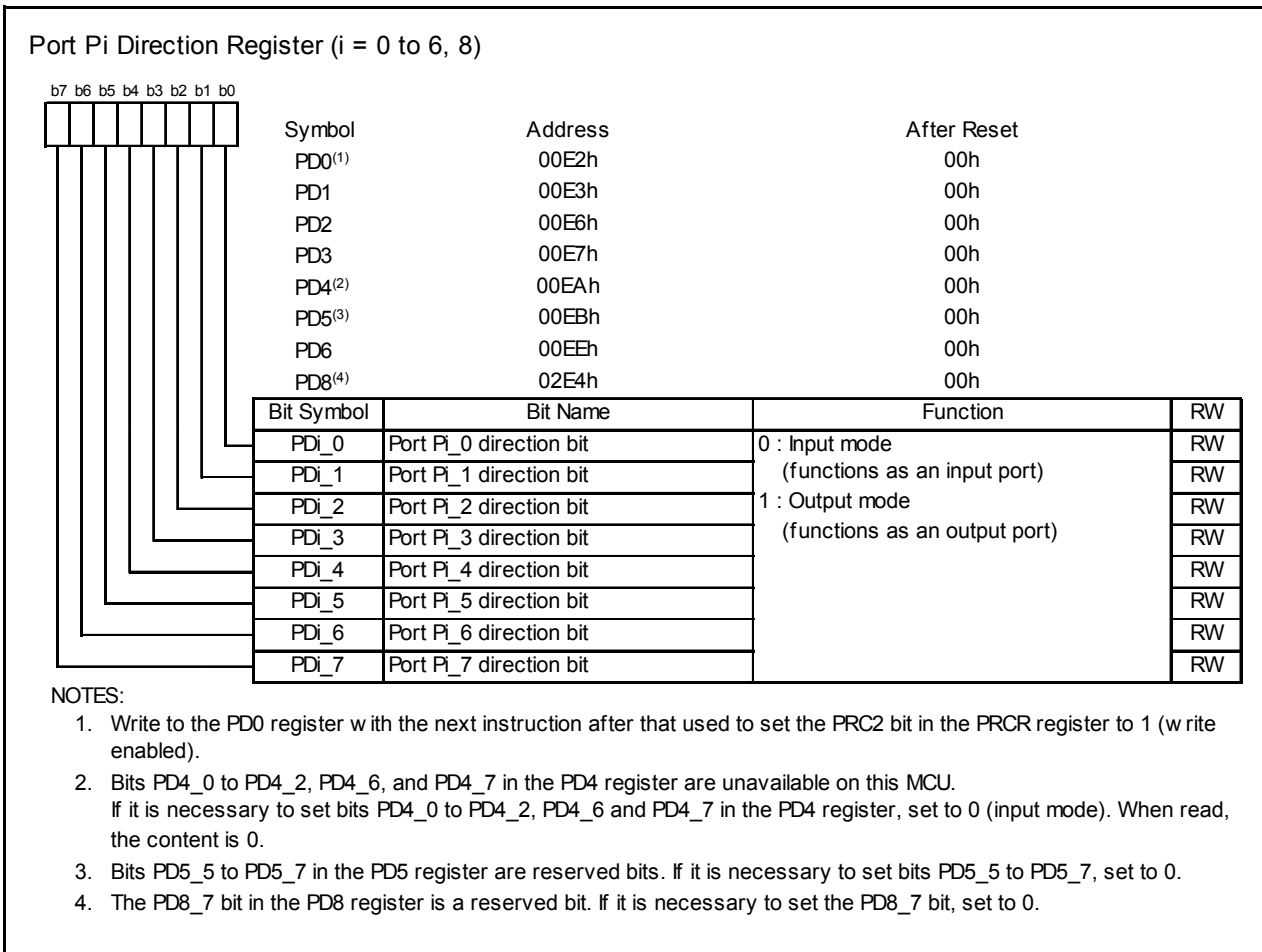


Figure 7.12 PDi (i = 0 to 6 and 8) Registers

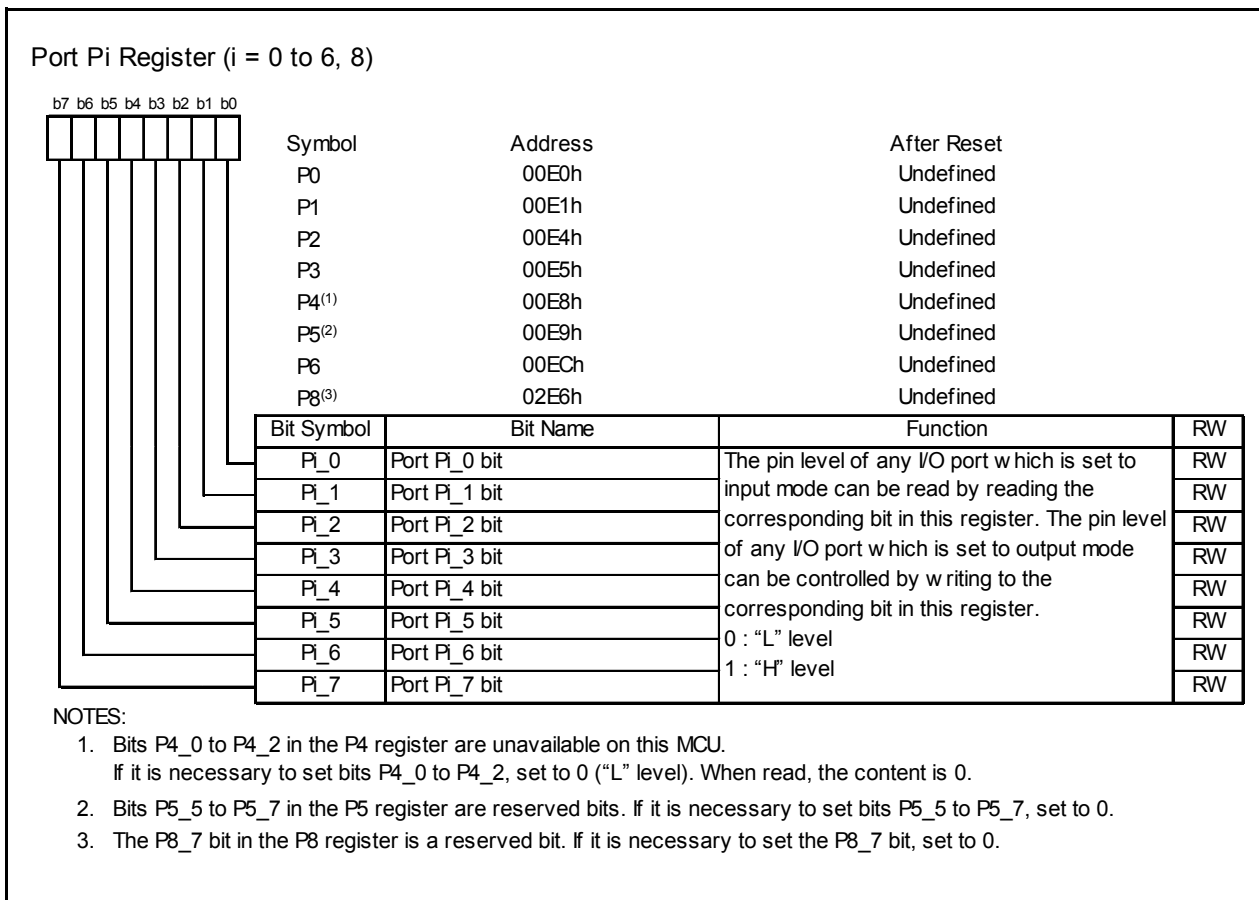


Figure 7.13 Pi (i = 0 to 6 and 8) Registers

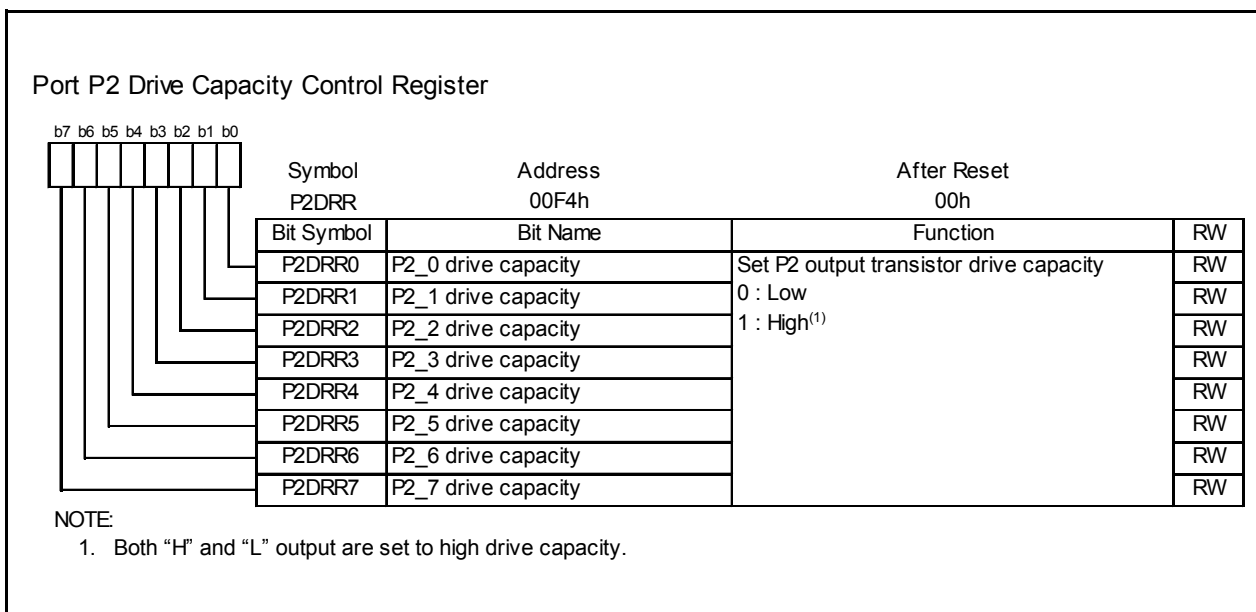


Figure 7.14 P2DRR Register

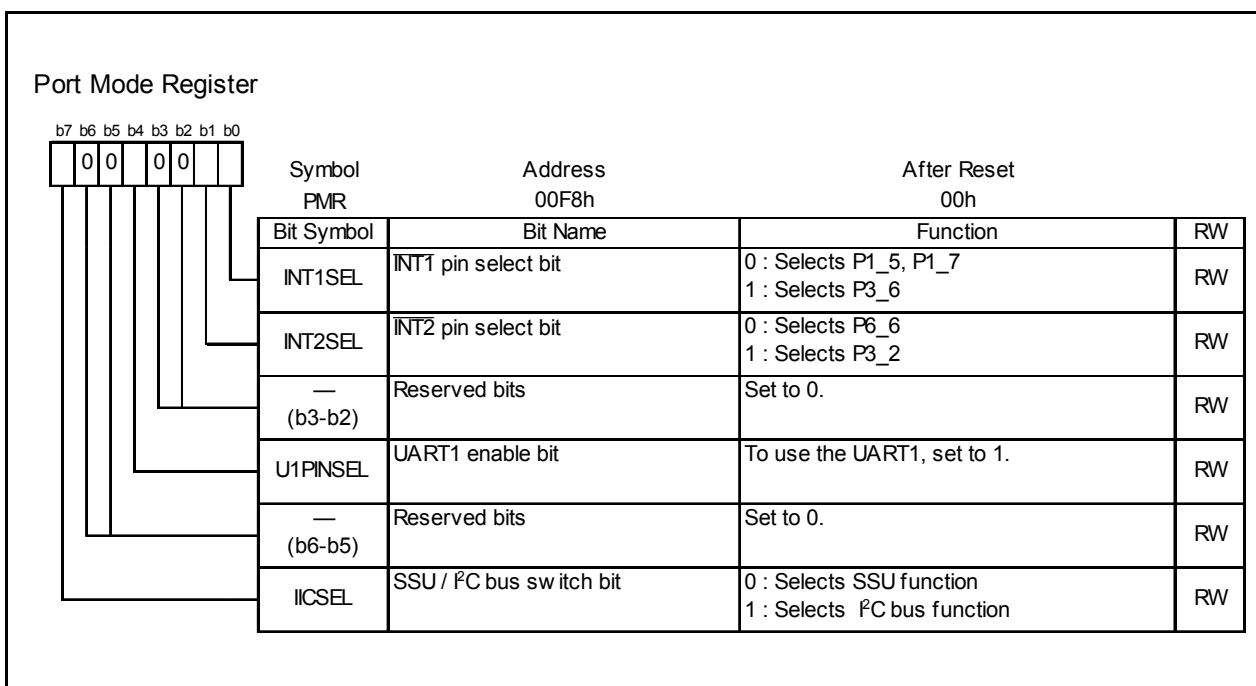


Figure 7.15 PMR Register

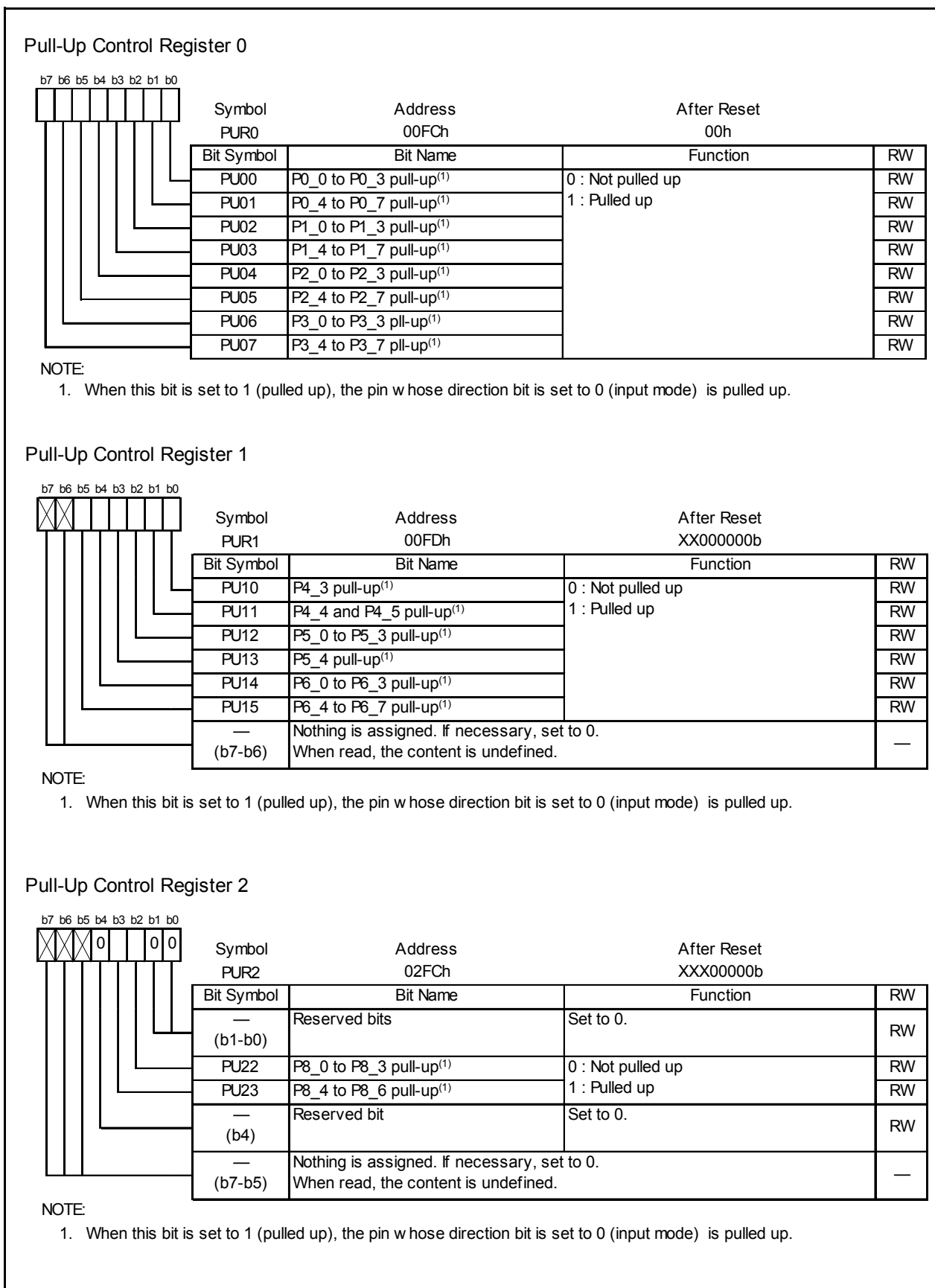


Figure 7.16 Registers PUR0, PUR1, and PUR2

7.4 Port settings

Tables 7.4 to 7.65 list the port settings.

Table 7.4 Port P0_0/AN7

| Register | PD0 | ADCON0 | | | ADCON2 | | Function |
|---------------|-------|--------|-----|-----|---------|---------|---------------------------|
| Bit | PD0_0 | CH2 | CH1 | CH0 | ADGSEL1 | ADGSEL0 | |
| Setting Value | 0 | X | X | X | X | X | Input port ⁽¹⁾ |
| | 1 | X | X | X | X | X | Output port |
| | 0 | 1 | 1 | 1 | 0 | 0 | A/D converter input (AN7) |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU00 bit in the PUR0 register to 1.

Table 7.5 Port P0_1/AN6

| Register | PD0 | ADCON0 | | | ADCON2 | | Function |
|---------------|-------|--------|-----|-----|---------|---------|---------------------------|
| Bit | PD0_1 | CH2 | CH1 | CH0 | ADGSEL1 | ADGSEL0 | |
| Setting Value | 0 | X | X | X | X | X | Input port ⁽¹⁾ |
| | 1 | X | X | X | X | X | Output port |
| | 0 | 1 | 1 | 0 | 0 | 0 | A/D converter input (AN6) |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU00 bit in the PUR0 register to 1.

Table 7.6 Port P0_2/AN5

| Register | PD0 | ADCON0 | | | ADCON2 | | Function |
|---------------|-------|--------|-----|-----|---------|---------|---------------------------|
| Bit | PD0_2 | CH2 | CH1 | CH0 | ADGSEL1 | ADGSEL0 | |
| Setting Value | 0 | X | X | X | X | X | Input port ⁽¹⁾ |
| | 1 | X | X | X | X | X | Output port |
| | 0 | 1 | 0 | 1 | 0 | 0 | A/D converter input (AN5) |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU00 bit in the PUR0 register to 1.

Table 7.7 Port P0_3/AN4

| Register | PD0 | ADCON0 | | | ADCON2 | | Function |
|---------------|-------|--------|-----|-----|---------|---------|---------------------------|
| Bit | PD0_3 | CH2 | CH1 | CH0 | ADGSEL1 | ADGSEL0 | |
| Setting Value | 0 | X | X | X | X | X | Input port ⁽¹⁾ |
| | 1 | X | X | X | X | X | Output port |
| | 0 | 1 | 0 | 0 | 0 | 0 | A/D converter input (AN4) |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU00 bit in the PUR0 register to 1.

Table 7.8 Port P0_4/AN3

| Register | PD0 | ADCON0 | | | ADCON2 | | Function |
|---------------|-------|--------|-----|-----|---------|---------|---------------------------|
| Bit | PD0_4 | CH2 | CH1 | CH0 | ADGSEL1 | ADGSEL0 | |
| Setting Value | 0 | X | X | X | X | X | Input port ⁽¹⁾ |
| | 1 | X | X | X | X | X | Output port |
| | 0 | 0 | 1 | 1 | 0 | 0 | A/D converter input (AN3) |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU01 bit in the PUR0 register to 1.

Table 7.9 Port P0_5/AN2/CLK1

| Register | PD0 | ADCON0 | | | ADCON2 | | PMR | U1MR | | | | U1SR | | Function | |
|---------------|-------|--------|-----|-----|---------|---------|----------|-----------------|-----------------|------|-------|----------|---------------------------|------------------------------|-------------|
| Bit | PD0_5 | CH2 | CH1 | CH0 | ADGSEL1 | ADGSEL0 | U1PINSEL | SMD2 | SMD1 | SMD0 | CKDIR | CLK1PSEL | CLK10PSEL | | |
| Setting Value | 0 | X | X | X | X | X | X | Other than 001b | | | X | X | X | Input port ⁽¹⁾ | |
| | | | | | | | 0 | X | X | X | X | X | X | | |
| | | | | | | | X | X | X | X | 1 | X | X | | |
| | 1 | X | X | X | X | X | X | X | Other than 001b | | | X | X | X | Output port |
| | | | | | | | | 0 | X | X | X | X | X | X | |
| | | | | | | | | X | X | X | X | 0 | X | X | |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | X | X | X | X | X | X | A/D converter input (AN2) | | |
| 0 | X | X | X | X | X | X | 1 | X | X | X | 1 | 0 | 1 | CLK1 (external clock) input | |
| X | X | X | X | X | X | X | 1 | 0 | 0 | 1 | 0 | | | CLK1 (internal clock) output | |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU01 bit in the PUR0 register to 1.

Table 7.10 Port P0_6/AN1/DA0

| Register | PD0 | ADCON0 | | | ADCON2 | | DACON | Function |
|---------------|-------|--------|-----|-----|---------|---------|-------|----------------------------|
| Bit | PD0_6 | CH2 | CH1 | CH0 | ADGSEL1 | ADGSEL0 | DA0E | |
| Setting Value | 0 | X | X | X | X | X | X | Input port ⁽¹⁾ |
| | 1 | X | X | X | X | X | X | Output port |
| | 0 | 0 | 0 | 1 | 0 | 0 | X | A/D converter input (AN1) |
| | 0 | X | X | X | X | X | 1 | D/A converter output (DA0) |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU01 bit in the PUR0 register to 1.

Table 7.11 Port P0_7/AN0/DA1

| Register | PD0 | ADCON0 | | | ADCON2 | | DACON | Function |
|---------------|-------|--------|-----|-----|---------|---------|-------|----------------------------|
| Bit | PD0_7 | CH2 | CH1 | CH0 | ADGSEL1 | ADGSEL0 | DA1E | |
| Setting Value | 0 | X | X | X | X | X | X | Input port ⁽¹⁾ |
| | 1 | X | X | X | X | X | X | Output port |
| | 0 | 0 | 0 | 0 | 0 | 0 | X | A/D converter input (AN0) |
| | 0 | X | X | X | X | X | 1 | D/A converter output (DA1) |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU01 bit in the PUR0 register to 1.

Table 7.12 Port P1_0/KI0/AN8

| Register | PD1 | KIEN | ADCON0 | | | ADCON2 | | Function |
|---------------|-------|-------|--------|-----|-----|---------|---------|---------------------------|
| Bit | PD1_0 | KI0EN | CH2 | CH1 | CH0 | ADGSEL1 | ADGSEL0 | |
| Setting Value | 0 | X | X | X | X | X | X | Input port ⁽¹⁾ |
| | 1 | X | X | X | X | X | X | Output port |
| | 0 | 1 | X | X | X | X | X | KI0 input |
| | 0 | X | 1 | 0 | 0 | 0 | 1 | A/D converter input (AN8) |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU02 bit in the PUR0 register to 1.

Table 7.13 Port P1_1/KI1/AN9

| Register | PD1 | KIEN | ADCON0 | | | ADCON2 | | Function |
|---------------|-------|-------|--------|-----|-----|---------|---------|---------------------------|
| Bit | PD1_1 | KI1EN | CH2 | CH1 | CH0 | ADGSEL1 | ADGSEL0 | |
| Setting Value | 0 | X | X | X | X | X | X | Input port ⁽¹⁾ |
| | 1 | X | X | X | X | X | X | Output port |
| | 0 | 1 | X | X | X | X | X | KI1 input |
| | 0 | X | 1 | 0 | 1 | 0 | 1 | A/D converter input (AN9) |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU02 bit in the PUR0 register to 1.

Table 7.14 Port P1_2/KI2/AN10

| Register | PD1 | KIEN | ADCON0 | | | ADCON2 | | Function |
|---------------|-------|-------|--------|-----|-----|---------|---------|----------------------------|
| Bit | PD1_2 | KI2EN | CH2 | CH1 | CH0 | ADGSEL1 | ADGSEL0 | |
| Setting Value | 0 | X | X | X | X | X | X | Input port ⁽¹⁾ |
| | 1 | X | X | X | X | X | X | Output port |
| | 0 | 1 | X | X | X | X | X | KI2 input |
| | 0 | X | 1 | 1 | 0 | 0 | 1 | A/D converter input (AN10) |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU02 bit in the PUR0 register to 1.

Table 7.15 Port P1_3/KI3/AN11

| Register | PD1 | KIEN | ADCON0 | | | ADCON2 | | Function |
|---------------|-------|-------|--------|-----|-----|---------|---------|----------------------------|
| Bit | PD1_3 | KI3EN | CH2 | CH1 | CH0 | ADGSEL1 | ADGSEL0 | |
| Setting Value | 0 | X | X | X | X | X | X | Input port ⁽¹⁾ |
| | 1 | X | X | X | X | X | X | Output port |
| | 0 | 1 | X | X | X | X | X | KI3 input |
| | 0 | X | 1 | 1 | 1 | 0 | 1 | A/D converter input (AN11) |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU02 bit in the PUR0 register to 1.

Table 7.16 Port P1_4/TXD0

| Register | PD1 | U0MR | | | Function | |
|---------------|-------|------|------|------|---------------------------|----------------------------|
| Bit | PD1_4 | SMD2 | SMD1 | SMD0 | | |
| Setting Value | 0 | 0 | 0 | 0 | Input port ⁽¹⁾ | |
| | 1 | 0 | 0 | 0 | Output port | |
| | X | X | 0 | 0 | 1 | TXD0 output ⁽²⁾ |
| | | | 1 | 0 | 0 | |
| | | | 1 | 0 | 1 | |
| | | 1 | 1 | 0 | | |

X: 0 or 1

NOTES:

1. Pulled up by setting the PU03 bit in the PUR0 register to 1.
2. N-channel open-drain output by setting the NCH bit in the U0C0 register to 1.

Table 7.17 Port P1_5/RXD0/(TRAIO)/(INT1)

| Register | PD1 | TRAIOC | | TRAMR | | | INTEN | PMR | Function |
|---------------|-------|--------|-----------------|-----------------|-------|-------|--------|--------------------|---------------------------|
| Bit | PD1_5 | TIOSEL | TOPCR | TMOD2 | TMOD1 | TMOD0 | INT1EN | INT1SEL | |
| Setting Value | 0 | 0 | X | X | X | X | X | X | Input port ⁽¹⁾ |
| | | X | 1 | X | X | X | | | |
| | | X | X | Other than 001b | | | | | |
| | 1 | 0 | X | X | X | X | X | X | Output port |
| | | X | 1 | X | X | X | | | |
| | | X | X | Other than 001b | | | | | |
| | 0 | X | X | Other than 001b | | | X | X | RXD0 input ⁽¹⁾ |
| | | 0 | | 0 | 0 | 1 | | | |
| | 0 | 1 | X | Other than 001b | | | X | X | TRAIO input |
| 0 | 1 | X | Other than 001b | | | 1 | 0 | TRAIO/INT1 input | |
| X | 1 | 0 | 0 | 0 | 1 | X | X | TRAIO pulse output | |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU03 bit in the PUR0 register to 1.

Table 7.18 Port P1_6/CLK0

| Register | PD1 | UOMR | | | | Function |
|---------------|-------|-----------------|------|------|------------------------------|-----------------------------|
| Bit | PD1_6 | SMD2 | SMD1 | SMD0 | CKDIR | |
| Setting Value | 0 | Other than 001b | | | X | Input port ⁽¹⁾ |
| | | X | X | X | 1 | |
| | 1 | Other than 001b | | | X | Output port |
| | 0 | X | X | X | 1 | CLK0 (external clock) input |
| X | 0 | 0 | 1 | 0 | CLK0 (internal clock) output | |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU03 bit in the PUR0 register to 1.

Table 7.19 Port P1_7/TRAIO/INT1

| Register | PD1 | TRAIOC | | TRAMR | | | INTEN | PMR | Function |
|---------------|-------|--------|-------|-----------------|-------|-------|--------|---------|---------------------------|
| Bit | PD1_7 | TIOSEL | TOPCR | TMOD2 | TMOD1 | TMOD0 | INT1EN | INT1SEL | |
| Setting Value | 0 | 1 | X | X | X | X | X | X | Input port ⁽¹⁾ |
| | | X | 1 | X | X | X | | | |
| | | X | X | Other than 001b | | | | | |
| | 1 | 1 | X | X | X | X | X | X | Output port |
| | | X | 1 | X | X | X | | | |
| | | X | X | Other than 001b | | | | | |
| | 0 | 0 | X | Other than 001b | | | X | X | TRAIO input |
| | 0 | 0 | X | Other than 001b | | | 1 | 0 | TRAIO/INT1 input |
| | X | 0 | 0 | 0 | 0 | 1 | X | X | TRAIO pulse output |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU03 bit in the PUR0 register to 1.

Table 7.20 Port P2_0/TRDIOA0/TRDCLK

| Register | PD2 | TRDOER1 | TRDFCR | | | | TRDIOA0 | | | Function |
|---------------|-------|---------|--------|------|-------|------|---------|------|------|--|
| Bit | PD2_0 | EA0 | CMD1 | CMD0 | STCLK | PWM3 | IOA2 | IOA1 | IOA0 | |
| Setting Value | 0 | 1 | X | X | X | X | X | X | X | Input port ⁽¹⁾ |
| | 1 | 1 | X | X | X | X | X | X | X | Output port ⁽²⁾ |
| | 0 | X | 0 | 0 | 0 | 1 | 1 | X | X | Timer mode (input capture function) |
| | 0 | X | X | X | 1 | 1 | 0 | 0 | 0 | External clock input (TRDCLK) |
| | X | 0 | 0 | 0 | 0 | 0 | X | X | X | PWM3 mode waveform output ⁽²⁾ |
| | X | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| | | | | | | | 0 | 1 | X | |

X: 0 or 1

NOTES:

1. Pulled up by setting the PU04 bit in the PUR0 register to 1.
2. Output drive capacity high by setting the P2DRR0 bit in the P2DRR register to 1.

Table 7.21 Port P2_1/TRDIOB0

| Register | PD2 | TRDOER1 | TRDFCR | | | | TRDPMR | TRDIOA0 | | | Function |
|---------------|-------|---------|--------|------|------|-------|--------|---------|------|---|--|
| Bit | PD2_1 | EB0 | CMD1 | CMD0 | PWM3 | PWMB0 | IOB2 | IOB1 | IOB0 | | |
| Setting Value | 0 | 1 | X | X | X | X | X | X | X | Input port ⁽¹⁾ | |
| | 1 | 1 | X | X | X | X | X | X | X | Output port ⁽²⁾ | |
| | 0 | X | 0 | 0 | 1 | 0 | 1 | X | X | Timer mode (input capture function) | |
| | X | 0 | | 1 | 0 | X | X | X | X | X | Complementary PWM mode waveform output |
| | | | | 1 | 1 | | | | | | |
| | X | 0 | 0 | 1 | X | X | X | X | X | X | Reset synchronous PWM mode waveform output |
| | X | 0 | 0 | 0 | 0 | 0 | X | X | X | X | PWM3 mode waveform output ⁽²⁾ |
| | X | 0 | 0 | 0 | 1 | 1 | X | X | X | X | PWM mode waveform output ⁽²⁾ |
| X | 0 | 0 | 0 | 1 | 0 | | 0 | 0 | 1 | Timer mode waveform output (output compare function) ⁽²⁾ | |
| | | | | | | | 0 | 1 | X | | |

X: 0 or 1

NOTES:

1. Pulled up by setting the PU04 bit in the PUR0 register to 1.
2. Output drive capacity high by setting the P2DRR1 bit in the P2DRR register to 1.

Table 7.22 Port P2_2/TRDIOC0

| Register | PD2 | TRDOER1 | TRDFCR | | | | TRDPMR | TRDIOC0 | | | Function |
|---------------|-------|---------|--------|------|------|-------|--------|---------|------|-------------------------------------|---|
| Bit | PD2_2 | EC0 | CMD1 | CMD0 | PWM3 | PWMC0 | IOC2 | IOC1 | IOC0 | | |
| Setting Value | 0 | 1 | X | X | X | X | X | X | X | Input port ⁽¹⁾ | |
| | 1 | 1 | X | X | X | X | X | X | X | Output port ⁽²⁾ | |
| | 0 | X | 0 | 0 | 1 | 0 | 1 | X | X | Timer mode (input capture function) | |
| | X | 0 | | 1 | 0 | X | X | X | X | X | Complementary PWM mode waveform output ⁽²⁾ |
| | | | | 1 | 1 | | | | | | |
| | X | 0 | 0 | 1 | X | X | X | X | X | X | Reset synchronous PWM mode waveform output ⁽²⁾ |
| | X | 0 | 0 | 0 | 0 | 1 | 1 | X | X | X | PWM mode waveform output ⁽²⁾ |
| | X | 0 | 0 | 0 | 1 | 0 | | 0 | 0 | 1 | Timer mode waveform output (output compare function) ⁽²⁾ |
| 0 | | | | | | | | 1 | X | | |

X: 0 or 1

NOTES:

1. Pulled up by setting the PU04 bit in the PUR0 register to 1.
2. Output drive capacity high by setting the P2DRR2 bit in the P2DRR register to 1.

Table 7.23 Port P2_3/TRDIOD0

| Register | PD2 | TRDOER1 | TRDFCR | | | TRDPMR | TRDIORC0 | | | Function | |
|---------------|-------|---------|--------|------|------|--------|----------|------|------|---|---|
| Bit | PD2_3 | ED0 | CMD1 | CMD0 | PWM3 | PWMD0 | IOD2 | IOD1 | IOD0 | | |
| Setting Value | 0 | 1 | X | X | X | X | X | X | X | Input port ⁽¹⁾ | |
| | 1 | 1 | X | X | X | X | X | X | X | Output port ⁽²⁾ | |
| | 0 | X | 0 | 0 | 1 | 0 | 1 | X | X | Timer mode (input capture function) | |
| | X | 0 | 1 | 0 | X | X | X | X | X | X | Complementary PWM mode waveform output ⁽²⁾ |
| | | | 1 | 1 | | | | | | | |
| | X | 0 | 0 | 1 | X | X | X | X | X | Reset synchronous PWM mode waveform output ⁽²⁾ | |
| | X | 0 | 0 | 0 | 1 | 1 | X | X | X | PWM mode waveform output ⁽²⁾ | |
| X | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | X | Timer mode waveform output (output compare function) ⁽²⁾ | |
| | | | | | | 0 | 1 | | | | |

X: 0 or 1

NOTES:

1. Pulled up by setting the PU04 bit in the PUR0 register to 1.
2. Output drive capacity high by setting the P2DRR3 bit in the P2DRR register to 1.

Table 7.24 Port P2_4/TRDIOA1

| Register | PD2 | TRDOER1 | TRDFCR | | | TRDIOA1 | | | Function | |
|---------------|-------|---------|--------|------|------|---------|------|------|---|---|
| Bit | PD2_4 | EA1 | CMD1 | CMD0 | PWM3 | IOA2 | IOA1 | IOA0 | | |
| Setting Value | 0 | 1 | X | X | X | X | X | X | Input port ⁽¹⁾ | |
| | 1 | 1 | X | X | X | X | X | X | Output port ⁽²⁾ | |
| | 0 | X | 0 | 0 | 1 | 1 | X | X | Timer mode (input capture function) | |
| | X | 0 | 1 | 0 | X | X | X | X | X | Complementary PWM mode waveform output ⁽²⁾ |
| | | | 1 | 1 | | | | | | |
| | X | 0 | 0 | 1 | X | X | X | X | Reset synchronous PWM mode waveform output ⁽²⁾ | |
| | X | 0 | 0 | 0 | 1 | 0 | 0 | 1 | X | Timer mode waveform output (output compare function) ⁽²⁾ |
| 0 | | | | | | 1 | | | | |

X: 0 or 1

NOTES:

1. Pulled up by setting the PU05 bit in the PUR0 register to 1.
2. Output drive capacity high by setting the P2DRR4 bit in the P2DRR register to 1.

Table 7.25 Port P2_5/TRDIOB1

| Register | PD2 | TRDOER1 | TRDFCR | | | TRDPMR | TRDIOA1 | | | Function | |
|---------------|-------|---------|--------|------|------|--------|---------|------|------|---|---|
| Bit | PD2_5 | EB1 | CMD1 | CMD0 | PWM3 | PWMB1 | IOB2 | IOB1 | IOB0 | | |
| Setting Value | 0 | 1 | X | X | X | X | X | X | X | Input port ⁽¹⁾ | |
| | 1 | 1 | X | X | X | X | X | X | X | Output port ⁽²⁾ | |
| | 0 | X | 0 | 0 | 1 | 0 | 1 | X | X | Timer mode (input capture function) | |
| | X | 0 | 1 | 0 | X | X | X | X | X | X | Complementary PWM mode waveform output ⁽²⁾ |
| | | | 1 | 1 | | | | | | | |
| | X | 0 | 0 | 1 | X | X | X | X | X | Reset synchronous PWM mode waveform output ⁽²⁾ | |
| | X | 0 | 0 | 0 | 1 | 1 | X | X | X | PWM mode waveform output ⁽²⁾ | |
| X | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | X | Timer mode waveform output (output compare function) ⁽²⁾ | |
| | | | | | | 0 | 1 | | | | |

X: 0 or 1

NOTES:

1. Pulled up by setting the PU05 bit in the PUR0 register to 1.
2. Output drive capacity high by setting the P2DRR5 bit in the P2DRR register to 1.

Table 7.26 Port P2_6/TRDIOC1

| Register | PD2 | TRDOER1 | TRDFCR | | | TRDPMR | TRDIOC1 | | | Function | |
|---------------|-------|---------|--------|------|------|--------|---------|------|------|---|---|
| Bit | PD2_6 | EC1 | CMD1 | CMD0 | PWM3 | PWMC1 | IOC2 | IOC1 | IOC0 | | |
| Setting Value | 0 | 1 | X | X | X | X | X | X | X | Input port ⁽¹⁾ | |
| | 1 | 1 | X | X | X | X | X | X | X | Output port ⁽²⁾ | |
| | 0 | X | 0 | 0 | 1 | 0 | 1 | X | X | Timer mode (input capture function) | |
| | X | 0 | 1 | 0 | X | X | X | X | X | X | Complementary PWM mode waveform output ⁽²⁾ |
| | | | 1 | 1 | | | | | | | |
| | X | 0 | 0 | 1 | X | X | X | X | X | Reset synchronous PWM mode waveform output ⁽²⁾ | |
| | X | 0 | 0 | 0 | 1 | 1 | X | X | X | PWM mode waveform output ⁽²⁾ | |
| X | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | X | Timer mode waveform output (output compare function) ⁽²⁾ | |
| | | | | | | 0 | 1 | | | | |

X: 0 or 1

NOTES:

1. Pulled up by setting the PU05 bit in the PUR0 register to 1.
2. Output drive capacity high by setting the P2DRR6 bit in the P2DRR register to 1.

Table 7.27 Port P2_7/TRDIOD1

| Register | PD2 | TRDOER1 | TRDFCR | | | TRDPMR | TRDIOC1 | | | Function | |
|---------------|-------|---------|--------|------|------|--------|---------|------|------|---|---|
| Bit | PD2_7 | ED1 | CMD1 | CMD0 | PWM3 | PWMD1 | IOD2 | IOD1 | IOD0 | | |
| Setting Value | 0 | 1 | X | X | X | X | X | X | X | Input port ⁽¹⁾ | |
| | 1 | 1 | X | X | X | X | X | X | X | Output port ⁽²⁾ | |
| | 0 | X | 0 | 0 | 1 | 0 | 1 | X | X | Timer mode (input capture function) | |
| | X | 0 | 1 | 0 | X | X | X | X | X | X | Complementary PWM mode waveform output ⁽²⁾ |
| | | | 1 | 1 | | | | | | | |
| | X | 0 | 0 | 1 | X | X | X | X | X | Reset synchronous PWM mode waveform output ⁽²⁾ | |
| | X | 0 | 0 | 0 | 1 | 1 | X | X | X | PWM mode waveform output ⁽²⁾ | |
| X | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | X | Timer mode waveform output (output compare function) ⁽²⁾ | |
| | | | | | | 0 | 1 | | | | |

X: 0 or 1

NOTES:

1. Pulled up by setting the PU05 bit in the PUR0 register to 1.
2. Output drive capacity high by setting the P2DRR7 bit in the P2DRR register to 1.

Table 7.28 Port P3_0/TRAO

| Register | PD3 | TRAIOC | Function |
|---------------|-------|--------|---------------------------|
| Bit | PD3_0 | TOENA | |
| Setting Value | 0 | 0 | Input port ⁽¹⁾ |
| | 1 | 0 | Output port |
| | X | 1 | TRAO output |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU06 bit in the PUR0 register to 1.

Table 7.29 Port P3_1/TRBO

| Register | PD3 | TRBMR | | TRBIOC | Function |
|---------------|-------|----------------|-------|--------|---------------------------|
| Bit | PD3_1 | TMOD1 | TMOD0 | TOCNT | |
| Setting Value | 0 | 0 | 0 | X | Input port ⁽¹⁾ |
| | 1 | 0 | 0 | X | Output port |
| | X | 01b | | 1 | |
| | X | Other than 00b | | 0 | TRBO output |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU06 bit in the PUR0 register to 1.

Table 7.30 Port P3_2($\overline{\text{INT2}}$)

| Register | PD3 | INTEN | PMR | Function |
|---------------|-------|--------|---------|--------------------------------|
| Bit | PD3_2 | INT2EN | INT2SEL | |
| Setting Value | 0 | X | X | Input port ⁽¹⁾ |
| | 1 | X | X | Output port |
| | 0 | 1 | 1 | $\overline{\text{INT2}}$ input |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU06 bit in the PUR0 register to 1.

Table 7.31 Port P3_3/SSI

| Register | PD3 | Clock Synchronous Serial I/O with Chip Select (Refer to Table 16.4 Association between Communication Modes and I/O Pins.) | | PMR | Function |
|---------------|-------|--|-------------------|--------|----------------------------|
| Bit | PD3_3 | SSI output control | SSI input control | IICSEL | |
| Setting Value | 0 | 0 | 0 | 0 | Input port ⁽¹⁾ |
| | | X | X | 1 | |
| | 1 | 0 | 0 | 0 | Output port ⁽²⁾ |
| | | X | X | 1 | |
| | X | 0 | 1 | 0 | SSI input |
| | X | 1 | 0 | 0 | SSI output ⁽²⁾ |

X: 0 or 1

NOTES:

1. Pulled up by setting the PU06 bit in the PUR0 register to 1.
2. N-channel open-drain output by setting the SOOS bit in the SSMR2 register to 1 when this pin functions as output.

Table 7.32 Port P3_4/ $\overline{\text{SDA}}$ / $\overline{\text{SCS}}$

| Register | PD3 | SSMR2 | | PMR | ICCR1 | Function | |
|---------------|-------|-------|------|--------|-------|-------------------------------|---|
| Bit | PD3_4 | CSS1 | CSS0 | IICSEL | ICE | | |
| Setting Value | 0 | 0 | 0 | 0 | X | Input port ⁽¹⁾ | |
| | 0 | 0 | 0 | X | 0 | | |
| | 1 | 0 | 0 | 0 | X | Output port ⁽²⁾ | |
| | 1 | 0 | 0 | X | 0 | | |
| | X | 0 | 1 | 0 | X | $\overline{\text{SCS}}$ input | |
| | X | 1 | 0 | 0 | 0 | X | $\overline{\text{SCS}}$ output ⁽²⁾ |
| | | | 1 | 1 | | | |
| X | X | X | X | 1 | 1 | SDA input/output | |

X: 0 or 1

NOTES:

1. Pulled up by setting the PU07 bit in the PUR0 register to 1.
2. N-channel open-drain output by setting the CSOS bit in the SSMR2 register to 1 when this pin functions as output.

Table 7.33 Port P3_5/SCL/SSCK

| Register | PD3 | Clock Synchronous Serial I/O with Chip Select (Refer to Table 16.4 Association between Communication Modes and I/O Pins.) | | PMR | ICCR1 | Function | |
|---------------|-------|---|--------------------|--------|-------|----------------------------|----------------------------|
| Bit | PD3_5 | SSCK output control | SSCK input control | IICSEL | ICE | | |
| Setting Value | 0 | 0 | 0 | 0 | X | Input port ⁽¹⁾ | |
| | 0 | 0 | 0 | X | 0 | | |
| | 1 | 0 | 0 | 0 | X | Output port ⁽²⁾ | |
| | 1 | 0 | 0 | X | 0 | | |
| | X | 0 | 1 | 0 | 0 | 0 | SSCK input |
| | X | 1 | 0 | 0 | 0 | 0 | SSCK output ⁽²⁾ |
| | X | 1 | 0 | 0 | 1 | 1 | SCL input/output |

X: 0 or 1

NOTES:

1. Pulled up by setting the PU07 bit in the PUR0 register to 1.
2. N-channel open-drain output by setting the SCKOS bit in the SSMR2 register to 1 when this pin functions as output.

Table 7.34 Port P3_6/ $\overline{\text{INT1}}$

| Register | PD3 | INTEN | PMR | Function |
|---------------|-------|--------|---------|--------------------------------|
| Bit | PD3_6 | INT1EN | INT1SEL | |
| Setting Value | 0 | X | X | Input port ⁽¹⁾ |
| | 1 | X | X | Output port |
| | 0 | 1 | 1 | $\overline{\text{INT1}}$ input |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU07 bit in the PUR0 register to 1.

Table 7.35 Port P3_7/SSO

| Register | PD3 | Clock Synchronous Serial I/O with Chip Select (Refer to Table 16.4 Association between Communication Modes and I/O Pins.) | | SSMR2 | PMR | Function |
|---------------|-------|---|-------------------|-------|--------|--|
| Bit | PD3_7 | SSO output control | SSO input control | SOOS | IICSEL | |
| Setting Value | 0 | 0 | 0 | X | 0 | Input port ⁽¹⁾ |
| | | X | X | | 1 | |
| | 1 | 0 | 0 | 0 | 0 | Output port |
| | | X | X | | 1 | |
| | X | 0 | 1 | 0 | 0 | SSO input |
| | X | 1 | 0 | 0 | 0 | SSO output (CMOS output) |
| X | 1 | 0 | 0 | 1 | 0 | SSO output (N-channel open-drain output) |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU07 bit in the PUR0 register to 1.

Table 7.36 VREF

| Register | ADCON1 | Function |
|---------------|--------|----------------------|
| Bit | VCUT | |
| Setting Value | 0 | Not the pin function |
| | 1 | VREF input |

Table 7.37 Port P4_3/XCIN

| Register | PD4 | CM0 | CM1 | | Circuit specifications | | Function | |
|---------------|-------|------|------|------|------------------------|-------------------|---|-----------------------------|
| Bit | PD4_3 | CM04 | CM10 | CM12 | Oscillation buffer | Feedback resistor | | |
| Setting Value | 0 | 0 | X | X | OFF | OFF | Input port ⁽¹⁾ | |
| | 1 | 0 | X | X | OFF | OFF | Output port | |
| | X | 1 | 0 | 0 | ON | ON | XCIN-XCOUT oscillation (on-chip feedback resistor enabled) | |
| | X | 1 | 0 | 1 | ON | OFF | XCIN-XCOUT oscillation (on-chip feedback resistor disabled) | |
| | X | 1 | 1 | 0 | 1 | OFF | ON | XCIN-XCOUT oscillation stop |
| | | | | 1 | 1 | OFF | OFF | |
| X | 1 | 0 | 0 | 1 | ON | ON | External XCIN input | |
| | | | 1 | 1 | ON | OFF | | |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU10 bit in the PUR1 register to 1.

Table 7.38 Port P4_4/XCOUT

| Register | PD4 | CM0 | CM1 | | Circuit specifications | | Function | |
|---------------|-------|------|------|------|------------------------|-------------------|--|-----------------------------|
| Bit | PD4_4 | CM04 | CM10 | CM12 | Oscillation buffer | Feedback resistor | | |
| Setting Value | 0 | 0 | X | X | OFF | OFF | Input port ⁽¹⁾ | |
| | 1 | 0 | X | X | OFF | OFF | Output port | |
| | X | 1 | 0 | 0 | ON | ON | XCIN-XCOUT oscillation (on-chip feedback resistor enabled) | |
| | X | 1 | 0 | 1 | ON | OFF | XCIN-XCOUT oscillation (on-chip feedback resistor disabled) | |
| | X | 1 | 1 | 0 | 1 | OFF | ON | XCIN-XCOUT oscillation stop |
| | | | | 1 | 1 | OFF | OFF | |
| X | 1 | 0 | 0 | 1 | ON | ON | External XCOUT output (inverted output of XCIN) ⁽²⁾ | |
| | | | 1 | 1 | ON | OFF | | |

X: 0 or 1

NOTES:

1. Pulled up by setting the PU11 bit in the PUR1 register to 1.
2. Since the XCIN-XCOUT oscillation buffer operates with internal step-down power, the XCOUT output level cannot be used as the CMOS level signal directly.

Table 7.39 Port P4_5/ $\overline{\text{INT0}}$

| Register | PD4 | INTEN | Function |
|---------------|-------|--------|--------------------------------|
| Bit | PD4_5 | INT0EN | |
| Setting Value | 0 | X | Input port ⁽¹⁾ |
| | 1 | X | Output port |
| | 0 | 1 | $\overline{\text{INT0}}$ input |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU11 bit in the PUR1 register to 1.

Table 7.40 Port P4_6/XIN

| Register | CM1 | | CM0 | Circuit specifications | | Function | |
|---------------|-----|------|------|------------------------|--------------------|----------|---------------------------|
| | Bit | CM13 | CM10 | CM05 | Oscillation buffer | | Feedback resistor |
| Setting Value | 0 | | X | X | OFF | OFF | Input port |
| | 1 | | 0 | 0 | ON | ON | XIN-XOUT oscillation |
| | 1 | | 0 | 1 | OFF | ON | External XIN input |
| | 1 | | 1 | 0 | OFF | OFF | XIN-XOUT oscillation stop |
| | 1 | | 1 | 1 | OFF | OFF | XIN-XOUT oscillation stop |

X: 0 or 1

Table 7.41 Port P4_7/XOUT

| Register | CM1 | | CM0 | Circuit specifications | | Function | |
|---------------|-----|------|------|------------------------|--------------------|----------|---------------------------|
| | Bit | CM13 | CM10 | CM05 | Oscillation buffer | | Feedback resistor |
| Setting Value | 0 | | X | X | OFF | OFF | Input port |
| | 1 | | 0 | 0 | ON | ON | XIN-XOUT oscillation |
| | 1 | | 0 | 1 | OFF | ON | XOUT is "H" pull-up |
| | 1 | | 1 | 0 | OFF | OFF | XIN-XOUT oscillation stop |
| | 1 | | 1 | 1 | OFF | OFF | XIN-XOUT oscillation stop |

X: 0 or 1

Table 7.42 Port P5_0/TRCCLK

| Register | PD5 | TRCCR1 | | | Function |
|---------------|-------|-----------------|------|------|---------------------------|
| Bit | PD5_0 | TCK2 | TCK1 | TCK0 | |
| Setting Value | 0 | Other than 101b | | | Input port ⁽¹⁾ |
| | 1 | Other than 101b | | | Output port |
| | 0 | 1 | 0 | 1 | TRCCLK input |

NOTE:

1. Pulled up by setting the PU12 bit in the PUR1 register to 1.

Table 7.43 Port P5_1/TRCIOA/TRCTRG

| Register | PD5 | Timer RC Setting | Function |
|---------------|-------|---|---------------------------|
| Bit | PD5_1 | – | |
| Setting Value | 0 | Other than TRCIOA usage conditions | Input port ⁽¹⁾ |
| | 1 | | Output port |
| | X | Refer to Table 7.44 TRCIOA Pin Setting | TRCIOA output |
| | 0 | | TRCIOA input |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU12 bit in the PUR1 register to 1.

Table 7.44 TRCIOA Pin Setting

| Register | TRCOER | TRCMR | TRCIOR0 | | | TRCCR2 | | Function |
|------------------|--------|-------|---------|------|------|--------|-------|---|
| Bit | EA | PWM2 | IOA2 | IOA1 | IOA0 | TCEG1 | TCEG2 | |
| Setting value | 0 | 1 | 0 | 0 | 1 | X | X | Timer waveform output (output compare function) |
| | | | 0 | 1 | X | X | X | |
| | 1 | 1 | 1 | X | X | X | X | Timer mode (input capture function) |
| | | | | | | | | |
| | 1 | 0 | X | X | X | 1 | X | PWM2 mode TRCTRG input |
| | | | | | | | | |
| Other than above | | | | | | | | Other than TRCIOA usage conditions |

X: 0 or 1

Table 7.45 Port P5_2/TRCIOB

| Register | PD5 | Timer RC Setting | Function |
|---------------|-------|---|---------------------------|
| Bit | PD5_2 | – | |
| Setting Value | 0 | Other than TRCIOB usage conditions | Input port ⁽¹⁾ |
| | 1 | | Output port |
| | X | Refer to Table 7.46 TRCIOB Pin Setting | TRCIOB output |
| | 0 | | TRCIOB input |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU12 bit in the PUR1 register to 1.

Table 7.46 TRCIOB Pin Setting

| Register | TRCOER | TRCMR | | TRCIOR0 | | | Function | |
|------------------|--------|-------|------|---------|------|------|------------------------------------|---|
| Bit | EB | PWM2 | PWMB | IOB2 | IOB1 | IOB0 | | |
| Setting value | 0 | 0 | X | X | X | X | PWM2 mode waveform output | |
| | 0 | 1 | 1 | X | X | X | PWM mode waveform output | |
| | 0 | 1 | 0 | 0 | 0 | 0 | 1 | Timer waveform output (output compare function) |
| | | | | | 0 | 1 | X | |
| | 1 | 1 | 0 | 1 | X | X | X | Timer mode (input capture function) |
| | | | | | | | | |
| Other than above | | | | | | | Other than TRCIOB usage conditions | |

X: 0 or 1

Table 7.47 Port P5_3/TRCIOC

| Register | PD5 | Timer RC Setting | Function |
|---------------|-------|---|---------------------------|
| Bit | PD5_3 | – | |
| Setting Value | 0 | Other than TRCIOC usage conditions | Input port ⁽¹⁾ |
| | 1 | | Output port |
| | X | Refer to Table 7.48 TRCIOC Pin Setting | TRCIOC output |
| | 0 | | TRCIOC input |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU12 bit in the PUR1 register to 1.

Table 7.48 TRCIOC Pin Setting

| Register | TRCOER | TRCMR | | TRCIOR1 | | | Function |
|------------------|--------|-------|------|---------|------|------|---|
| Bit | EC | PWM2 | PWMC | IOC2 | IOC1 | IOC0 | |
| Setting value | 0 | 1 | 1 | X | X | X | PWM mode waveform output |
| | 0 | 1 | 0 | 0 | 0 | 1 | Timer waveform output (output compare function) |
| | | | | 0 | 1 | X | |
| | 0 | 1 | 0 | 1 | X | X | Timer mode (input capture function) |
| | 1 | | | | | | |
| Other than above | | | | | | | Other than TRCIOC usage conditions |

X: 0 or 1

Table 7.49 Port P5_4/TRCIOD

| Register | PD5 | Timer RC Setting | Function |
|---------------|-------|---|---------------------------|
| Bit | PD5_4 | – | |
| Setting Value | 0 | Other than TRCIOD usage conditions | Input port ⁽¹⁾ |
| | 1 | | Output port |
| | X | Refer to Table 7.50 TRCIOD Pin Setting | TRCIOD output |
| | 0 | | TRCIOD input |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU13 bit in the PUR1 register to 1.

Table 7.50 TRCIOD Pin Setting

| Register | TRCOER | TRCMR | | TRCIOR1 | | | Function |
|------------------|--------|-------|------|---------|------|------|---|
| Bit | EC | PWM2 | PWMD | IOD2 | IOD1 | IOD0 | |
| Setting value | 0 | 1 | 1 | X | X | X | PWM mode waveform output |
| | 0 | 1 | 0 | 0 | 0 | 1 | Timer waveform output (output compare function) |
| | | | | 0 | 1 | X | |
| | 0 | 1 | 0 | 1 | X | X | Timer mode (input capture function) |
| | 1 | | | | | | |
| Other than above | | | | | | | Other than TRCIOD usage conditions |

X: 0 or 1

Table 7.51 Port P6_0/TREO

| Register | PD6 | TRECR1 | Function |
|---------------|-------|--------|---------------------------|
| Bit | PD6_0 | TOENA | |
| Setting Value | 0 | 0 | Input port ⁽¹⁾ |
| | 1 | 0 | Output port |
| | X | 1 | TREO output |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU14 bit in the PUR1 register to 1.

Table 7.52 Port P6_1

| Register | PD6 | Function |
|---------------|-------|---------------------------|
| Bit | PD6_1 | |
| Setting Value | 0 | Input port ⁽¹⁾ |
| | 1 | Output port |

NOTE:

1. Pulled up by setting the PU14 bit in the PUR1 register to 1.

Table 7.53 Port P6_2

| Register | PD6 | Function |
|---------------|-------|---------------------------|
| Bit | PD6_2 | |
| Setting Value | 0 | Input port ⁽¹⁾ |
| | 1 | Output port |

NOTE:

1. Pulled up by setting the PU14 bit in the PUR1 register to 1.

Table 7.54 Port P6_3/TXD2

| Register | PD6 | U2MR | | | U2C0 | Function | |
|---------------|-------|------|------|------|------|---------------------------|---|
| Bit | PD6_3 | SMD2 | SMD1 | SMD0 | NCH | | |
| Setting Value | 0 | 0 | 0 | 0 | X | Input port ⁽¹⁾ | |
| | | X | X | X | | | |
| | 1 | 0 | 0 | 0 | X | Output port | |
| | | X | X | X | | | |
| | X | X | 0 | 0 | 1 | 0 | TXD2 output (CMOS output) |
| | | | 1 | 0 | 0 | | |
| | | | 1 | 0 | 1 | | |
| | | | 1 | 1 | 0 | | |
| | X | X | 0 | 0 | 1 | 1 | TXD2 output (N-channel open-drain output) |
| | | | 1 | 0 | 0 | | |
| | | | 1 | 0 | 1 | | |
| | | | 1 | 1 | 0 | | |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU14 bit in the PUR1 register to 1.

Table 7.55 Port P6_4/RXD2

| Register | PD6 | Function |
|---------------|-------|---------------------------|
| Bit | PD6_4 | |
| Setting Value | 0 | Input port ⁽¹⁾ |
| | 1 | Output port |
| | 0 | RXD2 input ⁽¹⁾ |

NOTE:

1. Pulled up by setting the PU15 bit in the PUR1 register to 1.

Table 7.56 Port P6_5/(CLK1)/CLK2

| Register | PD6 | PMR | U1MR | | | | U1SR | | Function |
|---------------|-------|----------|-----------------|------|------|-------|-----------|------------------------------|------------------------------|
| Bit | PD6_5 | U1PINSEL | SMD2 | SMD1 | SMD0 | CKDIR | CLK11PSEL | CLK10PSEL | |
| Setting Value | 0 | X | Other than 001b | | | X | X | X | Input port ⁽¹⁾ |
| | | 0 | X | X | X | X | X | X | |
| | | X | X | X | X | 1 | X | X | |
| | 1 | X | Other than 001b | | | X | X | X | Output port |
| | | 0 | X | X | X | | X | X | |
| | | X | X | X | X | 0 | X | X | |
| | 0 | 1 | X | X | X | 1 | 1 | 0 | CLK1 (external clock) input |
| | X | 1 | 0 | 0 | 1 | 0 | | | CLK1 (internal clock) output |
| 0 | X | X | X | X | 1 | 0 | X | CLK2 (external clock) input | |
| X | X | 0 | 0 | 1 | 0 | | | CLK2 (internal clock) output | |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU15 bit in the PUR1 register to 1.

Table 7.57 Port P6_6/ $\overline{\text{INT2}}$ /TXD1

| Register | PD6 | PMR | U1MR | | | U1C0 | INTEN | PMR | Function |
|---------------|-------|----------|------|------|------|------|--------|---------|------------------------------------|
| Bit | PD6_6 | U1PINSEL | SMD2 | SMD1 | SMD0 | NCH | INT2EN | INT2SEL | |
| Setting Value | 0 | X | 0 | 0 | 0 | X | X | X | Input port ⁽¹⁾ |
| | | 0 | X | X | X | | | | |
| | 1 | X | 0 | 0 | 0 | X | X | X | Output port |
| | | 0 | X | X | X | | | | |
| | 0 | X | X | X | X | X | 1 | 0 | $\overline{\text{INT2}}$ input |
| | X | 1 | 0 | 0 | 1 | 0 | X | X | TXD1 output (CMOS output) |
| | | | 1 | 0 | 0 | | | | |
| | | | 1 | 0 | 1 | | | | |
| | | | 1 | 1 | 0 | | | | |
| | X | 1 | 0 | 0 | 1 | 1 | X | X | TXD1 output (N-channel open-drain) |
| | | | 1 | 0 | 0 | | | | |
| | | | 1 | 0 | 1 | | | | |
| 1 | | | 1 | 0 | | | | | |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU15 bit in the PUR1 register to 1.

Table 7.58 Port P6_7/ $\overline{\text{INT3}}$ /RXD1

| Register | PD6 | PMR | INTEN | Function |
|---------------|-------|----------|--------|--------------------------------|
| Bit | PD6_7 | U1PINSEL | INT3EN | |
| Setting Value | 0 | X | X | Input port ⁽¹⁾ |
| | 1 | X | X | Output port |
| | 0 | X | 1 | $\overline{\text{INT3}}$ input |
| | 0 | 1 | X | RXD1 input ⁽¹⁾ |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU15 bit in the PUR1 register to 1.

Table 7.59 Port P8_0/TRFO00

| Register | PD8 | TRFOUT | P8 | Function |
|---------------|-------|---------|------|---------------------------|
| Bit | PD8_0 | TRFOUT0 | P8_0 | |
| Setting Value | 0 | 0 | X | Input port ⁽¹⁾ |
| | 1 | 0 | X | Output port |
| | X | 1 | 0 | |
| | X | 1 | 1 | TRFO00 output |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU22 bit in the PUR2 register to 1.

Table 7.60 Port P8_1/TRFO01

| Register | PD8 | TRFOUT | P8 | Function |
|---------------|-------|---------|------|---------------------------|
| Bit | PD8_1 | TRFOUT1 | P8_1 | |
| Setting Value | 0 | 0 | X | Input port ⁽¹⁾ |
| | 1 | 0 | X | Output port |
| | X | 1 | 0 | |
| | X | 1 | 1 | TRFO01 output |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU22 bit in the PUR2 register to 1.

Table 7.61 Port P8_2/TRFO02

| Register | PD8 | TRFOUT | P8 | Function |
|---------------|-------|---------|------|---------------------------|
| Bit | PD8_2 | TRFOUT2 | P8_2 | |
| Setting Value | 0 | 0 | X | Input port ⁽¹⁾ |
| | 1 | 0 | X | Output port |
| | X | 1 | 0 | |
| | X | 1 | 1 | TRFO02 output |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU22 bit in the PUR2 register to 1.

Table 7.62 Port P8_3/TRFO10/TRFI

| Register | PD8 | TRFOUT | P8 | Function |
|---------------|-------|---------|------|---------------------------|
| Bit | PD8_3 | TRFOUT3 | P8_3 | |
| Setting Value | 0 | 0 | X | Input port ⁽¹⁾ |
| | 1 | 0 | X | Output port |
| | X | 1 | 0 | |
| | X | 1 | 1 | TRFO02 output |
| | 0 | 0 | X | TRFI input |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU22 bit in the PUR2 register to 1.

Table 7.63 Port P8_4/TRFO11

| Register | PD8 | TRFOUT | P8 | Function |
|---------------|-------|---------|------|---------------------------|
| Bit | PD8_4 | TRFOUT4 | P8_4 | |
| Setting Value | 0 | 0 | X | Input port ⁽¹⁾ |
| | 1 | 0 | X | Output port |
| | X | 1 | 0 | |
| | X | 1 | 1 | TRFO11 output |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU23 bit in the PUR2 register to 1.

Table 7.64 Port P8_5/TRFO12

| Register | PD8 | TRFOUT | P8 | Function |
|---------------|-------|---------|------|---------------------------|
| Bit | PD8_5 | TRFOUT5 | P8_5 | |
| Setting Value | 0 | 0 | X | Input port ⁽¹⁾ |
| | 1 | 0 | X | Output port |
| | X | 1 | 0 | |
| | X | 1 | 1 | TRFO12output |

X: 0 or 1

NOTE:

1. Pulled up by setting the PU23 bit in the PUR2 register to 1.

Table 7.65 Port P8_6

| Register | PD8 | Function |
|---------------|-------|---------------------------|
| Bit | PD8_6 | |
| Setting Value | 0 | Input port ⁽¹⁾ |
| | 1 | Output port |

NOTE:

1. Pulled up by setting the PU23 bit in the PUR2 register to 1.

7.5 Unassigned Pin Handling

Table 7.66 lists Unassigned Pin Handling.

Table 7.66 Unassigned Pin Handling

| Pin Name | Connection |
|---|--|
| Ports P0 to P3, P4_3 to P4_5, P5_0 to P5_4, P6, P8_0 to P8_6 | <ul style="list-style-type: none"> • After setting to input mode, connect each pin to VSS via a resistor (pull-down) or connect each pin to VCC via a resistor (pull-up).(2) • After setting to output mode, leave these pins open.(1,2) |
| Ports P4_6, P4_7 | Connect to VCC via a pull-up resistor(2) |
| VREF | Connect to VCC |
| RESET ⁽³⁾ | Connect to VCC via a pull-up resistor(2) |

NOTES:

1. If these ports are set to output mode and left open, they remain in input mode until they are switched to output mode by a program. The voltage level of these pins may be undefined and the power current may increase while the ports remain in input mode. The content of the direction registers may change due to noise or program runaway caused by noise. In order to enhance program reliability, the program should periodically repeat the setting of the direction registers.
2. Connect these unassigned pins to the MCU using the shortest wire length (2 cm or less) possible.
3. When the power-on reset function is in use.

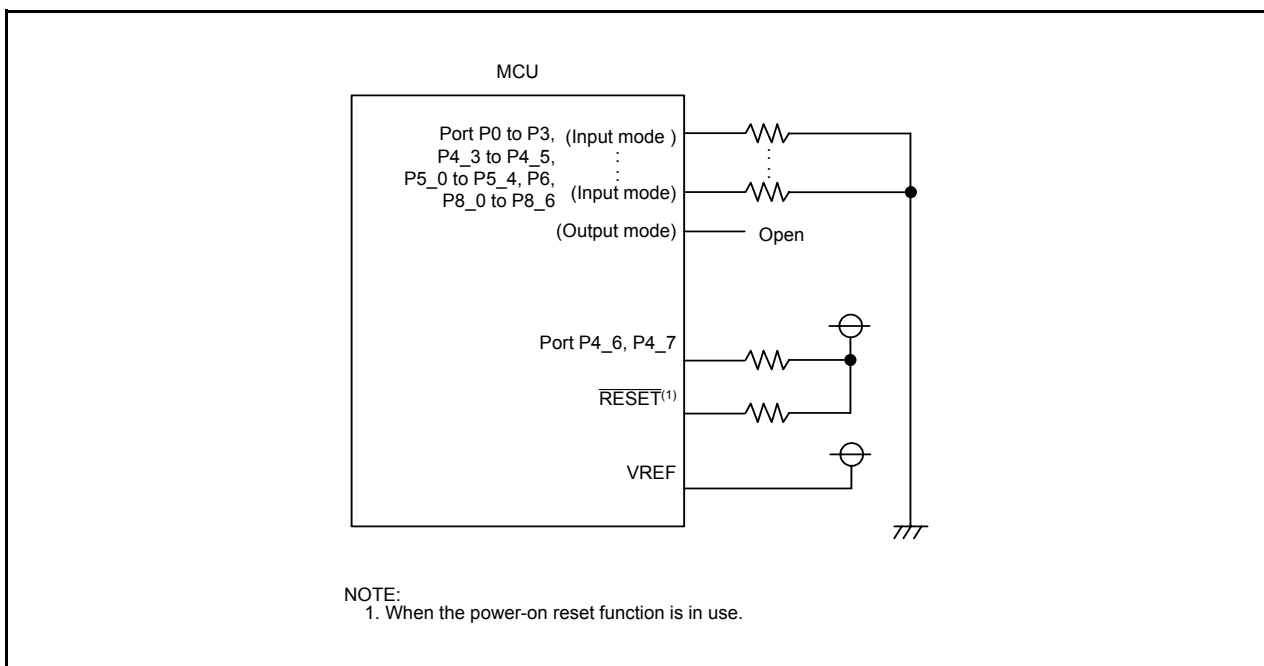


Figure 7.17 Unassigned Pin Handling

8. Processor Mode

8.1 Processor Modes

Single-chip mode can be selected as the processor mode.

Table 8.1 lists Features of Processor Mode. Figure 8.1 shows the PM0 Register and Figure 8.2 shows the PM1 Register.

Table 8.1 Features of Processor Mode

| Processor Mode | Accessible Areas | Pins Assignable as I/O Port Pins |
|------------------|---------------------------------|--|
| Single-chip mode | SFR, internal RAM, internal ROM | All pins are I/O ports or peripheral function I/O pins |

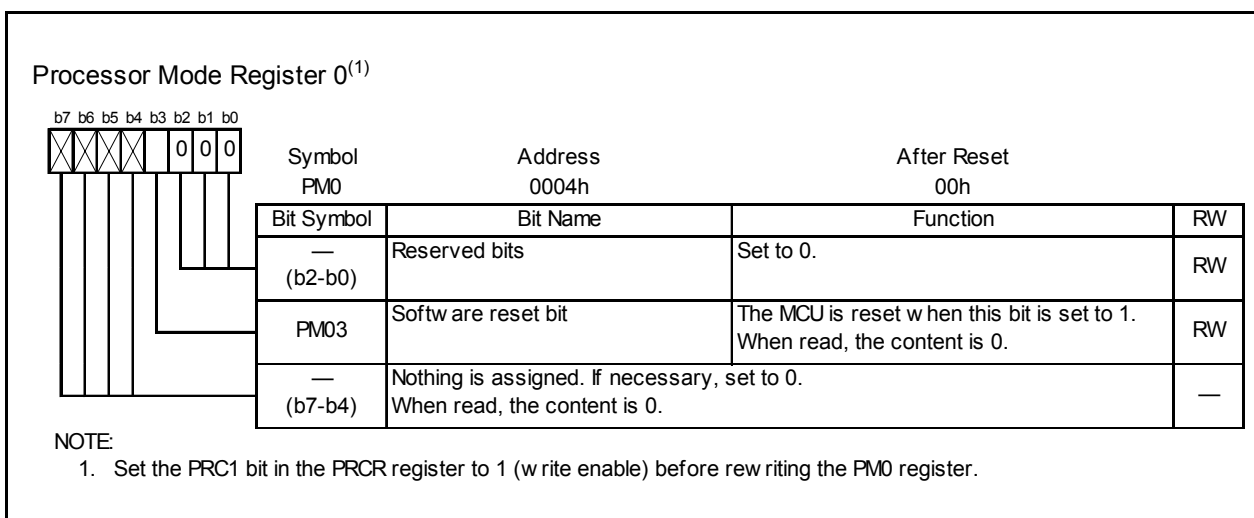


Figure 8.1 PM0 Register

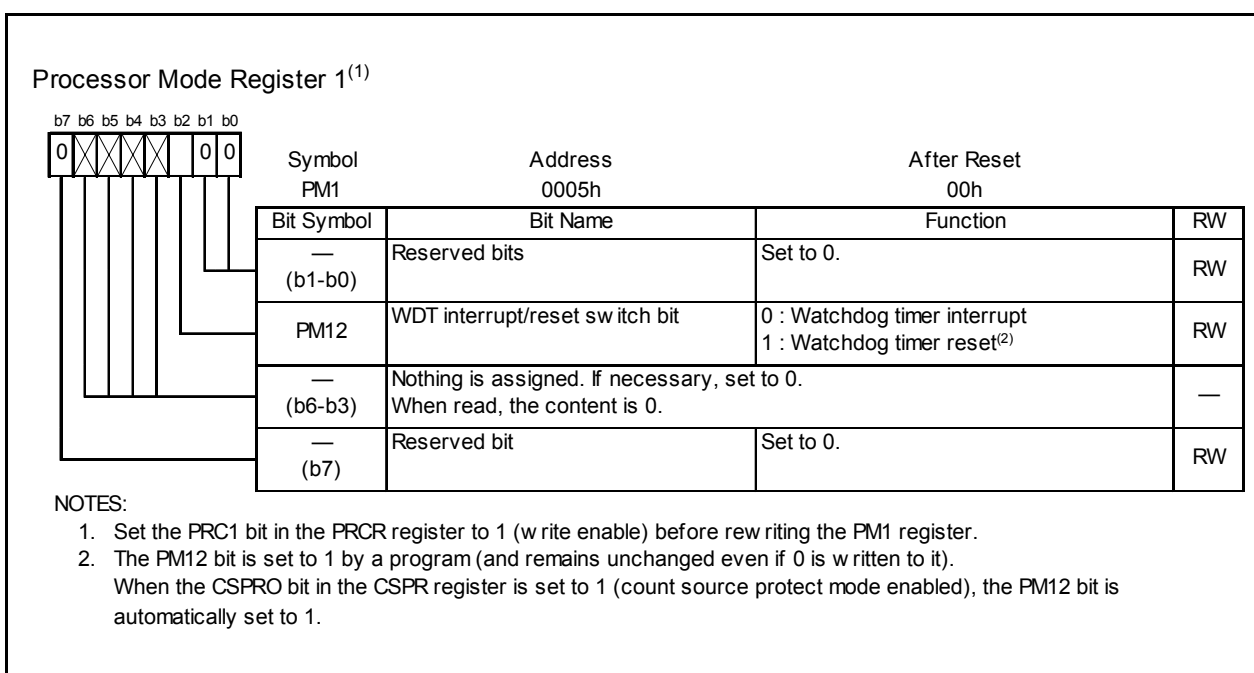


Figure 8.2 PM1 Register

9. Bus

The bus cycles differ when accessing ROM/RAM, and when accessing SFR.

Table 9.1 lists Bus Cycles by Access Space of the R8C/2A Group and Table 9.2 lists Bus Cycles by Access Space of the R8C/2B Group.

ROM/RAM and SFR are connected to the CPU by an 8-bit bus. When accessing in word (16-bit) units, these areas are accessed twice in 8-bit units.

Table 9.3 lists Access Units and Bus Operations and Table 9.4 lists Access Units and Bus Operations of SFR (address 0200h to 02FFh).

Table 9.1 Bus Cycles by Access Space of the R8C/2A Group

| Access Area | Bus Cycle |
|------------------------------|-----------------------|
| SFR (address 0000h to 01FFh) | 2 cycles of CPU clock |
| SFR (address 0200h to 02FFh) | 3 cycles of CPU clock |
| ROM/RAM | 1 cycle of CPU clock |

Table 9.2 Bus Cycles by Access Space of the R8C/2B Group

| Access Area | Bus Cycle |
|---|-----------------------|
| SFR (address 0000h to 01FFh)/Data flash | 2 cycles of CPU clock |
| SFR (address 0200h to 02FFh) | 3 cycles of CPU clock |
| Program ROM/RAM | 1 cycle of CPU clock |

Table 9.3 Access Units and Bus Operations



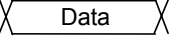

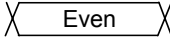


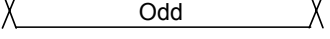


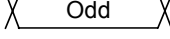


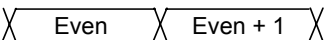


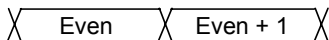


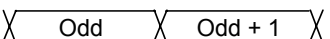


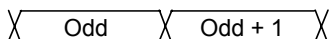


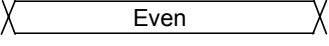


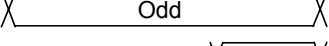
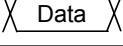

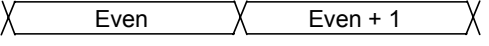

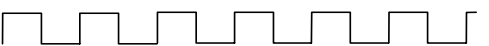


| Area | SFR (address 0000h to 01FFh), data flash | ROM (program ROM), RAM |
|-----------------------------|--|--|
| Even address Byte access | CPU clock  Address  Data  | CPU clock  Address  Data  |
| Odd address Byte access | CPU clock  Address  Data  | CPU clock  Address  Data  |
| Even address Word access | CPU clock  Address  Data  | CPU clock  Address  Data  |
| Odd address Word access | CPU clock  Address  Data  | CPU clock  Address  Data  |

Table 9.4 Access Units and Bus Operations of SFR (address 0200h to 02FFh)

| Area | SFR (address 0200h to 02FFh) |
|-----------------------------|--|
| Even address Byte access | CPU clock  Address  Data  |
| Odd address Byte access | CPU clock  Address  Data  |
| Even address Word access | CPU clock  Address  Data  |
| Odd address Word access | CPU clock  Address  Data  |

However, only following SFRs are connected with the 16-bit bus:

Timer RC: registers TRC, TRCGRA, TRCGRB, TRCGRC, and TRCGRD

Timer RD: registers TRDi (i = 0,1), TRDGRAi, TRDGRBi, TRDGRCi, and TRDGRDi

Therefore, they are accessed once in 16-bit units. The bus operation is the same as “Area: SFR, data flash, even address byte access” in Table 9.3 Access Units and Bus Operations, and 16-bit data is accessed at a time.

10. Clock Generation Circuit

The clock generation circuit has:

- XIN clock oscillation circuit
- XCIN clock oscillation circuit
- Low-speed on-chip oscillator
- High-speed on-chip oscillator

Table 10.1 lists Specifications of Clock Generation Circuit. Figure 10.1 shows a Clock Generation Circuit. Figure 10.2 shows a Peripheral Function Clock. Figures 10.3 to 10.9 show clock associated registers. Figure 10.10 shows a Procedure for Enabling Reduced Internal Power Consumption Using VCA20 bit.

Table 10.1 Specifications of Clock Generation Circuit

| Item | XIN Clock Oscillation Circuit | XCIN Clock Oscillation Circuit | On-Chip Oscillator | |
|------------------------------------|--|--|--|--|
| | | | High-Speed On-Chip Oscillator | Low-Speed On-Chip Oscillator |
| Applications | <ul style="list-style-type: none"> • CPU clock source • Peripheral function clock source | <ul style="list-style-type: none"> • CPU clock source • Timer RA and timer RE clock source | <ul style="list-style-type: none"> • CPU clock source • Peripheral function clock source • CPU and peripheral function clock sources when XIN clock stops oscillating | <ul style="list-style-type: none"> • CPU clock source • Peripheral function clock source • CPU and peripheral function clock sources when XIN clock stops oscillating |
| Clock frequency | 0 to 20 MHz | 32.768 kHz | Approx. 40 MHz ⁽⁴⁾ | Approx. 125 kHz |
| Connectable oscillator | <ul style="list-style-type: none"> • Ceramic resonator • Crystal oscillator | • Crystal oscillator | – | – |
| Oscillator connect pins | XIN, XOUT ⁽¹⁾ | XCIN, XCOU ⁽²⁾ | – ⁽¹⁾ | – ⁽¹⁾ |
| Oscillation stop, restart function | Usable | Usable | Usable | Usable |
| Oscillator status after reset | Stop | Stop | Stop | Oscillate |
| Others | Externally generated clock can be input ⁽³⁾ | <ul style="list-style-type: none"> • Externally generated clock can be input • On-chip feedback resistor R_f (connected/not connected, selectable) | – | – |

NOTES:

1. These pins can be used as P4_6 or P4_7 when using the on-chip oscillator clock as the CPU clock while the XIN clock oscillation circuit is not used.
2. These pins can be used as P4_3 and P4_4 when using the XIN clock oscillation circuit and on-chip oscillator clock for a CPU clock while the XCIN clock oscillation circuit is not used.
3. Set the CM05 bit in the CM0 register to 1 (XIN clock stopped) and the CM13 bit in the CM1 register to 1 (XIN-XOUT pin) when an external clock is input.
4. The clock frequency is automatically set to up to 20 MHz by a divider when using the high-speed on-chip oscillator as the CPU clock source.

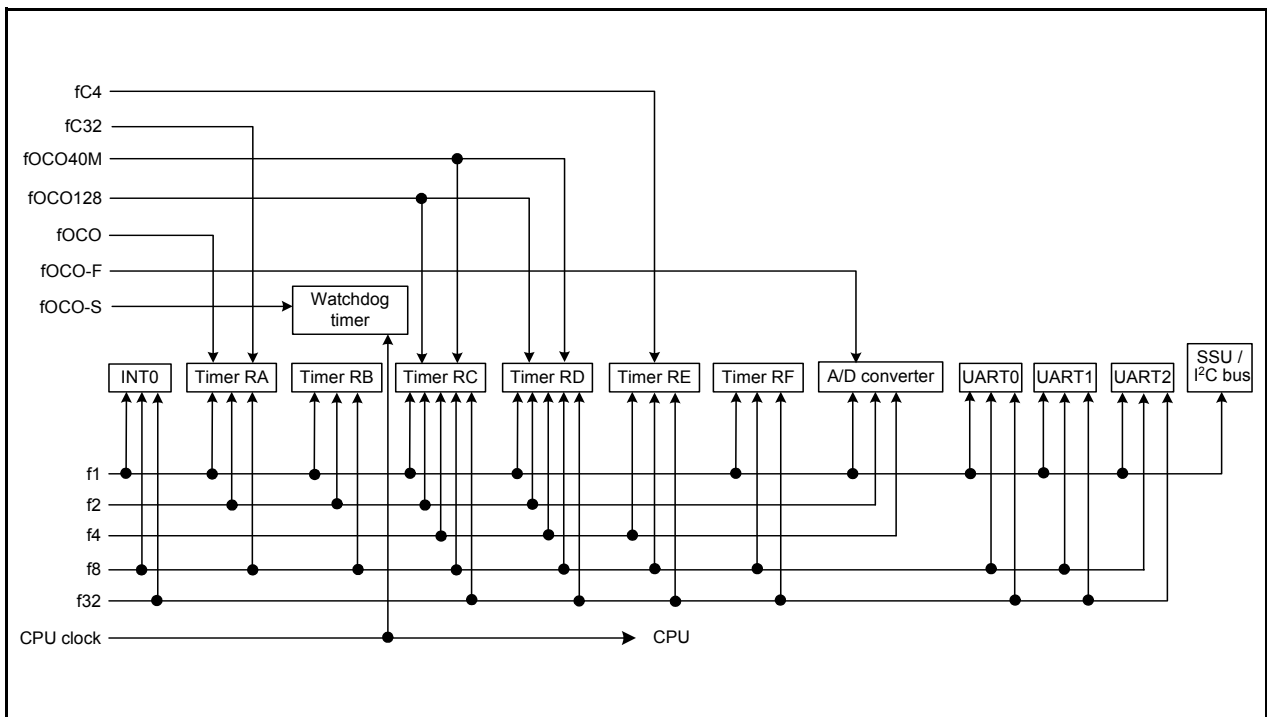


Figure 10.2 Peripheral Function Clock

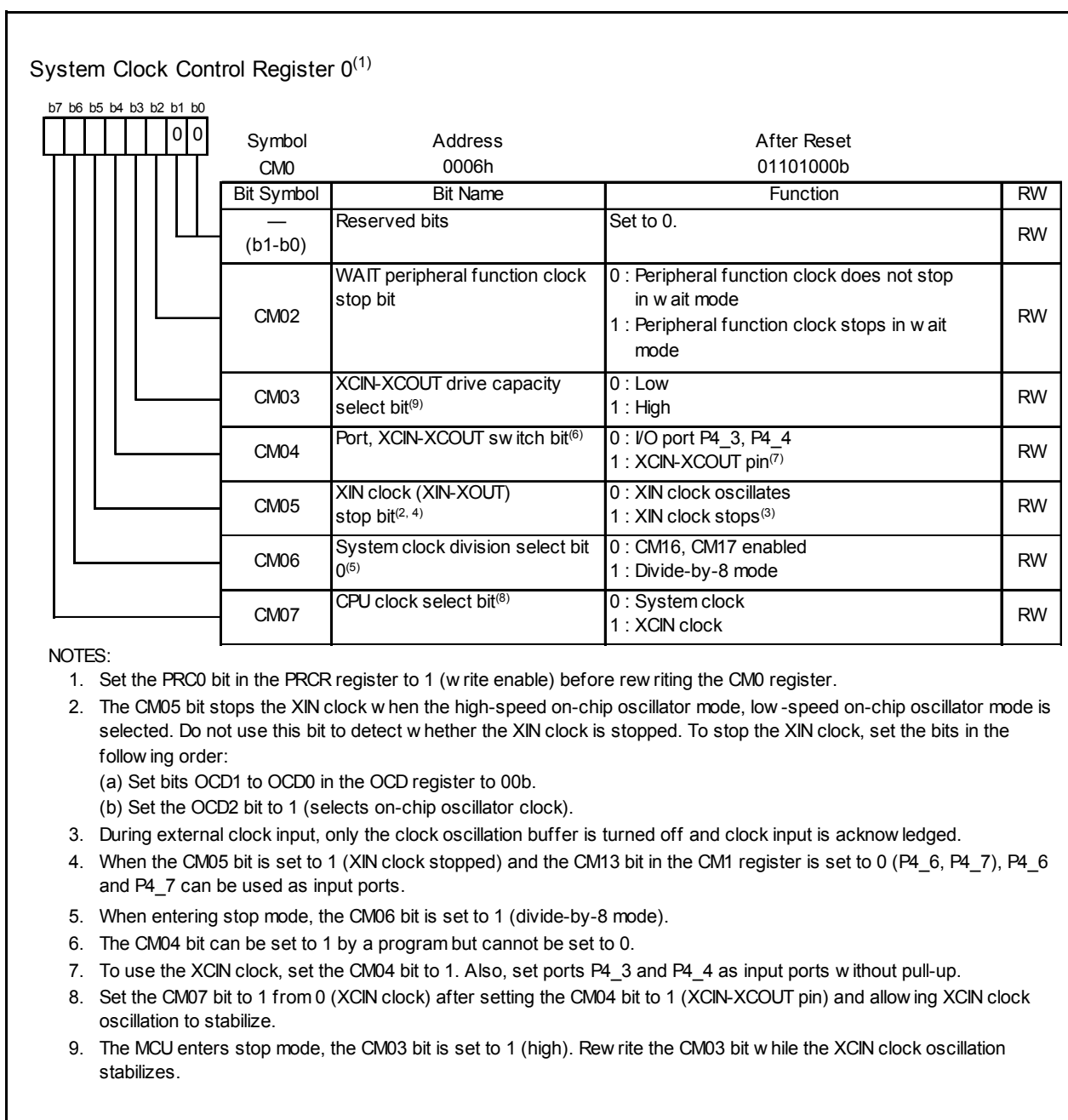


Figure 10.3 CM0 Register

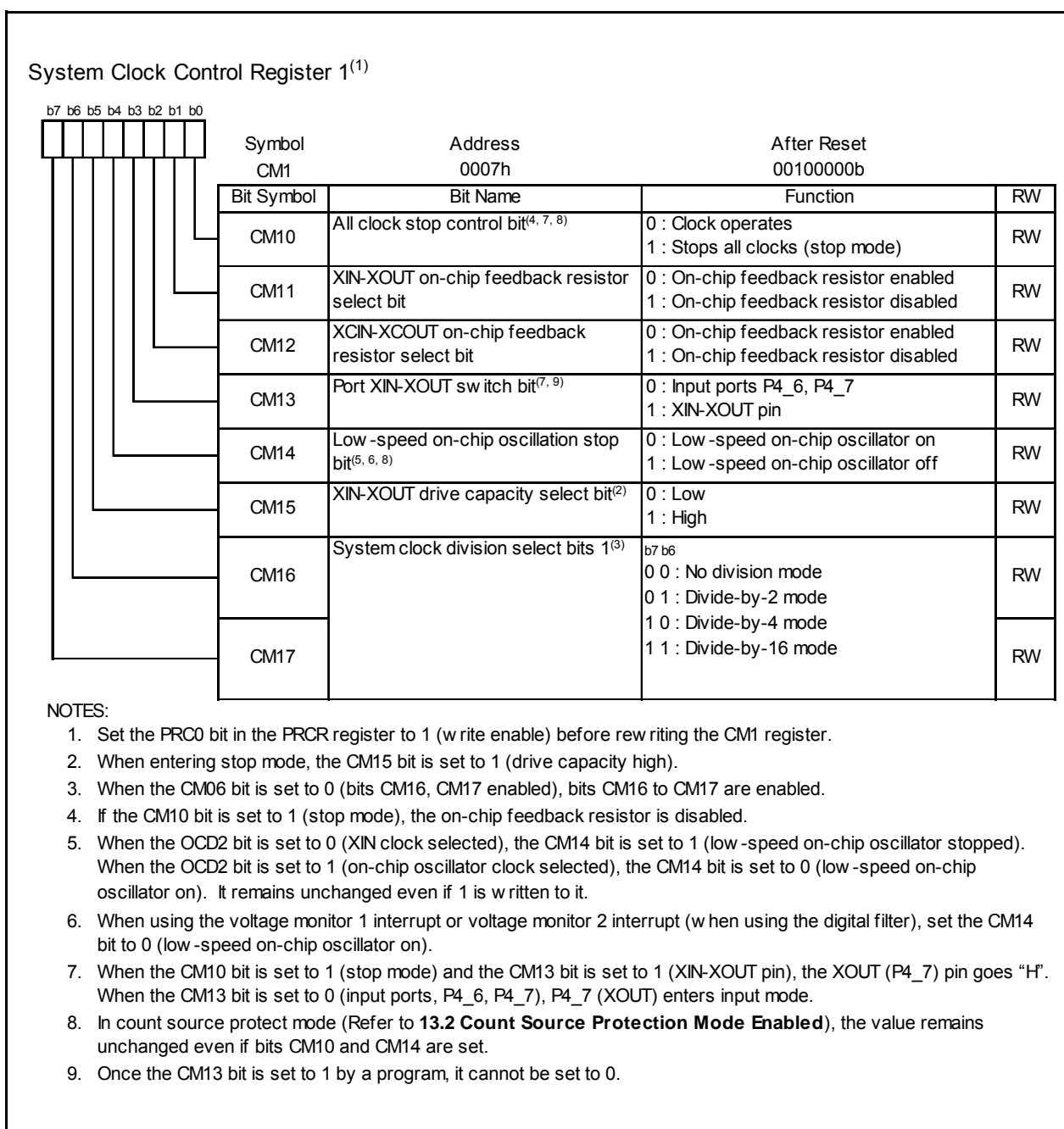


Figure 10.4 CM1 Register

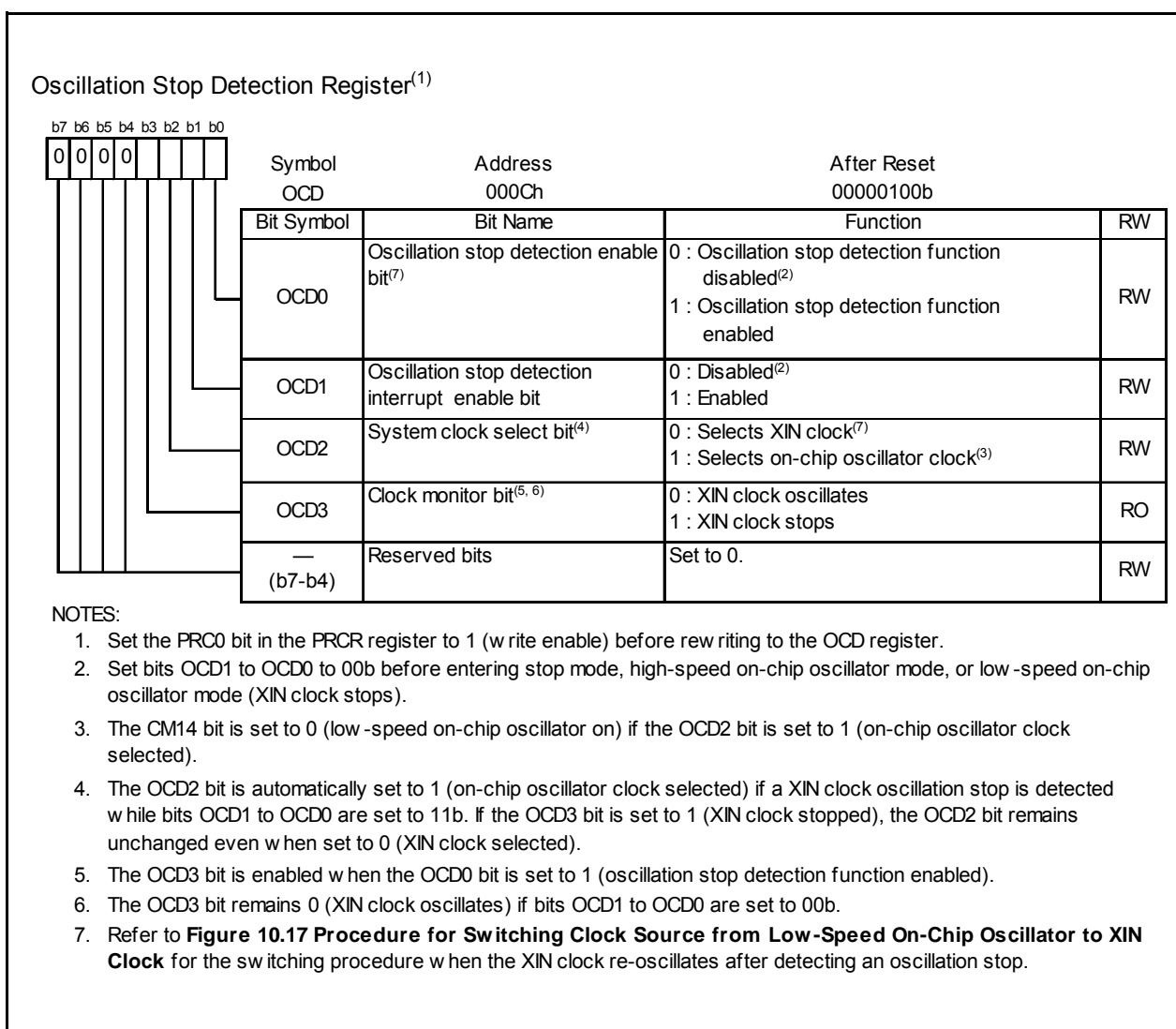


Figure 10.5 OCD Register

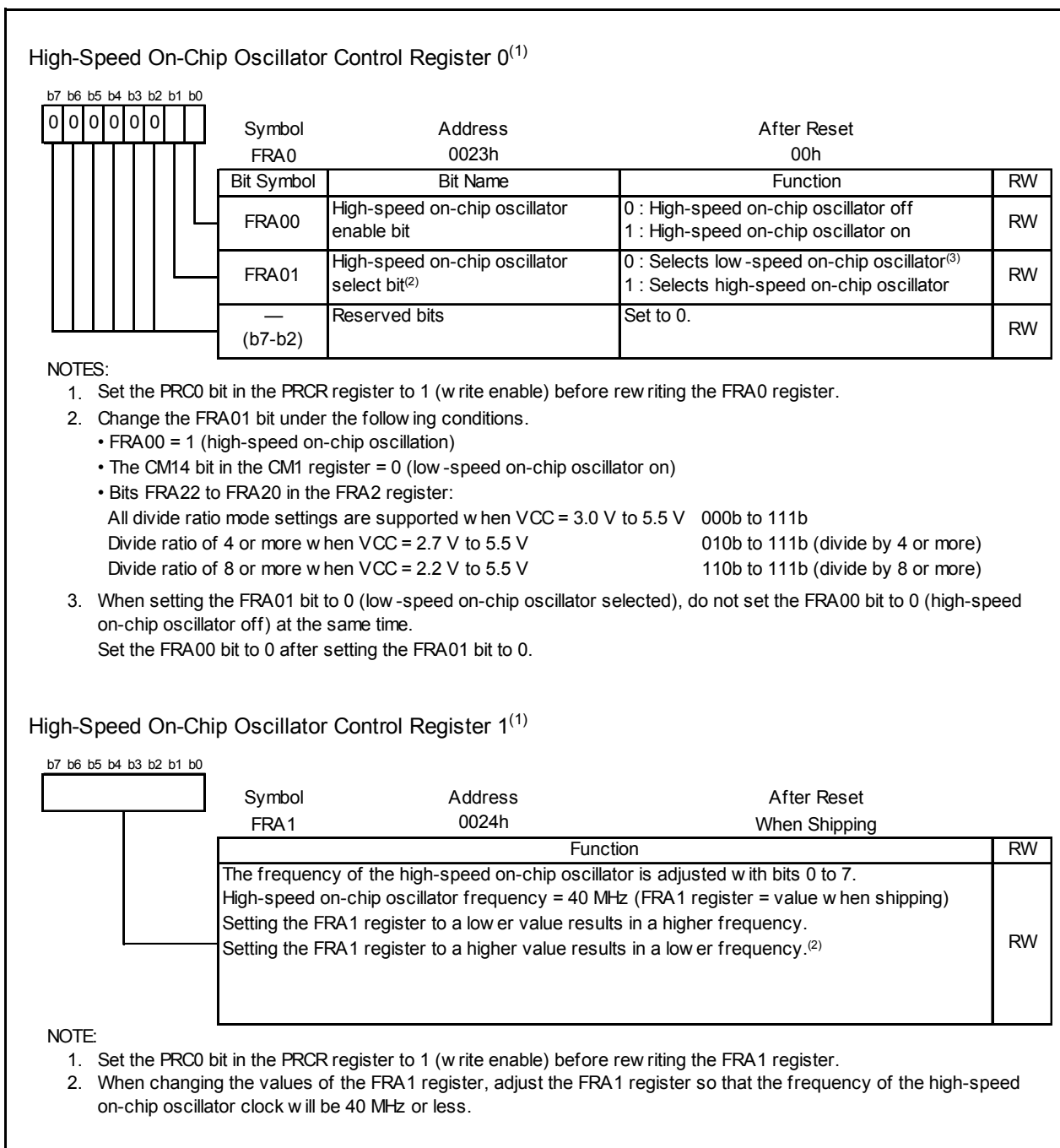


Figure 10.6 Registers FRA0 and FRA1

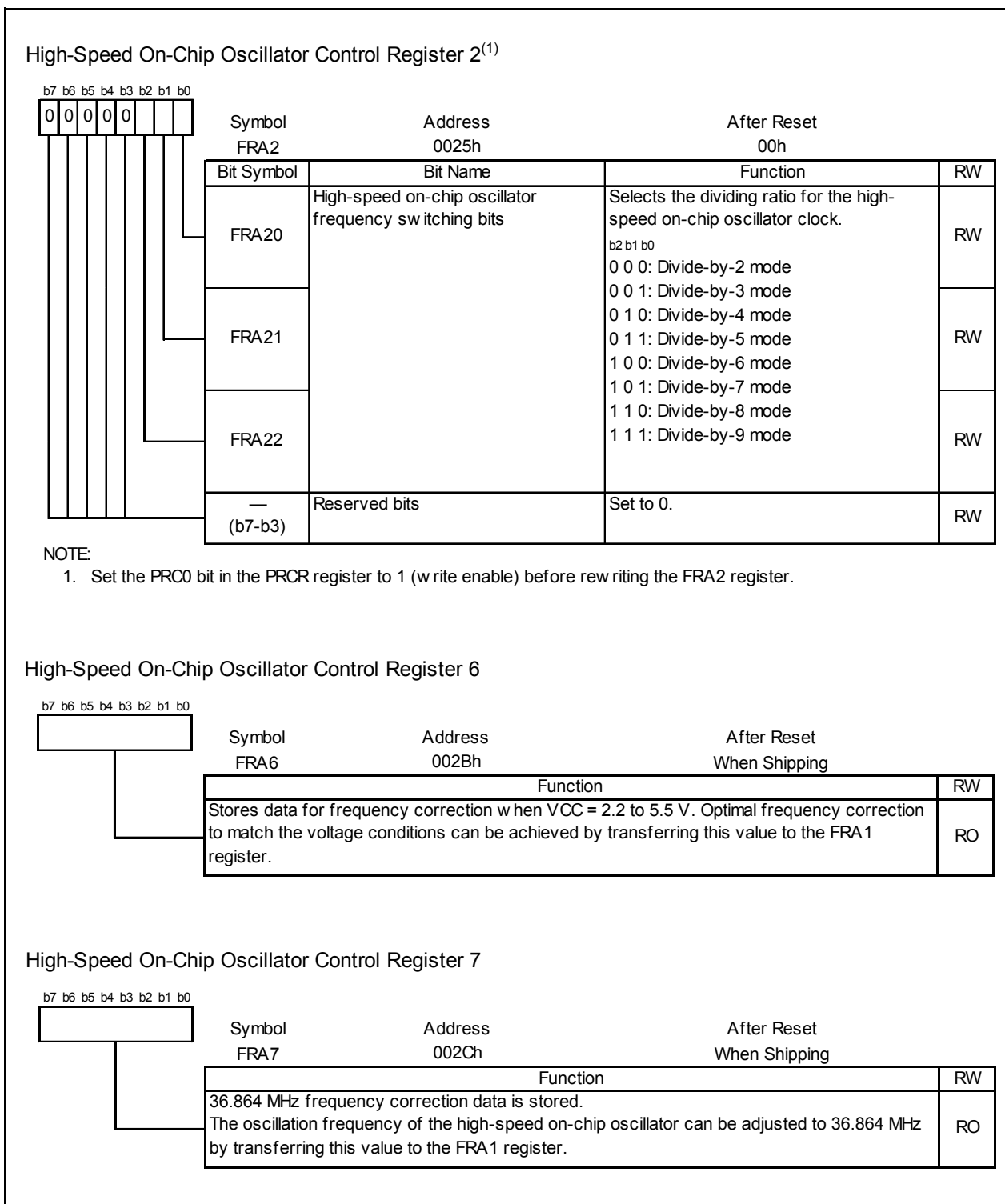


Figure 10.7 Registers FRA2, FRA6 and FRA7

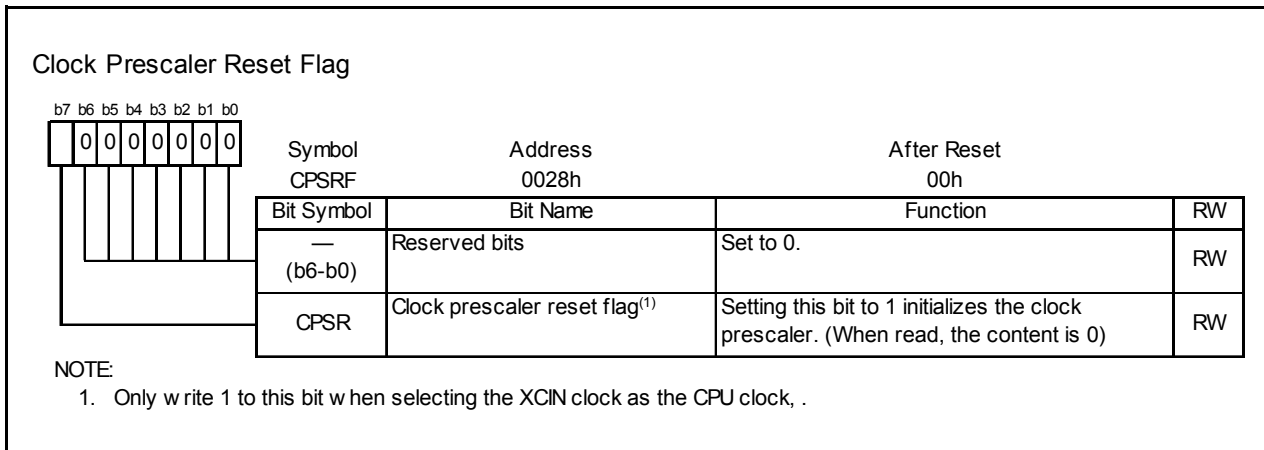


Figure 10.8 CPSRF Register

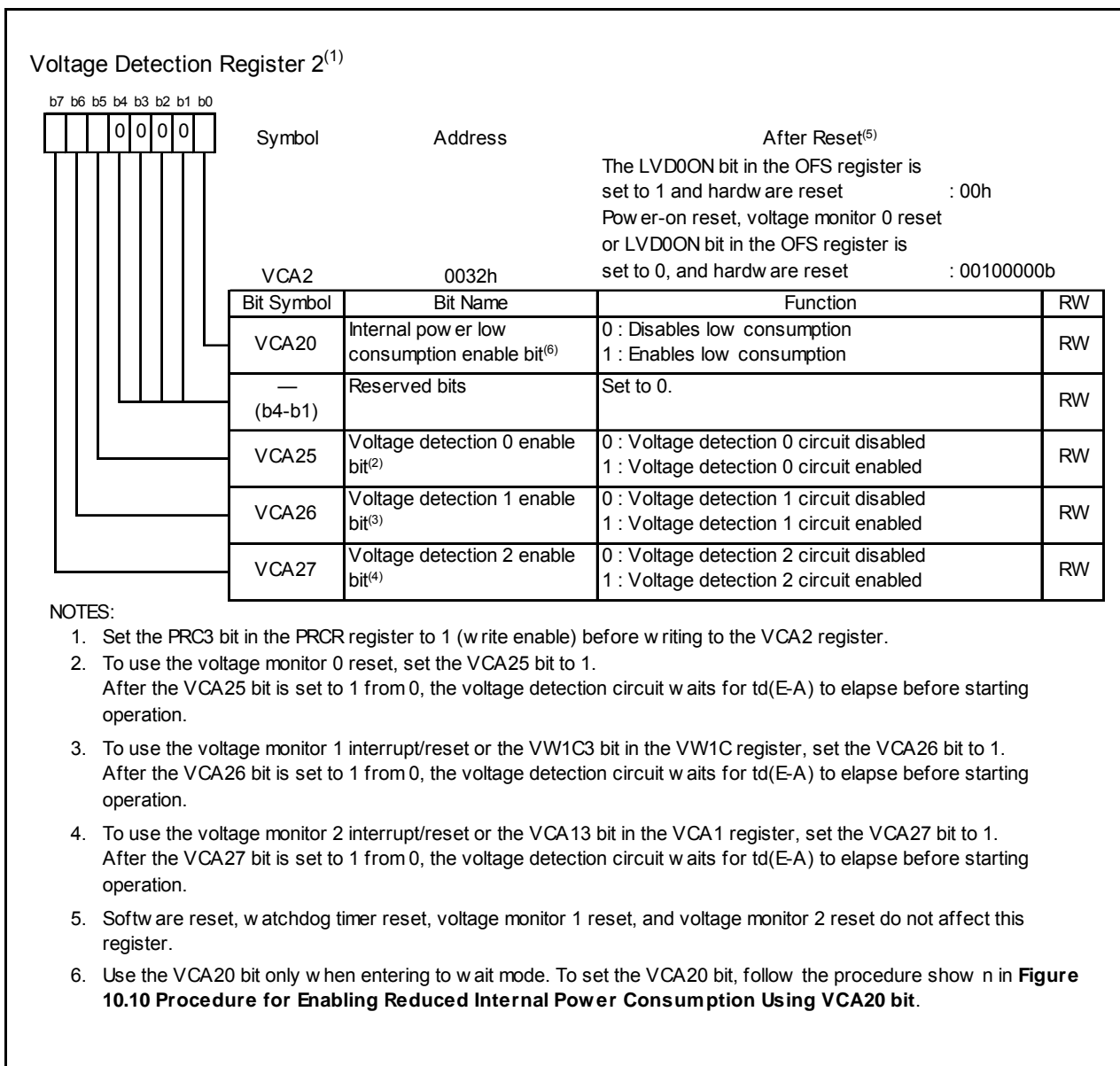


Figure 10.9 VCA2 Register

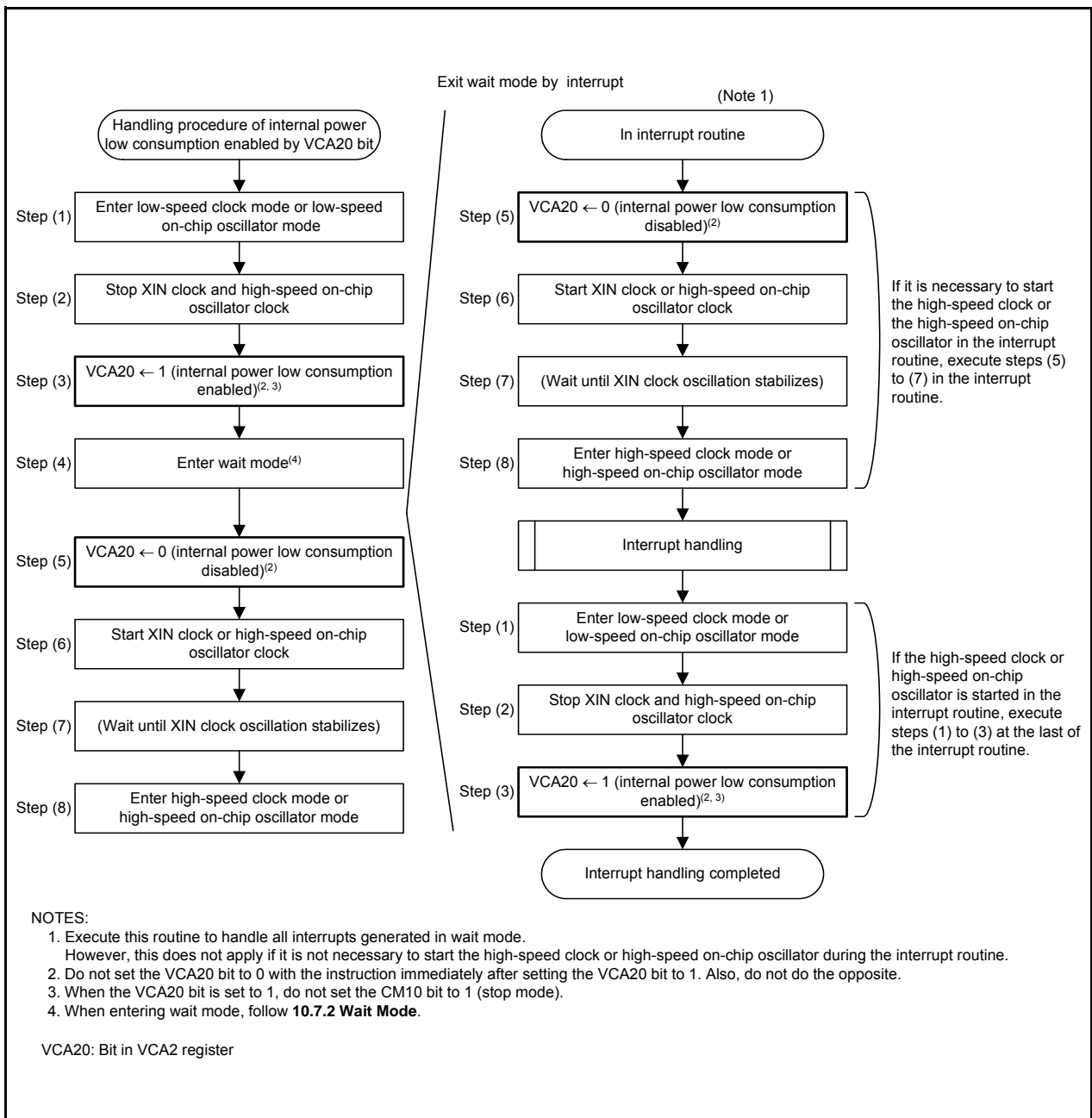


Figure 10.10 Procedure for Enabling Reduced Internal Power Consumption Using VCA20 bit

The clocks generated by the clock generation circuits are described below.

10.1 XIN Clock

This clock is supplied by the XIN clock oscillation circuit. This clock is used as the clock source for the CPU and peripheral function clocks. The XIN clock oscillation circuit is configured by connecting a resonator between the XIN and XOUT pins. The XIN clock oscillation circuit includes an on-chip feedback resistor, which is disconnected from the oscillation circuit in stop mode in order to reduce the amount of power consumed by the chip. The XIN clock oscillation circuit may also be configured by feeding an externally generated clock to the XIN pin.

Figure 10.11 shows Examples of XIN Clock Connection Circuit.

In reset and after reset, the XIN clock stops.

The XIN clock starts oscillating when the CM05 bit in the CM0 register is set to 0 (XIN clock oscillates) after setting the CM13 bit in the CM1 register to 1 (XIN- XOUT pin).

To use the XIN clock for the CPU clock source, set the OCD2 bit in the OCD register to 0 (select XIN clock) after the XIN clock is oscillating stably.

The power consumption can be reduced by setting the CM05 bit in the CM0 register to 1 (XIN clock stops) if the OCD2 bit is set to 1 (select on-chip oscillator clock).

When an external clock is input to the XIN pin are input, the XIN clock does not stop if the CM05 bit is set to 1. If necessary, use an external circuit to stop the clock.

In stop mode, all clocks including the XIN clock stop. Refer to **10.5 Power Control** for details.

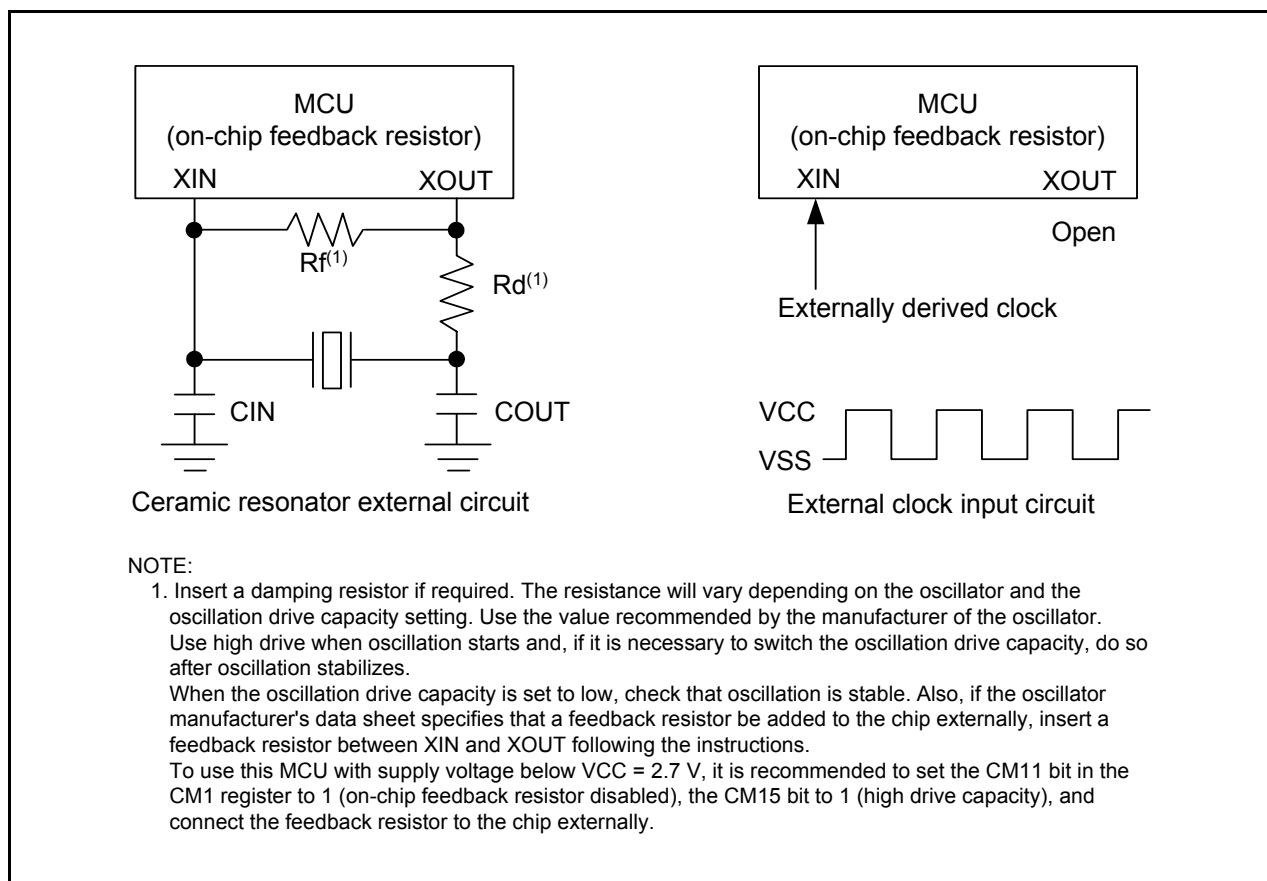


Figure 10.11 Examples of XIN Clock Connection Circuit

10.2 On-Chip Oscillator Clocks

These clocks are supplied by the on-chip oscillators (high-speed on-chip oscillator and a low-speed on-chip oscillator). The on-chip oscillator clock is selected by the FRA01 bit in the FRA0 register.

10.2.1 Low-Speed On-Chip Oscillator Clock

The clock generated by the low-speed on-chip oscillator is used as the clock source for the CPU clock, peripheral function clock, fOCO, and fOCO-S.

After reset, the on-chip oscillator clock generated by the low-speed on-chip oscillator divided by 8 is selected as the CPU clock.

If the XIN clock stops oscillating when bits OCD1 to OCD0 in the OCD register are set to 11b, the low-speed on-chip oscillator automatically starts operating, supplying the necessary clock for the MCU.

The frequency of the low-speed on-chip oscillator varies depending on the supply voltage and the operating ambient temperature. Application products must be designed with sufficient margin to allow for frequency changes.

10.2.2 High-Speed On-Chip Oscillator Clock

The clock generated by the high-speed on-chip oscillator is used as the clock source for the CPU clock, peripheral function clock, fOCO, fOCO-F, and fOCO40M.

To use the high-speed on-chip oscillator clock as the clock source for the CPU clock, peripheral clock, fOCO, and fOCO-F, set bits FRA20 to FRA22 in the FRA2 register as follows:

- All divide ratio mode settings are supported when VCC = 3.0 V to 5.5 V 000b to 111b
- Divide ratio of 4 or more when VCC = 2.7 V to 5.5 V 010b to 111b (divide by 4 or more)
- Divide ratio of 8 or more when VCC = 2.2 V to 5.5 V 110b to 111b (divide by 8 or more)

After reset, the on-chip oscillator clock generated by the high-speed on-chip oscillator stops. Oscillation is started by setting the FRA00 bit in the FRA0 register to 1 (high-speed on-chip oscillator on). The frequency can be adjusted by registers FRA1 and FRA2.

Furthermore, frequency correction data corresponding to the supply voltage ranges VCC = 2.2 V to 5.5 V is stored in FRA6 register. To use separate correction values to match this voltage ranges, transfer them from the FRA6 register to the FRA1 register.

The frequency correction data of 36.864 MHz is stored in the FRA7 register. To set the frequency of the high-speed on-chip oscillator to 36.864 MHz, transfer the correction value in the FRA7 register to the FRA1 register before use.

Since there are differences in the amount of frequency adjustment among the bits in the FRA1 register, make adjustments by changing the settings of individual bits. Adjust the FRA1 register so that the frequency of the high-speed on-chip oscillator clock will be 40 MHz or less.

10.3 XCIN Clock

This clock is supplied by the XCIN clock oscillation circuit. This clock is used as the clock source for the CPU clock, timer RA, and timer RE. The XCIN clock oscillation circuit is configured by connecting a resonator between the XCIN and XCOOUT pins. The XCIN clock oscillation circuit includes an on-chip feedback resistor, which is disconnected from the oscillation circuit in stop mode in order to reduce the amount of power consumed in the chip. The XCIN clock oscillation circuit may also be configured by feeding an externally generated clock to the XCIN pin.

Figure 10.12 shows Examples of XCIN Clock Connection Circuits.

During and after reset, the XCIN clock stops.

The XCIN clock starts oscillating when the CM04 bit in the CM0 register is set to 1 (XCIN-XCOOUT pin).

To use the XCIN clock for the CPU clock source, set the CM07 bit in the CM0 register to 1 (XCIN clock) after the XCIN clock is oscillating stably. To input an external clock to the XCIN pin, set the CM04 bit in the CM0 register to 1 (XCIN-XCOOUT pin) and leave the XCOOUT pin open.

This MCU has an on-chip feedback resistor and on-chip resistor disable/enable switching is possible by the CM12 bit in the CM1 register.

In stop mode, all clocks including the XCIN clock stop. Refer to **10.5 Power Control** for details.

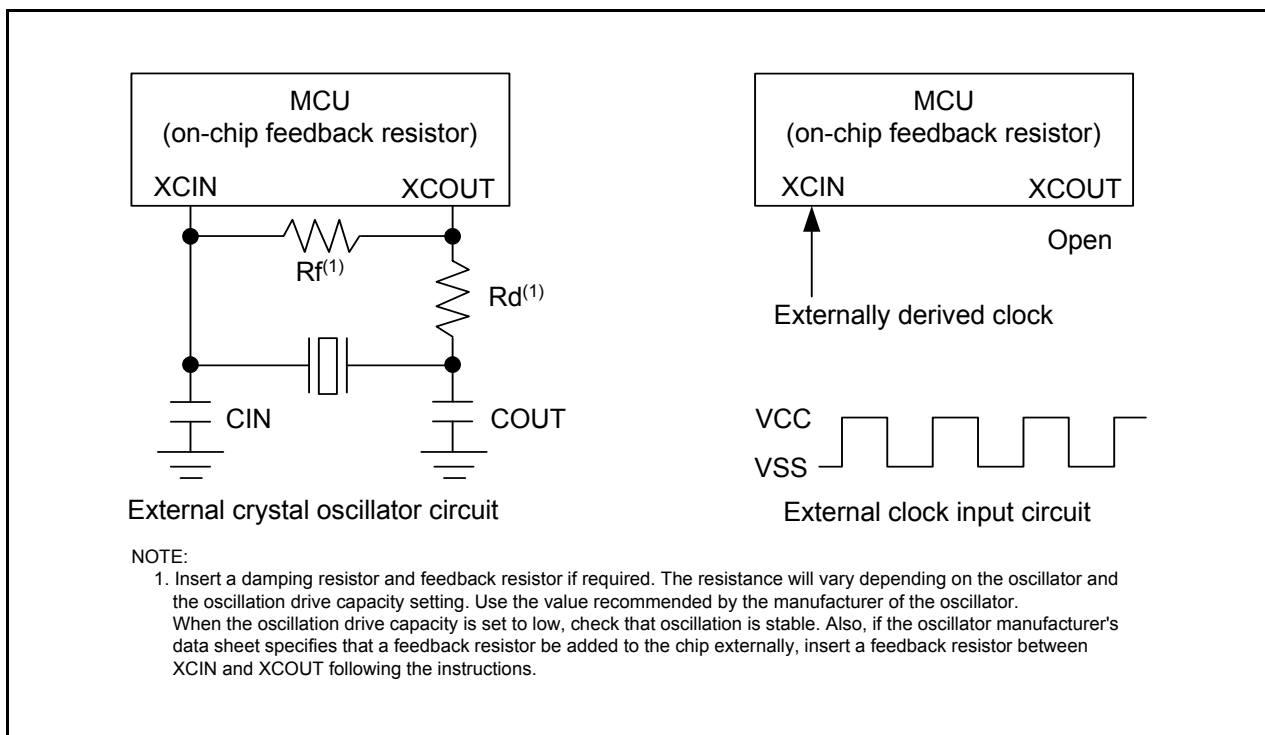


Figure 10.12 Examples of XCIN Clock Connection Circuits

10.4 CPU Clock and Peripheral Function Clock

There are a CPU clock to operate the CPU and a peripheral function clock to operate the peripheral functions. Refer to **Figure 10.1 Clock Generation Circuit**.

10.4.1 System Clock

The system clock is the clock source for the CPU and peripheral function clocks. Either the XIN clock or the on-chip oscillator clock can be selected.

10.4.2 CPU Clock

The CPU clock is an operating clock for the CPU and watchdog timer.

When the CM07 bit in the CM0 register is set to 0 (system clock), the system clock can be divided by 1 (no division), 2, 4, 8, or 16 to produce the CPU clock. Use the CM06 bit in the CM0 register and bits CM16 to CM17 in the CM1 register to select the value of the division.

When the CM07 bit in the CM0 register is set to 1 (XCIN clock), the XCIN clock is used for the CPU clock. Use the XCIN clock while the XCIN clock oscillation stabilizes.

After reset, the low-speed on-chip oscillator clock divided by 8 provides the CPU clock.

When entering stop mode from high-speed clock mode, the CM06 bit is set to 1 (divide-by-8 mode).

10.4.3 Peripheral Function Clock (f1, f2, f4, f8, and f32)

The peripheral function clock is the operating clock for the peripheral functions.

The clock f_i ($i = 1, 2, 4, 8,$ and 32) is generated by the system clock divided by i . The clock f_i is used for timers RA, RB, RC, RD, and RE, the serial interface and the A/D converter. The f_1 , f_8 , and f_{32} clock are used for timer RF.

When the WAIT instruction is executed after setting the CM02 bit in the CM0 register to 1 (peripheral function clock stops in wait mode), the clock f_i stop.

10.4.4 fOCO

fOCO is an operating clock for the peripheral functions.

fOCO runs at the same frequency as the on-chip oscillator clock and can be used as the source for timer RA.

When the WAIT instruction is executed, the clocks fOCO does not stop.

10.4.5 fOCO40M

fOCO40M is used as the count source for timer RC and timer RD. fOCO40M is generated by the high-speed on-chip oscillator and supplied by setting the FRA00 bit to 1.

When the WAIT instruction is executed, the clock fOCO40M does not stop.

fOCO40M can be used with supply voltage $V_{CC} = 3.0$ to 5.5 V.

10.4.6 fOCO-F

fOCO-F is used as the count source for the A/D converter. fOCO-F is generated by the high-speed on-chip oscillator and supplied by setting the FRA00 bit to 1.

When the WAIT instruction is executed, the clock fOCO-F does not stop.

10.4.7 fOCO-S

fOCO-S is an operating clock for the watchdog timer and voltage detection circuit. fOCO-S is supplied by setting the CM14 bit to 0 (low-speed on-chip oscillator on) and uses the clock generated by the low-speed on-chip oscillator. When the WAIT instruction is executed or in count source protect mode of the watchdog timer, fOCO-S does not stop.

10.4.8 fOCO128

fOCO128 is generated by fOCO divided by 128.

The clock fOCO128 is used for capture signal of timer RD (channel 0).

10.4.9 fC4 and fC32

The clock fC4 and fC32 are used for timer RA and timer RE.
Use fC4 and fC32 while the XCIN clock oscillation stabilizes.

10.5 Power Control

There are three power control modes. All modes other than wait mode and stop mode are referred to as standard operating mode.

10.5.1 Standard Operating Mode

Standard operating mode is further separated into four modes.

In standard operating mode, the CPU clock and the peripheral function clock are supplied to operate the CPU and the peripheral function clocks. Power consumption control is enabled by controlling the CPU clock frequency. The higher the CPU clock frequency, the more processing power increases. The lower the CPU clock frequency, the more power consumption decreases. When unnecessary oscillator circuits stop, power consumption is further reduced.

Before the clock sources for the CPU clock can be switched over, the new clock source needs to be oscillating and stable. If the new clock source is the XIN clock or XCIN clock, allow sufficient wait time in a program until oscillation is stabilized before exiting.

Table 10.2 Settings and Modes of Clock Associated Bits

| Modes | | OCD Register | CM1 Register | | | CM0 Register | | | | FRA0 Register | |
|------------------------------------|--------------|--------------|--------------|------|------|--------------|------|------|------|---------------|-------|
| | | OCD2 | CM17, CM16 | CM14 | CM13 | CM07 | CM06 | CM05 | CM04 | FRA01 | FRA00 |
| High-speed clock mode | No division | 0 | 00b | – | 1 | 0 | 0 | 0 | – | – | – |
| | Divide-by-2 | 0 | 01b | – | 1 | 0 | 0 | 0 | – | – | – |
| | Divide-by-4 | 0 | 10b | – | 1 | 0 | 0 | 0 | – | – | – |
| | Divide-by-8 | 0 | – | – | 1 | 0 | 1 | 0 | – | – | – |
| | Divide-by-16 | 0 | 11b | – | 1 | 0 | 0 | 0 | – | – | – |
| Low-speed clock mode | No division | – | – | – | – | 1 | – | – | 1 | – | – |
| High-speed on-chip oscillator mode | No division | 1 | 00b | – | – | 0 | 0 | – | – | 1 | 1 |
| | Divide-by-2 | 1 | 01b | – | – | 0 | 0 | – | – | 1 | 1 |
| | Divide-by-4 | 1 | 10b | – | – | 0 | 0 | – | – | 1 | 1 |
| | Divide-by-8 | 1 | – | – | – | 0 | 1 | – | – | 1 | 1 |
| | Divide-by-16 | 1 | 11b | – | – | 0 | 0 | – | – | 1 | 1 |
| Low-speed on-chip oscillator mode | No division | 1 | 00b | 0 | – | 0 | 0 | – | – | 0 | – |
| | Divide-by-2 | 1 | 01b | 0 | – | 0 | 0 | – | – | 0 | – |
| | Divide-by-4 | 1 | 10b | 0 | – | 0 | 0 | – | – | 0 | – |
| | Divide-by-8 | 1 | – | 0 | – | 0 | 1 | – | – | 0 | – |
| | Divide-by-16 | 1 | 11b | 0 | – | 0 | 0 | – | – | 0 | – |

–: can be 0 or 1, no change in outcome

10.5.1.1 High-Speed Clock Mode

The XIN clock divided by 1 (no division), 2, 4, 8, or 16 provides the CPU clock. Set the CM06 bit to 1 (divide-by-8 mode) when transiting to high-speed on-chip oscillator mode, low-speed on-chip oscillator mode. If the CM14 bit is set to 0 (low-speed on-chip oscillator on) or the FRA00 bit in the FRA0 register is set to 1 (high-speed on-chip oscillator on), fOCO can be used as timer RA. When the FRA00 bit is set to 1, fOCO40M can be used as timer RC and timer RD. When the CM14 bit is set to 0 (low-speed on-chip oscillator on), fOCO-S can be used for the watchdog timer and voltage detection circuit.

10.5.1.2 Low-Speed Clock Mode

The XCIN clock divided by 1 (no division) provides the CPU clock.

In this mode, stopping the XIN clock and high-speed on-chip oscillator, and setting the FMR47 bit in the FMR4 register to 1 (flash memory low consumption current read mode enabled) enables low consumption operation.

When the FRA00 bit is set to 1, fOCO40M can be used as timer RC and timer RD. When the CM14 bit is set to 0 (low-speed on-chip oscillator on), fOCO-S can be used for the watchdog timer and voltage detection circuit.

To enter wait mode from low-speed clock mode, setting the VCA20 bit in the VCA2 register to 1 (internal power low consumption enabled) enables lower consumption current in wait mode.

When enabling reduced internal power consumption using the VCA20 bit, follow **Figure 10.14 Procedure for Enabling Reduced Internal Power Consumption Using VCA20 bit**.

10.5.1.3 High-Speed On-Chip Oscillator Mode

The high-speed on-chip oscillator is used as the on-chip oscillator clock when the FRA00 bit in the FRA0 register is set to 1 (high-speed on-chip oscillator on) and the FRA01 bit in the FRA0 register is set to 1. The on-chip oscillator divided by 1 (no division), 2, 4, 8, or 16 provides the CPU clock. Set the CM06 bit to 1 (divide-by-8 mode) when transiting to high-speed clock mode. If the FRA00 bit is set to 1, fOCO40M can be used as timer RC and timer RD. When the CM14 bit is set to 0 (low-speed on-chip oscillator on), fOCO-S can be used for the watchdog timer and voltage detection circuit.

10.5.1.4 Low-Speed On-Chip Oscillator Mode

If the CM14 bit in the CM1 register is set to 0 (low-speed on-chip oscillator on) or the FRA01 bit in the FRA0 register is set to 0, the low-speed on-chip oscillator provides the on-chip oscillator clock.

The on-chip oscillator clock divided by 1 (no division), 2, 4, 8 or 16 provides the CPU clock. The on-chip oscillator clock is also the clock source for the peripheral function clocks. Set the CM06 bit to 1 (divide-by-8 mode) when transiting to high-speed clock mode. When the FRA00 bit is set to 1, fOCO40M can be used as timer RC and timer RD. When the CM14 bit is set to 0 (low-speed on-chip oscillator on), fOCO-S can be used as the watchdog timer and voltage detection circuit.

In this mode, stopping the XIN clock and high-speed on-chip oscillator, and setting the FMR47 bit in the FMR4 register to 1 (flash memory low consumption current read mode enabled) enables low consumption operation. To enter wait mode from low-speed on-chip oscillator mode, setting the VCA20 bit in the VCA2 register to 1 (internal power low consumption enabled) enables lower consumption current in wait mode.

When enabling reduced internal power consumption using the VCA20 bit, follow **Figure 10.14 Procedure for Enabling Reduced Internal Power Consumption Using VCA20 bit**.

10.5.2 Wait Mode

Since the CPU clock stops in wait mode, the CPU, which operates using the CPU clock, and the watchdog timer, when count source protection mode is disabled, stop. The XIN clock, XCIN clock, and on-chip oscillator clock do not stop and the peripheral functions using these clocks continue operating.

10.5.2.1 Peripheral Function Clock Stop Function

If the CM02 bit is set to 1 (peripheral function clock stops in wait mode), the f1, f2, f4, f8, and f32 clocks stop in wait mode. This reduces power consumption.

10.5.2.2 Entering Wait Mode

The MCU enters wait mode when the WAIT instruction is executed.

When the OCD2 bit in the OCD register is set to 1 (on-chip oscillator selected as system clock), set the OCD1 bit in the OCD register to 0 (oscillation stop detection interrupt disabled) before executing the WAIT instruction.

If the MCU enters wait mode while the OCD1 bit is set to 1 (oscillation stop detection interrupt enabled), current consumption is not reduced because the CPU clock does not stop.

10.5.2.3 Pin Status in Wait Mode

The I/O port is the status before wait mode was entered is maintained.

10.5.2.4 Exiting Wait Mode

The MCU exits wait mode by a reset or a peripheral function interrupt.

The peripheral function interrupts are affected by the CM02 bit. When the CM02 bit is set to 0 (peripheral function clock does not stop in wait mode), all peripheral function interrupts can be used to exit wait mode. When the CM02 bit is set to 1 (peripheral function clock stops in wait mode), the peripheral functions using the peripheral function clock stop operating and the peripheral functions operated by external signals or on-chip oscillator clock can be used to exit wait mode.

Table 10.3 lists Interrupts to Exit Wait Mode and Usage Conditions.

Table 10.3 Interrupts to Exit Wait Mode and Usage Conditions

| Interrupt | CM02 = 0 | CM02 = 1 |
|--|---|---|
| Serial interface interrupt | Usable when operating with internal or external clock | Usable when operating with external clock |
| Clock synchronous serial I/O with chip select interrupt / I ² C bus interface interrupt | Usable in all modes | (Do not use) |
| Key input interrupt | Usable | Usable |
| A/D conversion interrupt | Usable in one-shot mode | (Do not use) |
| Timer RA interrupt | Usable in all modes | Can be used if there is no filter in event counter mode. Usable by selecting fOCO or fC32 as count source. |
| Timer RB interrupt | Usable in all modes | (Do not use) |
| Timer RC interrupt | Usable in all modes | (Do not use) |
| Timer RD interrupt | Usable in all modes | Usable by selecting fOCO40M as count source |
| Timer RE interrupt | Usable in all modes | Usable when operating in real time clock mode |
| Timer RF interrupt | Usable in all modes | (Do not use) |
| INT interrupt | Usable | Usable (INT0 to INT3 can be used if there is no filter.) |
| Voltage monitor 1 interrupt | Usable | Usable |
| Voltage monitor 2 interrupt | Usable | Usable |
| Oscillation stop detection interrupt | Usable | (Do not use) |

Figure 10.13 shows the Time from Wait Mode to Interrupt Routine Execution.

When using a peripheral function interrupt to exit wait mode, set up the following before executing the WAIT instruction.

- (1) Set the interrupt priority level in bits ILVL2 to ILVL0 in the interrupt control registers of the peripheral function interrupts to be used for exiting wait mode. Set bits ILVL2 to ILVL0 of the peripheral function interrupts that are not to be used for exiting wait mode to 000b (interrupt disabled).
- (2) Set the I flag to 1.
- (3) Operate the peripheral function to be used for exiting wait mode.

When exiting by a peripheral function interrupt, the time (number of cycles) between interrupt request generation and interrupt routine execution is determined by the settings of the FMSTP bit in the FMR0 register and the CM07 bit in the CM0 register, as described in Figure 10.13.

The CPU clock, when exiting wait mode by a peripheral function interrupt, is the same clock as the CPU clock when the WAIT instruction is executed.

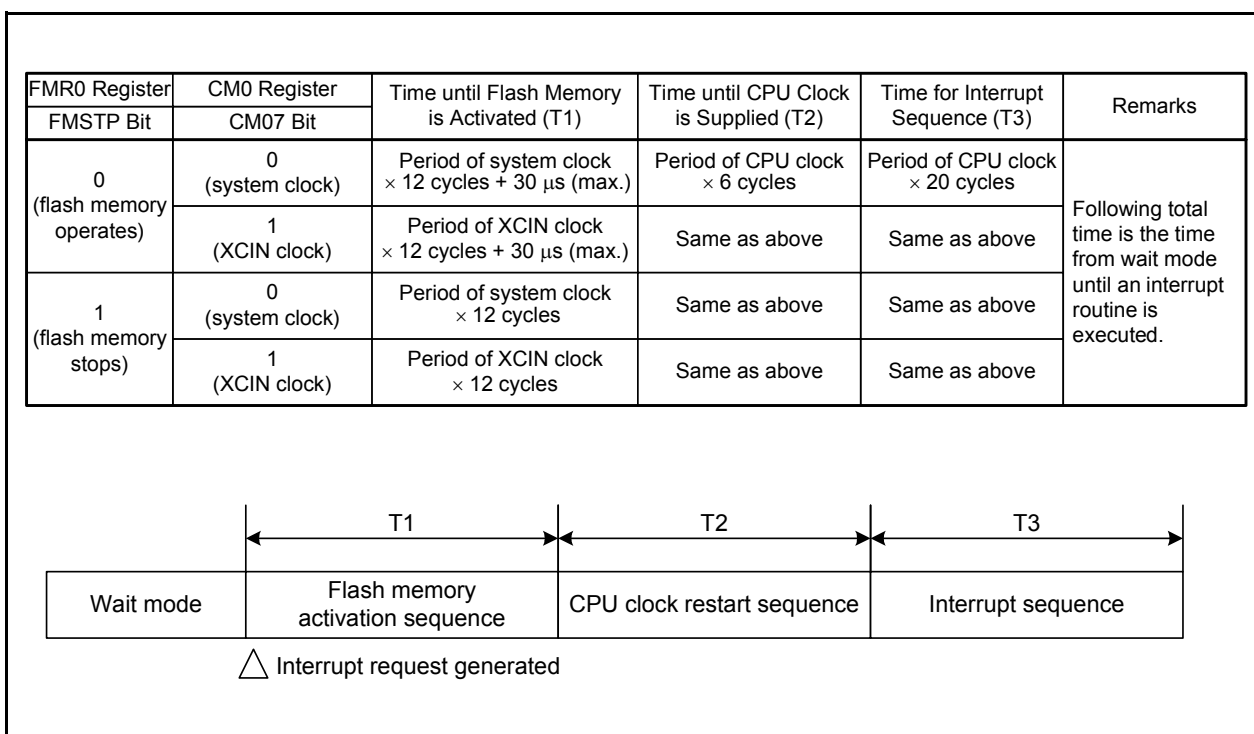


Figure 10.13 Time from Wait Mode to Interrupt Routine Execution

10.5.2.5 Reducing Internal Power Consumption

Internal power consumption can be reduced by using low-speed clock mode or low-speed on-chip oscillator mode. Figure 10.14 shows the Procedure for Enabling Reduced Internal Power Consumption Using VCA20 bit. When enabling reduced internal power consumption using the VCA20 bit, follow **Figure 10.14 Procedure for Enabling Reduced Internal Power Consumption Using VCA20 bit**.

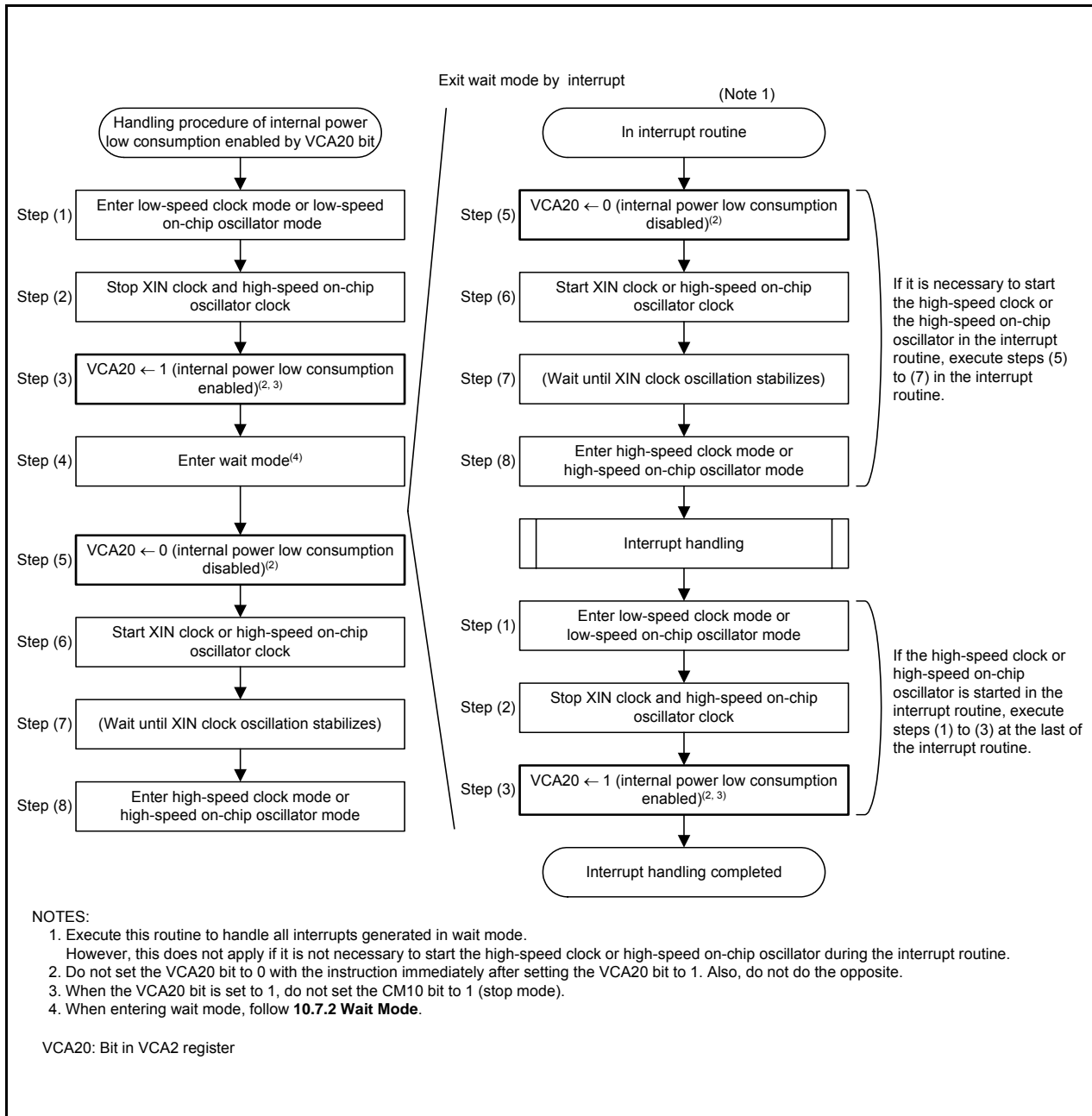


Figure 10.14 Procedure for Enabling Reduced Internal Power Consumption Using VCA20 bit

10.5.3 Stop Mode

Since the oscillator circuits stop in stop mode, the CPU clock and peripheral function clock stop and the CPU and peripheral functions that use these clocks stop operating. The least power required to operate the MCU is in stop mode. If the voltage applied to the VCC pin is V_{RAM} or more, the contents of internal RAM is maintained.

The peripheral functions clocked by external signals continue operating.

Table 10.4 lists Interrupts to Exit Stop Mode and Usage Conditions.

Table 10.4 Interrupts to Exit Stop Mode and Usage Conditions

| Interrupt | Usage Conditions |
|--|--|
| Key input interrupt | – |
| $\overline{\text{INT0}}$ to $\overline{\text{INT3}}$ interrupt | Can be used if there is no filter |
| Timer RA interrupt | Can be used if there is no filter when external pulse is counted in event counter mode |
| Serial interface interrupt | When external clock is selected |
| Voltage monitor 1 interrupt | Usable in digital filter disabled mode (VW1C1 bit in VW1C register is set to 1) |
| Voltage monitor 2 interrupt | Usable in digital filter disabled mode (VW2C1 bit in VW2C register is set to 1) |

10.5.3.1 Entering Stop Mode

The MCU enters stop mode when the CM10 bit in the CM1 register is set to 1 (all clocks stop). At the same time, the CM06 bit in the CM0 register is set to 1 (divide-by-8 mode) and the CM15 bit in the CM1 register is set to 1 (XIN clock oscillator circuit drive capacity high).

When using stop mode, set bits OCD1 to OCD0 to 00b before entering stop mode.

10.5.3.2 Pin Status in Stop Mode

The status before wait mode was entered is maintained.

However, when the CM13 bit in the CM1 register is set to 1 (XIN-XOUT pins), the XOUT(P4_7) pin is held "H". When the CM13 bit is set to 0 (input ports P4_6 and P4_7), the P4_7(XOUT pin) is held in input status.

10.5.3.3 Exiting Stop Mode

The MCU exits stop mode by a reset or peripheral function interrupt.

Figure 10.15 shows the Time from Stop Mode to Interrupt Routine Execution.

When using a peripheral function interrupt to exit stop mode, set up the following before setting the CM10 bit to 1.

- (1) Set the interrupt priority level in bits ILVL2 to ILVL0 of the peripheral function interrupts to be used for exiting stop mode. Set bits ILVL2 to ILVL0 of the peripheral function interrupts that are not to be used for exiting stop mode to 000b (interrupt disabled).
- (2) Set the I flag to 1.
- (3) Operates the peripheral function to be used for exiting stop mode.

When exiting by a peripheral function interrupt, the interrupt sequence is executed when an interrupt request is generated and the CPU clock supply is started.

If the clock used immediately before stop mode is a system clock and stop mode is exited by a peripheral function interrupt, the CPU clock becomes the previous system clock divided by 8.

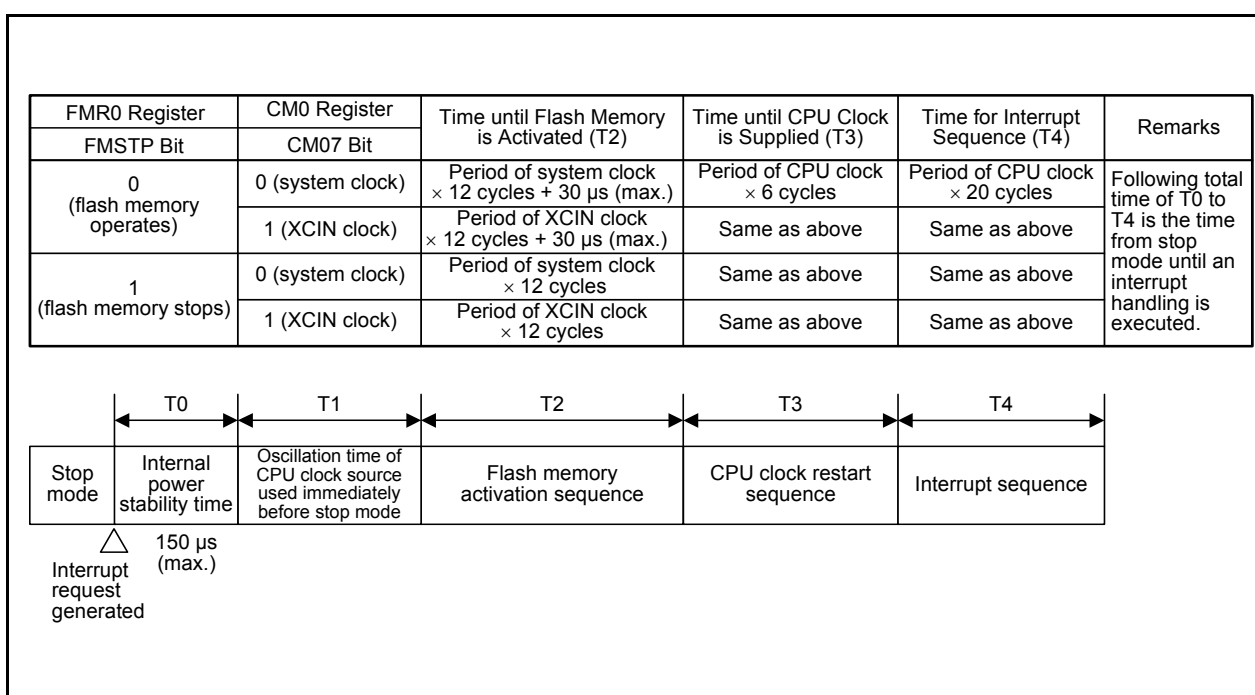


Figure 10.15 Time from Stop Mode to Interrupt Routine Execution

Figure 10.16 shows the State Transitions in Power Control Mode.

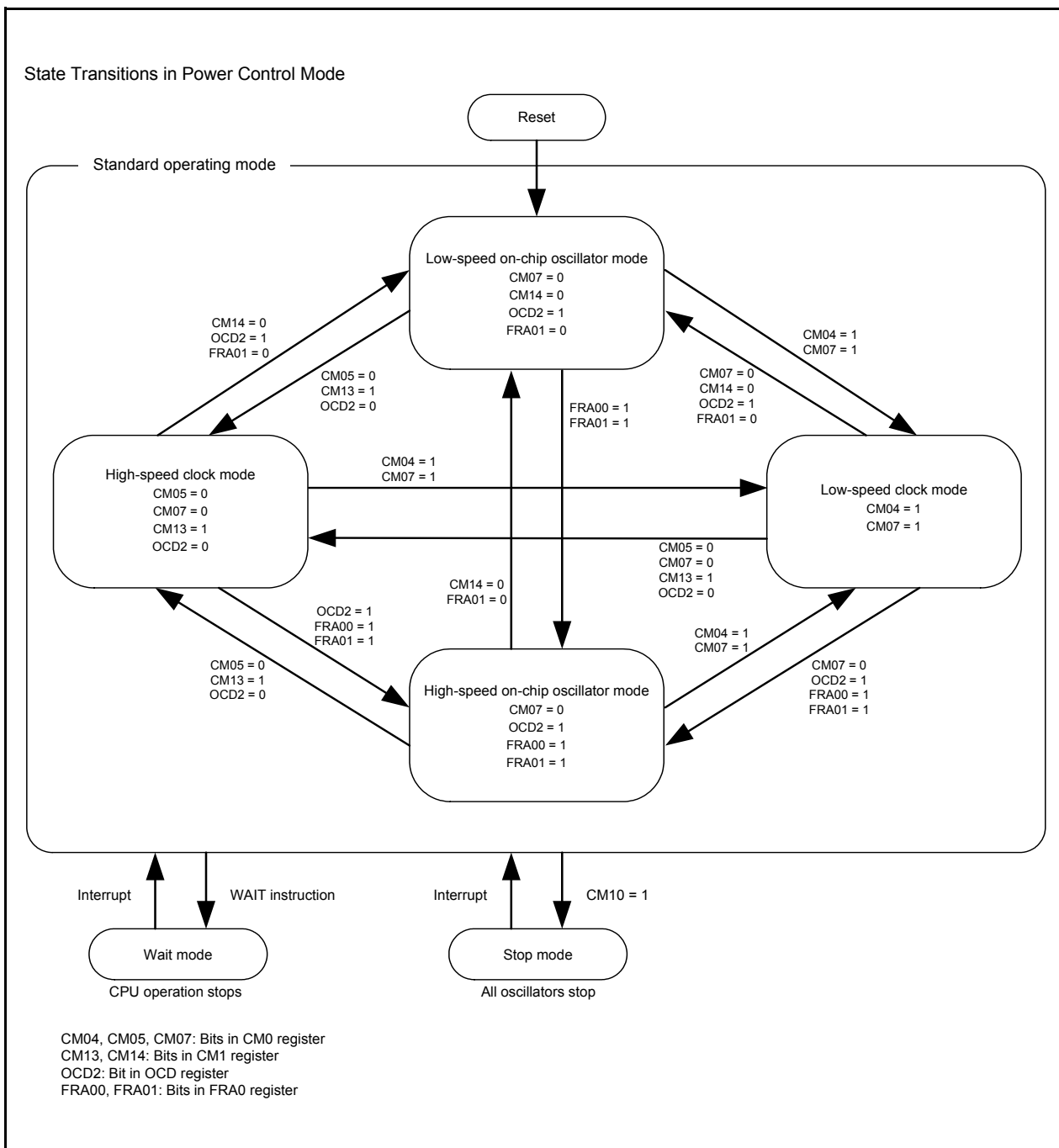


Figure 10.16 State Transitions in Power Control Mode

10.6 Oscillation Stop Detection Function

The oscillation stop detection function detects the stop of the XIN clock oscillating circuit. The oscillation stop detection function can be enabled and disabled by the OCD0 bit in the OCD register.

Table 10.5 lists the Specifications of Oscillation Stop Detection Function.

When the XIN clock is the CPU clock source and bits OCD1 to OCD0 are set to 11b, the system is placed in the following state if the XIN clock stops.

- OCD2 bit in OCD register = 1 (on-chip oscillator clock selected)
- OCD3 bit in OCD register = 1 (XIN clock stops)
- CM14 bit in CM1 register = 0 (low-speed on-chip oscillator oscillates)
- Oscillation stop detection interrupt request is generated.

Table 10.5 Specifications of Oscillation Stop Detection Function

| Item | Specification |
|---|---|
| Oscillation stop detection clock and frequency bandwidth | $f(\text{XIN}) \geq 2 \text{ MHz}$ |
| Enabled condition for oscillation stop detection function | Set bits OCD1 to OCD0 to 11b |
| Operation at oscillation stop detection | Oscillation stop detection interrupt is generated |

10.6.1 How to Use Oscillation Stop Detection Function

- The oscillation stop detection interrupt shares a vector with the voltage monitor 1 interrupt, the voltage monitor 2 interrupt, and the watchdog timer interrupt. When using the oscillation stop detection interrupt and watchdog timer interrupt, the interrupt source needs to be determined.
Table 10.6 lists Determining Interrupt Source for Oscillation Stop Detection, Watchdog Timer, Voltage Monitor 1, and Voltage Monitor 2 Interrupts. Figure 10.18 shows an Example of Determining Interrupt Source for Oscillation Stop Detection, Watchdog Timer, Voltage Monitor 1, or Voltage Monitor 2 Interrupt.
- When the XIN clock restarts after oscillation stop, switch the XIN clock to the clock source of the CPU clock and peripheral functions by a program.
Figure 10.17 shows the Procedure for Switching Clock Source from Low-Speed On-Chip Oscillator to XIN Clock.
- To enter wait mode while using the oscillation stop detection function, set the CM02 bit to 0 (peripheral function clock does not stop in wait mode).
- Since the oscillation stop detection function is a function for cases where the XIN clock is stopped by an external cause, set bits OCD1 to OCD0 to 00b when the XIN clock stops or is started by a program, (stop mode is selected or the CM05 bit is changed).
- This function cannot be used when the XIN clock frequency is 2 MHz or below. In this case, set bits OCD1 to OCD0 to 00b.
- To use the low-speed on-chip oscillator clock for the CPU clock and clock sources of peripheral functions after detecting the oscillation stop, set the FRA01 bit in the FRA0 register to 0 (low-speed on-chip oscillator selected) and bits OCD1 to OCD0 to 11b.
To use the high-speed on-chip oscillator clock for the CPU clock and clock sources of peripheral functions after detecting the oscillation stop, set the FRA00 bit to 1 (high-speed on-chip oscillator on) and the FRA01 bit to 1 (high-speed on-chip oscillator selected) and then set bits OCD1 to OCD0 to 11b.

Table 10.6 Determining Interrupt Source for Oscillation Stop Detection, Watchdog Timer, Voltage Monitor 1, and Voltage Monitor 2 Interrupts

| Generated Interrupt Source | Bit Showing Interrupt Cause |
|---|--|
| Oscillation stop detection ((a) or (b)) | (a) OCD3 bit in OCD register = 1 |
| | (b) OCD1 to OCD0 bits in OCD register = 11b and OCD2 bit = 1 |
| Watchdog timer | VW2C3 bit in VW2C register = 1 |
| Voltage monitor 1 | VW1C2 bit in VW1C register = 1 |
| Voltage monitor 2 | VW2C2 bit in VW2C register = 1 |

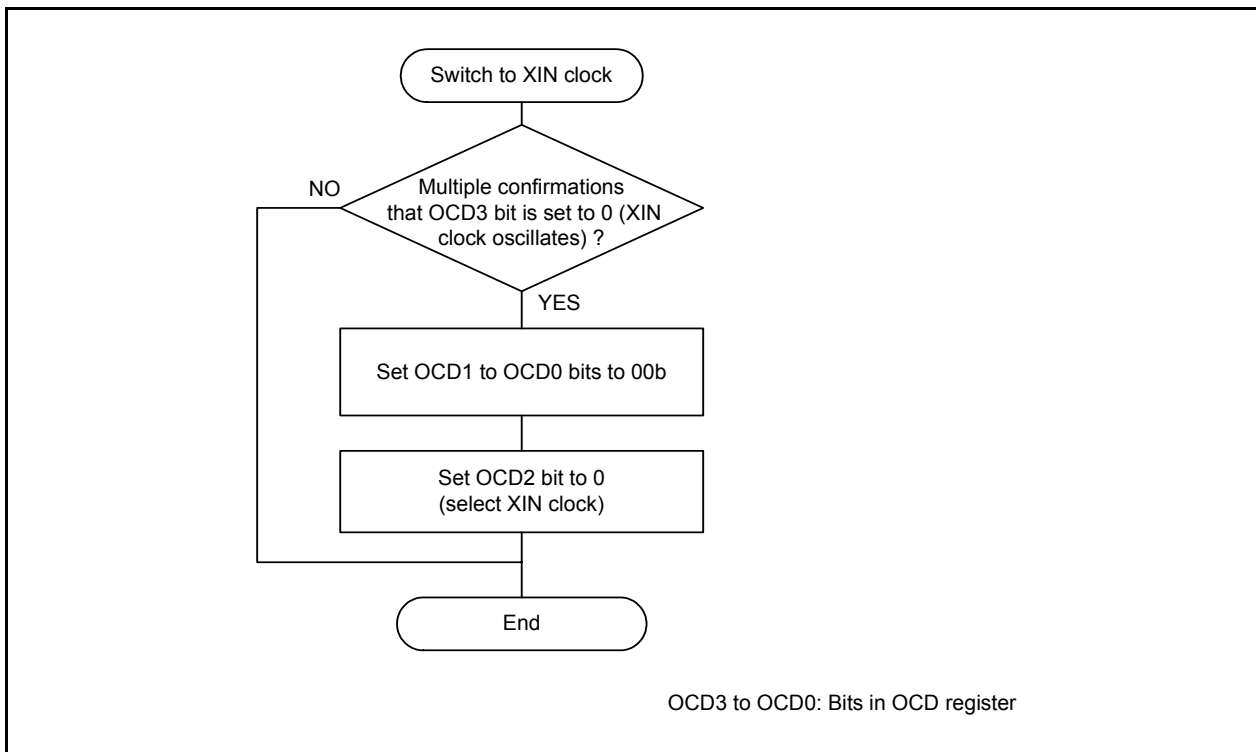


Figure 10.17 Procedure for Switching Clock Source from Low-Speed On-Chip Oscillator to XIN Clock

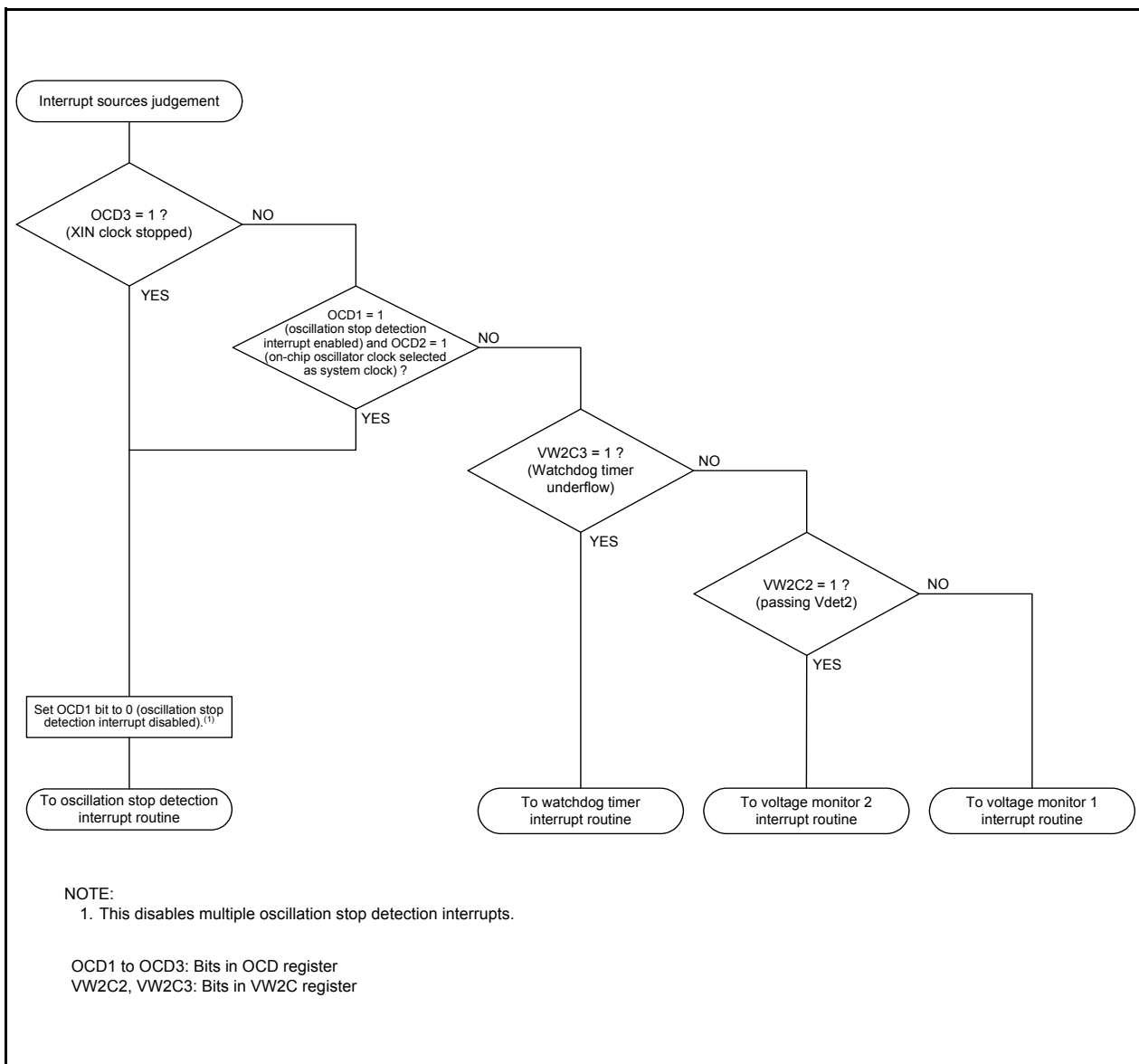


Figure 10.18 Example of Determining Interrupt Source for Oscillation Stop Detection, Watchdog Timer, Voltage Monitor 1, or Voltage Monitor 2 Interrupt

10.7 Notes on Clock Generation Circuit

10.7.1 Stop Mode

When entering stop mode, set the FMR01 bit in the FMR0 register to 0 (CPU rewrite mode disabled) and the CM10 bit in the CM1 register to 1 (stop mode). An instruction queue pre-reads 4 bytes from the instruction which sets the CM10 bit to 1 (stop mode) and the program stops.

Insert at least 4 NOP instructions following the JMP.B instruction after the instruction which sets the CM10 bit to 1.

- Program example to enter stop mode

```

BCLR    1,FMR0      ; CPU rewrite mode disabled
BSET    0,PRCR     ; Protect disabled
FSET    I          ; Enable interrupt
BSET    0,CM1      ; Stop mode
JMP.B   LABEL_001
LABEL_001 :
NOP
NOP
NOP
NOP

```

10.7.2 Wait Mode

When entering wait mode, set the FMR01 bit in the FMR0 register to 0 (CPU rewrite mode disabled) and execute the WAIT instruction. An instruction queue pre-reads 4 bytes from the WAIT instruction and the program stops. Insert at least 4 NOP instructions after the WAIT instruction.

- Program example to execute the WAIT instruction

```

BCLR    1,FMR0      ; CPU rewrite mode disabled
FSET    I          ; Enable interrupt
WAIT    ; Wait mode
NOP
NOP
NOP
NOP

```

10.7.3 Oscillation Stop Detection Function

Since the oscillation stop detection function cannot be used if the XIN clock frequency is 2 MHz or below, set bits OCD1 to OCD0 to 00b.

10.7.4 Oscillation Circuit Constants

Ask the manufacturer of the oscillator to specify the best oscillation circuit constants for your system.

To use this MCU with supply voltage below $VCC = 2.7$ V, it is recommended to set the CM11 bit in the CM1 register to 1 (on-chip feedback resistor disabled), the CM15 bit to 1 (high drive capacity), and connect the feedback resistor to the chip externally.

11. Protection

The protection function protects important registers from being easily overwritten when a program runs out of control. Figure 11.1 shows the PRCR Register. The registers protected by the PRCR register are listed below.

- Registers protected by PRC0 bit: Registers CM0, CM1, OCD, FRA0, FRA1, and FRA2
- Registers protected by PRC1 bit: Registers PM0 and PM1
- Registers protected by PRC2 bit: PD0 register
- Registers protected by PRC3 bit: Registers VCA2, VW0C, VW1C, and VW2C

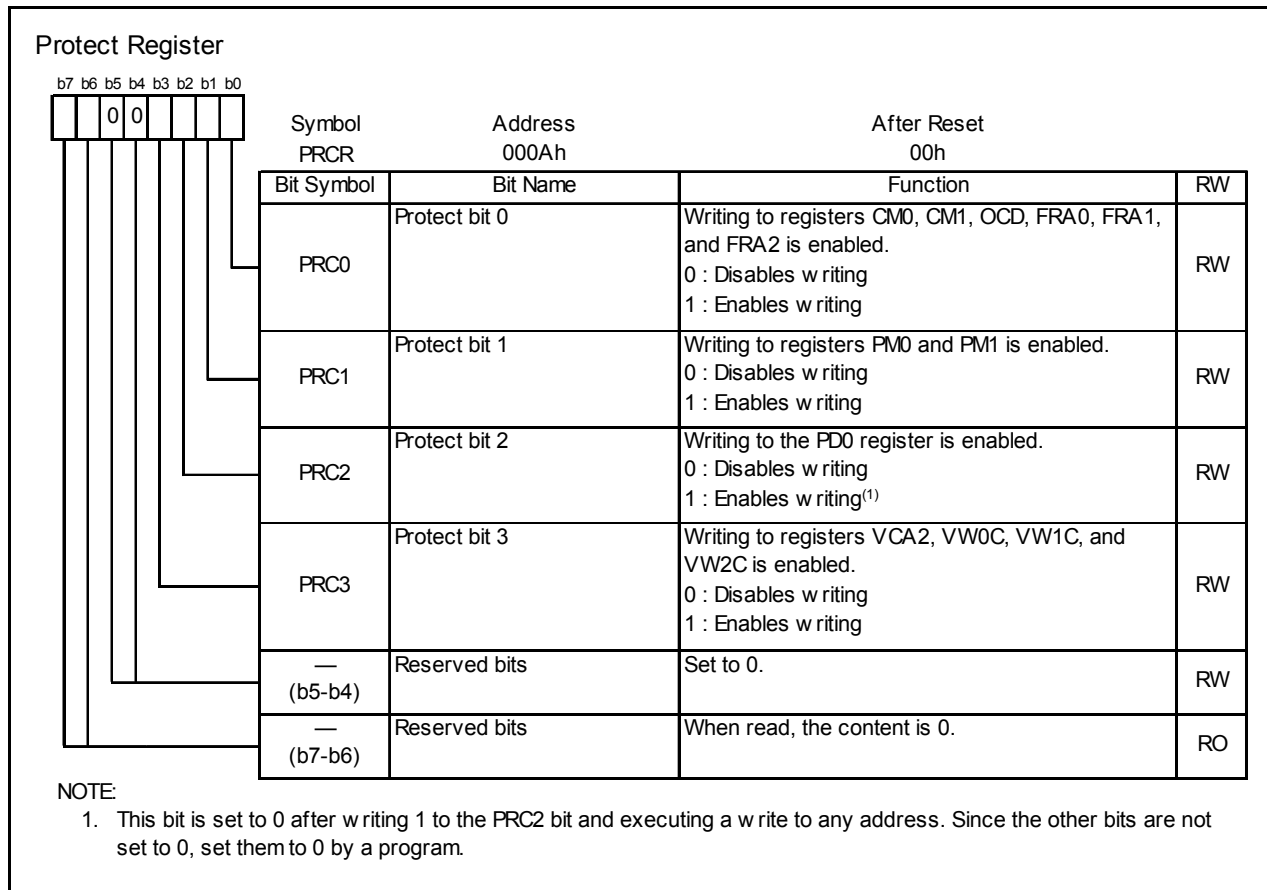


Figure 11.1 PRCR Register

12. Interrupts

12.1 Interrupt Overview

12.1.1 Types of Interrupts

Figure 12.1 shows the types of Interrupts.

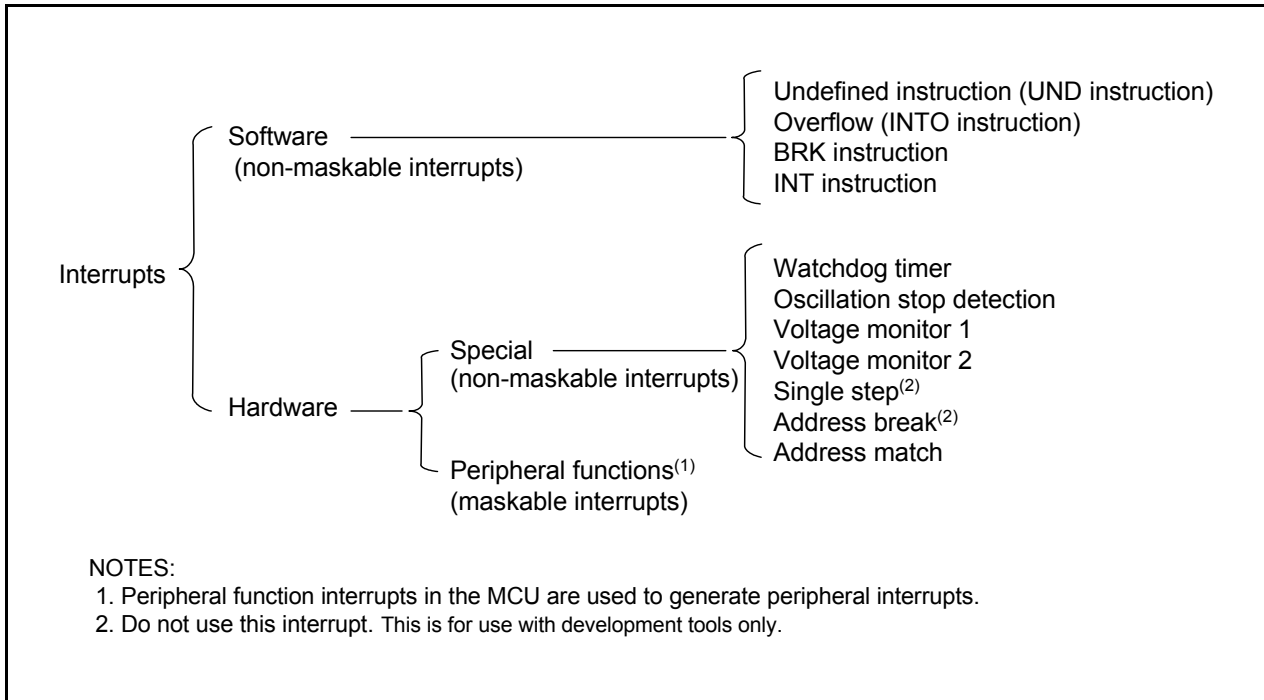


Figure 12.1 Interrupts

- **Maskable Interrupts:** The interrupt enable flag (I flag) enables or disables these interrupts. The interrupt priority order can be changed based on the interrupt priority level.
- **Non-Maskable Interrupts:** The interrupt enable flag (I flag) does not enable or disable these interrupts. The interrupt priority order cannot be changed based on interrupt priority level.

12.1.2 Software Interrupts

A software interrupt is generated when an instruction is executed. Software interrupts are non-maskable.

12.1.2.1 Undefined Instruction Interrupt

The undefined instruction interrupt is generated when the UND instruction is executed.

12.1.2.2 Overflow Interrupt

The overflow interrupt is generated when the O flag is set to 1 (arithmetic operation overflow) and the INTO instruction is executed. Instructions that set the O flag are: ABS, ADC, ADCF, ADD, CMP, DIV, DIVU, DIVX, NEG, RMPA, SBB, SHA, and SUB.

12.1.2.3 BRK Interrupt

A BRK interrupt is generated when the BRK instruction is executed.

12.1.2.4 INT Instruction Interrupt

An INT instruction interrupt is generated when the INT instruction is executed. The INT instruction can select software interrupt numbers 0 to 63. Software interrupt numbers 3 to 31 are assigned to the peripheral function interrupt. Therefore, the MCU executes the same interrupt routine when the INT instruction is executed as when a peripheral function interrupt is generated. For software interrupt numbers 0 to 31, the U flag is saved to the stack during instruction execution and the U flag is set to 0 (ISP selected) before the interrupt sequence is executed. The U flag is restored from the stack when returning from the interrupt routine. For software interrupt numbers 32 to 63, the U flag does not change state during instruction execution, and the selected SP is used.

12.1.3 Special Interrupts

Special interrupts are non-maskable.

12.1.3.1 Watchdog Timer Interrupt

The watchdog timer interrupt is generated by the watchdog timer. For details, refer to **13. Watchdog Timer**.

12.1.3.2 Oscillation Stop Detection Interrupt

The oscillation stop detection interrupt is generated by the oscillation stop detection function. For details of the oscillation stop detection function, refer to **10. Clock Generation Circuit**.

12.1.3.3 Voltage Monitor 1 Interrupt

The voltage monitor 1 interrupt is generated by the voltage detection circuit. For details of the voltage detection circuit, refer to **6. Voltage Detection Circuit**.

12.1.3.4 Voltage Monitor 2 Interrupt

The voltage monitor 2 interrupt is generated by the voltage detection circuit. For details of the voltage detection circuit, refer to **6. Voltage Detection Circuit**.

12.1.3.5 Single-Step Interrupt, and Address Break Interrupt

Do not use these interrupts. They are for use by development tools only.

12.1.3.6 Address Match Interrupt

The address match interrupt is generated immediately before executing an instruction that is stored at an address indicated by registers RMAD0 to RMAD1 when the AIER0 or AIER1 bit in the AIER register is set to 1 (address match interrupt enable). For details of the address match interrupt, refer to **12.4 Address Match Interrupt**.

12.1.4 Peripheral Function Interrupt

The peripheral function interrupt is generated by the internal peripheral function of the MCU and is a maskable interrupt. Refer to **Table 12.2 Relocatable Vector Tables** for sources of the peripheral function interrupt. For details of peripheral functions, refer to the descriptions of individual peripheral functions.

12.1.5 Interrupts and Interrupt Vectors

There are 4 bytes in each vector. Set the starting address of an interrupt routine in each interrupt vector. When an interrupt request is acknowledged, the CPU branches to the address set in the corresponding interrupt vector. Figure 12.2 shows an Interrupt Vector.

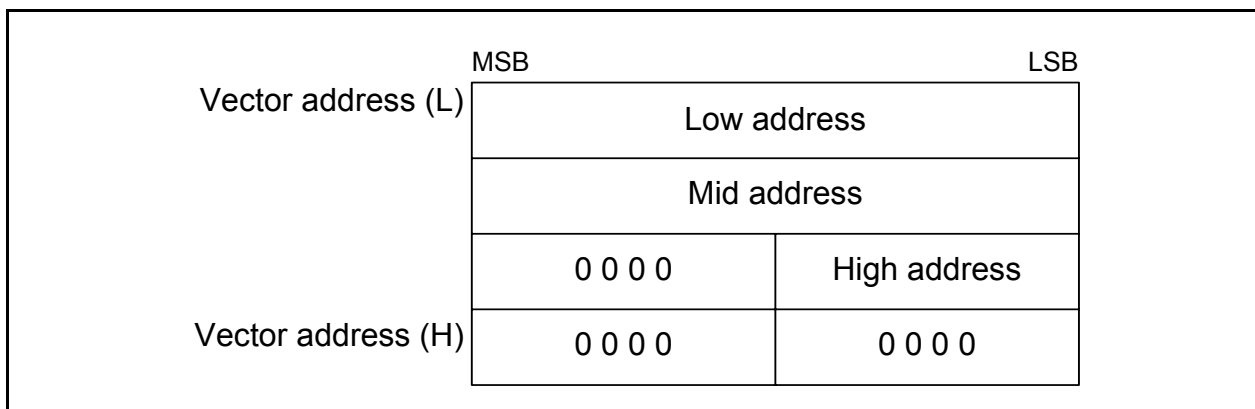


Figure 12.2 Interrupt Vector

12.1.5.1 Fixed Vector Tables

The fixed vector tables are allocated addresses 0FFDCh to 0FFFFh.

Table 12.1 lists the Fixed Vector Tables. The vector addresses (H) of fixed vectors are used by the ID code check function. For details, refer to **20.3 Functions to Prevent Rewriting of Flash Memory**.

Table 12.1 Fixed Vector Tables

| Interrupt Source | Vector Addresses Address (L) to (H) | Remarks | Reference |
|--|-------------------------------------|---|--|
| Undefined instruction | 0FFDCh to 0FFDFh | Interrupt on UND instruction | R8C/Tiny Series Software Manual |
| Overflow | 0FFE0h to 0FFE3h | Interrupt on INTO instruction | |
| BRK instruction | 0FFE4h to 0FFE7h | If the content of address 0FFE7h is FFh, program execution starts from the address shown by the vector in the relocatable vector table. | |
| Address match | 0FFE8h to 0FFEBh | | 12.4 Address Match Interrupt |
| Single step ⁽¹⁾ | 0FFEC h to 0FFEFh | | |
| Watchdog timer, Oscillation stop detection, Voltage monitor 1, Voltage monitor 2 | 0FFF0h to 0FFF3h | | 13. Watchdog Timer 10. Clock Generation Circuit 6. Voltage Detection Circuit |
| Address break ⁽¹⁾ | 0FFF4h to 0FFF7h | | |
| (Reserved) | 0FFF8h to 0FFFBh | | |
| Reset | 0FFFCh to 0FFFFh | | 5. Resets |

NOTES:

1. Do not use these interrupts. They are for use by development tools only.

12.1.5.2 Relocatable Vector Tables

The relocatable vector tables occupy 256 bytes beginning from the starting address set in the INTB register. Table 12.2 lists the Relocatable Vector Tables.

Table 12.2 Relocatable Vector Tables

| Interrupt Source | Vector Addresses ⁽¹⁾ Address (L) to Address (H) | Software Interrupt Number | Interrupt Control Register | Reference |
|--|---|---------------------------------|-------------------------------|--|
| BRK instruction ⁽³⁾ | +0 to +3 (0000h to 0003h) | 0 | – | R8C/Tiny Series Software Manual |
| (Reserved) | | 1 to 2 | – | |
| (Reserved) | | 3 to 6 | – | |
| Timer RC | +28 to +31 (001Ch to 001Fh) | 7 | TRCIC | 14.3 Timer RC |
| Timer RD (channel 0) | +32 to +35 (0020h to 0023h) | 8 | TRD0IC | 14.4 Timer RD |
| Timer RD (channel 1) | +36 to +39 (0024h to 0027h) | 9 | TRD1IC | |
| Timer RE | +40 to +43 (0028h to 002Bh) | 10 | TREIC | 14.5 Timer RE |
| UART2 transmit | +44 to +47 (002Ch to 002Fh) | 11 | S2TIC | 15. Serial Interface |
| UART2 receive | +48 to +51 (0030h to 0033h) | 12 | S2RIC | |
| Key input | +52 to +55 (0034h to 0037h) | 13 | KUPIC | 12.3 Key Input Interrupt |
| (Reserved) | | 14 | – | – |
| Clock synchronous serial I/O with chip select / I ² C bus interface ⁽²⁾ | +60 to +63 (003Ch to 003Fh) | 15 | SSUIC/IICIC | 16.2 Clock Synchronous Serial I/O with Chip Select (SSU), 16.3 I ² C bus Interface |
| Compare 1 | +64 to +67 (0040h to 0043h) | 16 | CMP1IC | 14.6 Timer RF |
| UART0 transmit | +68 to +71 (0044h to 0047h) | 17 | S0TIC | 15. Serial Interface |
| UART0 receive | +72 to +75 (0048h to 004Bh) | 18 | S0RIC | |
| UART1 transmit | +76 to +79 (004Ch to 004Fh) | 19 | S1TIC | |
| UART1 receive | +80 to +83 (0050h to 0053h) | 20 | S1RIC | |
| $\overline{\text{INT}}_2$ | +84 to +87 (0054h to 0057h) | 21 | INT2IC | |
| Timer RA | +88 to +91 (0058h to 005Bh) | 22 | TRAIC | 14.1 Timer RA |
| (Reserved) | | 23 | – | – |
| Timer RB | +96 to +99 (0060h to 0063h) | 24 | TRBIC | 14.2 Timer RB |
| $\overline{\text{INT}}_1$ | +100 to +103 (0064h to 0067h) | 25 | INT1IC | 12.2 $\overline{\text{INT}}$ Interrupt |
| $\overline{\text{INT}}_3$ | +104 to +107 (0068h to 006Bh) | 26 | INT3IC | |
| Timer RF | +108 to +111 (006Ch to 006Fh) | 27 | TRFIC | 14.6 Timer RF |
| Compare 0 | +112 to +115 (0070h to 0073h) | 28 | CMP0IC | |
| $\overline{\text{INT}}_0$ | +116 to +119 (0074h to 0077h) | 29 | INT0IC | 12.2 $\overline{\text{INT}}$ Interrupt |
| A/D | +120 to +123 (0078h to 007Bh) | 30 | ADIC | 18. A/D Converter |
| Capture | +124 to +127 (007Ch to 007Fh) | 31 | CAPIC | 14.6 Timer RF |
| Software interrupt ⁽³⁾ | +128 to +131 (0080h to 0083h) to +252 to +255 (00FCh to 00FFh) | 32 to 63 | – | R8C/Tiny Series Software Manual |

NOTES:

1. These addresses are relative to those in the INTB register.
2. The IICSEL bit in the PMR register switches functions.
3. The I flag does not disable these interrupts.

12.1.6 Interrupt Control

The following describes enabling and disabling the maskable interrupts and setting the priority for acknowledgement. The explanation does not apply to nonmaskable interrupts.

Use the I flag in the FLG register, IPL, and bits ILVL2 to ILVL0 in each interrupt control register to enable or disable maskable interrupts. Whether an interrupt is requested is indicated by the IR bit in each interrupt control register.

Figure 12.3 shows the Interrupt Control Register, Figure 12.4 shows Registers TRCIC, TRD0IC, TRD1IC, SSUIC, and IICIC and Figure 12.5 shows the INTiC Register.

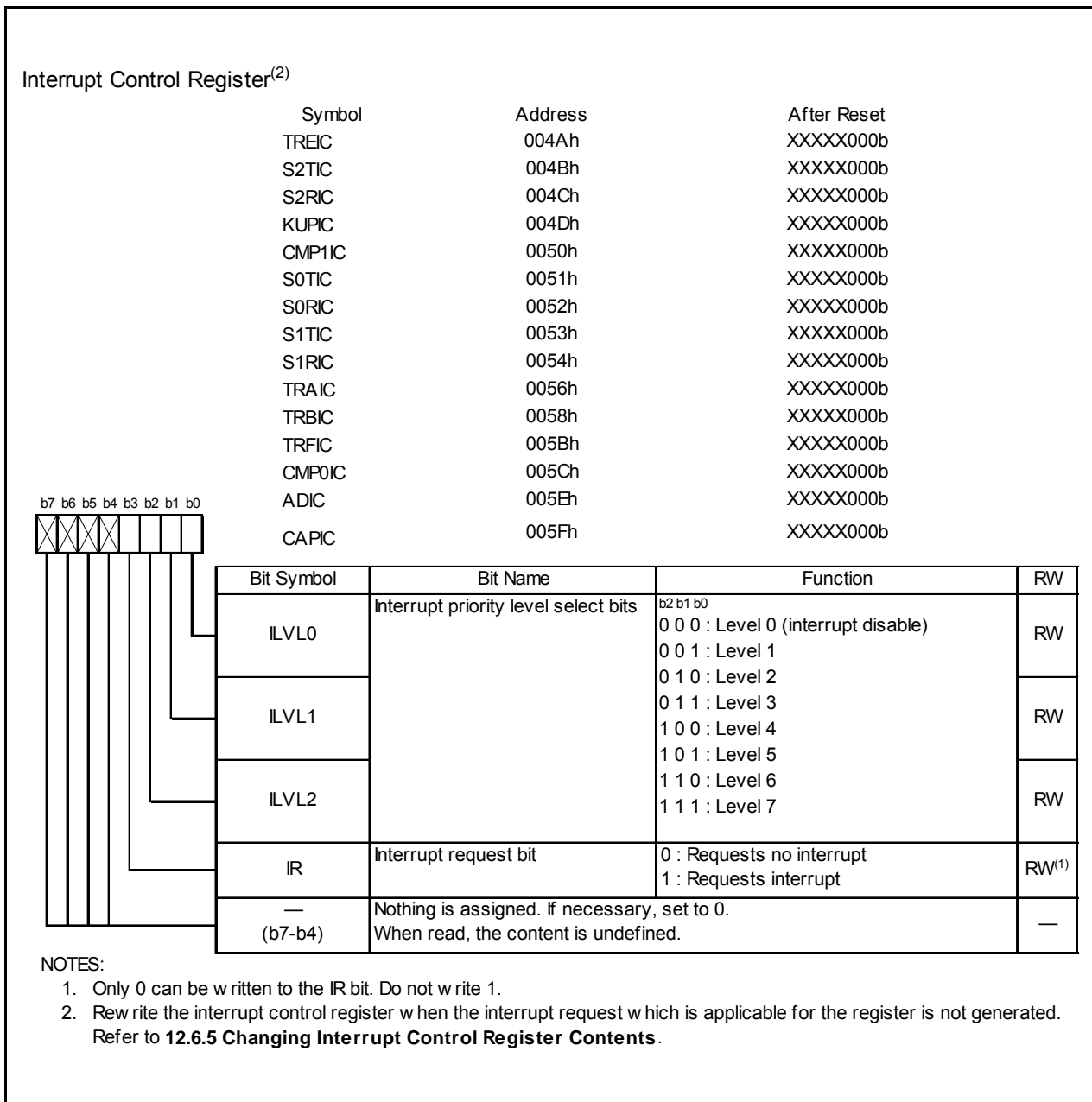


Figure 12.3 Interrupt Control Register

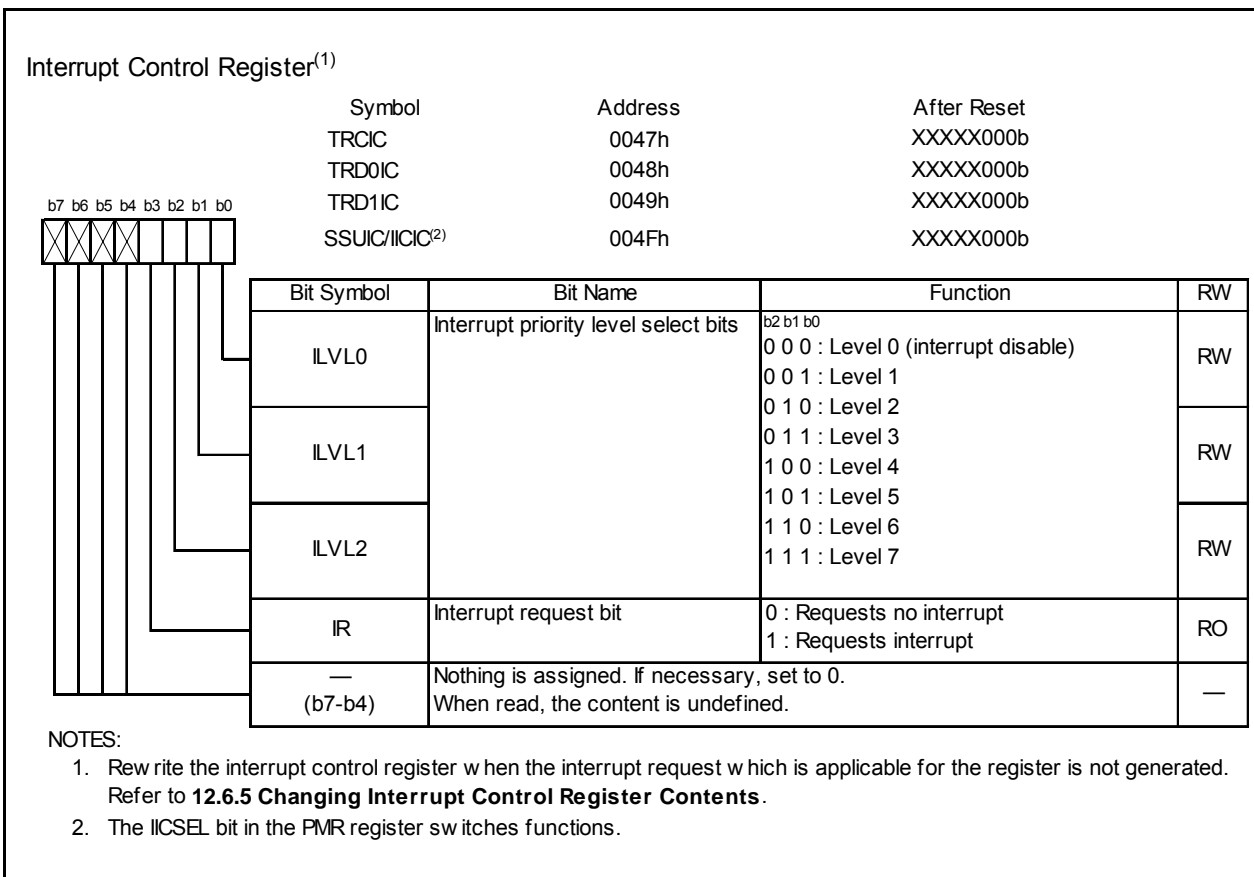


Figure 12.4 Registers TRCIC, TRD0IC, TRD1IC, SSUIC, and IICIC

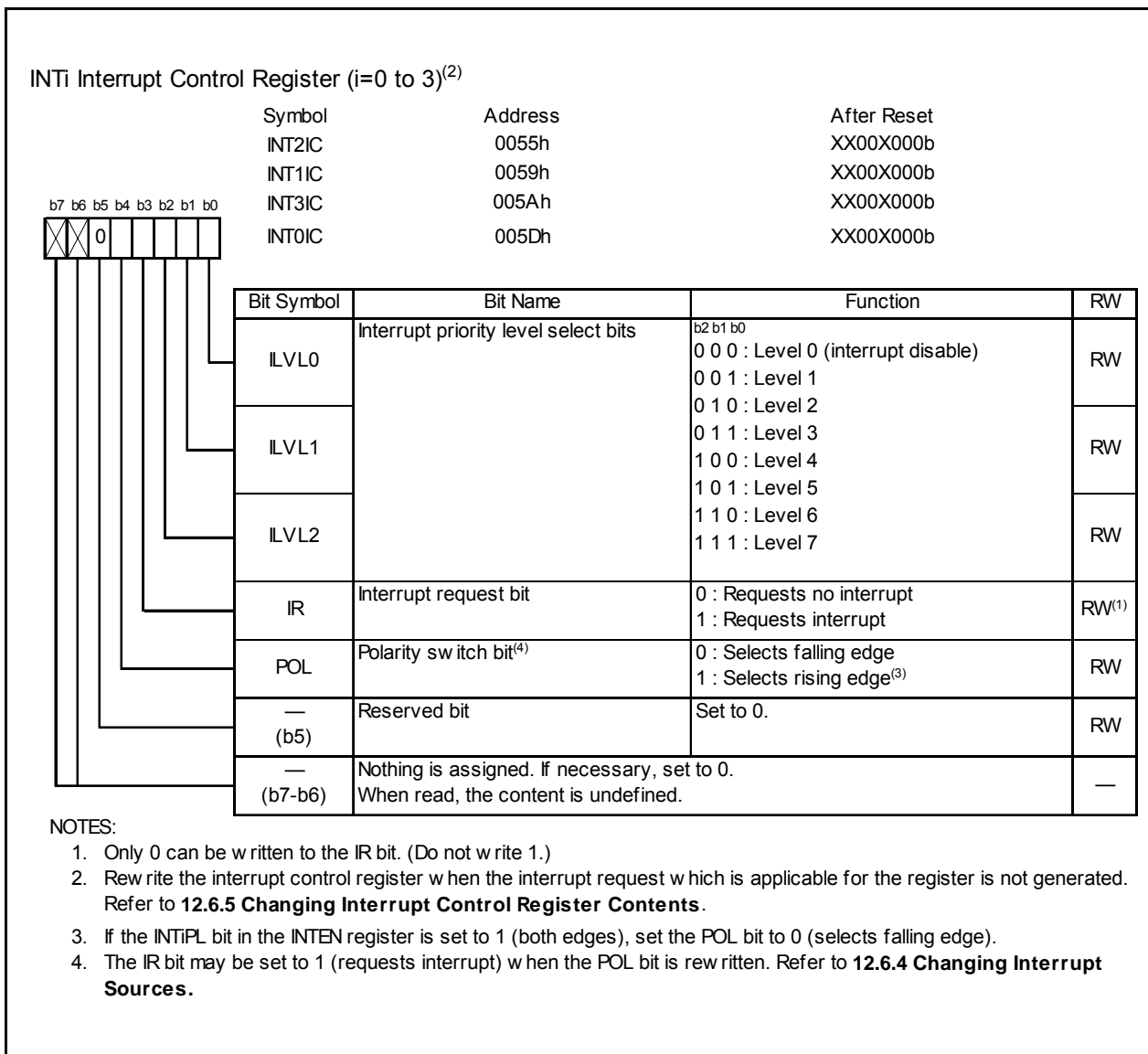


Figure 12.5 INTiIC Register

12.1.6.1 I Flag

The I flag enables or disables maskable interrupts. Setting the I flag to 1 (enabled) enables maskable interrupts. Setting the I flag to 0 (disabled) disables all maskable interrupts.

12.1.6.2 IR Bit

The IR bit is set to 1 (interrupt requested) when an interrupt request is generated. Then, when the interrupt request is acknowledged and the CPU branches to the corresponding interrupt vector, the IR bit is set to 0 (= interrupt not requested).

The IR bit can be set to 0 by a program. Do not write 1 to this bit.

However, the IR bit operations of the timer RD Interrupt, Clock Synchronous Serial I/O with Chip Select Interrupt and the I²C bus Interface Interrupt are different. Refer to **12.5 Timer RC Interrupt, Timer RD Interrupt, Clock Synchronous Serial I/O with Chip Select Interrupts, and I²C bus Interface Interrupt (Interrupts with Multiple Interrupt Request Sources)**.

12.1.6.3 Bits ILVL2 to ILVL0 and IPL

Interrupt priority levels can be set using bits ILVL2 to ILVL0.

Table 12.3 lists the Settings of Interrupt Priority Levels and Table 12.4 lists the Interrupt Priority Levels Enabled by IPL.

The following are conditions under which an interrupt is acknowledged:

- I flag = 1
- IR bit = 1
- Interrupt priority level > IPL

The I flag, IR bit, bits ILVL2 to ILVL0, and IPL are independent of each other. They do not affect one another.

Table 12.3 Settings of Interrupt Priority Levels


| ILVL2 to ILVL0 Bits | Interrupt Priority Level | Priority Order |
|---------------------|------------------------------|--|
| 000b | Level 0 (interrupt disabled) | – |
| 001b | Level 1 | Low  High |
| 010b | Level 2 | |
| 011b | Level 3 | |
| 100b | Level 4 | |
| 101b | Level 5 | |
| 110b | Level 6 | |
| 111b | Level 7 | |

Table 12.4 Interrupt Priority Levels Enabled by IPL

| IPL | Enabled Interrupt Priority Levels |
|------|--------------------------------------|
| 000b | Interrupt level 1 and above |
| 001b | Interrupt level 2 and above |
| 010b | Interrupt level 3 and above |
| 011b | Interrupt level 4 and above |
| 100b | Interrupt level 5 and above |
| 101b | Interrupt level 6 and above |
| 110b | Interrupt level 7 and above |
| 111b | All maskable interrupts are disabled |

12.1.6.4 Interrupt Sequence

An interrupt sequence is performed between an interrupt request acknowledgement and interrupt routine execution.

When an interrupt request is generated while an instruction is being executed, the CPU determines its interrupt priority level after the instruction is completed. The CPU starts the interrupt sequence from the following cycle. However, for the SMOVB, SMOVF, SSTR, or RMPA instruction if an interrupt request is generated while the instruction is being executed, the MCU suspends the instruction to start the interrupt sequence. The interrupt sequence is performed as indicated below.

Figure 12.6 shows the Time Required for Executing Interrupt Sequence.

- (1) The CPU gets interrupt information (interrupt number and interrupt request level) by reading address 00000h. The IR bit for the corresponding interrupt is set to 0 (interrupt not requested).⁽²⁾
- (2) The FLG register is saved to a temporary register⁽¹⁾ in the CPU immediately before entering the interrupt sequence.
- (3) The I, D and U flags in the FLG register are set as follows:
The I flag is set to 0 (interrupts disabled).
The D flag is set to 0 (single-step interrupt disabled).
The U flag is set to 0 (ISP selected).
However, the U flag does not change state if an INT instruction for software interrupt number 32 to 63 is executed.
- (4) The CPU's internal temporary register⁽¹⁾ is saved to the stack.
- (5) The PC is saved to the stack.
- (6) The interrupt priority level of the acknowledged interrupt is set in the IPL.
- (7) The starting address of the interrupt routine set in the interrupt vector is stored in the PC.

After the interrupt sequence is completed, instructions are executed from the starting address of the interrupt routine.

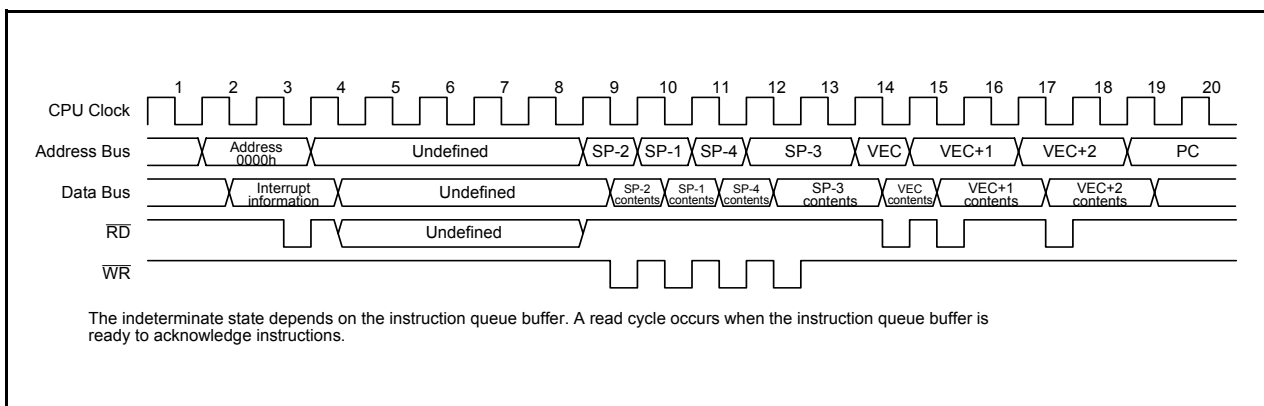


Figure 12.6 Time Required for Executing Interrupt Sequence

NOTES:

1. This register cannot be accessed by the user.
2. Refer to **12.5 Timer RC Interrupt, Timer RD Interrupt, Clock Synchronous Serial I/O with Chip Select Interrupts, and I²C bus Interface Interrupt (Interrupts with Multiple Interrupt Request Sources)** for the IR bit operations of the timer RC Interrupt, timer RD Interrupt, Clock Synchronous Serial I/O with Chip Select Interrupt, and the I²C bus Interface Interrupt.

12.1.6.5 Interrupt Response Time

Figure 12.7 shows the Interrupt Response Time. The interrupt response time is the period between an interrupt request generation and the execution of the first instruction in the interrupt routine. The interrupt response time includes the period between interrupt request generation and the completion of execution of the instruction (refer to (a) in Figure 12.7) and the period required to perform the interrupt sequence (20 cycles, refer to (b) in Figure 12.7).

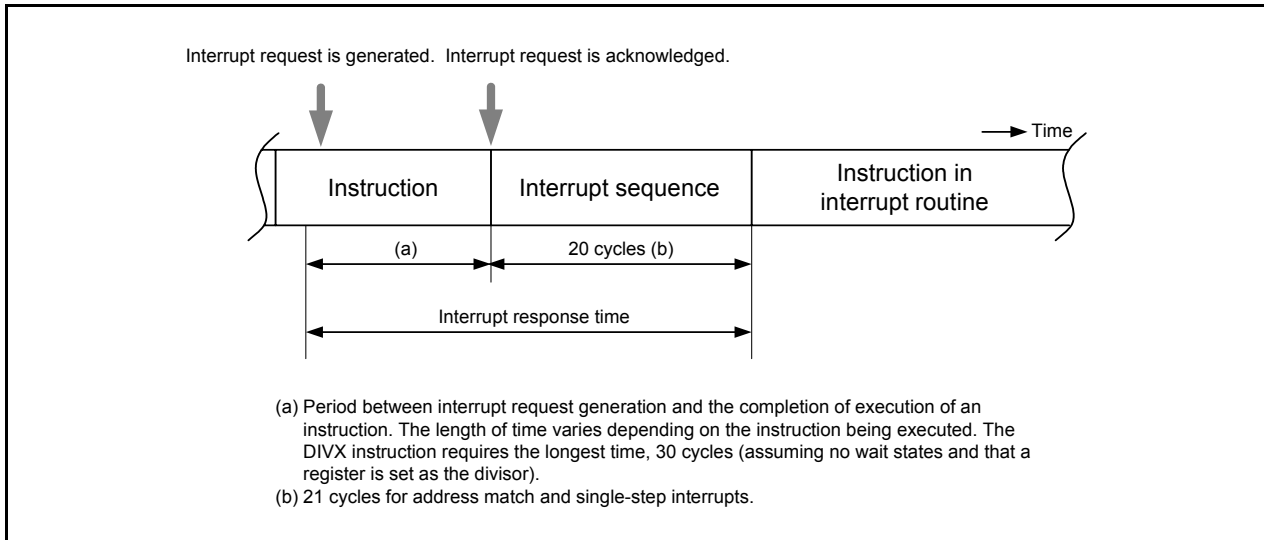


Figure 12.7 Interrupt Response Time

12.1.6.6 IPL Change when Interrupt Request is Acknowledged

When an interrupt request of a maskable interrupt is acknowledged, the interrupt priority level of the acknowledged interrupt is set in the IPL.

When a software interrupt or special interrupt request is acknowledged, the level listed in Table 12.5 is set in the IPL.

Table 12.5 lists the IPL Value When Software or Special Interrupt Is Acknowledged.

Table 12.5 IPL Value When Software or Special Interrupt Is Acknowledged

| Interrupt Source | Value Set in IPL |
|---|------------------|
| Watchdog timer, oscillation stop detection, voltage monitor 1, voltage monitor 2, Address break | 7 |
| Software, address match, single-step | Not changed |

12.1.6.8 Returning from an Interrupt Routine

When the REIT instruction is executed at the end of an interrupt routine, the FLG register and PC, which have been saved to the stack, are automatically restored. The program, that was running before the interrupt request was acknowledged, starts running again.

Restore registers saved by a program in an interrupt routine using the POPM instruction or others before executing the REIT instruction.

12.1.6.9 Interrupt Priority

If two or more interrupt requests are generated while a single instruction is being executed, the interrupt with the higher priority is acknowledged.

Set bits ILVL2 to ILVL0 to select the desired priority level for maskable interrupts (peripheral functions). However, if two or more maskable interrupts have the same priority level, their interrupt priority is resolved by hardware, and the higher priority interrupts acknowledged.

The priority levels of special interrupts, such as reset (reset has the highest priority) and watchdog timer, are set by hardware.

Figure 12.10 shows the Priority Levels of Hardware Interrupts.

The interrupt priority does not affect software interrupts. The MCU jumps to the interrupt routine when the instruction is executed.

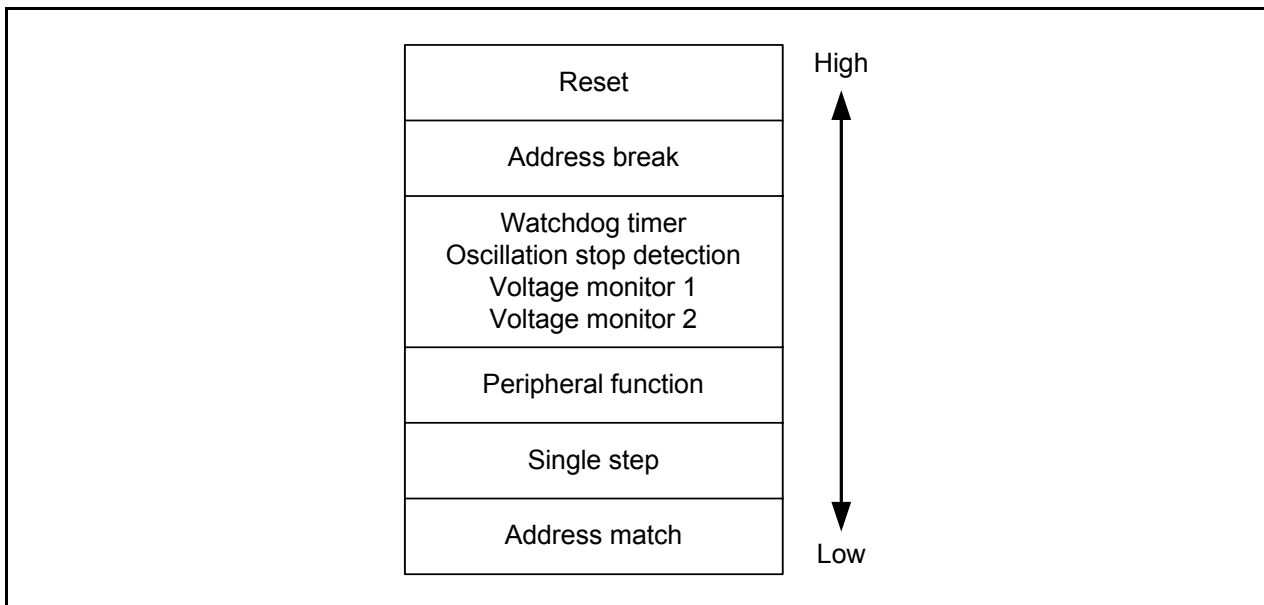


Figure 12.10 Priority Levels of Hardware Interrupts

12.1.6.10 Interrupt Priority Judgement Circuit

The interrupt priority judgement circuit selects the highest priority interrupt, as shown in Figure 12.11.

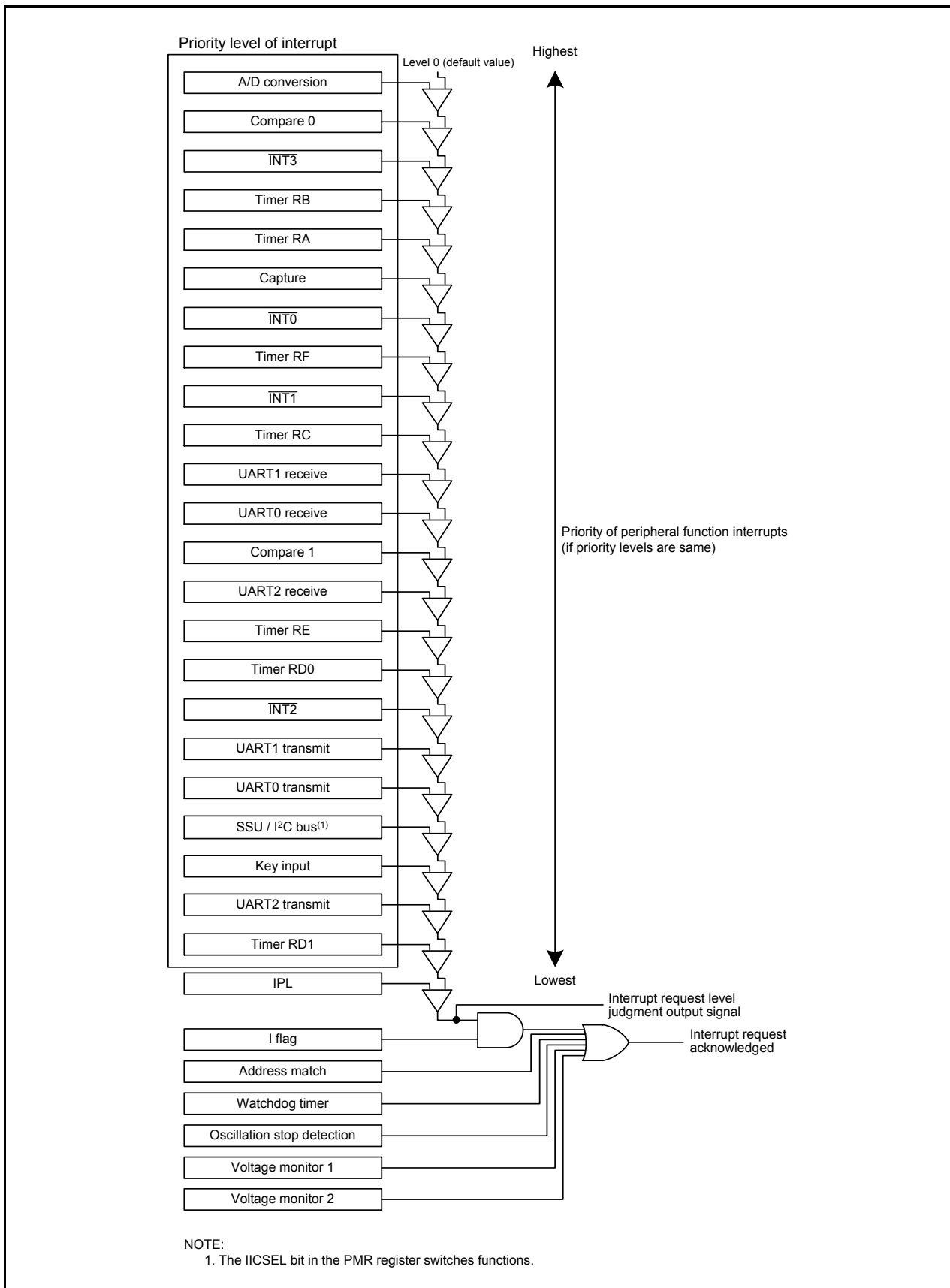


Figure 12.11 Interrupt Priority Level Judgement Circuit

12.2 $\overline{\text{INT}}$ Interrupt

12.2.1 $\overline{\text{INT}}_i$ Interrupt ($i = 0$ to 3)

The $\overline{\text{INT}}_i$ interrupt is generated by an $\overline{\text{INT}}_i$ input. When using the $\overline{\text{INT}}_i$ interrupt, the INT_iEN bit in the INTEN register is set to 1 (enable). The edge polarity is selected using the INT_iPL bit in the INTEN register and the POL bit in the INT_iIC register. The $\overline{\text{INT}}_1$ input and the $\overline{\text{INT}}_2$ input can select the input pin.

Inputs can be passed through a digital filter with three different sampling clocks.

The $\overline{\text{INT}}_0$ pin is shared with the pulse output forced cutoff of timer RC and timer RD, and the external trigger input of timer RB.

Figure 12.12 shows the PMR Register, Figure 12.13 shows the INTEN Register, Figure 12.14 shows the INTF Register, and Figure 12.15 shows the TRAIOC Register

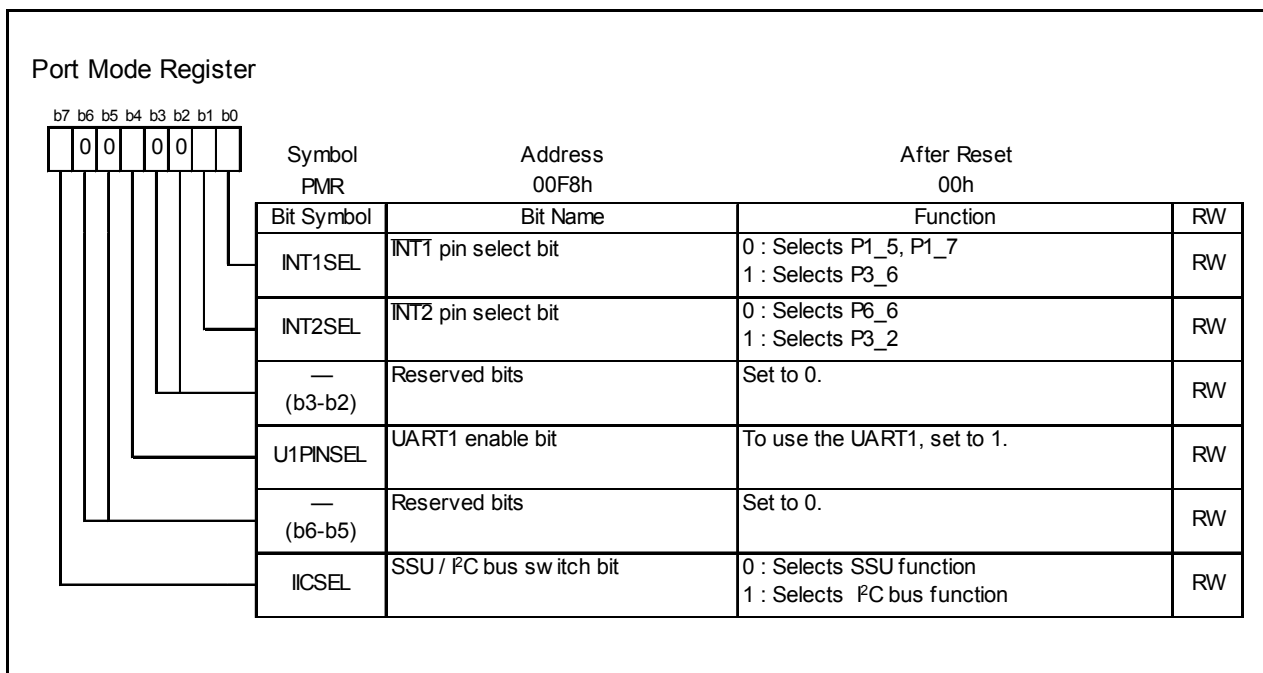


Figure 12.12 PMR Register

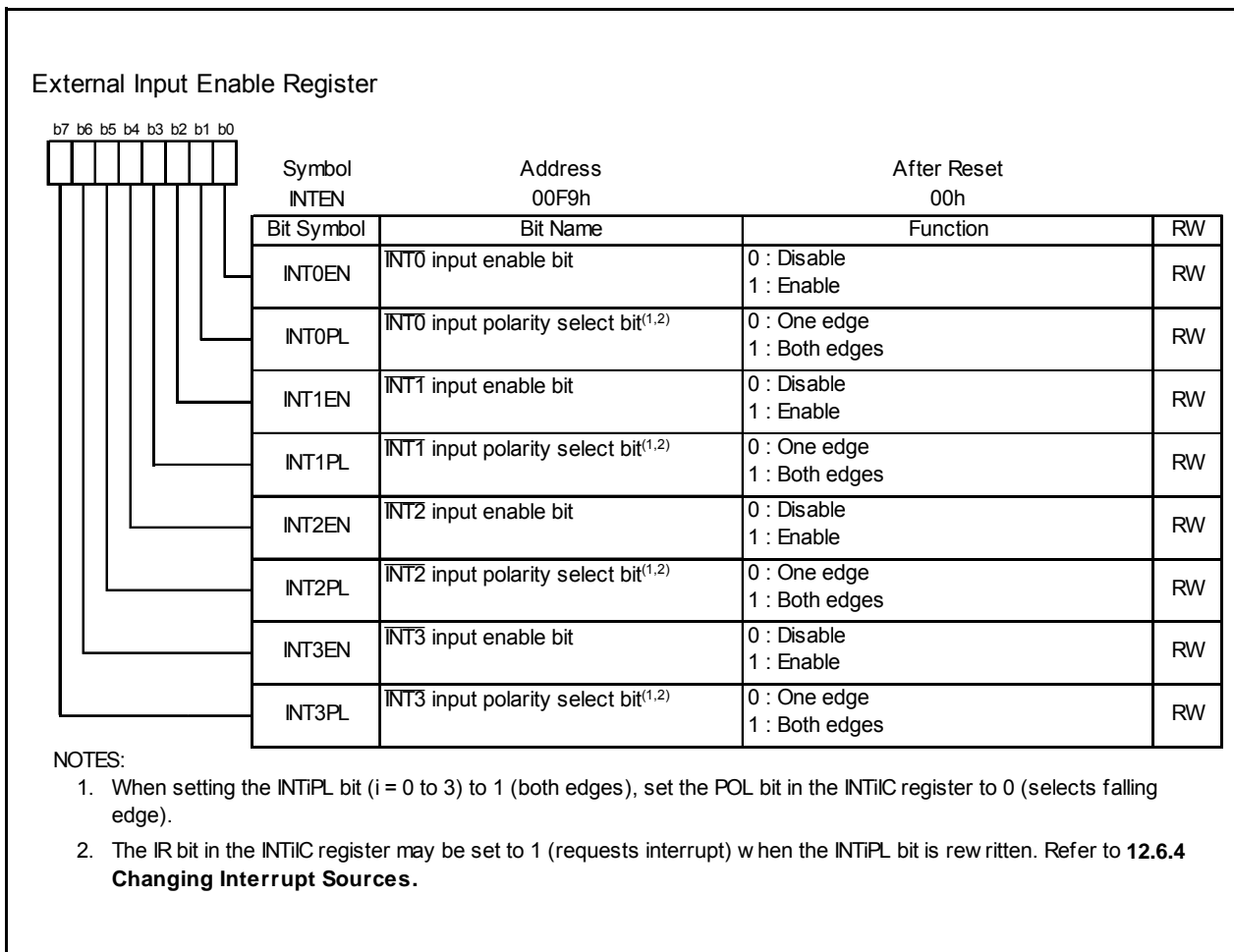


Figure 12.13 INTEN Register

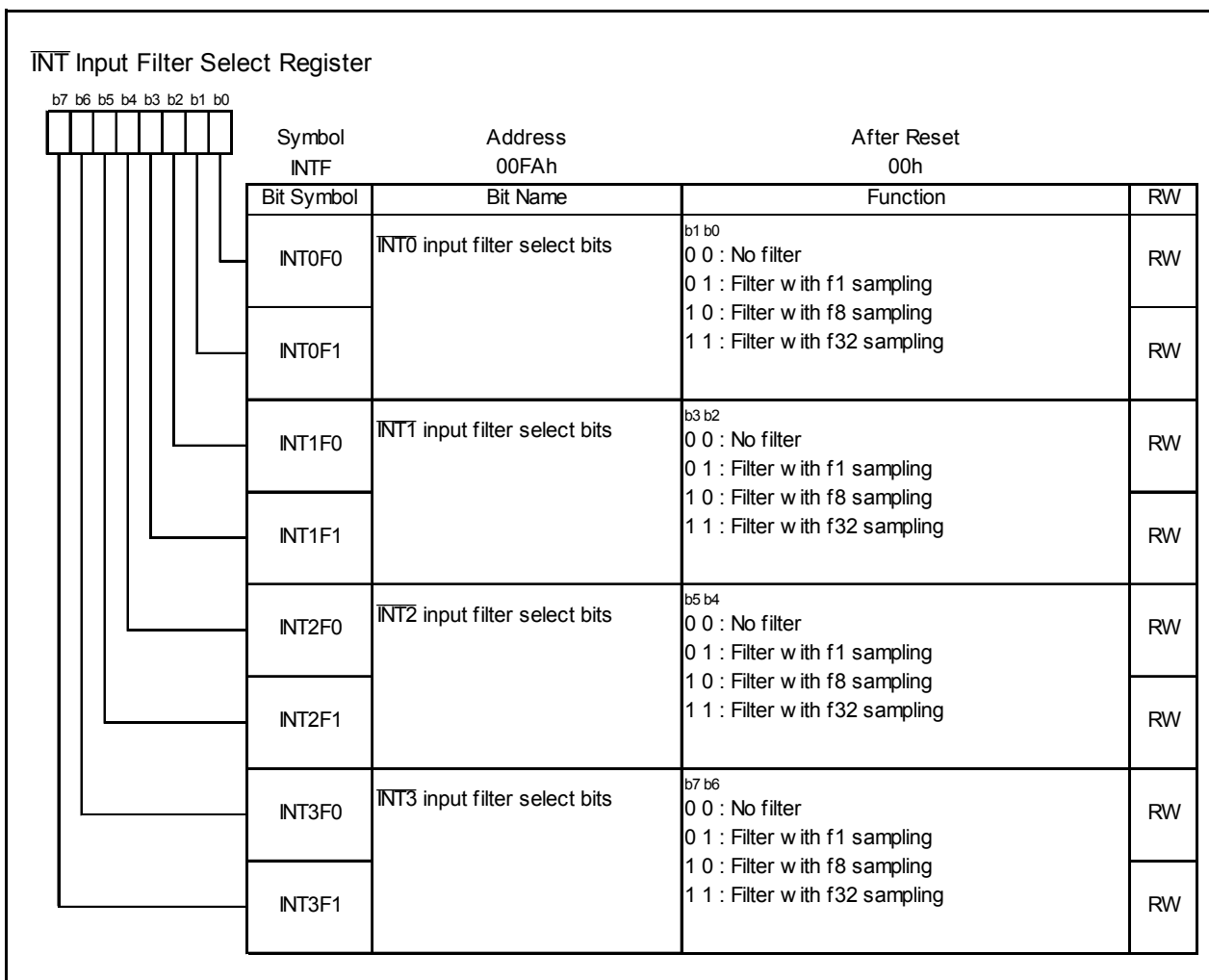


Figure 12.14 INTF Register

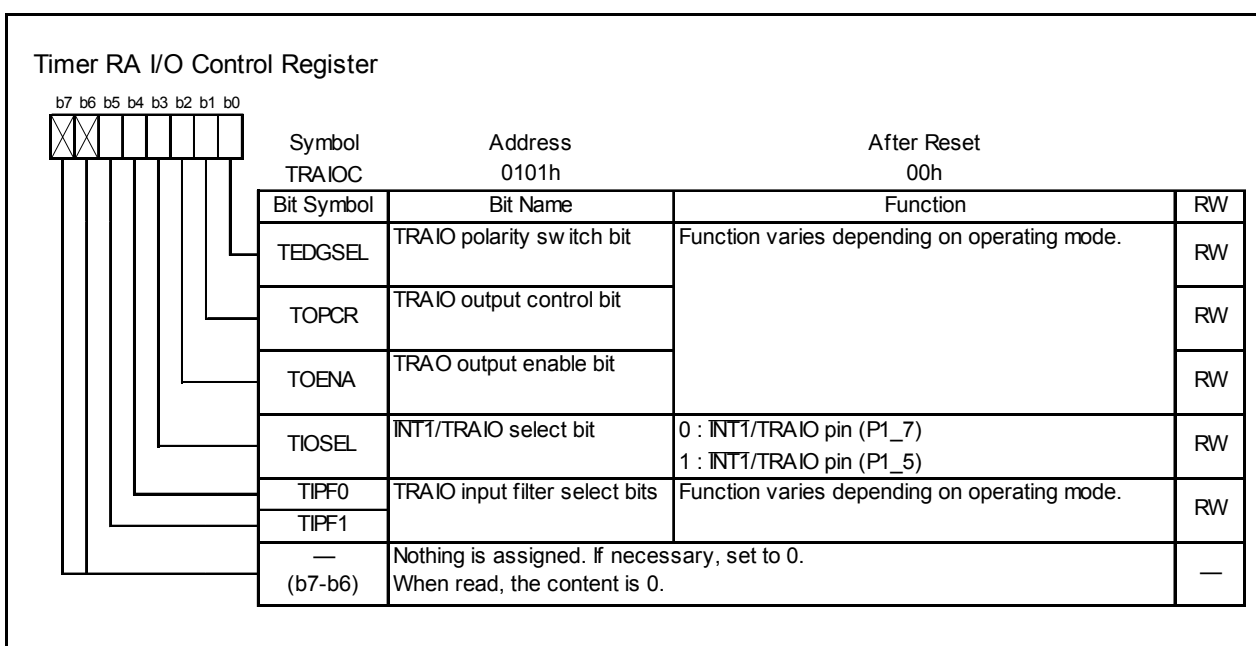


Figure 12.15 TRAIOC Register

12.2.2 $\overline{\text{INT}}_i$ Input Filter (i = 0 to 3)

The $\overline{\text{INT}}_i$ input contains a digital filter. The sampling clock is selected by bits INTiF1 to INTiF0 in the INTF register.

The $\overline{\text{INT}}_i$ level is sampled every sampling clock cycle and if the sampled input level matches three times, the IR bit in the INTiC register is set to 1 (interrupt requested).

Figure 12.16 shows the Configuration of $\overline{\text{INT}}_i$ Input Filter. Figure 12.17 shows an Operating Example of $\overline{\text{INT}}_i$ Input Filter.

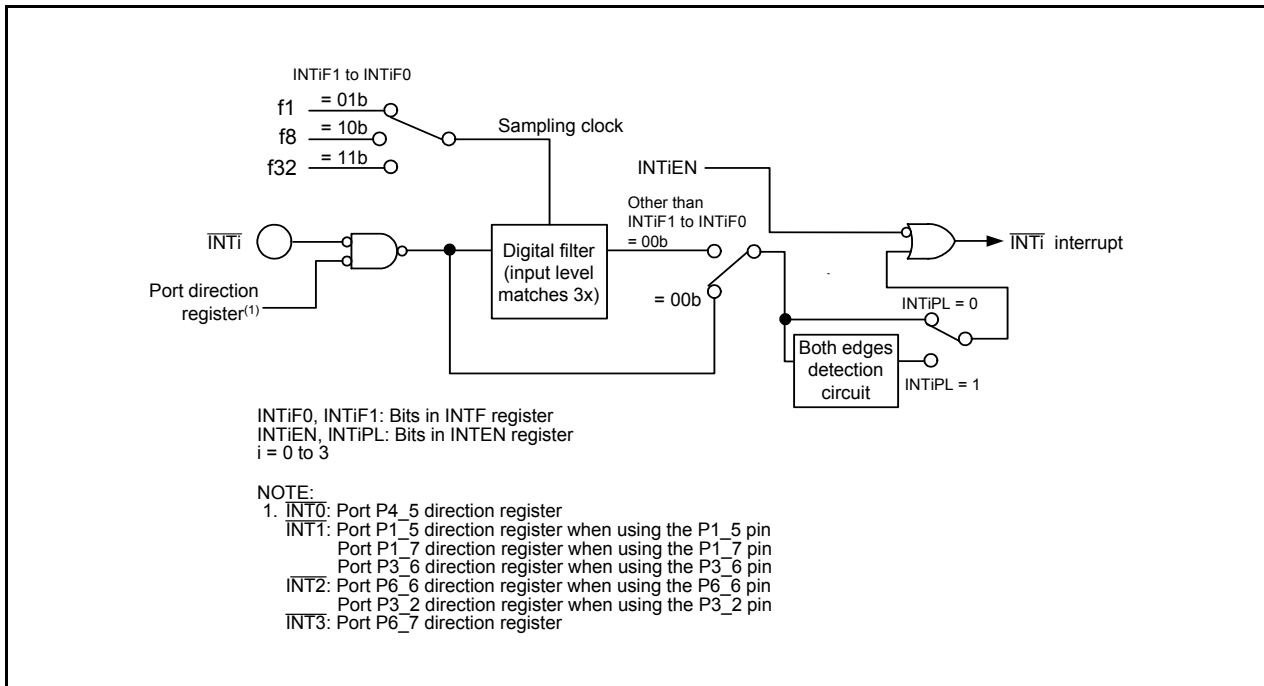


Figure 12.16 Configuration of $\overline{\text{INT}}_i$ Input Filter

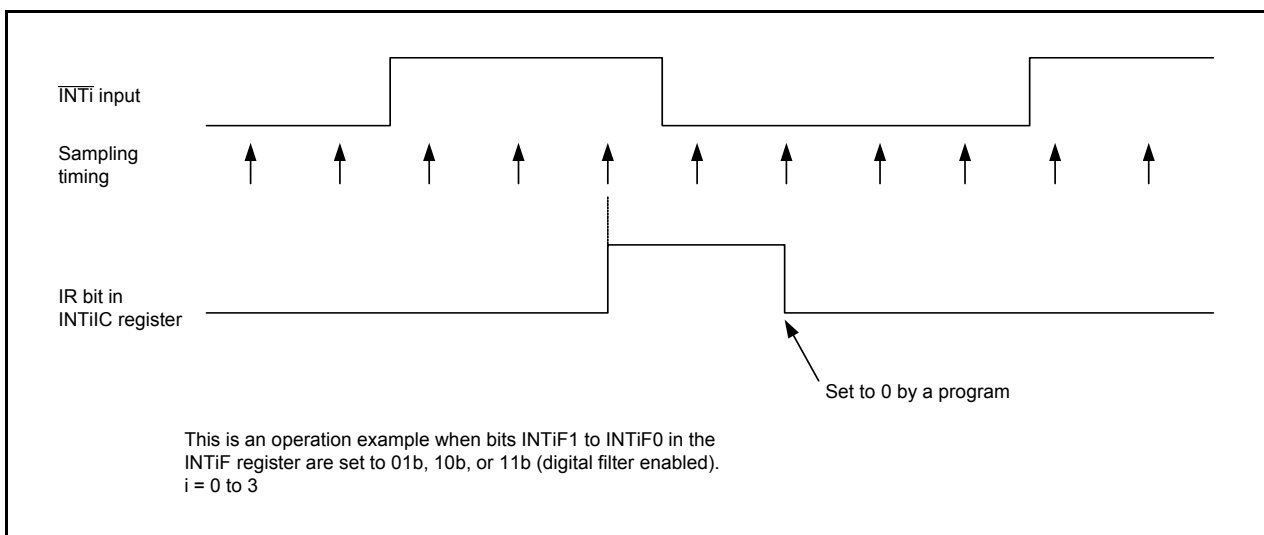


Figure 12.17 Operating Example of $\overline{\text{INT}}_i$ Input Filter

12.3 Key Input Interrupt

A key input interrupt request is generated by one of the input edges of pins $\overline{K10}$ to $\overline{K13}$. The key input interrupt can be used as a key-on wake-up function to exit wait or stop mode.

The $KIiEN$ ($i = 0$ to 3) bit in the $KIEN$ register can select whether or not the pins are used as \overline{KIi} input. The $KIiPL$ bit in the $KIEN$ register can select the input polarity.

When inputting “L” to the \overline{KIi} pin which sets the $KIiPL$ bit to 0 (falling edge), the input of the other pins $\overline{K10}$ to $\overline{K13}$ is not detected as interrupts. Also, when inputting “H” to the \overline{KIi} pin, which sets the $KIiPL$ bit to 1 (rising edge), the input of the other pins $\overline{K10}$ to $\overline{K13}$ is not detected as interrupts.

Figure 12.18 shows a Block Diagram of Key Input Interrupt.

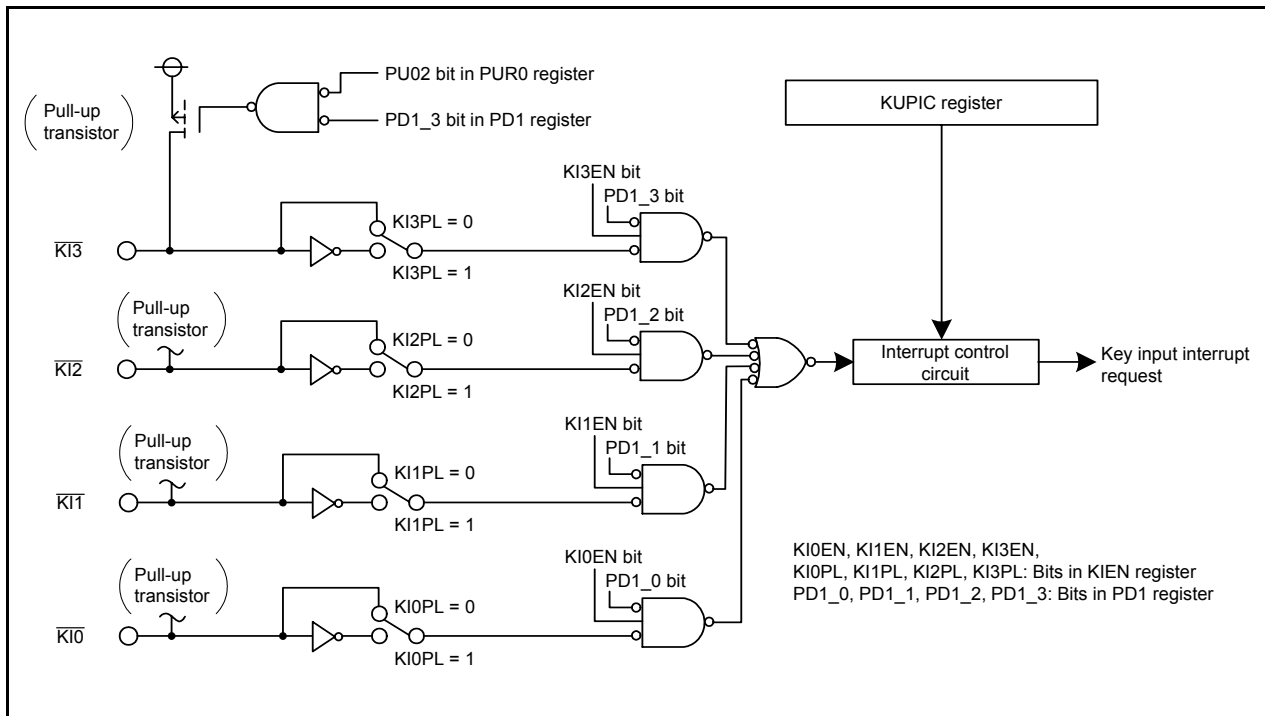


Figure 12.18 Block Diagram of Key Input Interrupt

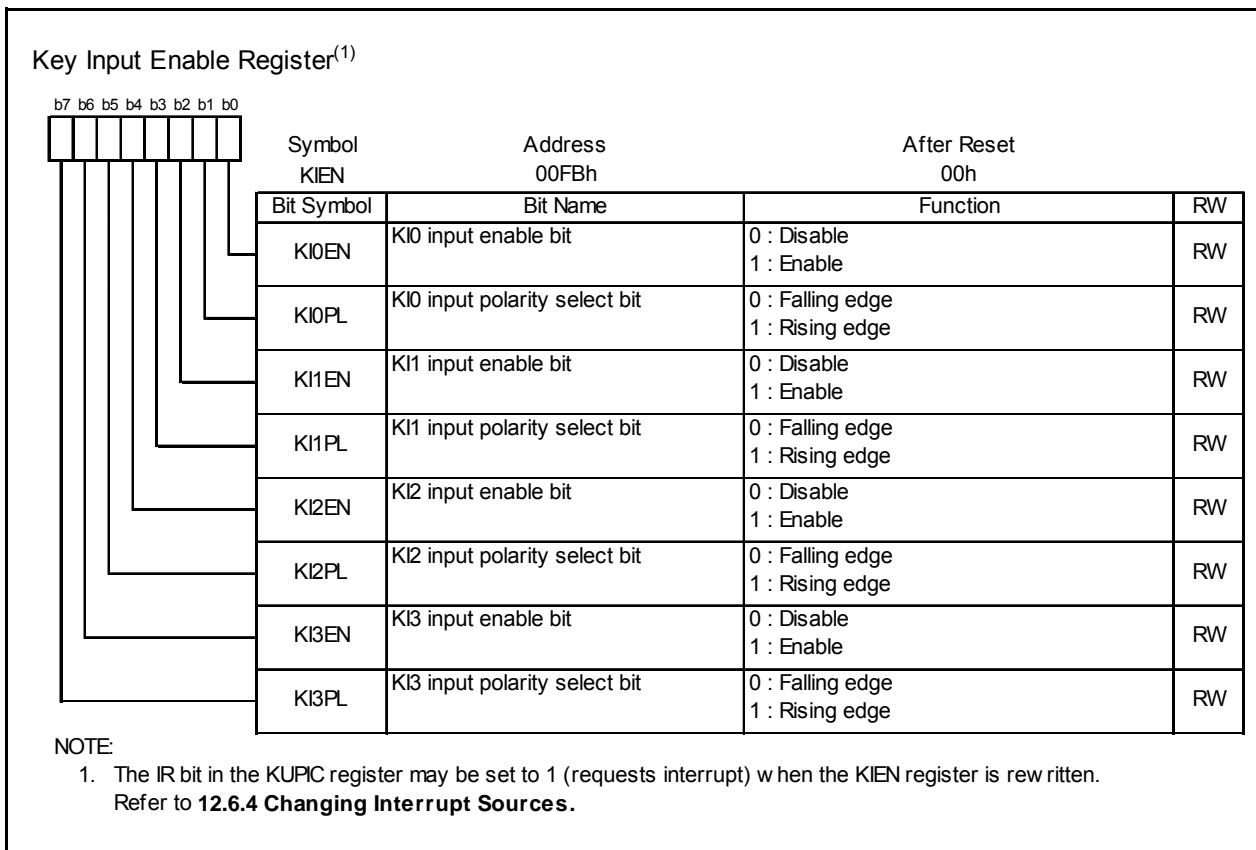


Figure 12.19 KIEN Register

12.4 Address Match Interrupt

An address match interrupt request is generated immediately before execution of the instruction at the address indicated by the RMADi register (i = 0 or 1). This interrupt is used as a break function by the debugger. When using the on-chip debugger, do not set an address match interrupt (registers of AIER, RMAD0, and RMAD1 and fixed vector tables) in a user system.

Set the starting address of any instruction in the RMADi register. Bits AIER0 and AIER1 in the AIER0 register can be used to select enable or disable of the interrupt. The I flag and IPL do not affect the address match interrupt.

The value of the PC (Refer to **12.1.6.7 Saving a Register** for the value of the PC) which is saved to the stack when an address match interrupt is acknowledged varies depending on the instruction at the address indicated by the RMADi register. (The appropriate return address is not saved on the stack.) When returning from the address match interrupt, return by one of the following means:

- Change the content of the stack and use the REIT instruction.
- Use an instruction such as POP to restore the stack as it was before the interrupt request was acknowledged. Then use a jump instruction.

Table 12.6 lists the Values of PC Saved to Stack when Address Match Interrupt is Acknowledged.

Figure 12.20 shows Registers AIER and RMAD0 to RMAD1.

Table 12.6 Values of PC Saved to Stack when Address Match Interrupt is Acknowledged

| Address Indicated by RMADi Register (i = 0 or 1) | PC Value Saved ⁽¹⁾ |
|--|---|
| <ul style="list-style-type: none"> • Instruction with 2-byte operation code⁽²⁾ • Instruction with 1-byte operation code⁽²⁾ ADD.B:S #IMM8,dest SUB.B:S #IMM8,dest AND.B:S #IMM8,dest OR.B:S #IMM8,dest MOV.B:S #IMM8,dest STZ #IMM8,dest STNZ #IMM8,dest STZX #IMM81,#IMM82,dest CMP.B:S #IMM8,dest PUSHM src POPM dest JMPS #IMM8 JSRS #IMM8 MOV.B:S #IMM,dest (however, dest = A0 or A1) | Address indicated by RMADi register + 2 |
| <ul style="list-style-type: none"> • Instructions other than the above | Address indicated by RMADi register + 1 |

NOTES:

1. Refer to the **12.1.6.7 Saving a Register** for the PC value saved.
2. Operation code: Refer to the **R8C/Tiny Series Software Manual (REJ09B0001)**.

Chapter 4. Instruction Code/Number of Cycles contains diagrams showing operation code below each syntax. Operation code is shown in the bold frame in the diagrams.

Table 12.7 Correspondence Between Address Match Interrupt Sources and Associated Registers

| Address Match Interrupt Source | Address Match Interrupt Enable Bit | Address Match Interrupt Register |
|--------------------------------|------------------------------------|----------------------------------|
| Address match interrupt 0 | AIER0 | RMAD0 |
| Address match interrupt 1 | AIER1 | RMAD1 |

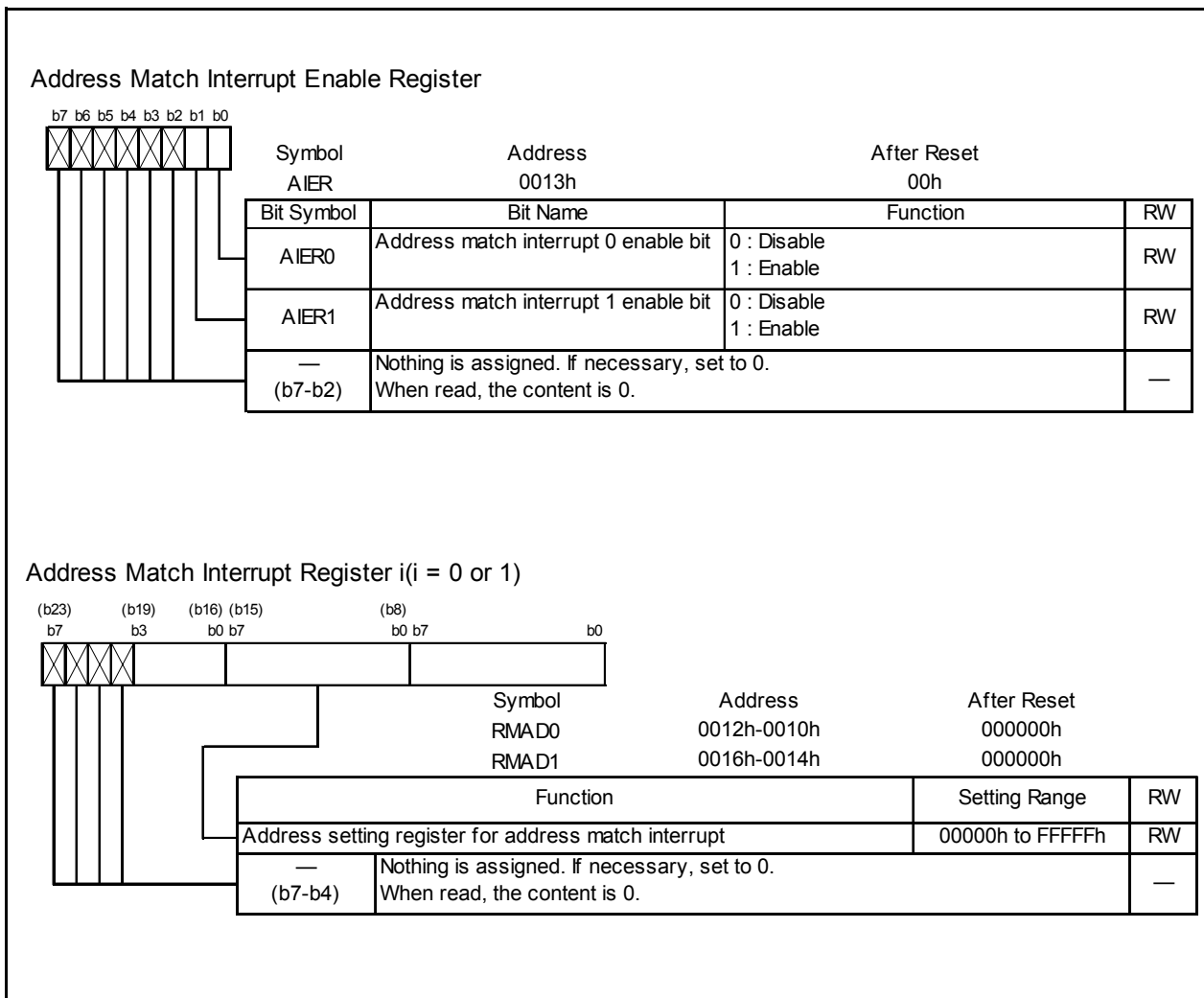


Figure 12.20 Registers AIER and RMAD0 to RMAD1

12.5 Timer RC Interrupt, Timer RD Interrupt, Clock Synchronous Serial I/O with Chip Select Interrupts, and I²C bus Interface Interrupt (Interrupts with Multiple Interrupt Request Sources)

The timer RC interrupt, timer RD (channel 0) interrupt, timer RD (channel 1) interrupt, clock synchronous serial I/O with chip select interrupt, and I²C bus interface interrupt each have multiple interrupt request sources. An interrupt request is generated by the logical OR of several interrupt request factors and is reflected in the IR bit in the corresponding interrupt control register. Therefore, each of these peripheral functions has its own interrupt request source status register (status register) and interrupt request source enable register (enable register) to control the generation of interrupt requests (change the IR bit in the interrupt control register). Table 12.8 lists the Registers Associated with Timer RC Interrupt, Timer RD Interrupt, Clock Synchronous Serial I/O with Chip Select Interrupt, and I²C bus Interface Interrupt and Figure 12.21 shows a Block Diagram of Timer RD Interrupt.

Table 12.8 Registers Associated with Timer RC Interrupt, Timer RD Interrupt, Clock Synchronous Serial I/O with Chip Select Interrupt, and I²C bus Interface Interrupt

| Peripheral Function Name | Status Register of Interrupt Request Source | Enable Register of Interrupt Request Source | Interrupt Control Register |
|---|---|---|----------------------------|
| Timer RC | TRCSR | TRCIER | TRCIC |
| Timer RD | Channel 0 | TRDSR0 | TRD0IC |
| | Channel 1 | TRDSR1 | TRD1IC |
| Clock synchronous serial I/O with chip select | SSSR | SSER | SSUIC |
| I ² C bus interface | ICSR | ICIER | IICIC |

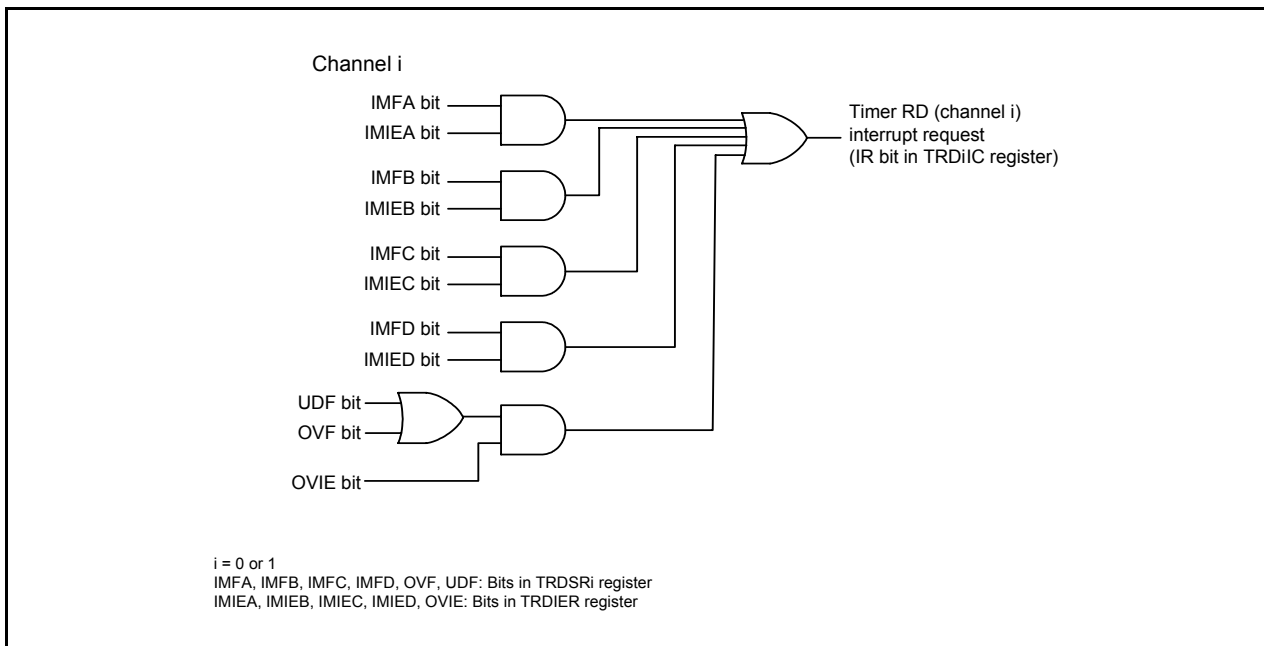


Figure 12.21 Block Diagram of Timer RD Interrupt

As with other maskable interrupts, the timer RC interrupt, timer RD (channel 0) interrupt, timer RD (channel 1) interrupt, clock synchronous serial I/O with chip select interrupt, and I²C bus interface interrupt are controlled by the combination of the I flag, IR bit, bits ILVL0 to ILVL2, and IPL. However, since each interrupt source is generated by a combination of multiple interrupt request sources, the following differences from other maskable interrupts apply:

- When bits in the enable register corresponding to bits set to 1 in the status register are set to 1 (enable interrupt), the IR bit in the interrupt control register is set to 1 (interrupt requested).
- When either bits in the status register or bits in the enable register corresponding to bits in the status register, or both, are set to 0, the IR bit is set to 0 (interrupt not requested). Basically, even though the interrupt is not acknowledged after the IR bit is set to 1, the interrupt request will not be maintained. Also, the IR bit is not set to 0 even if 0 is written to the IR bit.
- Individual bits in the status register are not automatically set to 0 even if the interrupt is acknowledged. Therefore, the IR bit is also not automatically set to 0 when the interrupt is acknowledged. Set each bit in the status register to 0 in the interrupt routine. Refer to the status register figure for how to set individual bits in the status register to 0.
- When multiple bits in the enable register are set to 1 and other request sources are generated after the IR bit is set to 1, the IR bit remains 1.
- When multiple bits in the enable register are set to 1, determine by the status register which request source causes an interrupt.

Refer to chapters of the individual peripheral functions (**14.3 Timer RC**, **14.4 Timer RD**, **16.2 Clock Synchronous Serial I/O with Chip Select (SSU)** and **16.3 I²C bus Interface**) for the status register and enable register.

Refer to **12.1.6 Interrupt Control** for the interrupt control register.

12.6 Notes on Interrupts

12.6.1 Reading Address 00000h

Do not read address 00000h by a program. When a maskable interrupt request is acknowledged, the CPU reads interrupt information (interrupt number and interrupt request level) from 00000h in the interrupt sequence. At this time, the acknowledged interrupt IR bit is set to 0.

If address 00000h is read by a program, the IR bit for the interrupt which has the highest priority among the enabled interrupts is set to 0. This may cause the interrupt to be canceled, or an unexpected interrupt to be generated.

12.6.2 SP Setting

Set any value in the SP before an interrupt is acknowledged. The SP is set to 0000h after reset. Therefore, if an interrupt is acknowledged before setting a value in the SP, the program may run out of control.

12.6.3 External Interrupt and Key Input Interrupt

Either “L” level or an “H” level of width shown in the Electrical Characteristics is necessary for the signal input to pins $\overline{\text{INT0}}$ to $\overline{\text{INT3}}$ and pins $\overline{\text{KI0}}$ to $\overline{\text{KI3}}$, regardless of the CPU clock.

For details, refer to **Table 21.22** (VCC = 5V), **Table 21.29** (VCC = 3V), **Table 21.36** (VCC = 2.2V) **External Interrupt INT_i (i = 0, 2, 3) Input** and **Table 21.19** (VCC = 5V), **Table 21.26** (VCC = 3V), **Table 21.33** (VCC = 2.2V) **TRAI0 Input, INT1 Input**.

12.6.4 Changing Interrupt Sources

The IR bit in the interrupt control register may be set to 1 (interrupt requested) when the interrupt source changes. When using an interrupt, set the IR bit to 0 (no interrupt requested) after changing the interrupt source. In addition, changes of interrupt sources include all factors that change the interrupt sources assigned to individual software interrupt numbers, polarities, and timing. Therefore, if a mode change of a peripheral function involves interrupt sources, edge polarities, and timing, set the IR bit to 0 (no interrupt requested) after the change. Refer to the individual peripheral function for its related interrupts.

Figure 12.22 shows an Example of Procedure for Changing Interrupt Sources.

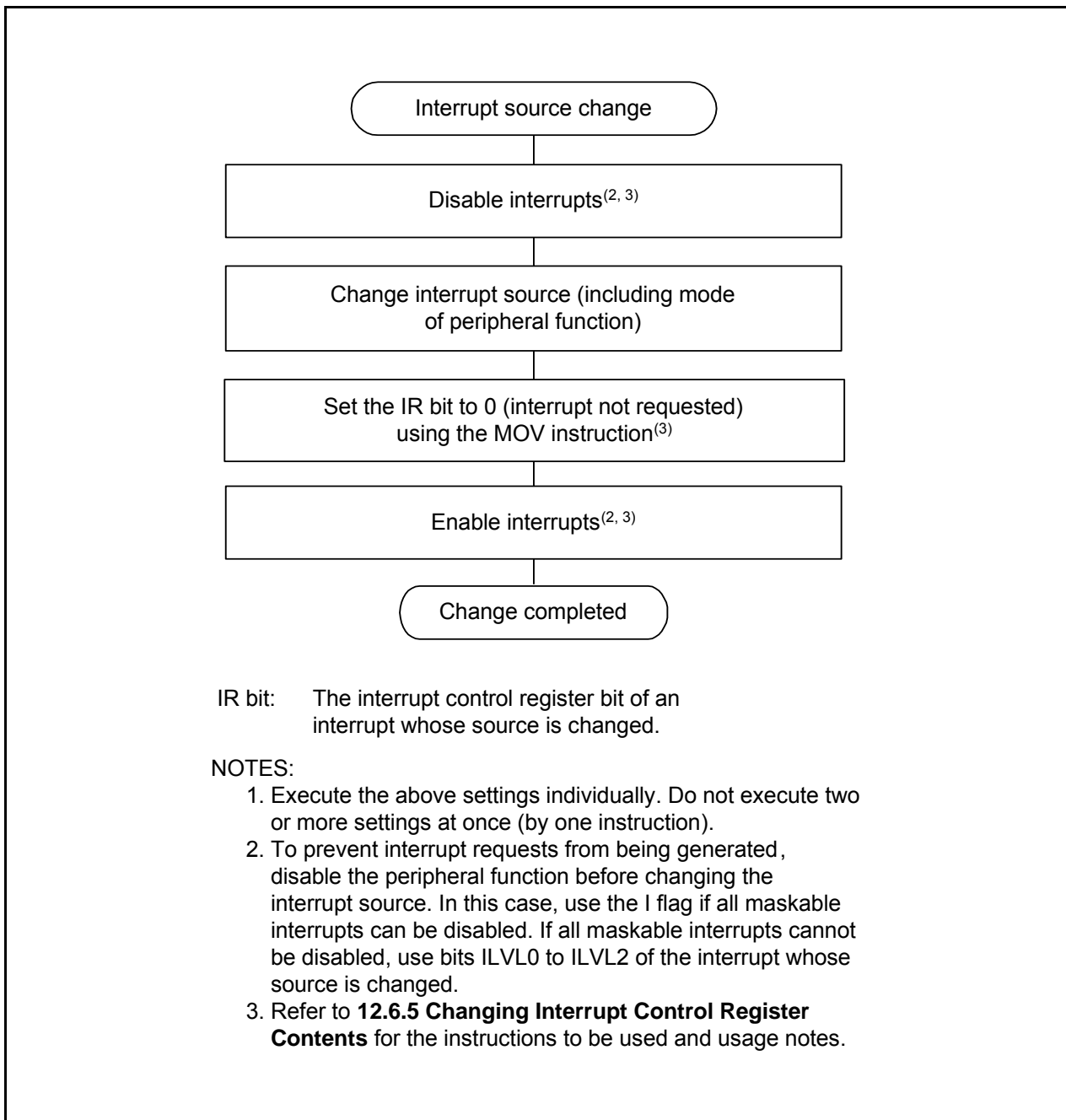


Figure 12.22 Example of Procedure for Changing Interrupt Sources

12.6.5 Changing Interrupt Control Register Contents

- (a) The contents of an interrupt control register can only be changed while no interrupt requests corresponding to that register are generated. If interrupt requests may be generated, disable interrupts before changing the interrupt control register contents.
- (b) When changing the contents of an interrupt control register after disabling interrupts, be careful to choose appropriate instructions.

Changing any bit other than IR bit

If an interrupt request corresponding to a register is generated while executing the instruction, the IR bit may not be set to 1 (interrupt requested), and the interrupt request may be ignored. If this causes a problem, use the following instructions to change the register: AND, OR, BCLR, BSET

Changing IR bit

If the IR bit is set to 0 (interrupt not requested), it may not be set to 0 depending on the instruction used. Therefore, use the MOV instruction to set the IR bit to 0.

- (c) When disabling interrupts using the I flag, set the I flag as shown in the sample programs below. Refer to (b) regarding changing the contents of interrupt control registers by the sample programs.

Sample programs 1 to 3 are for preventing the I flag from being set to 1 (interrupts enabled) before the interrupt control register is changed for reasons of the internal bus or the instruction queue buffer.

Example 1: Use NOP instructions to prevent I flag from being set to 1 before interrupt control register is changed

```
INT_SWITCH1:
  FCLR   I           ; Disable interrupts
  AND.B  #00H,0056H ; Set TRAIC register to 00h
  NOP                    ;
  NOP                    ;
  FSET   I           ; Enable interrupts
```

Example 2: Use dummy read to delay FSET instruction

```
INT_SWITCH2:
  FCLR   I           ; Disable interrupts
  AND.B  #00H,0056H ; Set TRAIC register to 00h
  MOV.W  MEM,R0      ; Dummy read
  FSET   I           ; Enable interrupts
```

Example 3: Use POPC instruction to change I flag

```
INT_SWITCH3:
  PUSHC  FLG
  FCLR   I           ; Disable interrupts
  AND.B  #00H,0056H ; Set TRAIC register to 00h
  POPC   FLG         ; Enable interrupts
```

13. Watchdog Timer

The watchdog timer is a function that detects when a program is out of control. Use of the watchdog timer is recommended to improve the reliability of the system. The watchdog timer contains a 15-bit counter and allows selection of count source protection mode enable or disable.

Table 13.1 lists information on the Watchdog Timer Specifications.

Refer to **5.6 Watchdog Timer Reset** for details on the watchdog timer.

Figure 13.1 shows the Block Diagram of Watchdog Timer. Figure 13.2 shows the Registers WDTR, WDTS, and WDC. Figure 13.3 shows the Registers CSPR and OFS.

Table 13.1 Watchdog Timer Specifications

| Item | Count Source Protection Mode Disabled | Count Source Protection Mode Enabled |
|------------------------------------|---|--------------------------------------|
| Count source | CPU clock | Low-speed on-chip oscillator clock |
| Count operation | Decrement | |
| Count start condition | Either of the following can be selected <ul style="list-style-type: none"> • After reset, count starts automatically • Count starts by writing to WDTS register | |
| Count stop condition | Stop mode, wait mode | None |
| Reset condition of watchdog timer | <ul style="list-style-type: none"> • Reset • Write 00h to the WDTR register before writing FFh • Underflow | |
| Operation at the time of underflow | Watchdog timer interrupt or watchdog timer reset | Watchdog timer reset |
| Select functions | <ul style="list-style-type: none"> • Division ratio of prescaler Selected by the WDC7 bit in the WDC register • Count source protection mode Whether count source protection mode is enabled or disabled after a reset can be selected by the CSPROINI bit in the OFS register (flash memory). If count source protection mode is disabled after a reset, it can be enabled or disabled by the CSPRO bit in the CSPR register (program). • Starts or stops of the watchdog timer after a reset Selected by the WDTON bit in the OFS register (flash memory). | |

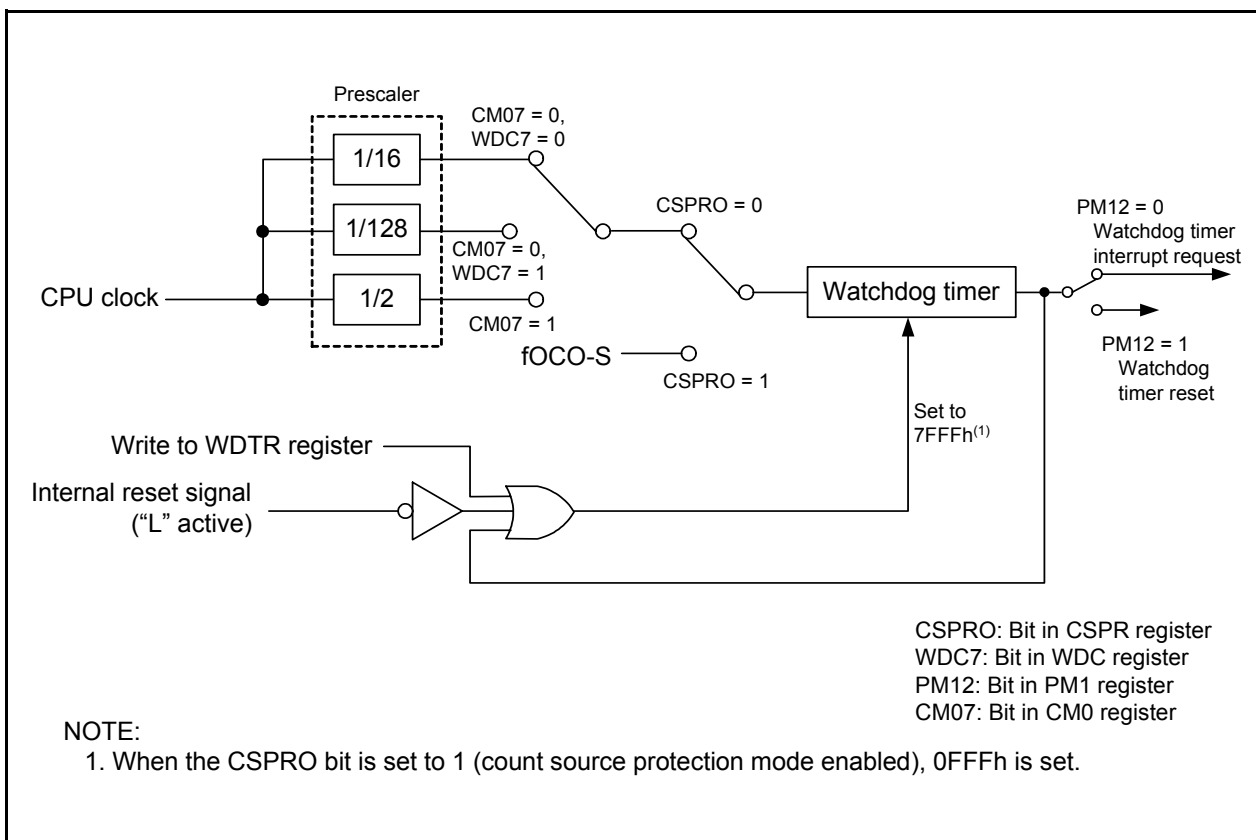


Figure 13.1 Block Diagram of Watchdog Timer

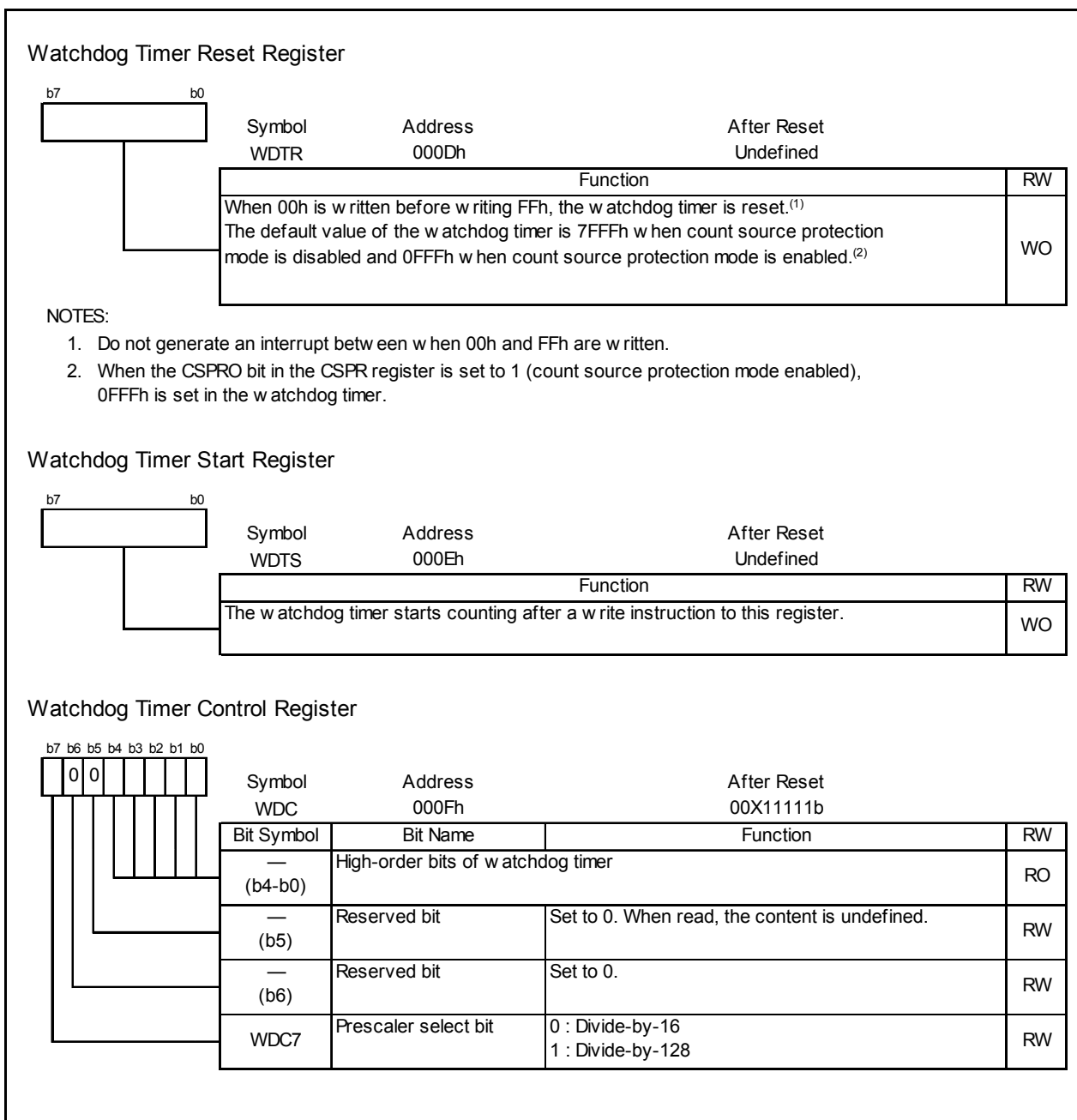


Figure 13.2 Registers WDTR, WDTS, and WDC

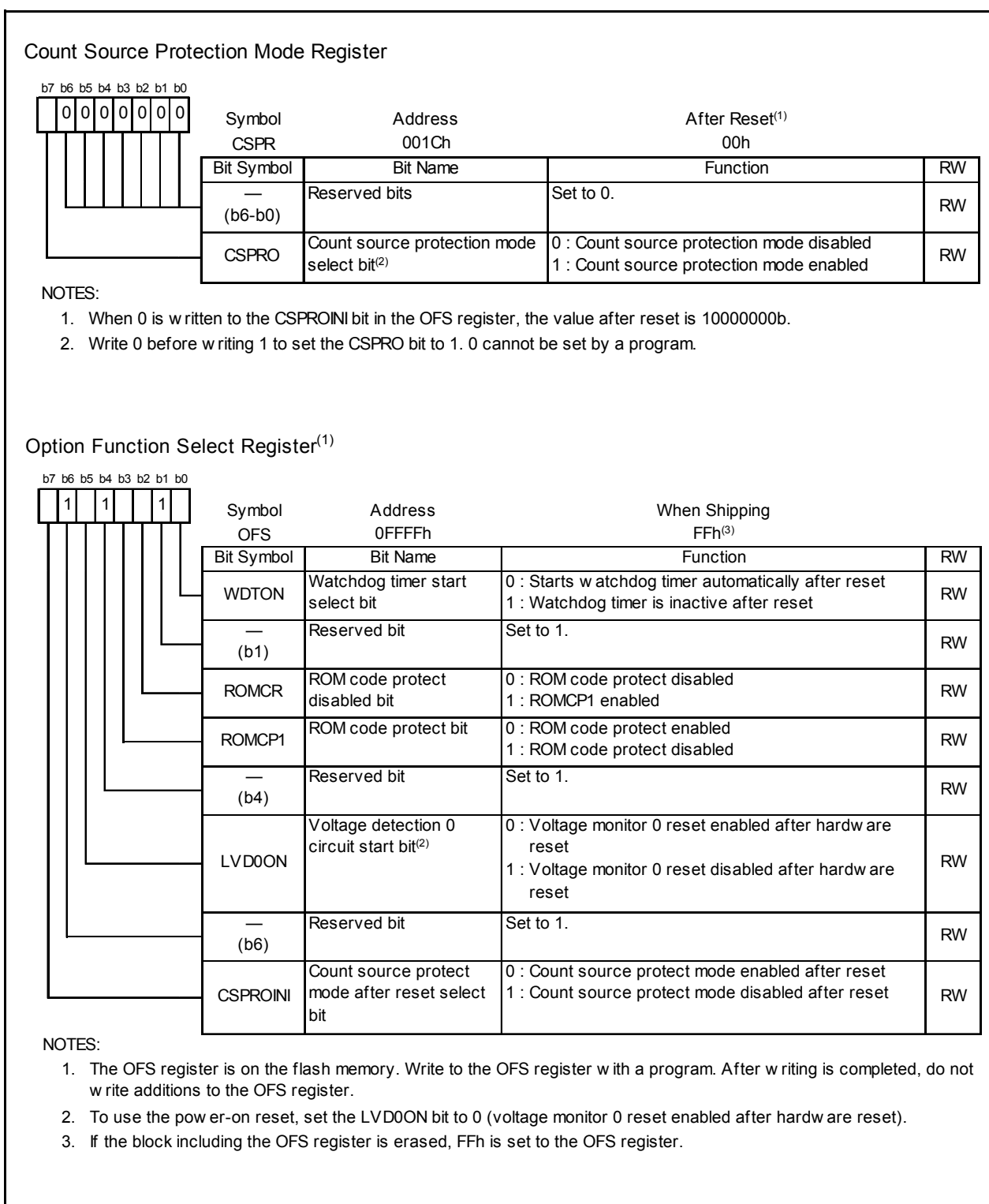


Figure 13.3 Registers CSPR and OFS

13.1 Count Source Protection Mode Disabled

The count source of the watchdog timer is the CPU clock when count source protection mode is disabled. Table 13.2 lists the Watchdog Timer Specifications (with Count Source Protection Mode Disabled).

Table 13.2 Watchdog Timer Specifications (with Count Source Protection Mode Disabled)

| Item | Specification |
|-----------------------------------|---|
| Count source | CPU clock |
| Count operation | Decrement |
| Period | $\text{Division ratio of prescaler (n)} \times \text{count value of watchdog timer (32768)}^{(1)}$ CPU clock n: 16 or 128 (selected by WDC7 bit in WDC register) Example: When the CPU clock frequency is 16 MHz and prescaler divided by 16, the period is approximately 32.8 ms |
| Reset condition of watchdog timer | <ul style="list-style-type: none"> • Reset • Write 00h to the WDTR register before writing FFh • Underflow |
| Count start condition | The WDTON bit ⁽²⁾ in the OFS register (0FFFFh) selects the operation of the watchdog timer after a reset <ul style="list-style-type: none"> • When the WDTON bit is set to 1 (watchdog timer is in stop state after reset) The watchdog timer and prescaler stop after a reset and the count starts when the WDTS register is written to • When the WDTON bit is set to 0 (watchdog timer starts automatically after exiting) • The watchdog timer and prescaler start counting automatically after a reset |
| Count stop condition | Stop and wait modes (inherit the count from the held value after exiting modes) |
| Operation at time of underflow | <ul style="list-style-type: none"> • When the PM12 bit in the PM1 register is set to 0 Watchdog timer interrupt • When the PM12 bit in the PM1 register is set to 1 Watchdog timer reset (refer to 5.6 Watchdog Timer Reset) |

NOTES:

1. The watchdog timer is reset when 00h is written to the WDTR register before FFh. The prescaler is reset after the MCU is reset. Some errors in the period of the watchdog timer may be caused by the prescaler.
2. The WDTON bit cannot be changed by a program. To set the WDTON bit, write 0 to bit 0 of address 0FFFFh with a flash programmer.

13.2 Count Source Protection Mode Enabled

The count source of the watchdog timer is the low-speed on-chip oscillator clock when count source protection mode is enabled. If the CPU clock stops when a program is out of control, the clock can still be supplied to the watchdog timer.

Table 13.3 lists the Watchdog Timer Specifications (with Count Source Protection Mode Enabled).

Table 13.3 Watchdog Timer Specifications (with Count Source Protection Mode Enabled)

| Item | Specification |
|-----------------------------------|--|
| Count source | Low-speed on-chip oscillator clock |
| Count operation | Decrement |
| Period | Count value of watchdog timer (4096) Low-speed on-chip oscillator clock Example: Period is approximately 32.8 ms when the low-speed on-chip oscillator clock frequency is 125 kHz |
| Reset condition of watchdog timer | <ul style="list-style-type: none"> • Reset • Write 00h to the WDTR register before writing FFh • Underflow |
| Count start condition | <p>The WDTON bit⁽¹⁾ in the OFS register (0FFFFh) selects the operation of the watchdog timer after a reset.</p> <ul style="list-style-type: none"> • When the WDTON bit is set to 1 (watchdog timer is in stop state after reset) The watchdog timer and prescaler stop after a reset and the count starts when the WDTS register is written to • When the WDTON bit is set to 0 (watchdog timer starts automatically after reset) The watchdog timer and prescaler start counting automatically after a reset |
| Count stop condition | None (The count does not stop in wait mode after the count starts. The MCU does not enter stop mode.) |
| Operation at time of underflow | Watchdog timer reset (Refer to 5.6 Watchdog Timer Reset.) |
| Registers, bits | <ul style="list-style-type: none"> • When setting the CSPPRO bit in the CSPR register to 1 (count source protection mode is enabled)⁽²⁾, the following are set automatically <ul style="list-style-type: none"> - Set 0FFFh to the watchdog timer - Set the CM14 bit in the CM1 register to 0 (low-speed on-chip oscillator on) - Set the PM12 bit in the PM1 register to 1 (The watchdog timer is reset when watchdog timer underflows) • The following conditions apply in count source protection mode <ul style="list-style-type: none"> - Writing to the CM10 bit in the CM1 register is disabled (It remains unchanged even if it is set to 1. The MCU does not enter stop mode.) - Writing to the CM14 bit in the CM1 register is disabled (It remains unchanged even if it is set to 1. The low-speed on-chip oscillator does not stop.) |

NOTES:

1. The WDTON bit cannot be changed by a program. To set the WDTON bit, write 0 to bit 0 of address 0FFFFh with a flash programmer.
2. Even if 0 is written to the CSPROINI bit in the OFS register, the CSPRO bit is set to 1. The CSPROINI bit cannot be changed by a program. To set the CSPROINI bit, write 0 to bit 7 of address 0FFFFh with a flash programmer.

14. Timers

The MCU has two 8-bit timers with 8-bit prescalers, three 16-bit timers, and a timer with a 4-bit counter and an 8-bit counter. The two 8-bit timers with 8-bit prescalers are timer RA and timer RB. These timers contain a reload register to store the default value of the counter. The three 16-bit timers is timer RC, timer RD, and timer RF, and have input capture and output compare functions. The 4-bit and 8-bit counters are timer RE, and has an output compare function. All the timers operate independently.

Tables 14.1 and 14.2 list Functional Comparison of Timers.

Table 14.1 Functional Comparison of Timers (1)

| Item | | Timer RA | Timer RB | Timer RC | Timer RD | Timer RE | Timer RF |
|---------------|---|--|--|--|---|--|---|
| Configuration | | 8-bit timer with 8-bit prescaler (with reload register) | 8-bit timer with 8-bit prescaler (with reload register) | 16-bit timer (with input capture and output compare) | 16-bit timer × 2 (with input capture and output compare) | 4-bit counter 8-bit counter | 16-bit timer (with input capture and output compare) |
| Count | | Decrement | Decrement | Increment | Increment/ Decrement | Increment | Increment |
| Count sources | | <ul style="list-style-type: none"> • f1 • f2 • f8 • fOCO • fC32 | <ul style="list-style-type: none"> • f1 • f2 • f8 • Timer RA underflow | <ul style="list-style-type: none"> • f1 • f2 • f4 • f8 • f32 • fOCO40M • TRCCLK | <ul style="list-style-type: none"> • f1 • f2 • f4 • f8 • f32 • fOCO40M • TRDIOAO | <ul style="list-style-type: none"> • f4 • f8 • f32 • fC4 | <ul style="list-style-type: none"> • f1 • f8 • f32 |
| Function | Count of the internal count source | Timer mode | Timer mode | Timer mode (output compare function) | Timer mode (output compare function) | — | Output compare mode |
| | Count of the external count source | Event counter mode | — | Timer mode (output compare function) | Timer mode (output compare function) | — | — |
| | External pulse width/period measurement | Pulse width measurement mode, pulse period measurement mode | — | Timer mode (input capture function; 4 pins) | Timer mode (input capture function; 2 channels × 4 pins) | — | Input capture mode |
| | PWM output | Pulse output mode ⁽¹⁾ , Event counter mode ⁽¹⁾ | Programmable waveform generation mode | Timer mode (output compare function; 4 pins) ⁽¹⁾ , PWM mode (3 pins), PWM2 mode (1 pin) | Timer mode (output compare function; 2 channels × 4 pins) ⁽¹⁾ , PWM mode (2 channels × 3 pins), PWM3 mode (2 channels × 2 pins) | Output compare mode ⁽¹⁾ | Output compare mode |
| | One-shot waveform output | — | Programmable one-shot generation mode, Programmable wait one-shot generation mode | PWM mode (3 pins) | PWM mode (2 channels × 3 pins) | — | — |
| | Three-phase waveforms output | — | — | — | Reset synchronous PWM mode (2 channels × 3 pins, Sawtooth wave modulation), Complementary PWM mode (2 channels × 3 pins, triangular wave modulation, dead time) | — | — |
| | Timer | Timer mode (only fC32 count) | — | — | — | Real-time clock mode | — |

NOTE:

1. Rectangular waves are output in these modes. Since the waves are inverted at each overflow, the “H” and “L” level widths of the pulses are the same.

Table 14.2 Functional Comparison of Timers (2)

| Item | Timer RA | Timer RB | Timer RC | Timer RD | Timer RE | Timer RF |
|-------------------|--|--|--|--|-----------------------|--|
| Input pin | TRAIO | $\overline{\text{INT0}}$ | $\overline{\text{INT0}}$, TRCCLK, TRCTRG, TRCIOA, TRCIOB, TRCIOC, TRCIOD | $\overline{\text{INT0}}$, TRDCLK, TRDIOA0, TRDIOA1, TRDIOB0, TRDIOB1, TRDIOC0, TRDIOC1, TRDIOD0, TRDIOD1 | — | TRFI |
| Output pin | TRAO TRAIO | TRBO | TRCIOA, TRCIOB, TRCIOC, TRCIOD | TRDIOA0, TRDIOA1, TRDIOB0, TRDIOB1, TRDIOC0, TRDIOC1, TRDIOD0, TRDIOD1 | TREO | TRFO00 to TRFO02, TRFO10 to TRFO12 |
| Related interrupt | Timer RA interrupt, INT1 interrupt | Timer RB interrupt, INT0 interrupt | Compare match/ input capture A to D interrupt, Overflow interrupt, INT0 interrupt | Compare match/ input capture A0 to D0 interrupt, Compare match/ input capture A1 to D1 interrupt, Overflow interrupt, Underflow interrupt ⁽¹⁾ , INT0 interrupt | Timer RE interrupt | Timer RF interrupt, Compare 0 interrupt, Compare 1 interrupt, |
| Timer stop | Provided | Provided | Provided | Provided | Provided | Provided |

NOTE:

1. The underflow interrupt can be set to channel 1.

14.1 Timer RA

Timer RA is an 8-bit timer with an 8-bit prescaler.

The prescaler and timer each consist of a reload register and counter. The reload register and counter are allocated at the same address, and can be accessed when accessing registers TRAPRE and TRA (refer to **Tables 14.3 to 14.7 the Specifications of Each Mode**).

The count source for timer RA is the operating clock that regulates the timing of timer operations such as counting and reloading.

Figure 14.1 shows a Block Diagram of Timer RA. Figures 14.2 and 14.3 show the registers associated with Timer RA.

Timer RA has the following five operating modes:

- Timer mode: The timer counts the internal count source.
- Pulse output mode: The timer counts the internal count source and outputs pulses of which polarity inverted by underflow of the timer.
- Event counter mode: The timer counts external pulses.
- Pulse width measurement mode: The timer measures the pulse width of an external pulse.
- Pulse period measurement mode: The timer measures the pulse period of an external pulse.

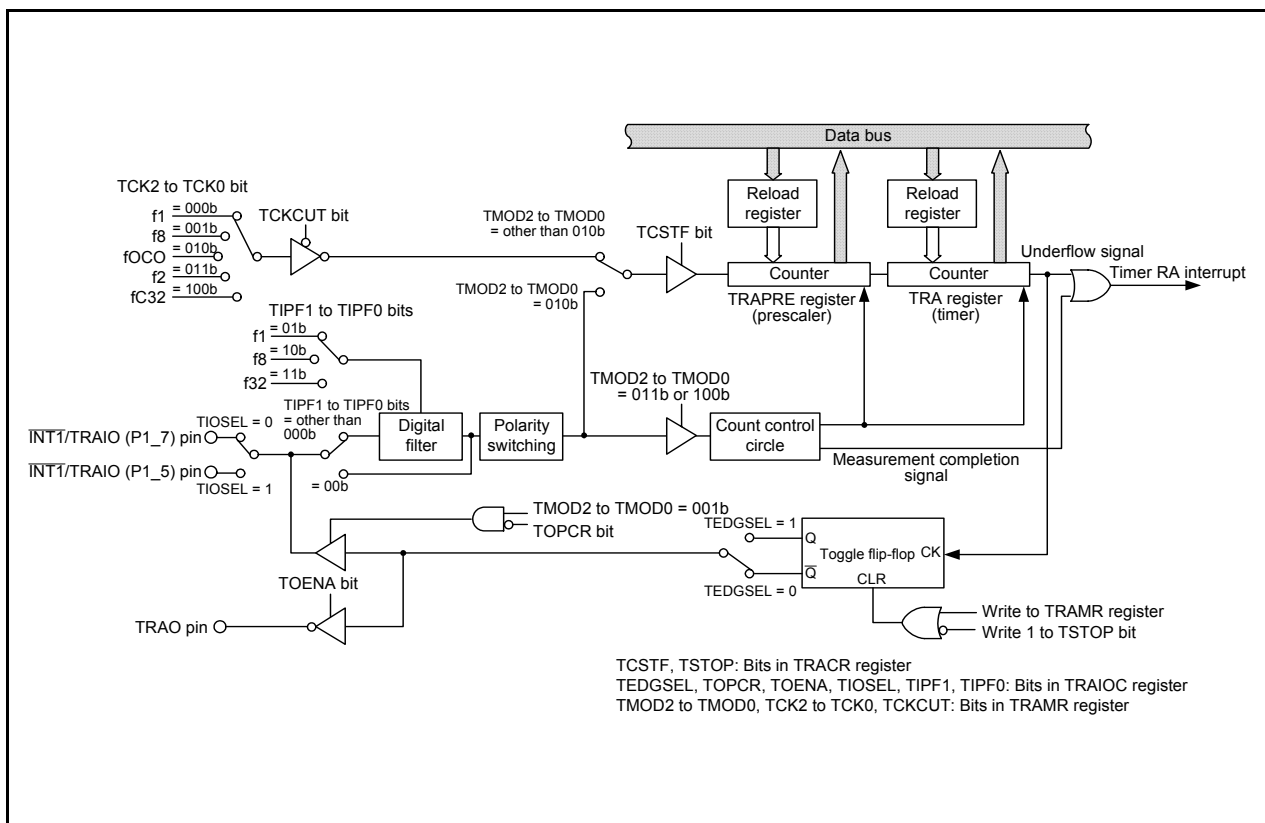


Figure 14.1 Block Diagram of Timer RA

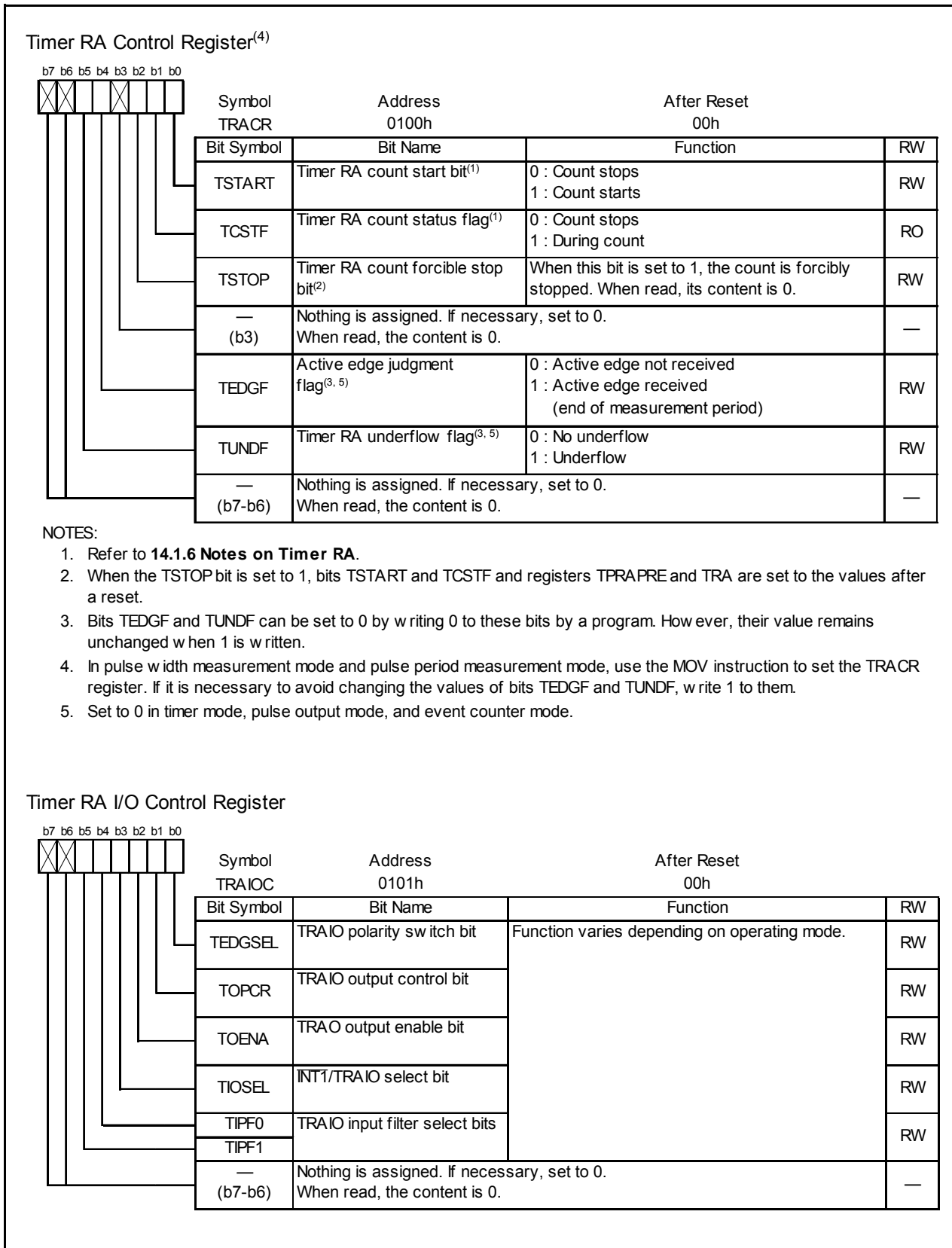


Figure 14.2 Registers TRACR and TRATIOC

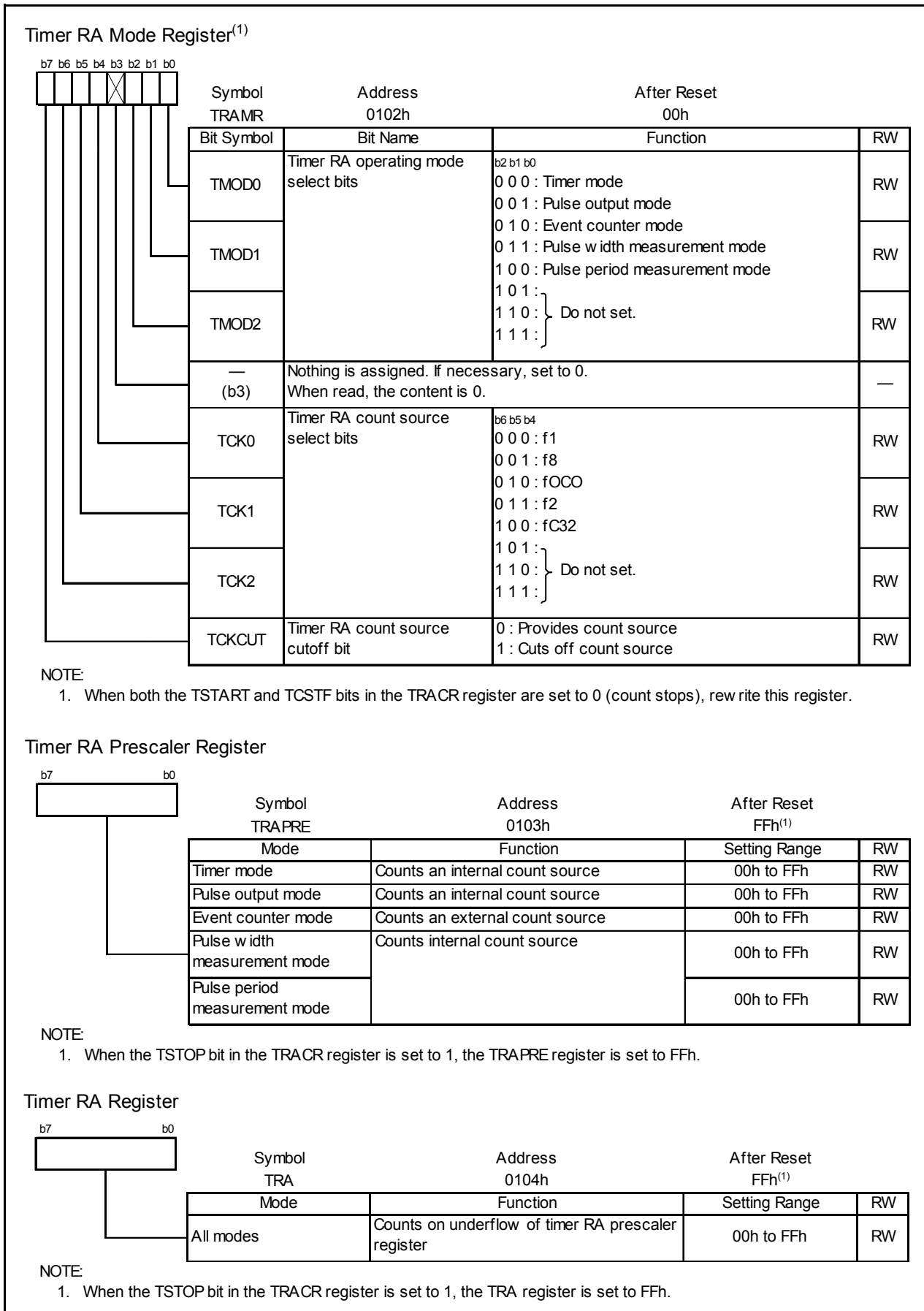


Figure 14.3 Registers TRAMR, TRAPRE, and TRA

14.1.1 Timer Mode

In this mode, the timer counts an internally generated count source (refer to **Table 14.3 Timer Mode Specifications**).

Figure 14.4 shows TRAIOC Register in Timer Mode.

Table 14.3 Timer Mode Specifications

| Item | Specification |
|-------------------------------------|---|
| Count sources | f1, f2, f8, fOCO, fC32 |
| Count operations | <ul style="list-style-type: none"> Decrement When the timer underflows, the contents of the reload register are reloaded and the count is continued. |
| Divide ratio | $1/(n+1)(m+1)$ n: Value set in TRAPRE register, m: Value set in TRA register |
| Count start condition | 1 (count starts) is written to the TSTART bit in the TRACR register. |
| Count stop conditions | <ul style="list-style-type: none"> 0 (count stops) is written to the TSTART bit in the TRACR register. 1 (count forcibly stops) is written to the TSTOP bit in the TRACR register. |
| Interrupt request generation timing | When timer RA underflows [timer RA interrupt]. |
| INT1/TRAI0 pin function | Programmable I/O port, or INT1 interrupt input |
| TRAO pin function | Programmable I/O port |
| Read from timer | The count value can be read by reading registers TRA and TRAPRE. |
| Write to timer | <ul style="list-style-type: none"> When registers TRAPRE and TRA are written while the count is stopped, values are written to both the reload register and counter. When registers TRAPRE and TRA are written during the count, values are written to the reload register and counter (refer to 14.1.1.1 Timer Write Control during Count Operation). |

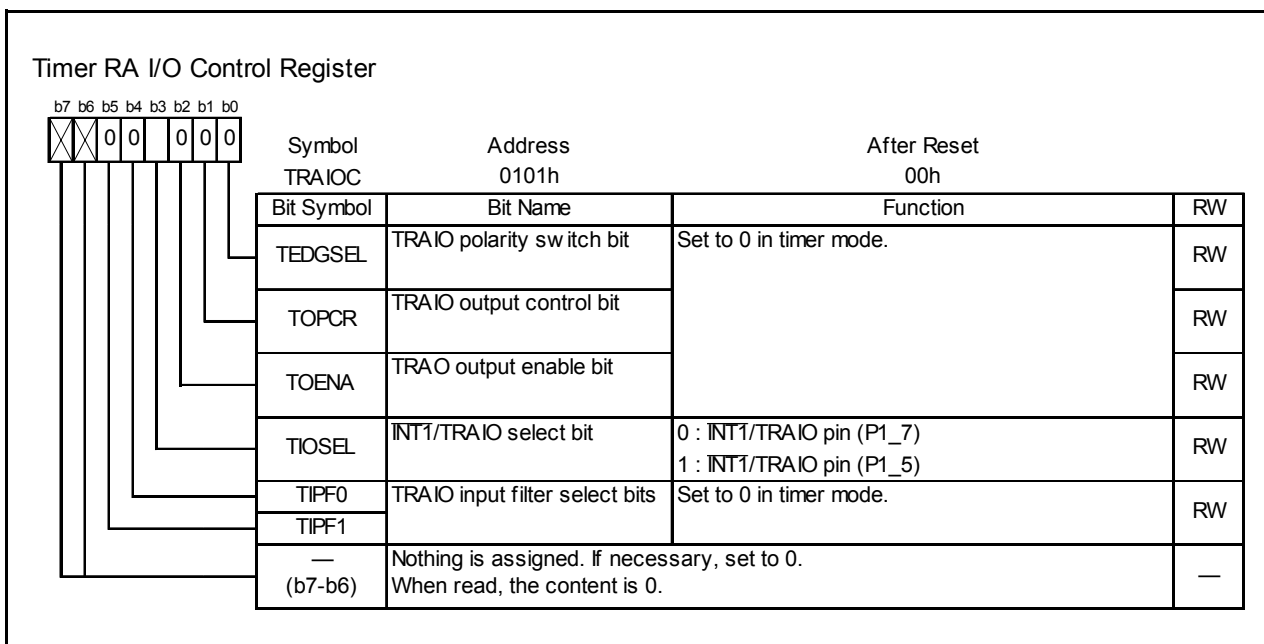


Figure 14.4 TRAIOC Register in Timer Mode

14.1.1.1 Timer Write Control during Count Operation

Timer RA has a prescaler and a timer (which counts the prescaler underflows). The prescaler and timer each consist of a reload register and a counter. When writing to the prescaler or timer, values are written to both the reload register and counter.

However, values are transferred from the reload register to the counter of the prescaler in synchronization with the count source. In addition, values are transferred from the reload register to the counter of the timer in synchronization with prescaler underflows. Therefore, if the prescaler or timer is written to when count operation is in progress, the counter value is not updated immediately after the WRITE instruction is executed. Figure 14.5 shows an Operating Example of Timer RA when Counter Value is Rewritten during Count Operation.

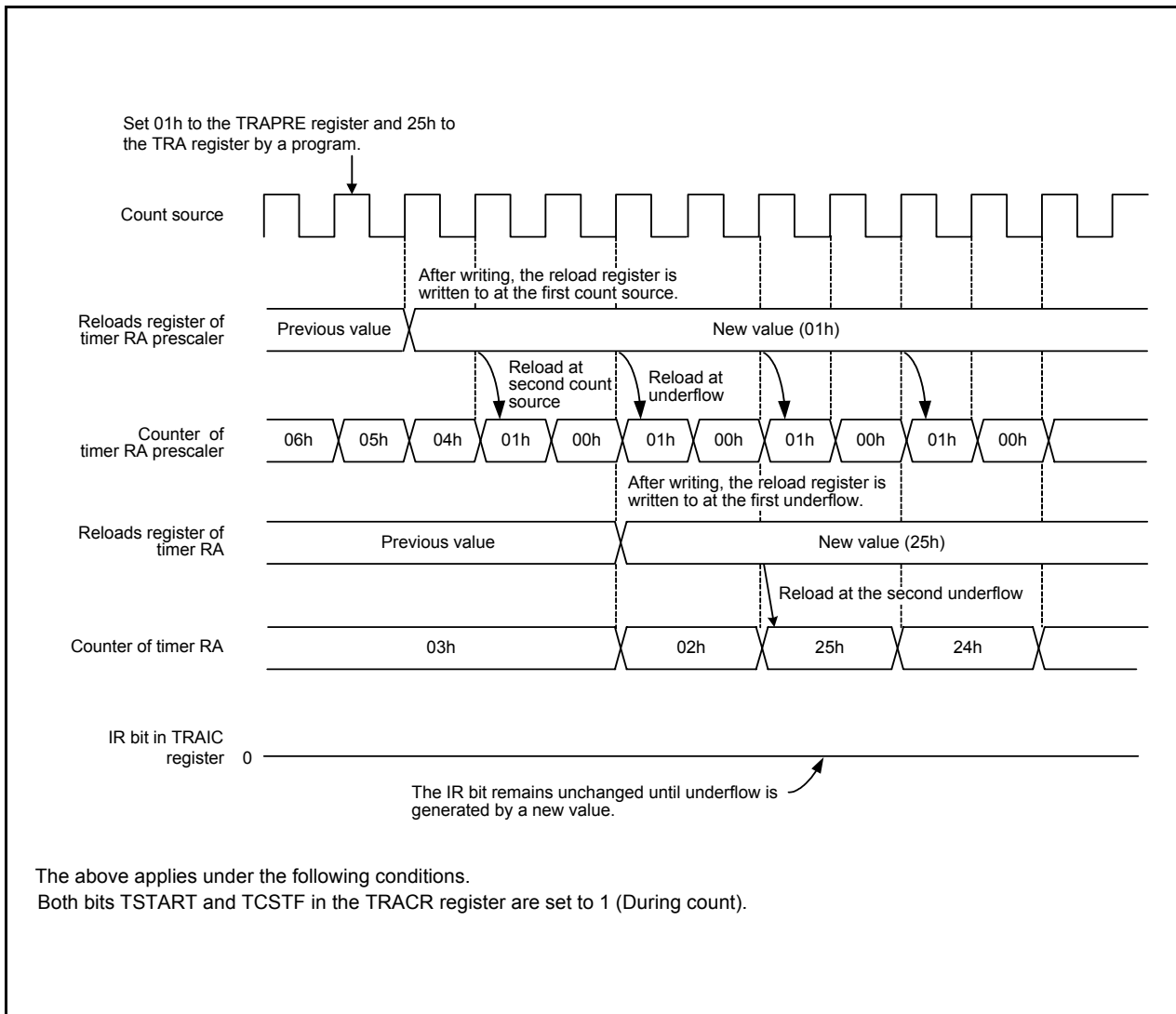


Figure 14.5 Operating Example of Timer RA when Counter Value is Rewritten during Count Operation

14.1.2 Pulse Output Mode

In pulse output mode, the internally generated count source is counted, and a pulse with inverted polarity is output from the TRAI0 pin each time the timer underflows (refer to **Table 14.4 Pulse Output Mode Specifications**).

Figure 14.6 shows TRAI0C Register in Pulse Output Mode.

Table 14.4 Pulse Output Mode Specifications

| Item | Specification |
|-------------------------------------|--|
| Count sources | f1, f2, f8, fOCO, fC32 |
| Count operations | <ul style="list-style-type: none"> • Decrement • When the timer underflows, the contents in the reload register is reloaded and the count is continued. |
| Divide ratio | $1/(n+1)(m+1)$ n: Value set in TRAPRE register, m: Value set in TRA register |
| Count start condition | 1 (count starts) is written to the TSTART bit in the TRACR register. |
| Count stop conditions | <ul style="list-style-type: none"> • 0 (count stops) is written to the TSTART bit in the TRACR register. • 1 (count forcibly stops) is written to the TSTOP bit in the TRACR register. |
| Interrupt request generation timing | When timer RA underflows [timer RA interrupt]. |
| INT1/TRAI0 pin function | Pulse output, programmable output port, or INT1 interrupt ⁽¹⁾ |
| TRAO pin function | Programmable I/O port or inverted output of TRAI0 ⁽¹⁾ |
| Read from timer | The count value can be read by reading registers TRA and TRAPRE. |
| Write to timer | <ul style="list-style-type: none"> • When registers TRAPRE and TRA are written while the count is stopped, values are written to both the reload register and counter. • When registers TRAPRE and TRA are written during the count, values are written to the reload register and counter (refer to 14.1.1.1 Timer Write Control during Count Operation). |
| Select functions | <ul style="list-style-type: none"> • TRAI0 output polarity switch function The TEDGSEL bit in the TRAI0C register selects the level at the start of pulse output.⁽¹⁾ • TRAO output function Pulses inverted from the TRAI0 output polarity can be output from the TRAO pin (selectable by the TOENA bit in the TRAI0C register). • Pulse output stop function Output from the TRAI0 pin is stopped by the TOPCR bit in the TRAI0C register. • INT1/TRAI0 pin select function P1_7 or P1_5 is selected by the TIOSEL bit in the TRAI0C register. |

NOTE:

1. The level of the output pulse becomes the level when the pulse output starts when the TRAMR register is written to.

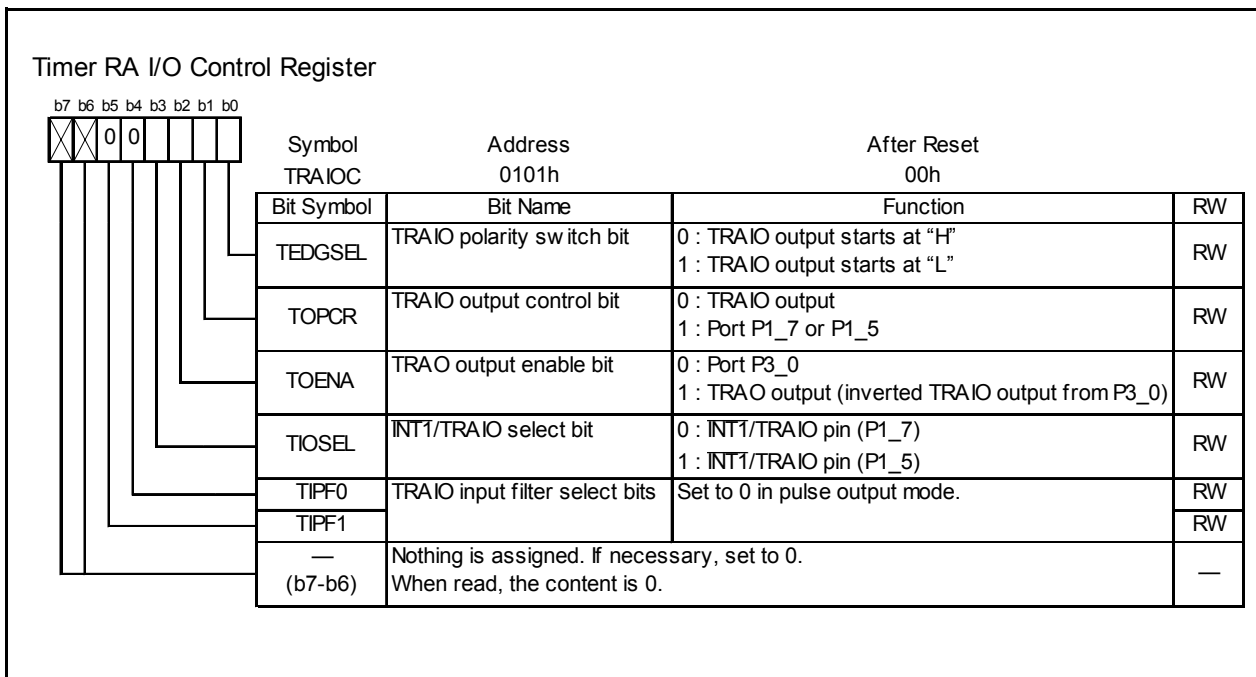


Figure 14.6 TRATIO Register in Pulse Output Mode

14.1.3 Event Counter Mode

In event counter mode, external signal inputs to the $\overline{\text{INT1}}$ /TRAIO pin are counted (refer to **Table 14.5 Event Counter Mode Specifications**).

Figure 14.7 shows TRAIOC Register in Event Counter Mode.

Table 14.5 Event Counter Mode Specifications

| Item | Specification |
|--|---|
| Count source | External signal which is input to TRAIO pin (active edge selectable by a program) |
| Count operations | <ul style="list-style-type: none"> • Decrement • When the timer underflows, the contents of the reload register are reloaded and the count is continued. |
| Divide ratio | $1/(n+1)(m+1)$ n: setting value of TRAPRE register, m: setting value of TRA register |
| Count start condition | 1 (count starts) is written to the TSTART bit in the TRACR register. |
| Count stop conditions | <ul style="list-style-type: none"> • 0 (count stops) is written to the TSTART bit in the TRACR register. • 1 (count forcibly stops) is written to the TSTOP bit in the TRACR register. |
| Interrupt request generation timing | <ul style="list-style-type: none"> • When timer RA underflows [timer RA interrupt]. |
| $\overline{\text{INT1}}$ /TRAIO pin function | Count source input ($\overline{\text{INT1}}$ interrupt input) |
| TRAIO pin function | Programmable I/O port or pulse output ⁽¹⁾ |
| Read from timer | The count value can be read by reading registers TRA and TRAPRE. |
| Write to timer | <ul style="list-style-type: none"> • When registers TRAPRE and TRA are written while the count is stopped, values are written to both the reload register and counter. • When registers TRAPRE and TRA are written during the count, values are written to the reload register and counter (refer to 14.1.1.1 Timer Write Control during Count Operation). |
| Select functions | <ul style="list-style-type: none"> • NT1 input polarity switch function The TEDGSEL bit in the TRAIOC register selects the active edge of the count source. • Count source input pin select function P1_7 or P1_5 is selected by the TIOSEL bit in the TRAIOC register. • Pulse output function Pulses of inverted polarity can be output from the TRAIO pin each time the timer underflows (selectable by the TOENA bit in the TRAIOC register)⁽¹⁾. • Digital filter function Bits TIPF0 and TIPF1 in the TRAIOC register enable or disable the digital filter and select the sampling frequency. |

NOTE:

1. The level of the output pulse becomes the level when the pulse output starts when the TRAMR register is written to.

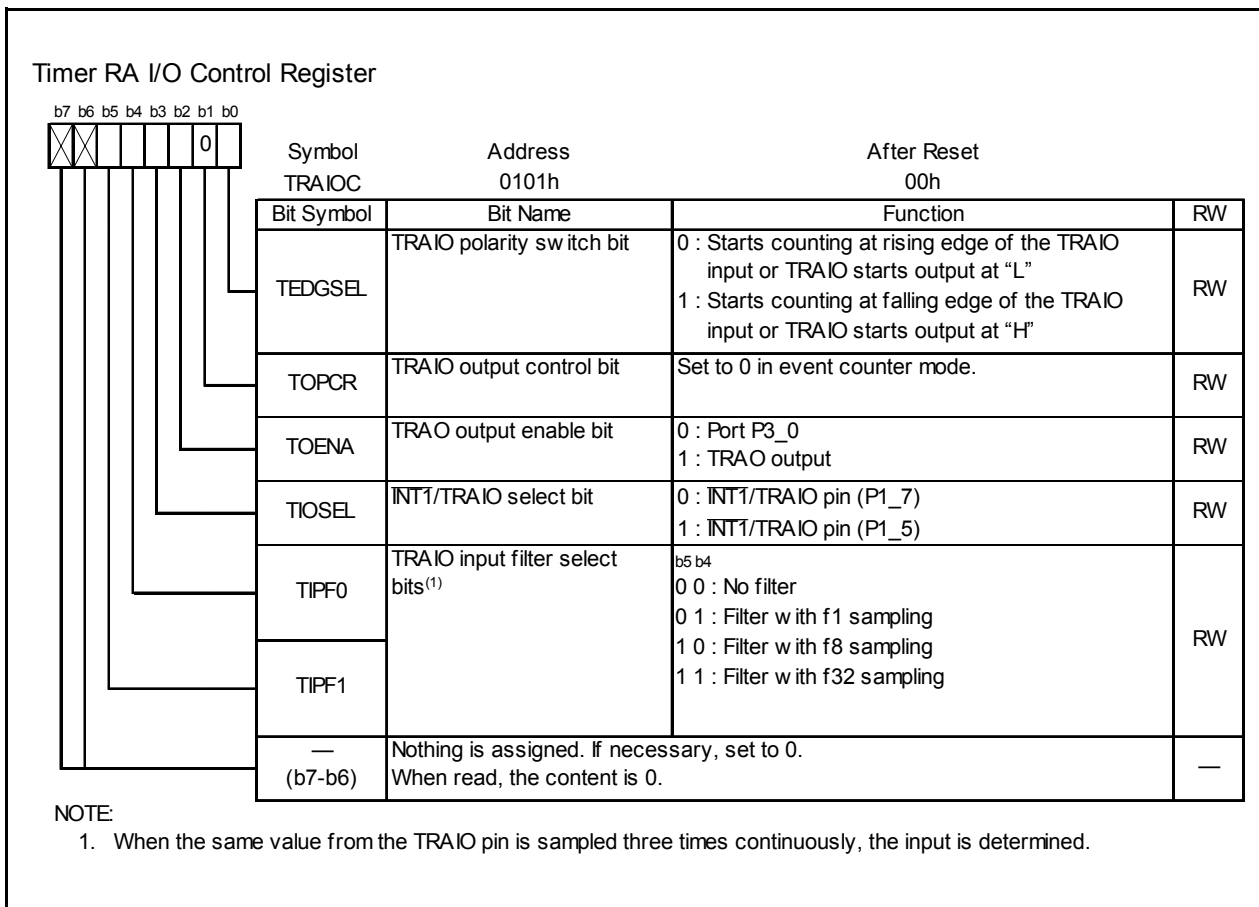


Figure 14.7 TRAI0C Register in Event Counter Mode

14.1.4 Pulse Width Measurement Mode

In pulse width measurement mode, the pulse width of an external signal input to the $\overline{\text{INT1}}$ /TRAIO pin is measured (refer to **Table 14.6 Pulse Width Measurement Mode Specifications**).

Figure 14.8 shows TRAIOC Register in Pulse Width Measurement Mode and Figure 14.9 shows an Operating Example of Pulse Width Measurement Mode.

Table 14.6 Pulse Width Measurement Mode Specifications

| Item | Specification |
|--|---|
| Count sources | f1, f2, f8, fOCO, fC32 |
| Count operations | <ul style="list-style-type: none"> • Decrement • Continuously counts the selected signal only when measurement pulse is “H” level, or conversely only “L” level. • When the timer underflows, the contents of the reload register are reloaded and the count is continued. |
| Count start condition | 1 (count starts) is written to the TSTART bit in the TRACR register. |
| Count stop conditions | <ul style="list-style-type: none"> • 0 (count stops) is written to the TSTART bit in the TRACR register. • 1 (count forcibly stops) is written to the TSTOP bit in the TRACR register. |
| Interrupt request generation timing | <ul style="list-style-type: none"> • When timer RA underflows [timer RA interrupt]. • Rising or falling of the TRAIO input (end of measurement period) [timer RA interrupt] |
| $\overline{\text{INT1}}$ /TRAIO pin function | Measured pulse input ($\overline{\text{INT1}}$ interrupt input) |
| TRAO pin function | Programmable I/O port |
| Read from timer | The count value can be read by reading registers TRA and TRAPRE. |
| Write to timer | <ul style="list-style-type: none"> • When registers TRAPRE and TRA are written while the count is stopped, values are written to both the reload register and counter. • When registers TRAPRE and TRA are written during the count, values are written to the reload register and counter (refer to 14.1.1.1 Timer Write Control during Count Operation). |
| Select functions | <ul style="list-style-type: none"> • Measurement level select • The TEDGSEL bit in the TRAIOC register selects the “H” or “L” level period. • Measured pulse input pin select function P1_7 or P1_5 is selected by the TIOSEL bit in the TRAIOC register. • Digital filter function Bits TIPF0 and TIPF1 in the TRAIOC register enable or disable the digital filter and select the sampling frequency. |

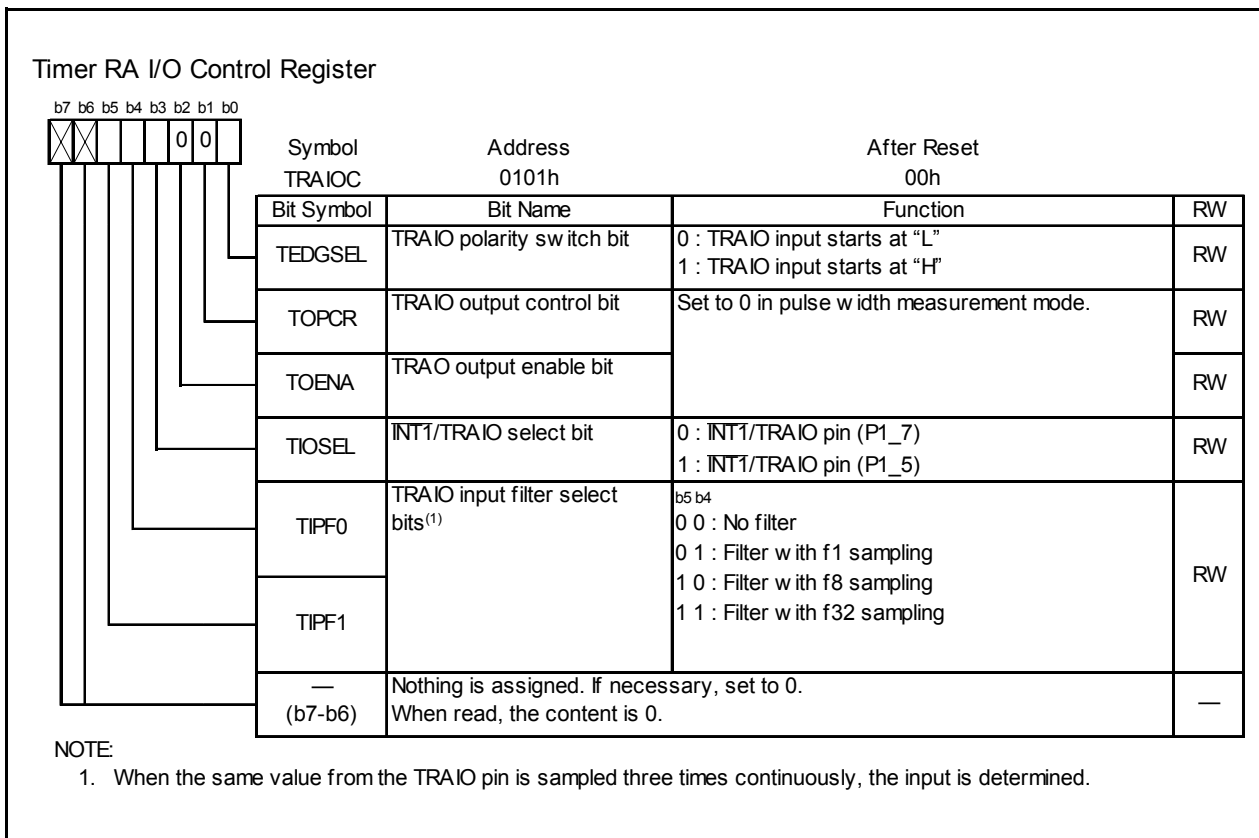


Figure 14.8 TRAI0C Register in Pulse Width Measurement Mode

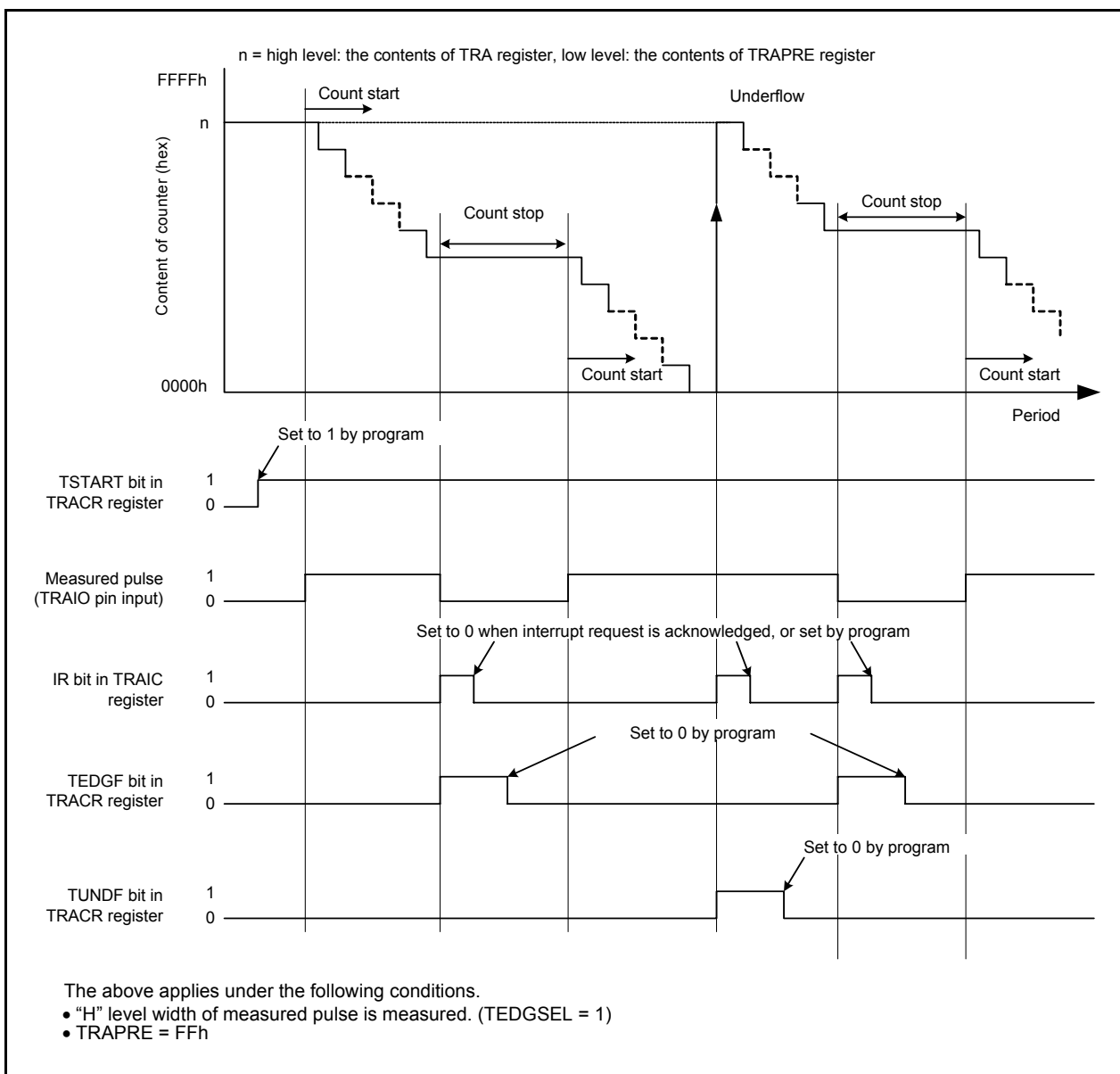


Figure 14.9 Operating Example of Pulse Width Measurement Mode

14.1.5 Pulse Period Measurement Mode

In pulse period measurement mode, the pulse period of an external signal input to the $\overline{\text{INT1}}$ /TRAIO pin is measured (refer to **Table 14.7 Pulse Period Measurement Mode Specifications**).

Figure 14.10 shows the TRAIOC Register in Pulse Period Measurement Mode and Figure 14.11 shows an Operating Example of Pulse Period Measurement Mode.

Table 14.7 Pulse Period Measurement Mode Specifications

| Item | Specification |
|--|---|
| Count sources | f1, f2, f8, fOCO, fC32 |
| Count operations | <ul style="list-style-type: none"> • Decrement • After the active edge of the measured pulse is input, the contents of the read-out buffer are retained at the first underflow of timer RA prescaler. Then timer RA reloads the contents in the reload register at the second underflow of timer RA prescaler and continues counting. |
| Count start condition | 1 (count starts) is written to the TSTART bit in the TRACR register. |
| Count stop conditions | <ul style="list-style-type: none"> • 0 (count stops) is written to TSTART bit in the TRACR register. • 1 (count forcibly stops) is written to the TSTOP bit in the TRACR register. |
| Interrupt request generation timing | <ul style="list-style-type: none"> • When timer RA underflows or reloads [timer RA interrupt]. • Rising or falling of the TRAIO input (end of measurement period) [timer RA interrupt] |
| $\overline{\text{INT1}}$ /TRAIO pin function | Measured pulse input ⁽¹⁾ ($\overline{\text{INT1}}$ interrupt input) |
| TRAIO pin function | Programmable I/O port |
| Read from timer | The count value can be read by reading registers TRA and TRAPRE. |
| Write to timer | <ul style="list-style-type: none"> • When registers TRAPRE and TRA are written while the count is stopped, values are written to both the reload register and counter. • When registers TRAPRE and TRA are written during the count, values are written to the reload register and counter (refer to 14.1.1.1 Timer Write Control during Count Operation). |
| Select functions | <ul style="list-style-type: none"> • Measurement period select The TEDGSEL bit in the TRAIOC register selects the measurement period of the input pulse. • Measured pulse input pin select function P1_7 or P1_5 is selected by the TIOSEL bit in the TRAIOC register. • Digital filter function Bits TIPF0 and TIPF1 in the TRAIOC register enable or disable the digital filter and select the sampling frequency. |

NOTE:

1. Input a pulse with a period longer than twice the timer RA prescaler period. Input a pulse with a longer "H" and "L" width than the timer RA prescaler period. If a pulse with a shorter period is input to the TRAIO pin, the input may be ignored.

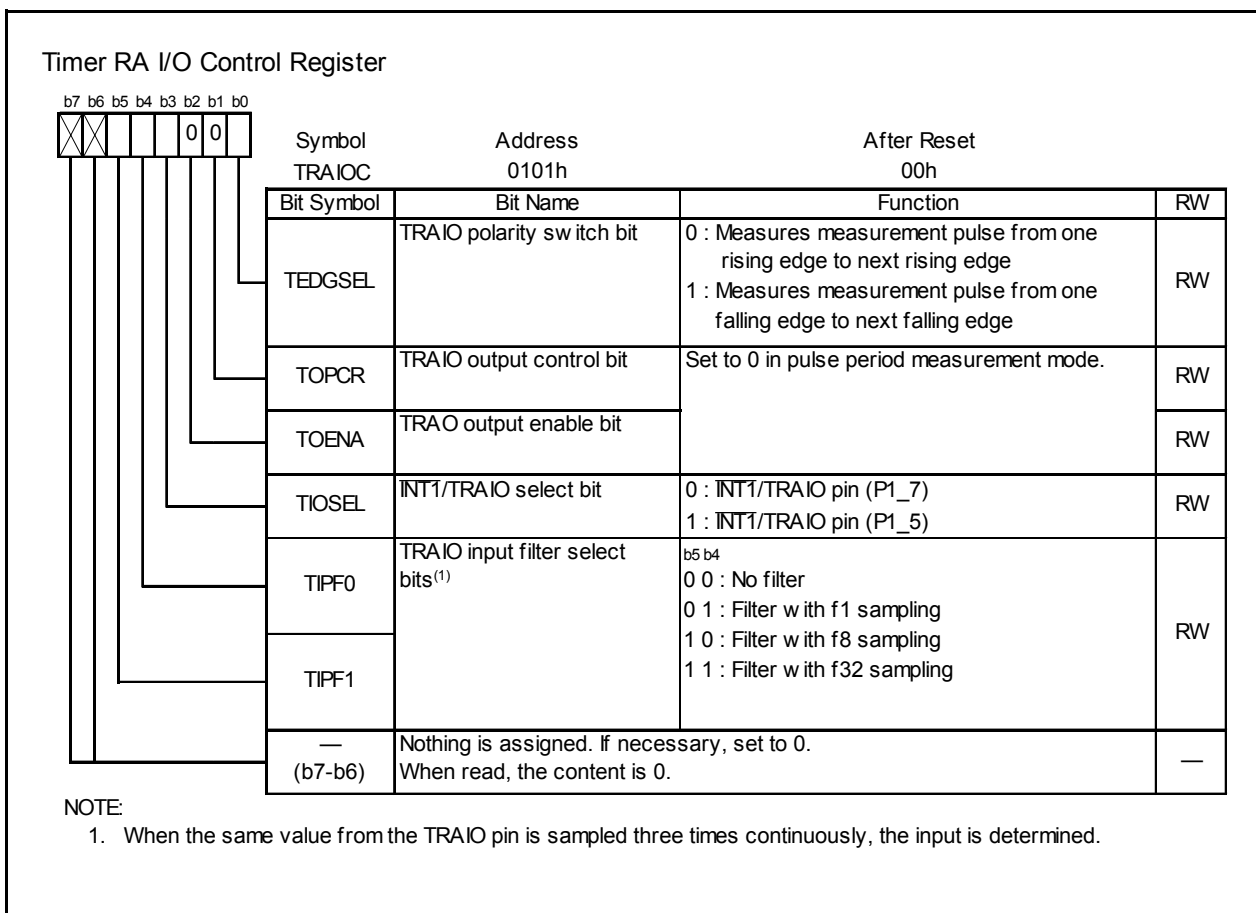


Figure 14.10 TRAI0C Register in Pulse Period Measurement Mode

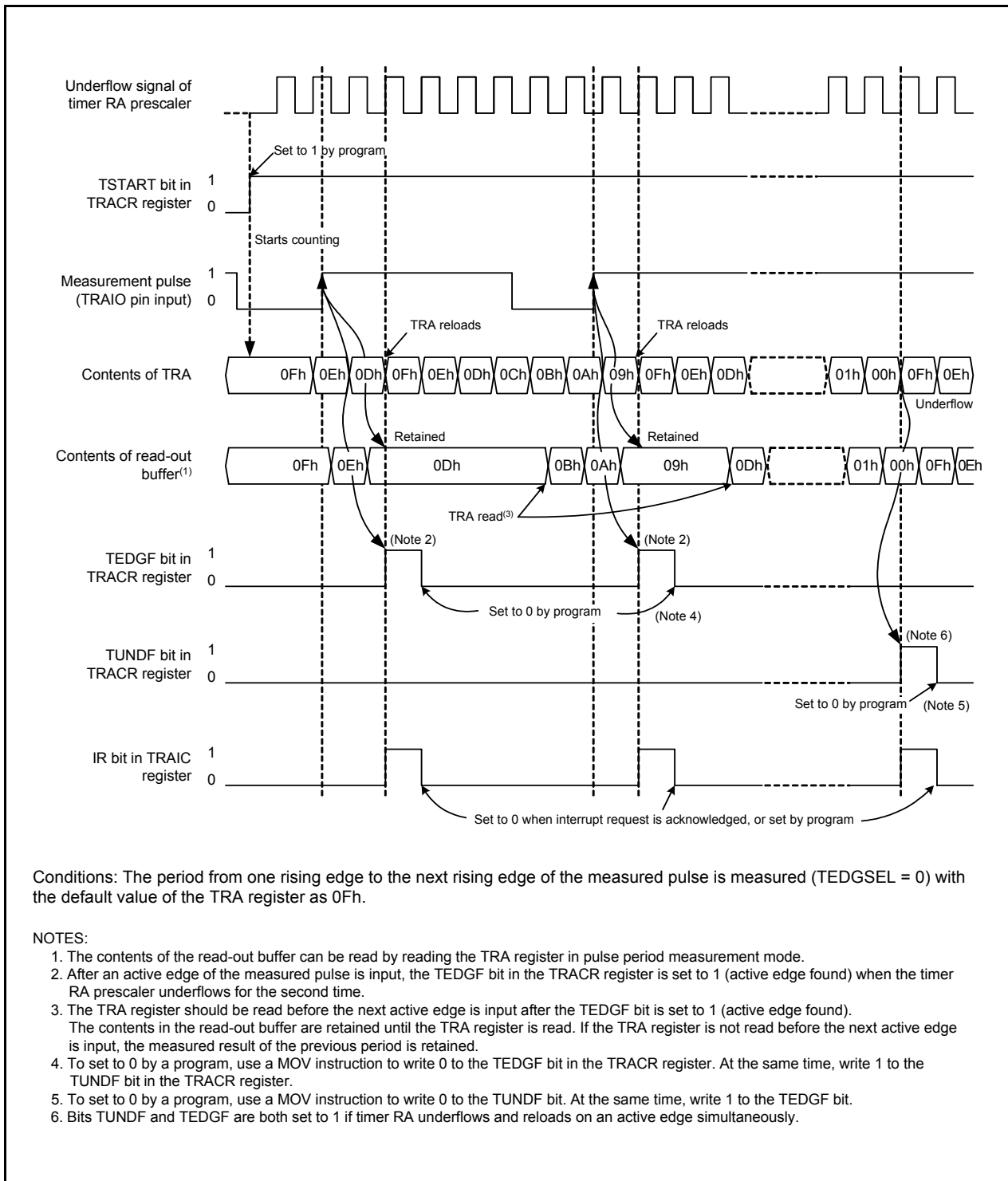


Figure 14.11 Operating Example of Pulse Period Measurement Mode

14.1.6 Notes on Timer RA

- Timer RA stops counting after a reset. Set the values in the timer RA and timer RA prescalers before the count starts.
- Even if the prescaler and timer RA are read out in 16-bit units, these registers are read 1 byte at a time by the MCU. Consequently, the timer value may be updated during the period when these two registers are being read.
- In pulse period measurement mode, bits TEDGF and TUNDF in the TRACR register can be set to 0 by writing 0 to these bits by a program. However, these bits remain unchanged if 1 is written. When using the READ-MODIFY-WRITE instruction for the TRACR register, the TEDGF or TUNDF bit may be set to 0 although these bits are set to 1 while the instruction is being executed. In this case, write 1 to the TEDGF or TUNDF bit which is not supposed to be set to 0 with the MOV instruction.
- When changing to pulse period measurement mode from another mode, the contents of bits TEDGF and TUNDF are undefined. Write 0 to bits TEDGF and TUNDF before the count starts.
- The TEDGF bit may be set to 1 by the first timer RA prescaler underflow generated after the count starts.
- When using the pulse period measurement mode, leave two or more periods of the timer RA prescaler immediately after the count starts, then set the TEDGF bit to 0.
- The TCSTF bit retains 0 (count stops) for 0 to 1 cycle of the count source after setting the TSTART bit to 1 (count starts) while the count is stopped.

During this time, do not access registers associated with timer RA⁽¹⁾ other than the TCSTF bit. Timer RA starts counting at the first valid edge of the count source after The TCSTF bit is set to 1 (during count). The TCSTF bit remains 1 for 0 to 1 cycle of the count source after setting the TSTART bit to 0 (count stops) while the count is in progress. Timer RA counting is stopped when the TCSTF bit is set to 0. During this time, do not access registers associated with timer RA⁽¹⁾ other than the TCSTF bit.

NOTE:

1. Registers associated with timer RA: TRACR, TRAIOC, TRAMR, TRAPRE, and TRA.

- When the TRAPRE register is continuously written during count operation (TCSTF bit is set to 1), allow three or more cycles of the count source clock for each write interval.
- When the TRA register is continuously written during count operation (TCSTF bit is set to 1), allow three or more cycles of the prescaler underflow for each write interval.

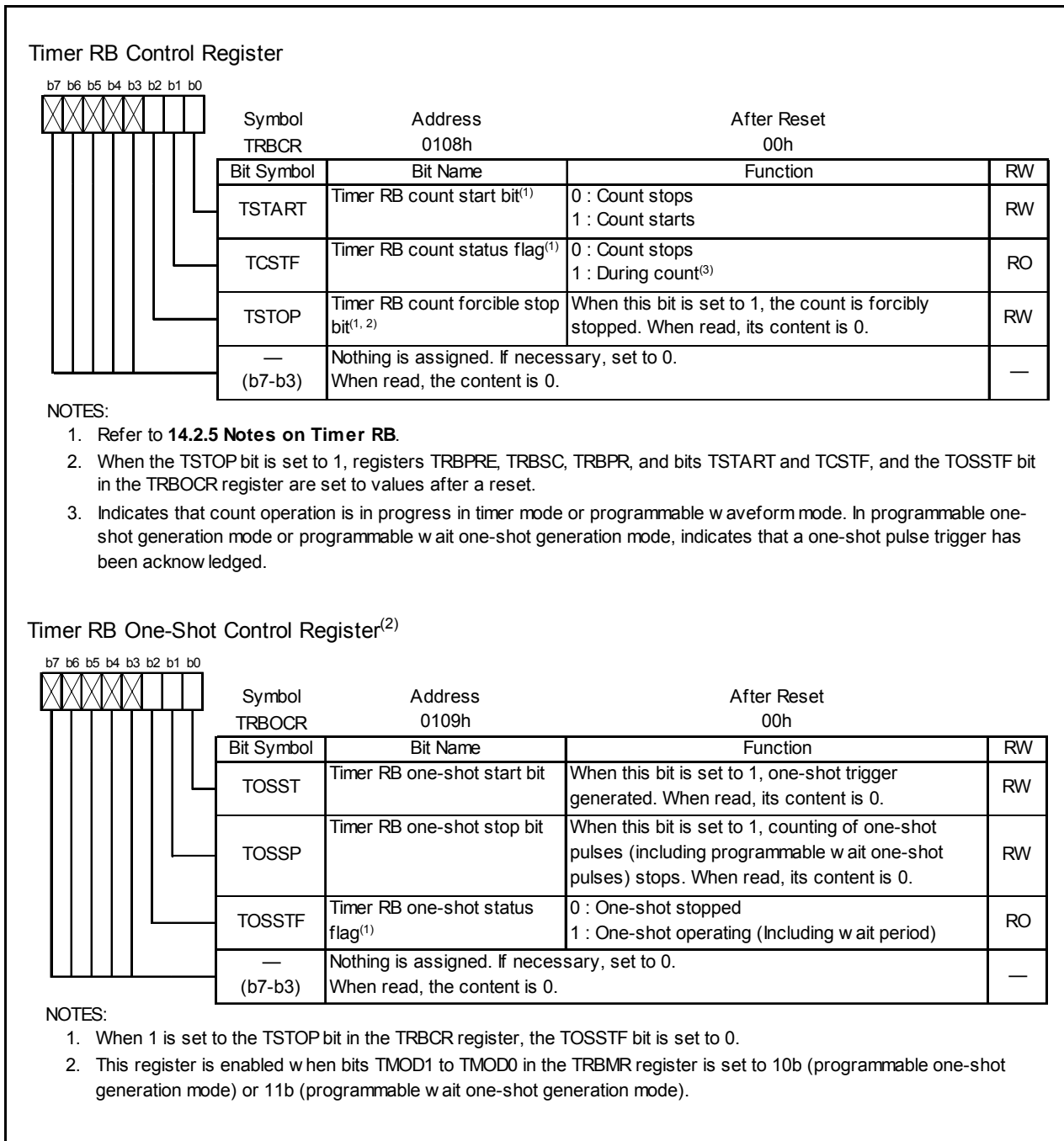


Figure 14.13 Registers TRBCR and TRBOCR

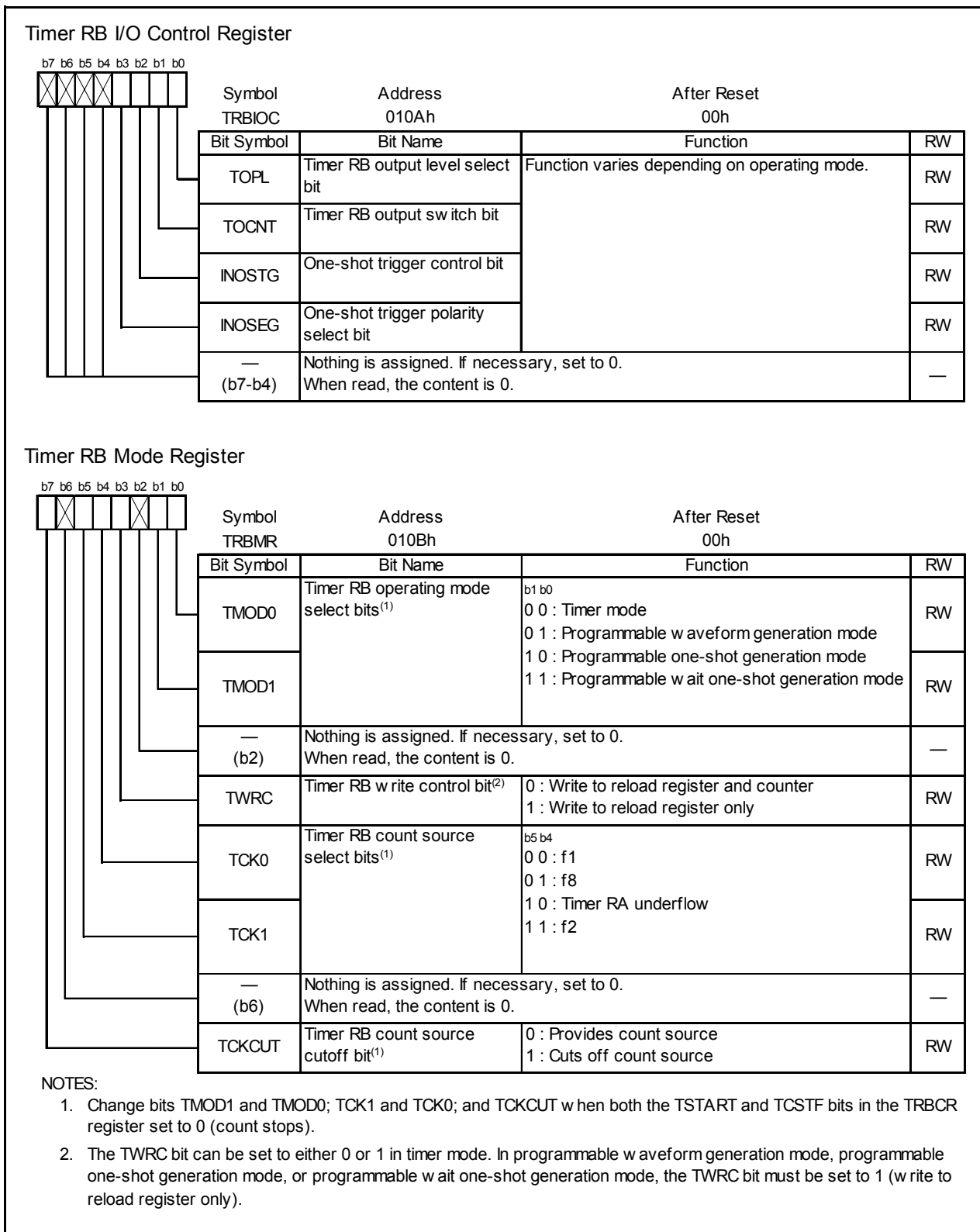


Figure 14.14 Registers TRBIOC and TRBMR

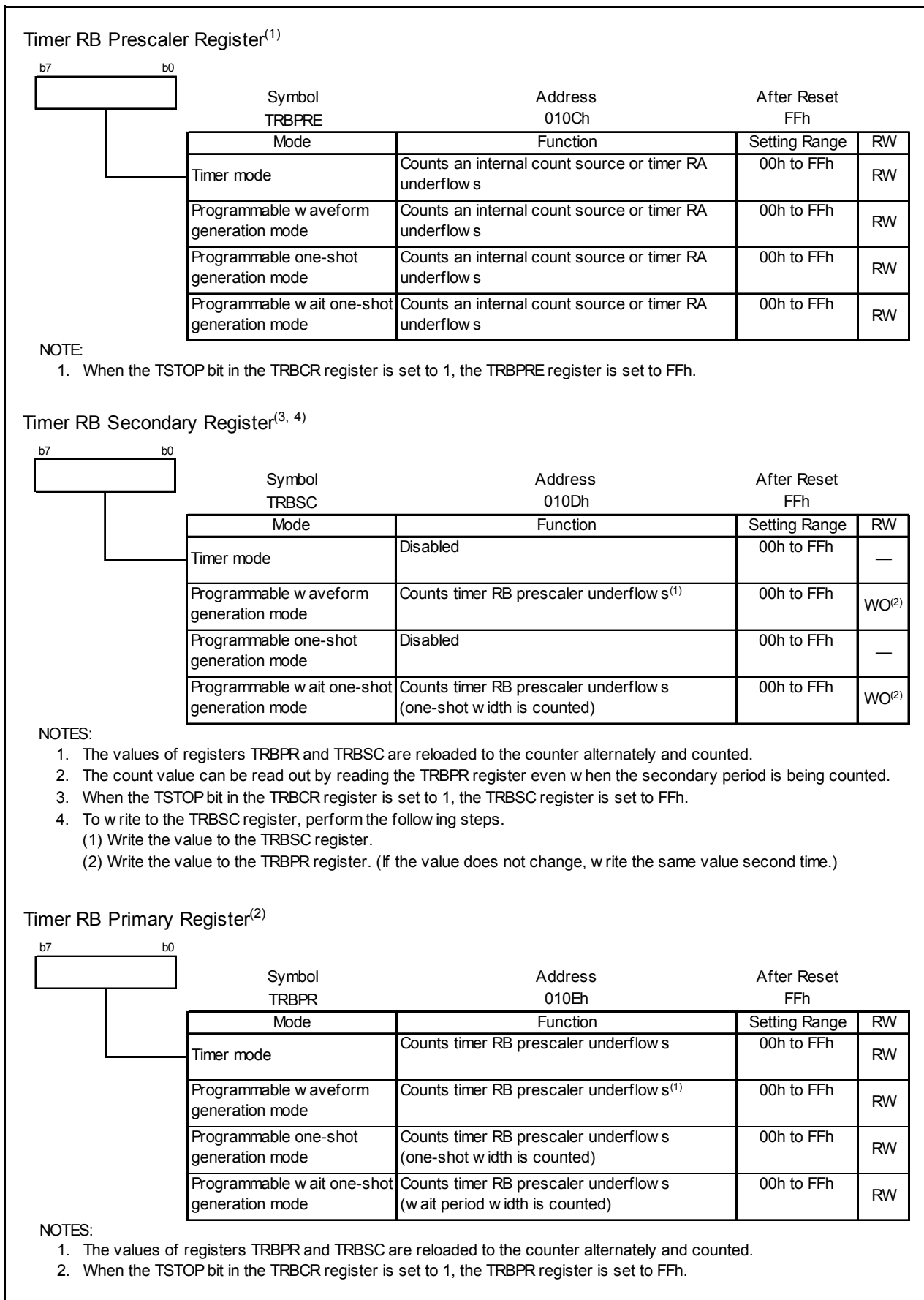


Figure 14.15 Registers TRBPRES, TRBSC, and TRBPR

14.2.1 Timer Mode

In timer mode, a count source which is internally generated or timer RA underflows are counted (refer to **Table 14.8 Timer Mode Specifications**). Registers TRBOCR and TRBSC are not used in timer mode.

Figure 14.16 shows TRBIOC Register in Timer Mode.

Table 14.8 Timer Mode Specifications

| Item | Specification |
|-------------------------------------|---|
| Count sources | f1, f2, f8, timer RA underflow |
| Count operations | <ul style="list-style-type: none"> Decrement When the timer underflows, it reloads the reload register contents before the count continues (when timer RB underflows, the contents of timer RB primary reload register is reloaded). |
| Divide ratio | $1/(n+1)(m+1)$ n: setting value in TRBPRES register, m: setting value in TRBPR register |
| Count start condition | 1 (count starts) is written to the TSTART bit in the TRBCR register. |
| Count stop conditions | <ul style="list-style-type: none"> 0 (count stops) is written to the TSTART bit in the TRBCR register. 1 (count forcibly stops) is written to the TSTOP bit in the TRBCR register. |
| Interrupt request generation timing | When timer RB underflows [timer RB interrupt]. |
| TRBO pin function | Programmable I/O port |
| INT0 pin function | Programmable I/O port or INT0 interrupt input |
| Read from timer | The count value can be read out by reading registers TRBPR and TRBPRES. |
| Write to timer | <ul style="list-style-type: none"> When registers TRBPRES and TRBPR are written while the count is stopped, values are written to both the reload register and counter. When registers TRBPRES and TRBPR are written to while count operation is in progress: If the TWRC bit in the TRBMR register is set to 0, the value is written to both the reload register and the counter. If the TWRC bit is set to 1, the value is written to the reload register only. (Refer to 14.2.1.1 Timer Write Control during Count Operation.) |

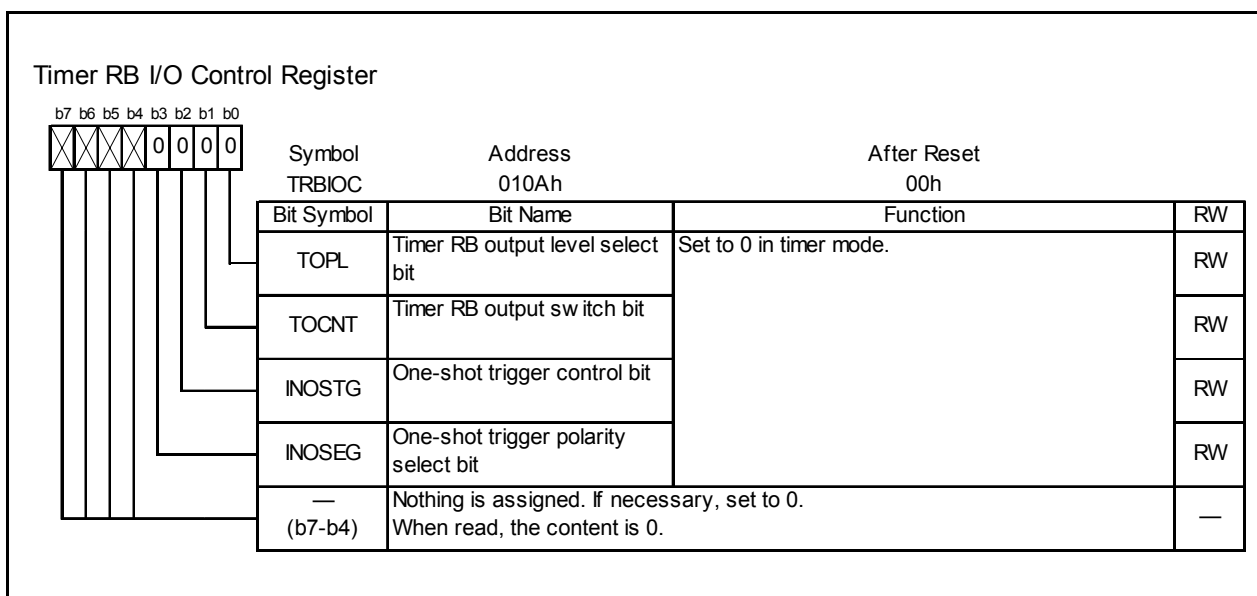


Figure 14.16 TRBIOC Register in Timer Mode

14.2.1.1 Timer Write Control during Count Operation

Timer RB has a prescaler and a timer (which counts the prescaler underflows). The prescaler and timer each consist of a reload register and a counter. In timer mode, the TWRC bit in the TRBMR register can be used to select whether writing to the prescaler or timer during count operation is performed to both the reload register and counter or only to the reload register.

However, values are transferred from the reload register to the counter of the prescaler in synchronization with the count source. In addition, values are transferred from the reload register to the counter of the timer in synchronization with prescaler underflows. Therefore, even if the TWRC bit is set for writing to both the reload register and counter, the counter value is not updated immediately after the WRITE instruction is executed. In addition, if the TWRC bit is set for writing to the reload register only, the synchronization of the writing will be shifted if the prescaler value changes. Figure 14.17 shows an Operating Example of Timer RB when Counter Value is Rewritten during Count Operation.

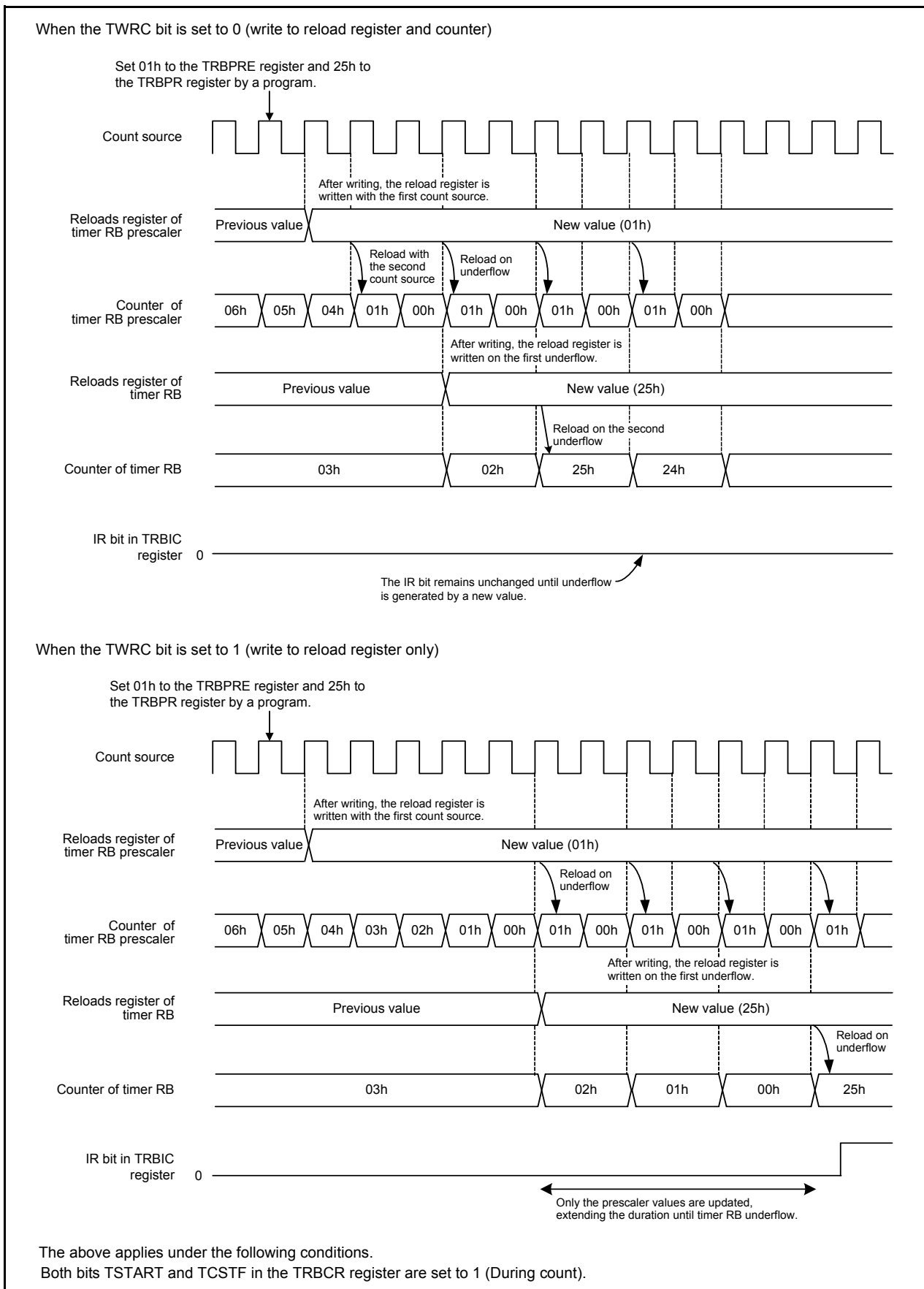


Figure 14.17 Operating Example of Timer RB when Counter Value is Rewritten during Count Operation

14.2.2 Programmable Waveform Generation Mode

In programmable waveform generation mode, the signal output from the TRBO pin is inverted each time the counter underflows, while the values in registers TRBPR and TRBSC are counted alternately (refer to **Table 14.9 Programmable Waveform Generation Mode Specifications**). Counting starts by counting the setting value in the TRBPR register. The TRBOCR register is unused in this mode.

Figure 14.18 shows TRBIOC Register in Programmable Waveform Generation Mode. Figure 14.19 shows an Operating Example of Timer RB in Programmable Waveform Generation Mode.

Table 14.9 Programmable Waveform Generation Mode Specifications

| Item | Specification |
|-------------------------------------|--|
| Count sources | f1, f2, f8, timer RA underflow |
| Count operations | <ul style="list-style-type: none"> • Decrement • When the timer underflows, it reloads the contents of the primary reload and secondary reload registers alternately before the count continues. |
| Width and period of output waveform | Primary period: $(n+1)(m+1)/f_i$ Secondary period: $(n+1)(p+1)/f_i$ Period: $(n+1)\{(m+1)+(p+1)\}/f_i$ f_i : Count source frequency n : Value set in TRBPRES register m : Value set in TRBPR register p : Value set in TRBSC register |
| Count start condition | 1 (count starts) is written to the TSTART bit in the TRBCR register. |
| Count stop conditions | <ul style="list-style-type: none"> • 0 (count stops) is written to the TSTART bit in the TRBCR register. • 1 (count forcibly stops) is written to the TSTOP bit in the TRBCR register. |
| Interrupt request generation timing | In half a cycle of the count source, after timer RB underflows during the secondary period (at the same time as the TRBO output change) [timer RB interrupt] |
| TRBO pin function | Programmable output port or pulse output |
| INT0 pin function | Programmable I/O port or INT0 interrupt input |
| Read from timer | The count value can be read out by reading registers TRBPR and TRBPRES ⁽¹⁾ . |
| Write to timer | <ul style="list-style-type: none"> • When registers TRBPRES, TRBSC, and TRBPR are written while the count is stopped, values are written to both the reload register and counter. • When registers TRBPRES, TRBSC, and TRBPR are written to during count operation, values are written to the reload registers only.⁽²⁾ |
| Select functions | <ul style="list-style-type: none"> • Output level select function The TOPL bit in the TRBIOC register selects the output level during primary and secondary periods. • TRBO pin output switch function Timer RB pulse output or P3_1 latch output is selected by the TOCNT bit in the TRBIOC register.⁽³⁾ |

NOTES:

1. Even when counting the secondary period, the TRBPR register may be read.
2. The set values are reflected in the waveform output beginning with the following primary period after writing to the TRBPR register.
3. The value written to the TOCNT bit is enabled by the following.
 - When count starts.
 - When a timer RB interrupt request is generated.
The contents after the TOCNT bit is changed are reflected from the output of the following primary period.

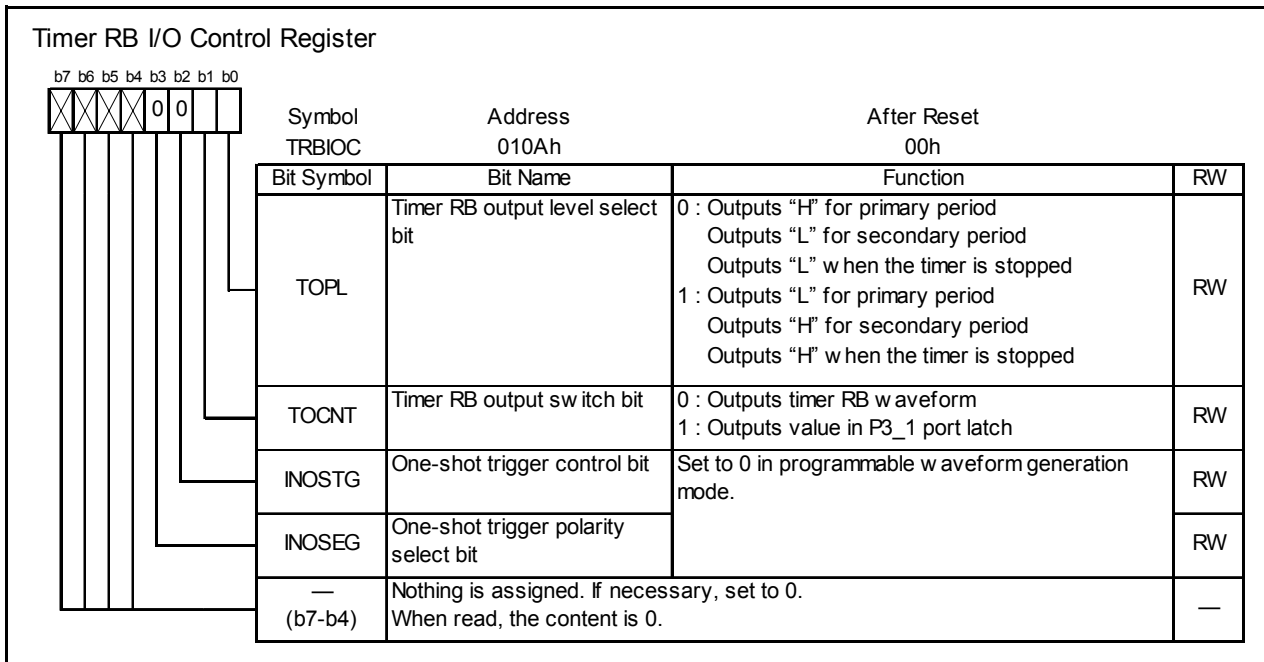


Figure 14.18 TRBIOC Register in Programmable Waveform Generation Mode

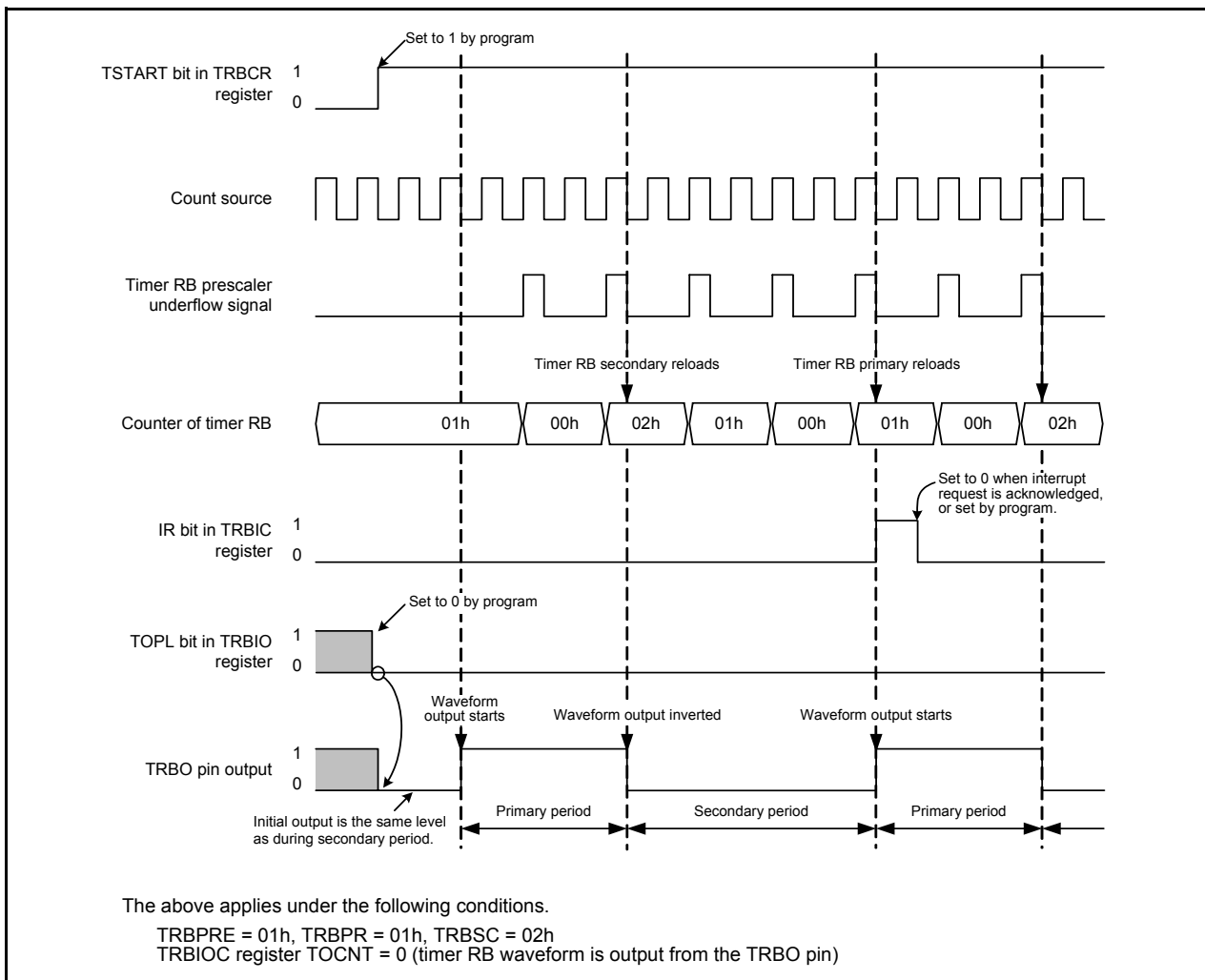


Figure 14.19 Operating Example of Timer RB in Programmable Waveform Generation Mode

14.2.3 Programmable One-shot Generation Mode

In programmable one-shot generation mode, a one-shot pulse is output from the TRBO pin by a program or an external trigger input (input to the INT0 pin) (refer to **Table 14.10 Programmable One-Shot Generation Mode Specifications**). When a trigger is generated, the timer starts operating from the point only once for a given period equal to the set value in the TRBPR register. The TRBSC register is not used in this mode.

Figure 14.20 shows TRBIOC Register in Programmable One-Shot Generation Mode. Figure 14.21 shows an Operating Example of Programmable One-Shot Generation Mode.

Table 14.10 Programmable One-Shot Generation Mode Specifications

| Item | Specification |
|-------------------------------------|---|
| Count sources | f1, f2, f8, timer RA underflow |
| Count operations | <ul style="list-style-type: none"> • Decrement the setting value in the TRBPR register • When the timer underflows, it reloads the contents of the reload register before the count completes and the TOSSTF bit is set to 0 (one-shot stops). • When the count stops, the timer reloads the contents of the reload register before it stops. |
| One-shot pulse output time | $(n+1)(m+1)/f_i$ f_i : Count source frequency, n : Setting value in TRBPRES register, m : Setting value in TRBPR register ⁽²⁾ |
| Count start conditions | <ul style="list-style-type: none"> • The TSTART bit in the TRBCR register is set to 1 (count starts) and the next trigger is generated. • Set the TOSST bit in the TRBOCR register to 1 (one-shot starts) • Input trigger to the INT0 pin |
| Count stop conditions | <ul style="list-style-type: none"> • When reloading completes after timer RB underflows during primary period. • When the TOSSP bit in the TRBOCR register is set to 1 (one-shot stops). • When the TSTART bit in the TRBCR register is set to 0 (count stops). • When the TSTOP bit in the TRBCR register is set to 1 (count forcibly stops). |
| Interrupt request generation timing | In half a cycle of the count source, after the timer underflows (at the same time as the TRBO output ends) [timer RB interrupt] |
| TRBP pin function | Pulse output |
| INT0 pin functions | <ul style="list-style-type: none"> • When the INOSTG bit in the TRBIOC register is set to 0 (INT0 one-shot trigger disabled): programmable I/O port or INT0 interrupt input • When the INOSTG bit in the TRBIOC register is set to 1 (INT0 one-shot trigger enabled): external trigger (INT0 interrupt input) |
| Read from timer | The count value can be read out by reading registers TRBPR and TRBPRES. |
| Write to timer | <ul style="list-style-type: none"> • When registers TRBPRES and TRBPR are written while the count is stopped, values are written to both the reload register and counter. • When registers TRBPRES and TRBPR are written during the count, values are written to the reload register only (the data is transferred to the counter at the following reload)⁽¹⁾. |
| Select functions | <ul style="list-style-type: none"> • Output level select function The TOPL bit in the TRBIOC register selects the output level of the one-shot pulse waveform. • One-shot trigger select function Refer to 14.2.3.1 One-Shot Trigger Selection. |

NOTES:

1. The set value is reflected at the following one-shot pulse after writing to the TRBPR register.
2. Do not set both the TRBPRES and TRBPR registers to 00h.

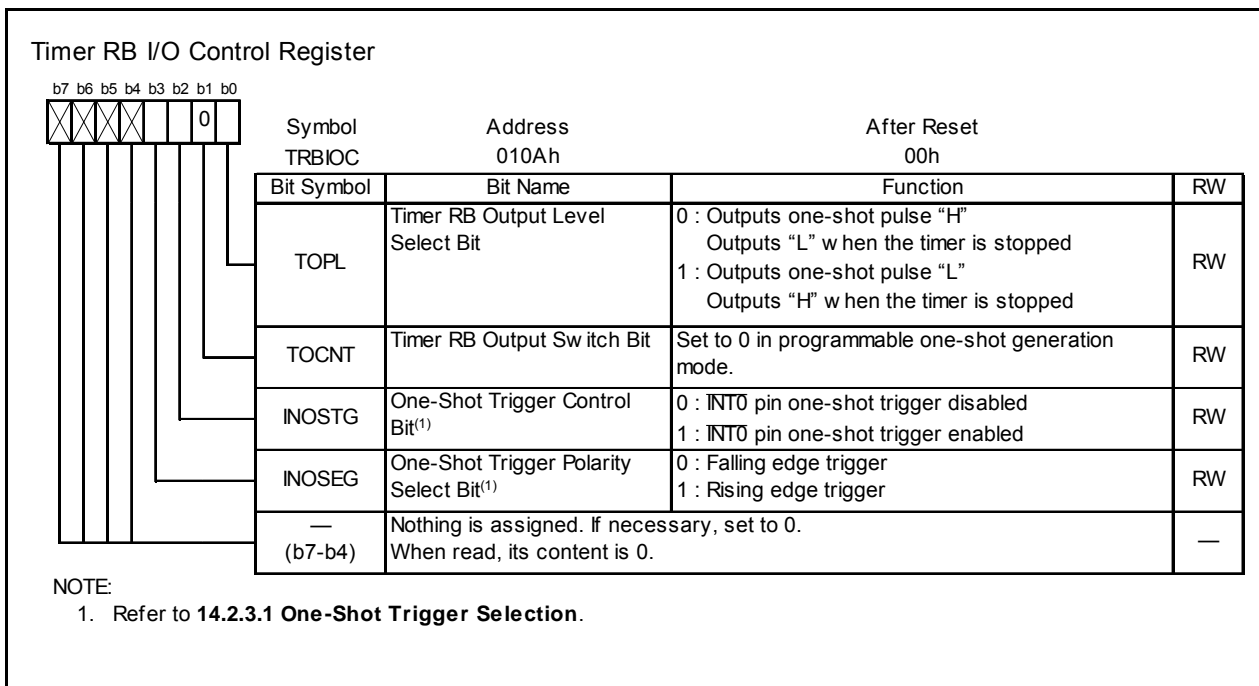


Figure 14.20 TRBIOC Register in Programmable One-Shot Generation Mode

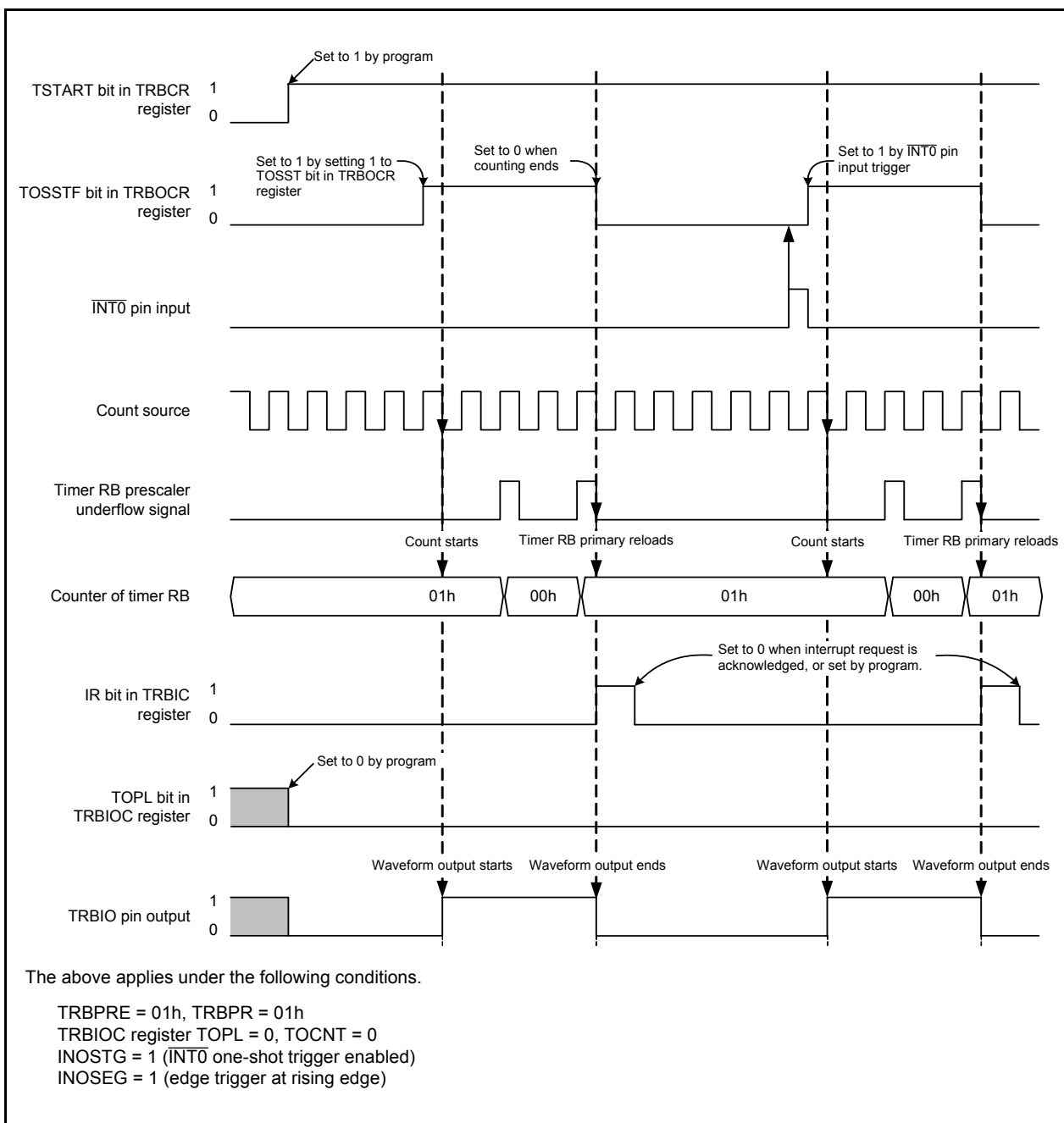


Figure 14.21 Operating Example of Programmable One-Shot Generation Mode

14.2.3.1 One-Shot Trigger Selection

In programmable one-shot generation mode and programmable wait one-shot generation mode, operation starts when a one-shot trigger is generated while the TCSTF bit in the TRBCR register is set to 1 (count starts).

A one-shot trigger can be generated by either of the following causes:

- 1 is written to the TOSST bit in the TRBOCR register by a program.
- Trigger input from the $\overline{\text{INT0}}$ pin.

When a one-shot trigger occurs, the TOSSTF bit in the TRBOCR register is set to 1 (one-shot operation in progress) after one or two cycles of the count source have elapsed. Then, in programmable one-shot generation mode, count operation begins and one-shot waveform output starts. (In programmable wait one-shot generation mode, count operation starts for the wait period.) If a one-shot trigger occurs while the TOSSTF bit is set to 1, no retriggering occurs.

To use trigger input from the $\overline{\text{INT0}}$ pin, input the trigger after making the following settings:

- Set the PD4_5 bit in the PD4 register to 0 (input port).
- Select the $\overline{\text{INT0}}$ digital filter with bits INT0F1 and INT0F0 in the INTF register.
- Select both edges or one edge with the INT0PL bit in INTEN register. If one edge is selected, further select falling or rising edge with the INOSEG bit in TRBIOC register.
- Set the INTOEN bit in the INTEN register to 0 (enabled).
- After completing the above, set the INOSTG bit in the TRBIOC register to 1 ($\overline{\text{INT}}$ pin one-shot trigger enabled).

Note the following points with regard to generating interrupt requests by trigger input from the $\overline{\text{INT0}}$ pin.

- Processing to handle the interrupts is required. Refer to **12. Interrupts**, for details.
- If one edge is selected, use the POL bit in the INT0IC register to select falling or rising edge. (The INOSEG bit in the TRBIOC register does not affect $\overline{\text{INT0}}$ interrupts).
- If a one-shot trigger occurs while the TOSSTF bit is set to 1, timer RB operation is not affected, but the value of the IR bit in the INTOIC register changes.

14.2.4 Programmable Wait One-Shot Generation Mode

In programmable wait one-shot generation mode, a one-shot pulse is output from the TRBO pin by a program or an external trigger input (input to the $\overline{\text{INT0}}$ pin) (refer to **Table 14.11 Programmable Wait One-Shot Generation Mode Specifications**). When a trigger is generated from that point, the timer outputs a pulse only once for a given length of time equal to the setting value in the TRBSC register after waiting for a given length of time equal to the setting value in the TRBPR register.

Figure 14.22 shows TRBIOC Register in Programmable Wait One-Shot Generation Mode. Figure 14.23 shows an Operating Example of Programmable Wait One-Shot Generation Mode.

Table 14.11 Programmable Wait One-Shot Generation Mode Specifications

| Item | Specification |
|-------------------------------------|---|
| Count sources | f1, f2, f8, timer RA underflow |
| Count operations | <ul style="list-style-type: none"> • Decrement the timer RB primary setting value. • When a count of the timer RB primary underflows, the timer reloads the contents of timer RB secondary before the count continues. • When a count of the timer RB secondary underflows, the timer reloads the contents of timer RB primary before the count completes and the TOSSTF bit is set to 0 (one-shot stops). • When the count stops, the timer reloads the contents of the reload register before it stops. |
| Wait time | $(n+1)(m+1)/f_i$ f_i : Count source frequency n : Value set in the TRBPRES register, m : Value set in the TRBPR register ⁽²⁾ |
| One-shot pulse output time | $(n+1)(p+1)/f_i$ f_i : Count source frequency n : Value set in the TRBPRES register, p : Value set in the TRBSC register |
| Count start conditions | <ul style="list-style-type: none"> • The TSTART bit in the TRBCR register is set to 1 (count starts) and the next trigger is generated. • Set the TOSST bit in the TRBOCR register to 1 (one-shot starts). • Input trigger to the INT0 pin |
| Count stop conditions | <ul style="list-style-type: none"> • When reloading completes after timer RB underflows during secondary period. • When the TOSSP bit in the TRBOCR register is set to 1 (one-shot stops). • When the TSTART bit in the TRBCR register is set to 0 (count starts). • When the TSTOP bit in the TRBCR register is set to 1 (count forcibly stops). |
| Interrupt request generation timing | In half a cycle of the count source after timer RB underflows during secondary period (complete at the same time as waveform output from the TRBO pin) [timer RB interrupt]. |
| TRBO pin function | Pulse output |
| INT0 pin functions | <ul style="list-style-type: none"> • When the INOSTG bit in the TRBIOC register is set to 0 ($\overline{\text{INT0}}$ one-shot trigger disabled): programmable I/O port or INT0 interrupt input • When the INOSTG bit in the TRBIOC register is set to 1 ($\overline{\text{INT0}}$ one-shot trigger enabled): external trigger (INT0 interrupt input) |
| Read from timer | The count value can be read out by reading registers TRBPR and TRBPRES. |
| Write to timer | <ul style="list-style-type: none"> • When registers TRBPRES, TRBSC, and TRBPR are written while the count stops, values are written to both the reload register and counter. • When registers TRBPRES, TRBSC, and TRBPR are written to during count operation, values are written to the reload registers only.⁽¹⁾ |
| Select functions | <ul style="list-style-type: none"> • Output level select function The TOPL bit in the TRBIOC register selects the output level of the one-shot pulse waveform. • One-shot trigger select function Refer to 14.2.3.1 One-Shot Trigger Selection. |

NOTES:

1. The set value is reflected at the following one-shot pulse after writing to registers TRBSC and TRBPR.
2. Do not set both the TRBPRES and TRBPR registers to 00h.

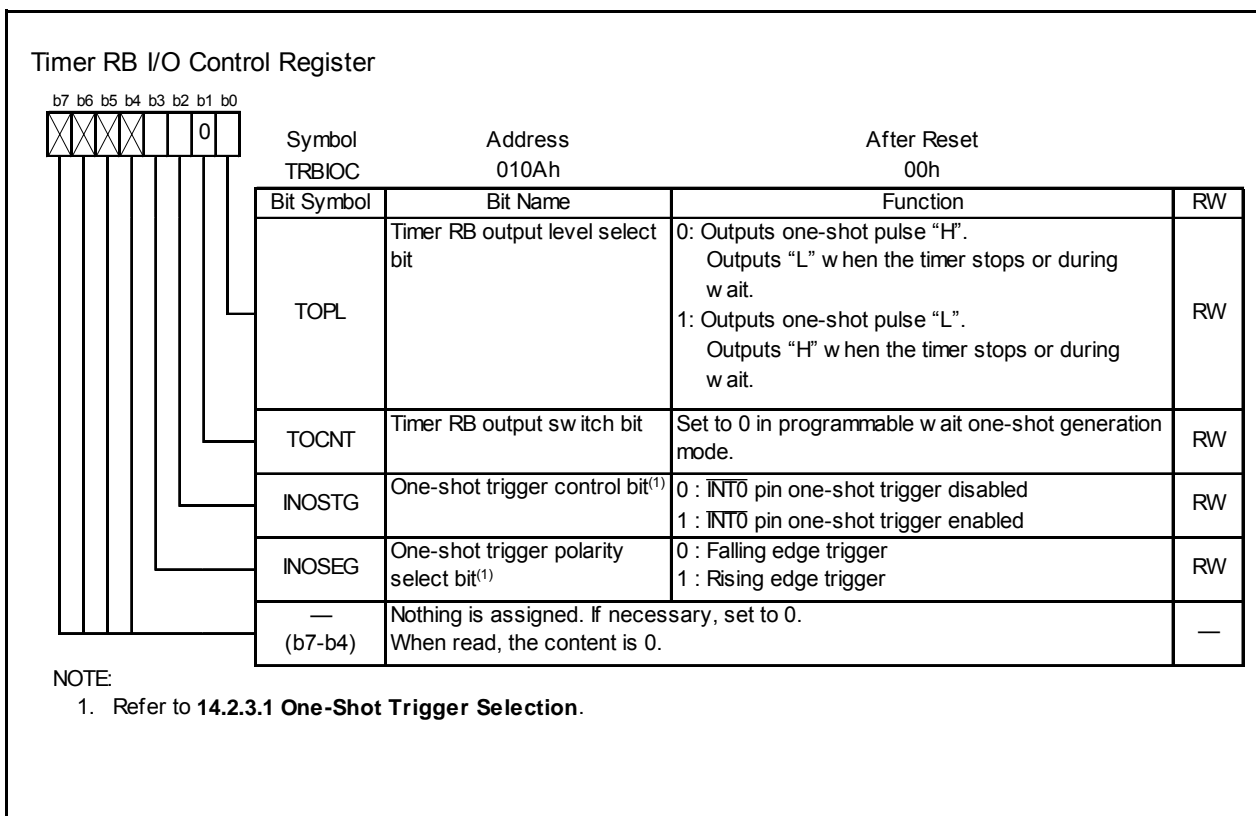


Figure 14.22 TRBIOC Register in Programmable Wait One-Shot Generation Mode

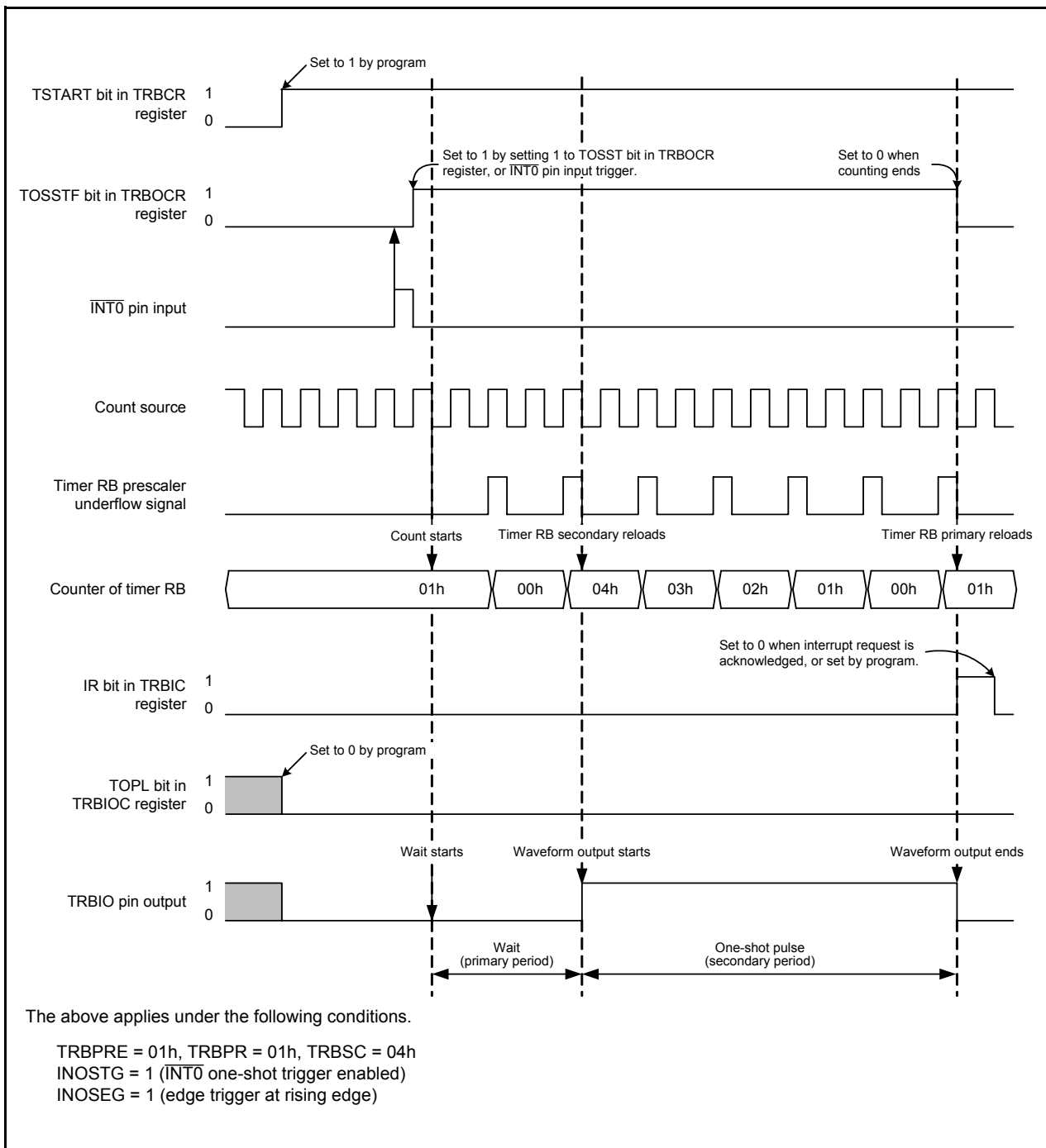


Figure 14.23 Operating Example of Programmable Wait One-Shot Generation Mode

14.2.5 Notes on Timer RB

- Timer RB stops counting after a reset. Set the values in the timer RB and timer RB prescalers before the count starts.
- Even if the prescaler and timer RB is read out in 16-bit units, these registers are read 1 byte at a time by the MCU. Consequently, the timer value may be updated during the period when these two registers are being read.
- In programmable one-shot generation mode and programmable wait one-shot generation mode, when setting the TSTART bit in the TRBCR register to 0 (count stops) or setting the TOSSP bit in the TRBOCR register to 1 (one-shot stops), the timer reloads the value of reload register and stops. Therefore, in programmable one-shot generation mode and programmable wait one-shot generation mode, read the timer count value before the timer stops.
- The TCSTF bit remains 0 (count stops) for 1 to 2 cycles of the count source after setting the TSTART bit to 1 (count starts) while the count is stopped.

During this time, do not access registers associated with timer RB⁽¹⁾ other than the TCSTF bit.

The TCSTF bit remains 1 for 1 to 2 cycles of the count source after setting the TSTART bit to 0 (count stops) while the count is in progress. Timer RB counting is stopped when the TCSTF bit is set to 0.

During this time, do not access registers associated with timer RB⁽¹⁾ other than the TCSTF bit.

NOTE:

1. Registers associated with timer RB: TRBCR, TRBOCR, TRBIOC, TRBMR, TRBPRES, TRBSC, and TRBPR.

- If the TSTOP bit in the TRBCR register is set to 1 during timer operation, timer RB stops immediately.
- If 1 is written to the TOSST or TOSSP bit in the TRBOCR register, the value of the TOSSTF bit changes after one or two cycles of the count source have elapsed. If the TOSSP bit is written to 1 during the period between when the TOSST bit is written to 1 and when the TOSSTF bit is set to 1, the TOSSTF bit may be set to either 0 or 1 depending on the content state. Likewise, if the TOSST bit is written to 1 during the period between when the TOSSP bit is written to 1 and when the TOSSTF bit is set to 0, the TOSSTF bit may be set to either 0 or 1.

14.2.5.1 Timer mode

The following workaround should be performed in timer mode.

To write to registers TRBPRES and TRBPR during count operation (TCSTF bit is set to 1), note the following points:

- When the TRBPRES register is written continuously, allow three or more cycles of the count source for each write interval.
- When the TRBPR register is written continuously, allow three or more cycles of the prescaler underflow for each write interval.

14.2.5.2 Programmable waveform generation mode

The following three workarounds should be performed in programmable waveform generation mode.

- (1) To write to registers TRBPRE and TRBPR during count operation (TCSTF bit is set to 1), note the following points:
 - When the TRBPRE register is written continuously, allow three or more cycles of the count source for each write interval.
 - When the TRBPR register is written continuously, allow three or more cycles of the prescaler underflow for each write interval.
- (2) To change registers TRBPRE and TRBPR during count operation (TCSTF bit is set to 1), synchronize the TRBO output cycle using a timer RB interrupt, etc. This operation should be performed only once in the same output cycle. Also, make sure that writing to the TRBPR register does not occur during period A shown in Figures 14.24 and 14.25.

The following shows the detailed workaround examples.

- Workaround example (a):
As shown in Figure 14.24, write to registers TRBSC and TRBPR in the timer RB interrupt routine. These write operations must be completed by the beginning of period A.

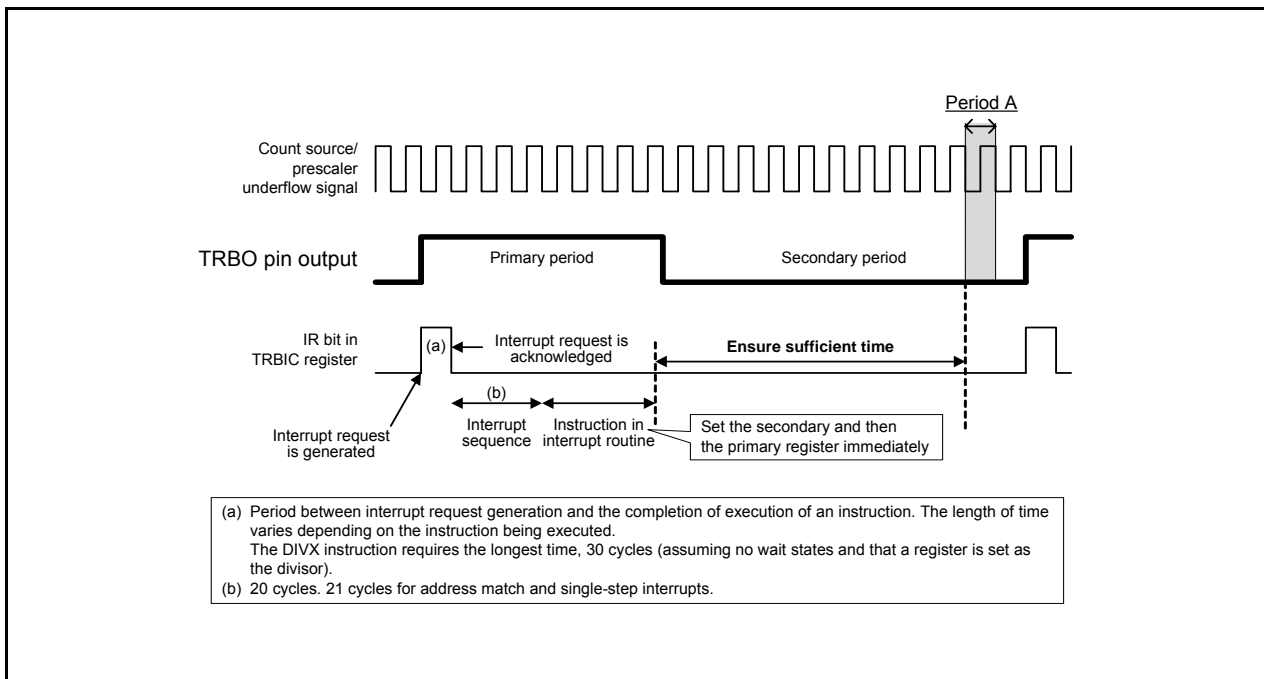


Figure 14.24 Workaround Example (a) When Timer RB interrupt is Used

- Workaround example (b):

As shown in Figure 14.25 detect the start of the primary period by the TRBO pin output level and write to registers TRBSC and TRBPR. These write operations must be completed by the beginning of period A.

If the port register's bit value is read after the port direction register's bit corresponding to the TRBO pin is set to 0 (input mode), the read value indicates the TRBO pin output value.

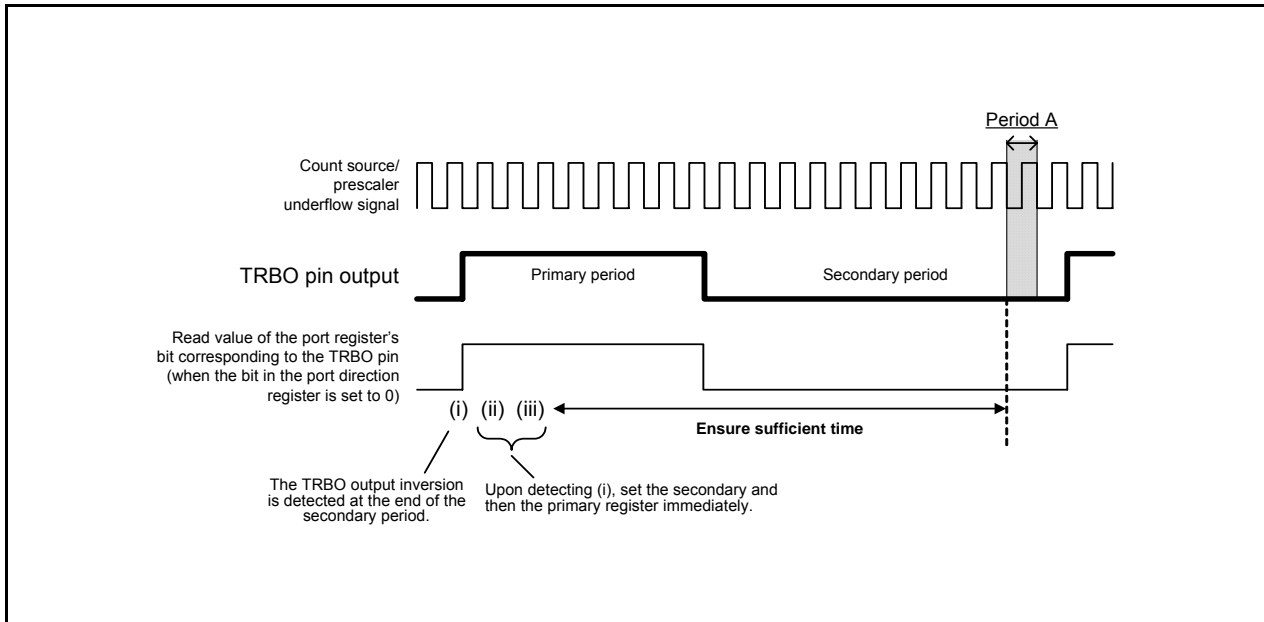


Figure 14.25 Workaround Example (b) When TRBO Pin Output Value is Read

- (3) To stop the timer counting in the primary period, use the TSTOP bit in the TRBCR register. In this case, registers TRBPRES and TRBPR are initialized and their values are set to the values after reset.

14.2.5.3 Programmable one-shot generation mode

The following two workarounds should be performed in programmable one-shot generation mode.

- (1) To write to registers TRBPRES and TRBPR during count operation (TCSTF bit is set to 1), note the following points:
 - When the TRBPRES register is written continuously during count operation (TCSTF bit is set to 1), allow three or more cycles of the count source for each write interval.
 - When the TRBPR register is written continuously during count operation (TCSTF bit is set to 1), allow three or more cycles of the prescaler underflow for each write interval.
- (2) Do not set both the TRBPRES and TRBPR registers to 00h.

14.2.5.4 Programmable wait one-shot generation mode

The following three workarounds should be performed in programmable wait one-shot generation mode.

- (1) To write to registers TRBPRE and TRBPR during count operation (TCSTF bit is set to 1), note the following points:
 - When the TRBPRE register is written continuously, allow three or more cycles of the count source for each write interval.
 - When the TRBPR register is written continuously, allow three or more cycles of the prescaler underflow for each write interval.
- (2) Do not set both the TRBPRE and TRBPR registers to 00h.
- (3) Set registers TRBSC and TRBPR using the following procedure.
 - (a) To use “ $\overline{\text{INT0}}$ pin one-shot trigger enabled” as the count start condition
Set the TRBSC register and then the TRBPR register. At this time, after writing to the TRBPR register, allow an interval of 0.5 or more cycles of the count source before trigger input from the $\overline{\text{INT0}}$ pin.
 - (b) To use “writing 1 to TOSST bit” as the start condition
Set the TRBSC register, the TRBPR register, and then TOSST bit. At this time, after writing to the TRBPR register, allow an interval of 0.5 or more cycles of the count source before writing to the TOSST bit.

14.3 Timer RC

14.3.1 Overview

Timer RC is a 16-bit timer with four I/O pins.

Timer RC uses either f1 or fOCO40M as its operation clock. Table 14.12 lists the Timer RC Operation Clock.

Table 14.12 Timer RC Operation Clock

| Condition | Timer RC Operation Clock |
|--|--------------------------|
| Count source is f1, f2, f4, f8, f32, or TRCCLK input (bits TCK2 to TCK0 in TRCCR1 register are set to a value from 000b to 101b) | f1 |
| Count source is fOCO40M (bits TCK2 to TCK0 in TRCCR1 register are set to 110b) | fOCO40M |

Table 14.13 lists the Timer RC I/O Pins, and Figure 14.26 shows a Timer RC Block Diagram.

Timer RC has three modes.

- Timer mode

- Input capture function The counter value is captured to a register, using an external signal as the trigger.
- Output compare function Matches between the counter and register values are detected. (Pin output state changes when a match is detected.)

The following two modes use the output compare function.

- PWM mode Pulses of a given width are output continuously.
- PWM2 mode A one-shot waveform or PWM waveform is output following the trigger after the wait time has elapsed.

Input capture function, output compare function, and PWM mode settings may be specified independently for each pin.

In PWM2 mode waveforms are output based on a combination of the counter or the register.

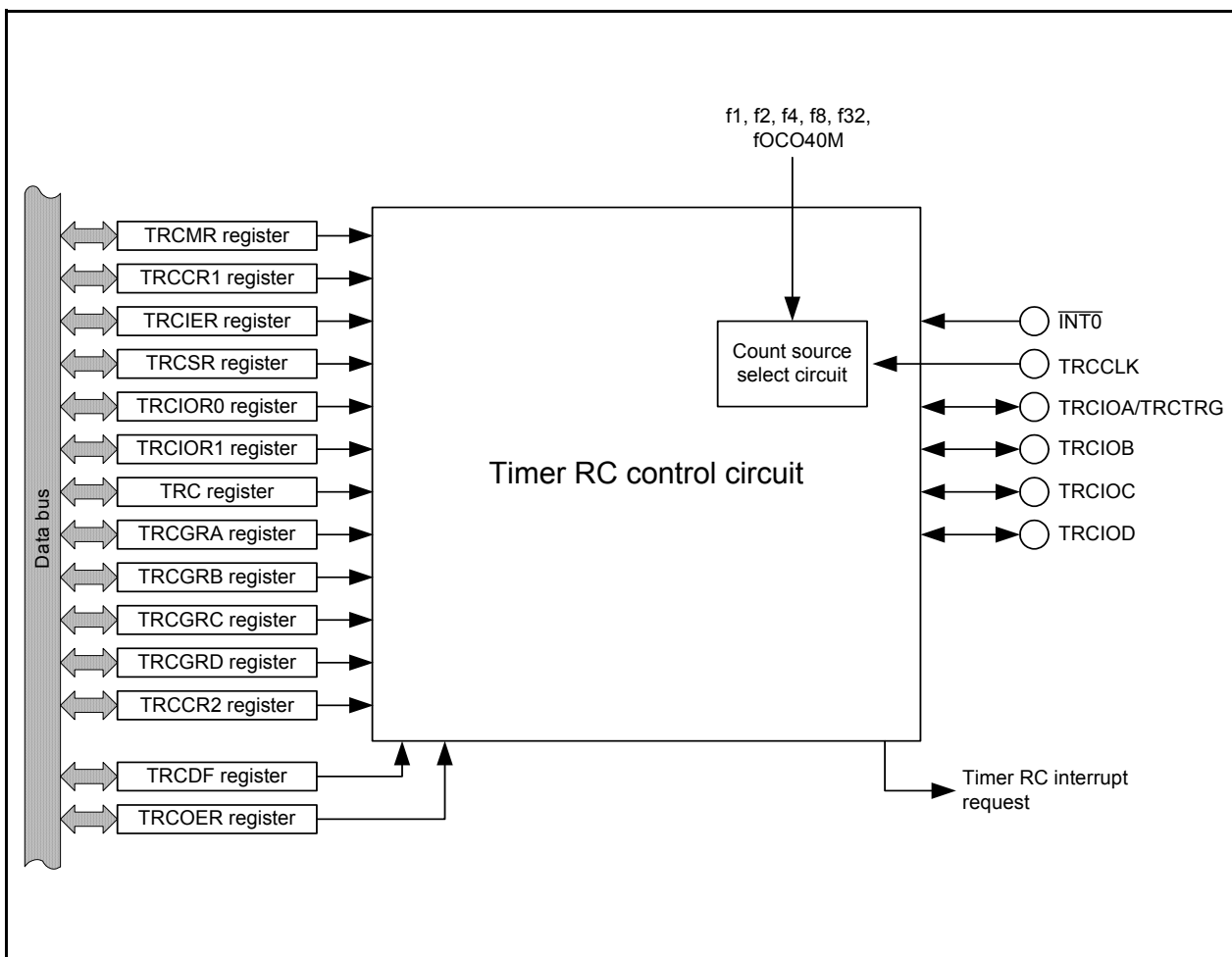


Figure 14.26 Timer RC Block Diagram

Table 14.13 Timer RC I/O Pins

| Pin Name | I/O | Function |
|--------------|-------|---|
| TRCIOA(P5_1) | I/O | Function differs according to the mode. Refer to descriptions of individual modes for details |
| TRCIOB(P5_2) | | |
| TRCIOC(P5_3) | | |
| TRCIOD(P5_4) | | |
| TRCCLK(P5_0) | Input | External clock input |
| TRCTRG(P5_1) | Input | PWM2 mode external trigger input |

14.3.2 Registers Associated with Timer RC

Table 14.14 lists the Registers Associated with Timer RC. Figures 14.27 to 14.37 show details of the registers associated with timer RC.

Table 14.14 Registers Associated with Timer RC

| Address | Symbol | Mode | | | | Related Information |
|----------------|---------|------------------------|-------------------------|-------|-------|--|
| | | Timer | | PWM | PWM2 | |
| | | Input Capture Function | Output Compare Function | | | |
| 0008h | MSTCR | Valid | Valid | Valid | Valid | Module operation enable register Figure 14.27 MSTCR Register |
| 0120h | TRCMR | Valid | Valid | Valid | Valid | Timer RC mode register Figure 14.28 TRCMR Register |
| 0121h | TRCCR1 | Valid | Valid | Valid | Valid | Timer RC control register 1 Figure 14.29 TRCCR1 Register Figure 14.50 TRCCR1 Register for Output Compare Function Figure 14.53 TRCCR1 Register in PWM Mode Figure 14.57 TRCCR1 Register in PWM2 Mode |
| 0122h | TRCIER | Valid | Valid | Valid | Valid | Timer RC interrupt enable register Figure 14.30 TRCIER Register |
| 0123h | TRCSR | Valid | Valid | Valid | Valid | Timer RC status register Figure 14.31 TRCSR Register |
| 0124h | TRCIOR0 | Valid | Valid | - | - | Timer RC I/O control register 0, timer RC I/O control register 1 Figure 14.37 Registers TRCIOR0 and TRCIOR1 Figure 14.44 TRCIOR0 Register for Input Capture Function Figure 14.45 TRCIOR1 Register for Input Capture Function Figure 14.48 TRCIOR0 Register for Output Compare Function Figure 14.49 TRCIOR1 Register for Output Compare Function |
| 0125h | TRCIOR1 | | | | | |
| 0126h 0127h | TRC | Valid | Valid | Valid | Valid | Timer RC counter Figure 14.32 TRC Register |
| 0128h 0129h | TRCGRA | Valid | Valid | Valid | Valid | Timer RC general registers A, B, C, and D Figure 14.33 Registers TRCGRA, TRCGRB, TRCGRC, and TRCGRD |
| 012Ah 012Bh | TRCGRB | | | | | |
| 012Ch 012Dh | TRCGRC | | | | | |
| 012Eh 012Fh | TRCGRD | | | | | |
| 0130h | TRCCR2 | - | - | - | Valid | Timer RC control register 2 Figure 14.34 TRCCR2 Register |
| 0131h | TRCDF | Valid | - | - | Valid | Timer RC digital filter function select register Figure 14.35 TRCDF Register |
| 0132h | TRCOER | - | Valid | Valid | Valid | Timer RC output master enable register Figure 14.36 TRCOER Register |

- : Invalid

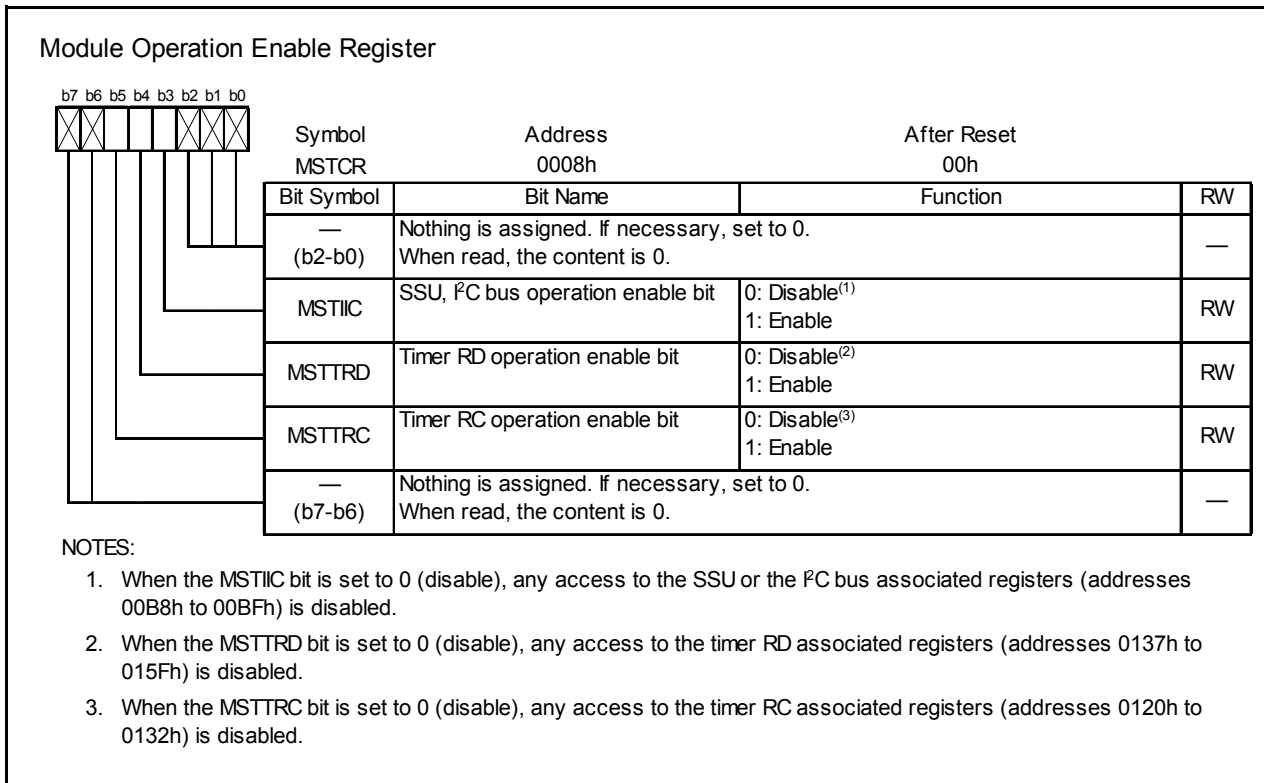


Figure 14.27 MSTCR Register

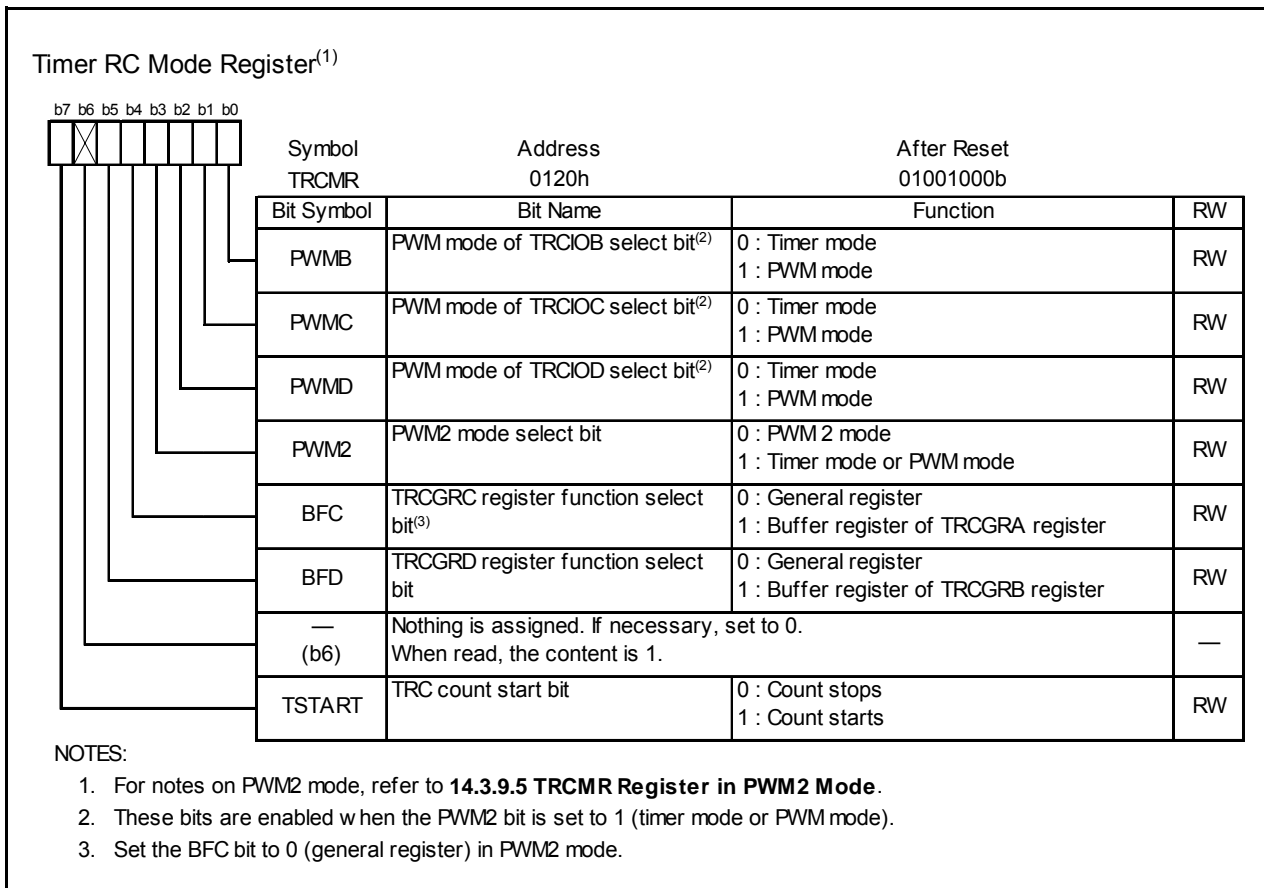


Figure 14.28 TRCMR Register

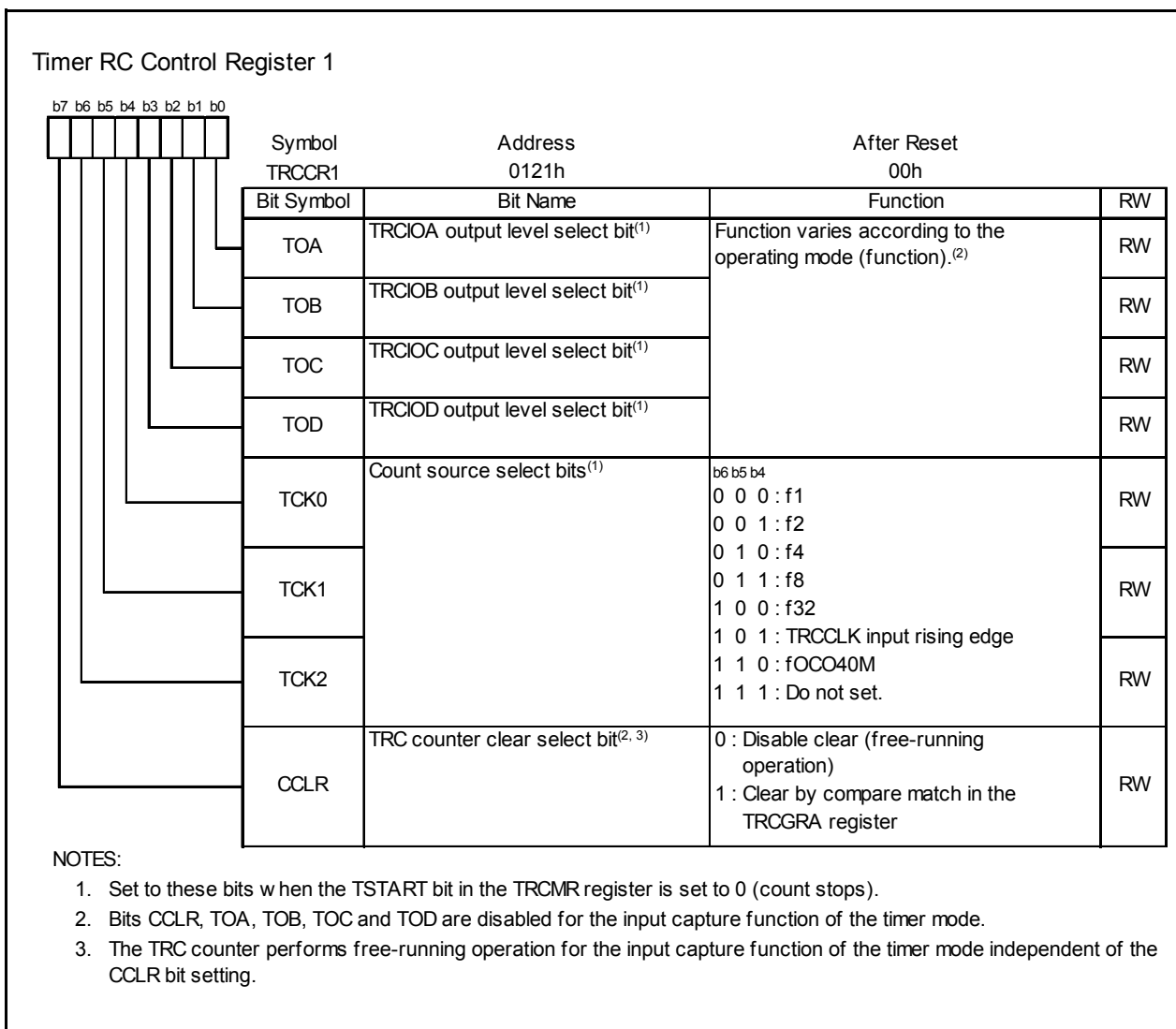


Figure 14.29 TRCCR1 Register

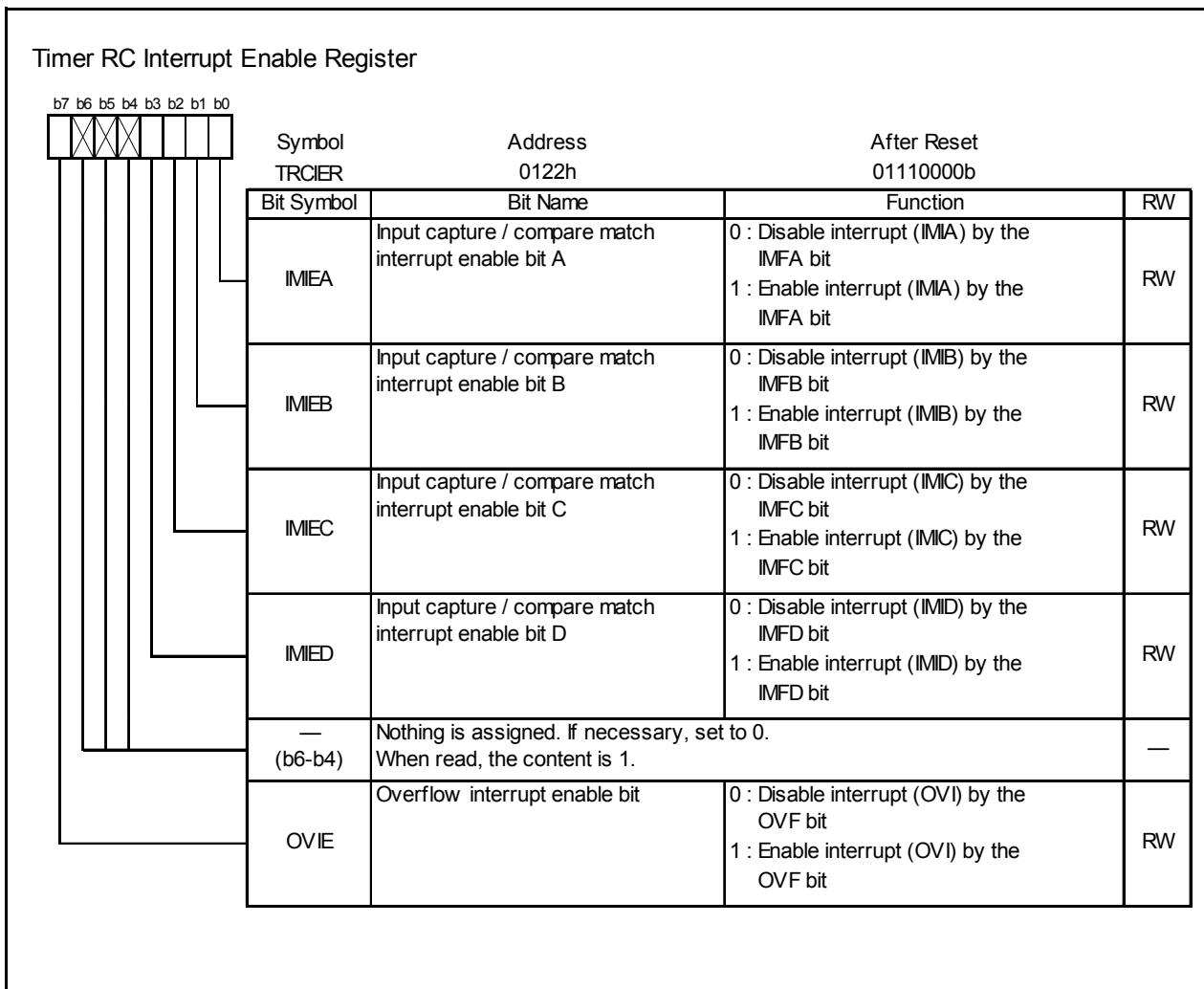


Figure 14.30 TRCIER Register

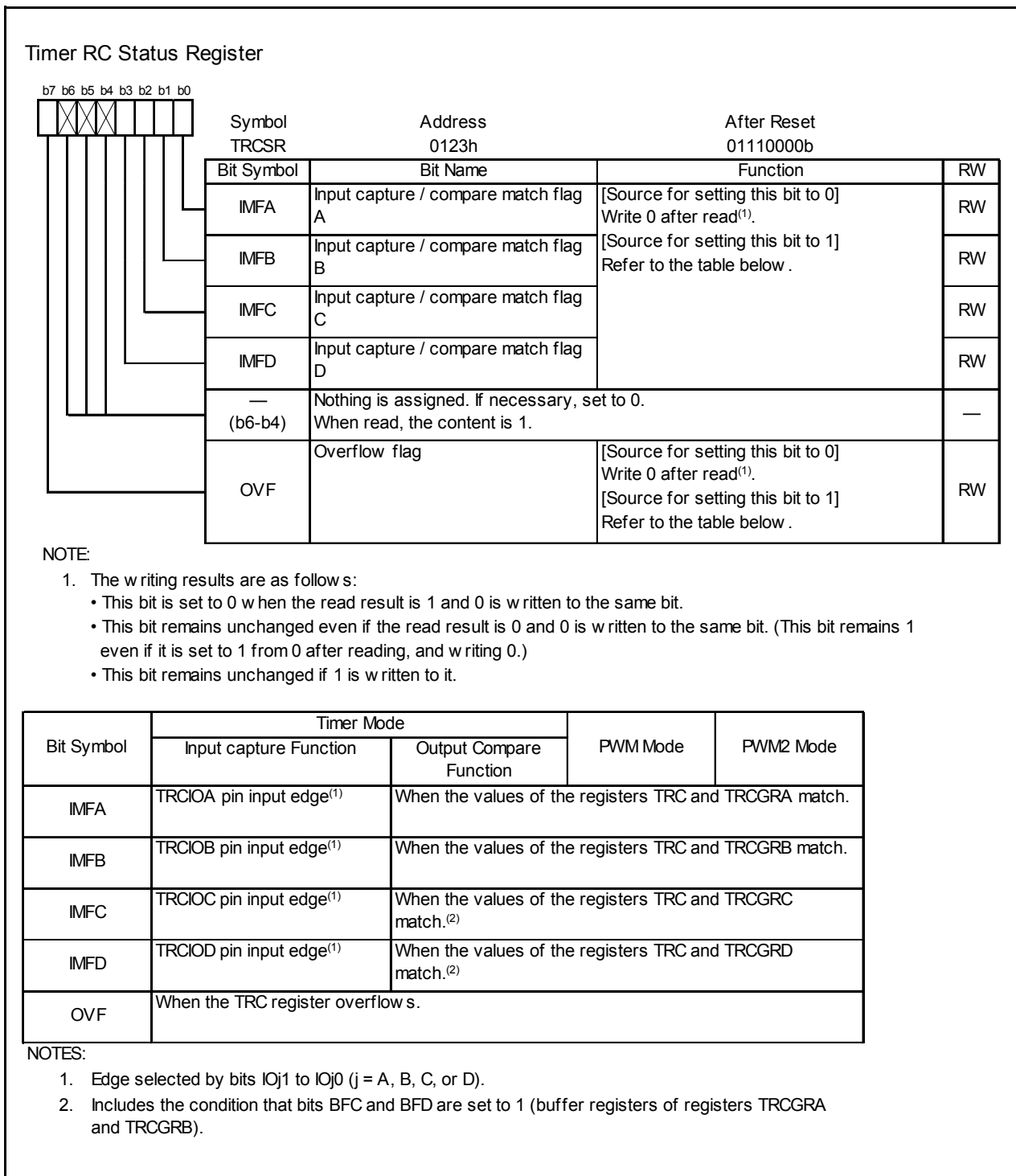


Figure 14.31 TRCSR Register

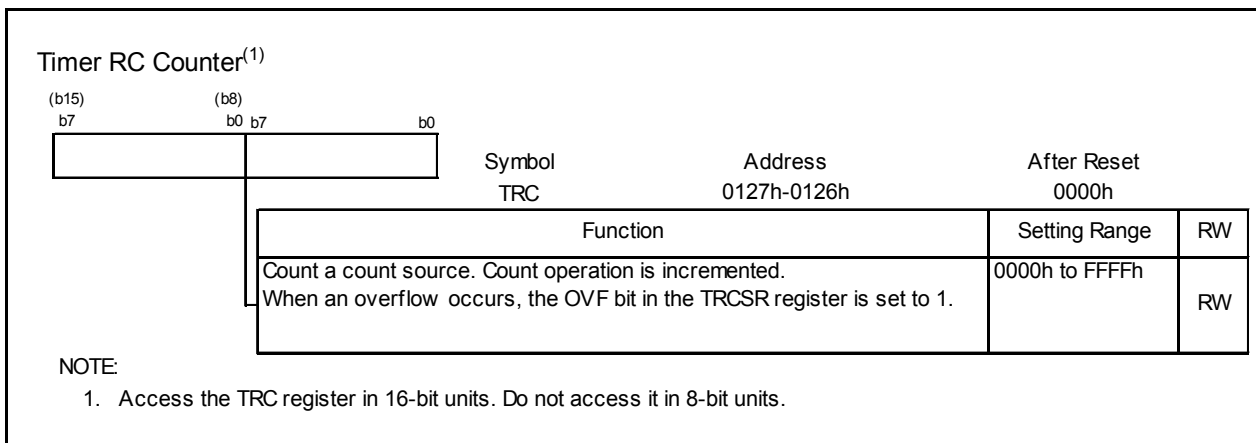


Figure 14.32 TRC Register

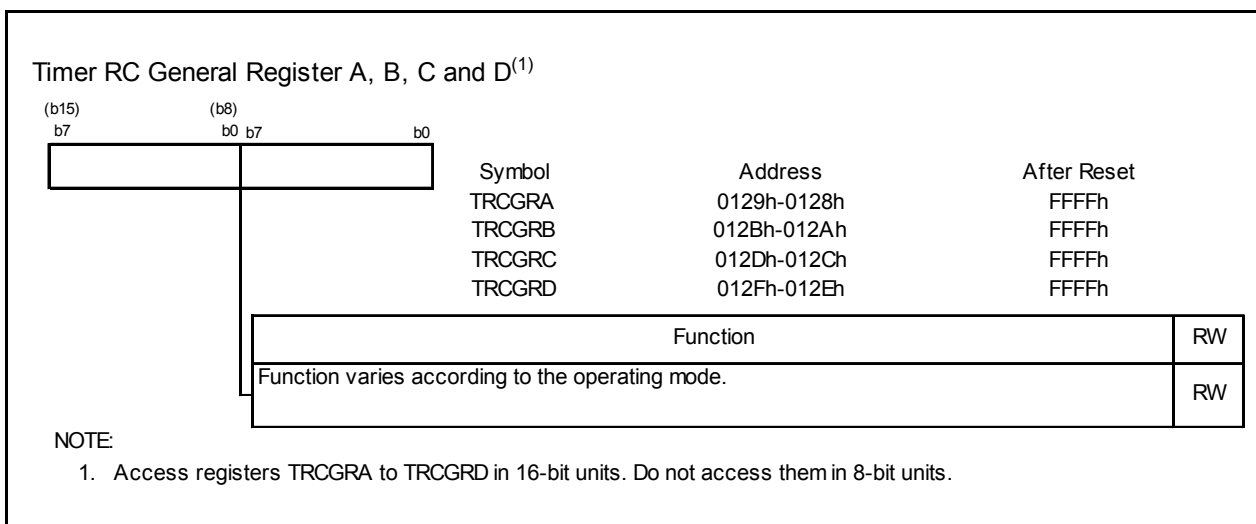


Figure 14.33 Registers TRCGRA, TRCGRB, TRCGRC, and TRCGRD

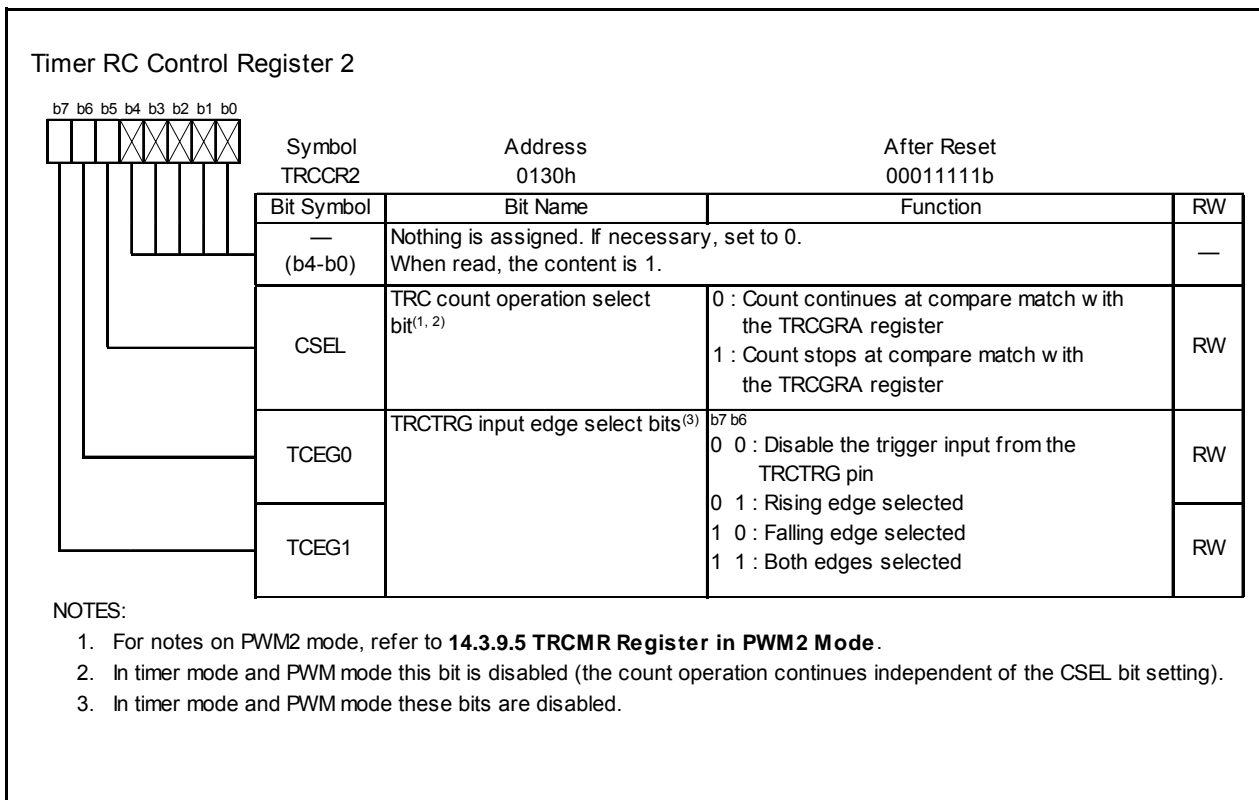


Figure 14.34 TRCCR2 Register

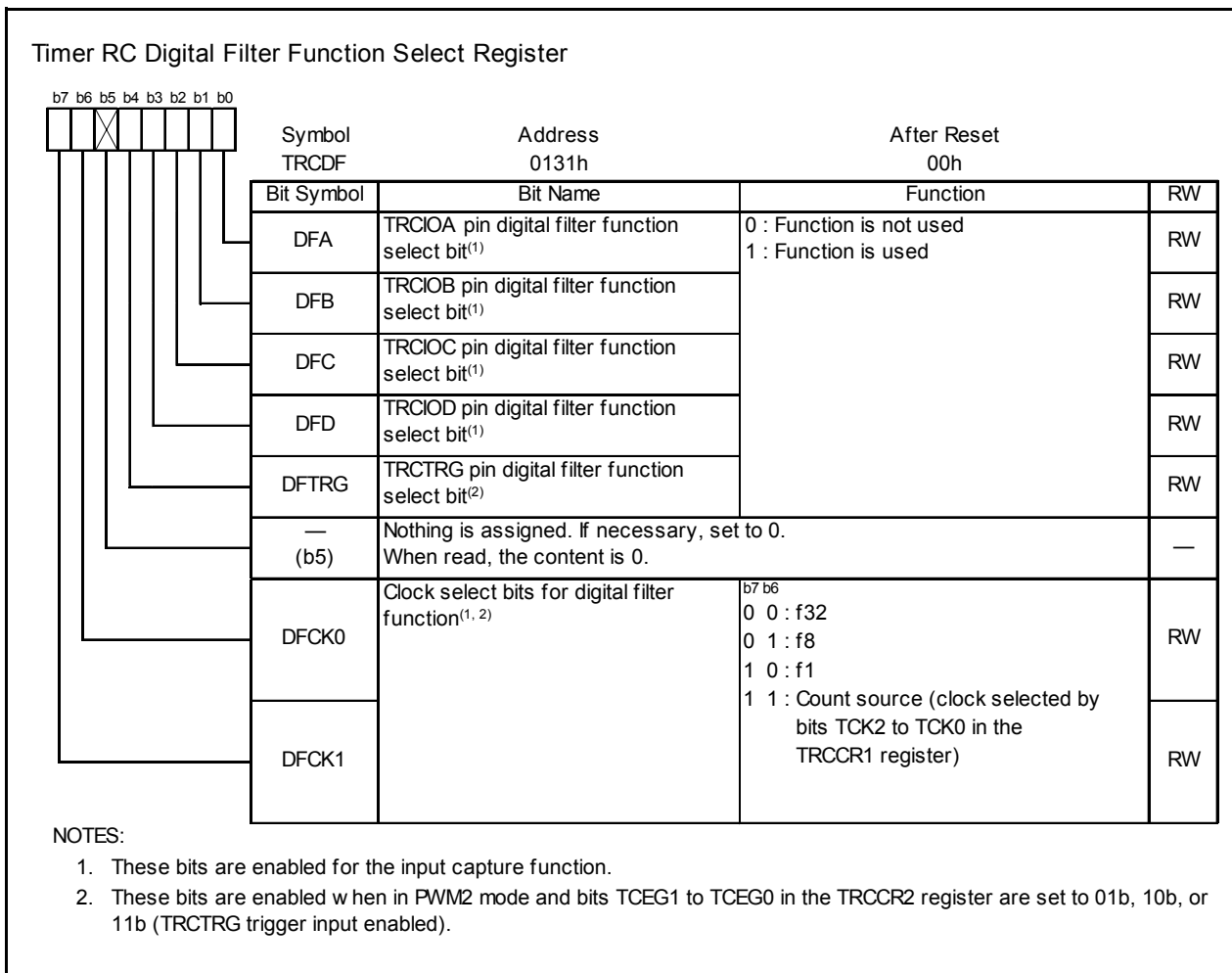


Figure 14.35 TRCDF Register

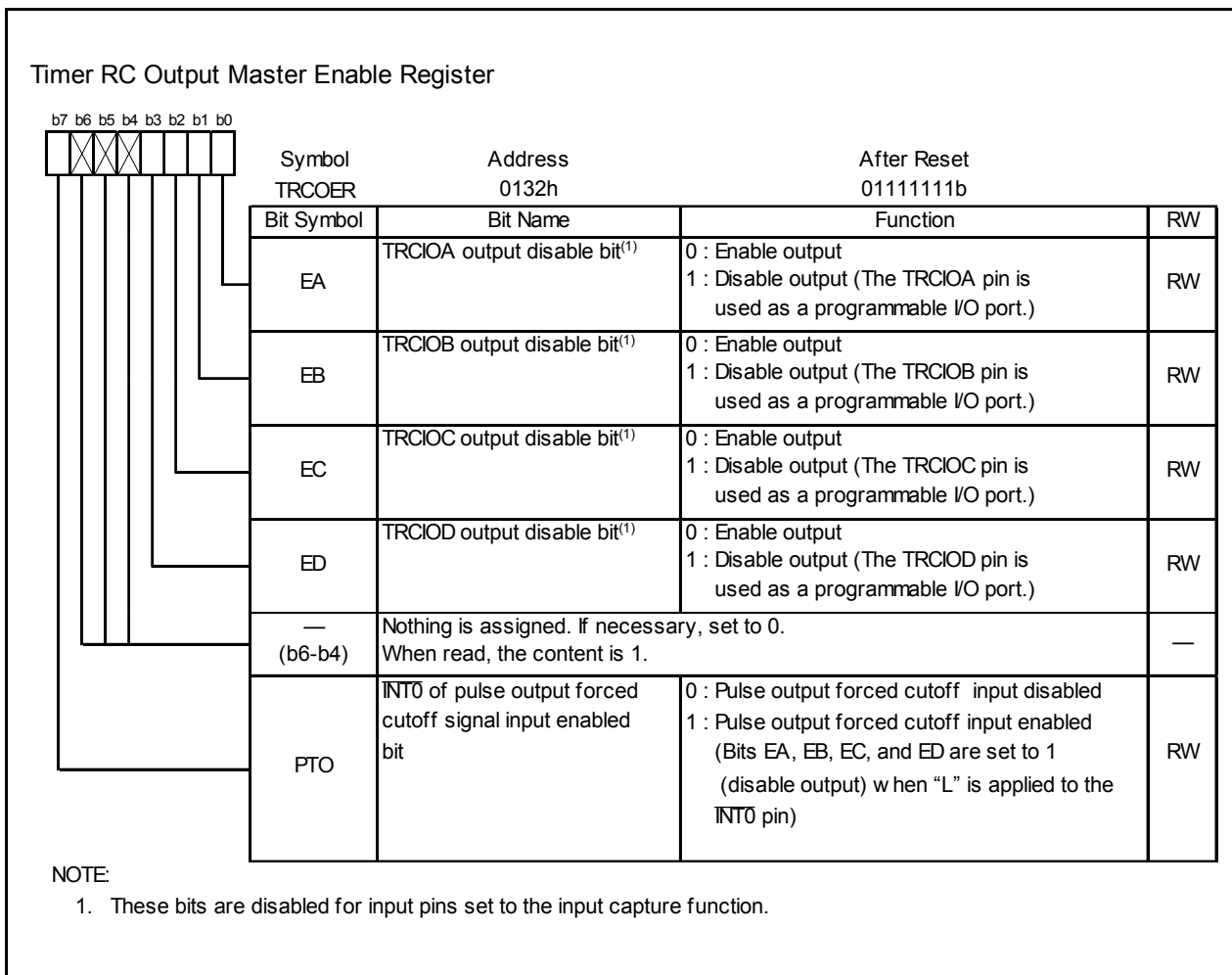


Figure 14.36 TRCOER Register

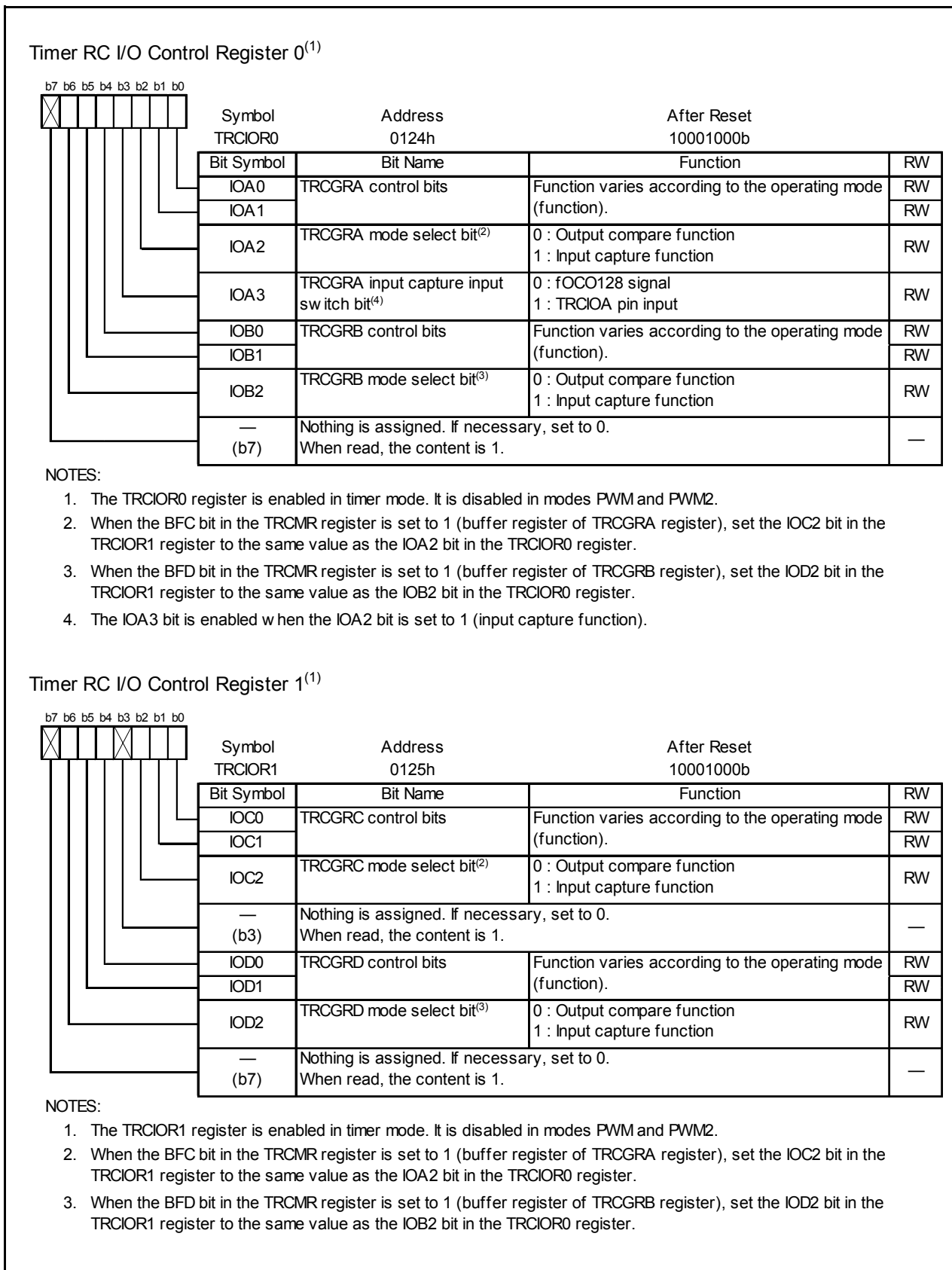


Figure 14.37 Registers TRCIOR0 and TRCIOR1

14.3.3 Common Items for Multiple Modes

14.3.3.1 Count Source

The method of selecting the count source is common to all modes.
 Table 14.15 lists the Count Source Selection, and Figure 14.38 shows a Count Source Block Diagram.

Table 14.15 Count Source Selection

| Count Source | Selection Method |
|-------------------------------------|---|
| f1, f2, f4, f8, f32 | Count source selected using bits TCK2 to TCK0 in TRCCR1 register |
| fOCO40M | FRA00 bit in FRA0 register set to 1 (high-speed on-chip oscillator on) and bits TCK2 to TCK0 in TRCCR1 register are set to 110b (fOCO40M) |
| External signal input to TRCCLK pin | Bits TCK2 to TCK0 in TRCCR1 register are set to 101b (count source is rising edge of external clock) and PD5_0 bit in PD5 register is set to 0 (input mode) |

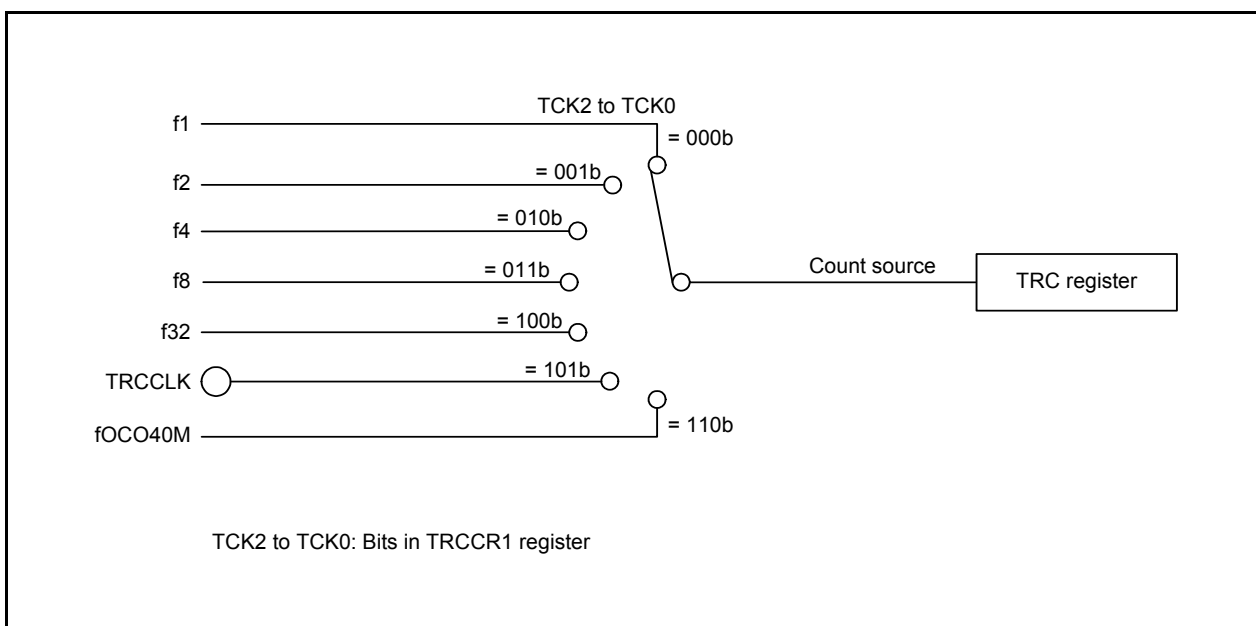


Figure 14.38 Count Source Block Diagram

The pulse width of the external clock input to the TRCCLK pin should be three cycles or more of the timer RC operation clock (see **Table 14.12 Timer RC Operation Clock**).

To select fOCO40M as the count source, set the FRA00 bit in the FRA0 register set to 1 (high-speed on-chip oscillator on), and then set bits TCK2 to TCK0 in the TRCCR1 register to 110b (fOCO40M).

14.3.3.2 Buffer Operation

Bits BFC and BFD in the TRCMR register are used to select the TRCGRC or TRCGRD register as the buffer register for the TRCGRA or TRCGRB register.

- Buffer register for TRCGRA register: TRCGRC register
- Buffer register for TRCGRB register: TRCGRD register

Buffer operation differs depending on the mode.

Table 14.16 lists the Buffer Operation in Each Mode, Figure 14.39 shows the Buffer Operation for Input Capture Function, and Figure 14.40 shows the Buffer Operation for Output Compare Function.

Table 14.16 Buffer Operation in Each Mode

| Function, Mode | Transfer Timing | Transfer Destination Register |
|-------------------------|--|---|
| Input capture function | Input capture signal input | Contents of TRCGRA (TRCGRB) register are transferred to buffer register |
| Output compare function | Compare match between TRC register and TRCGRA (TRCGRB) register | Contents of buffer register are transferred to TRCGRA (TRCGRB) register |
| PWM mode | | |
| PWM2 mode | <ul style="list-style-type: none"> • Compare match between TRC register and TRCGRA register • TRCTRГ pin trigger input | Contents of buffer register (TRCGRD) are transferred to TRCGRB register |

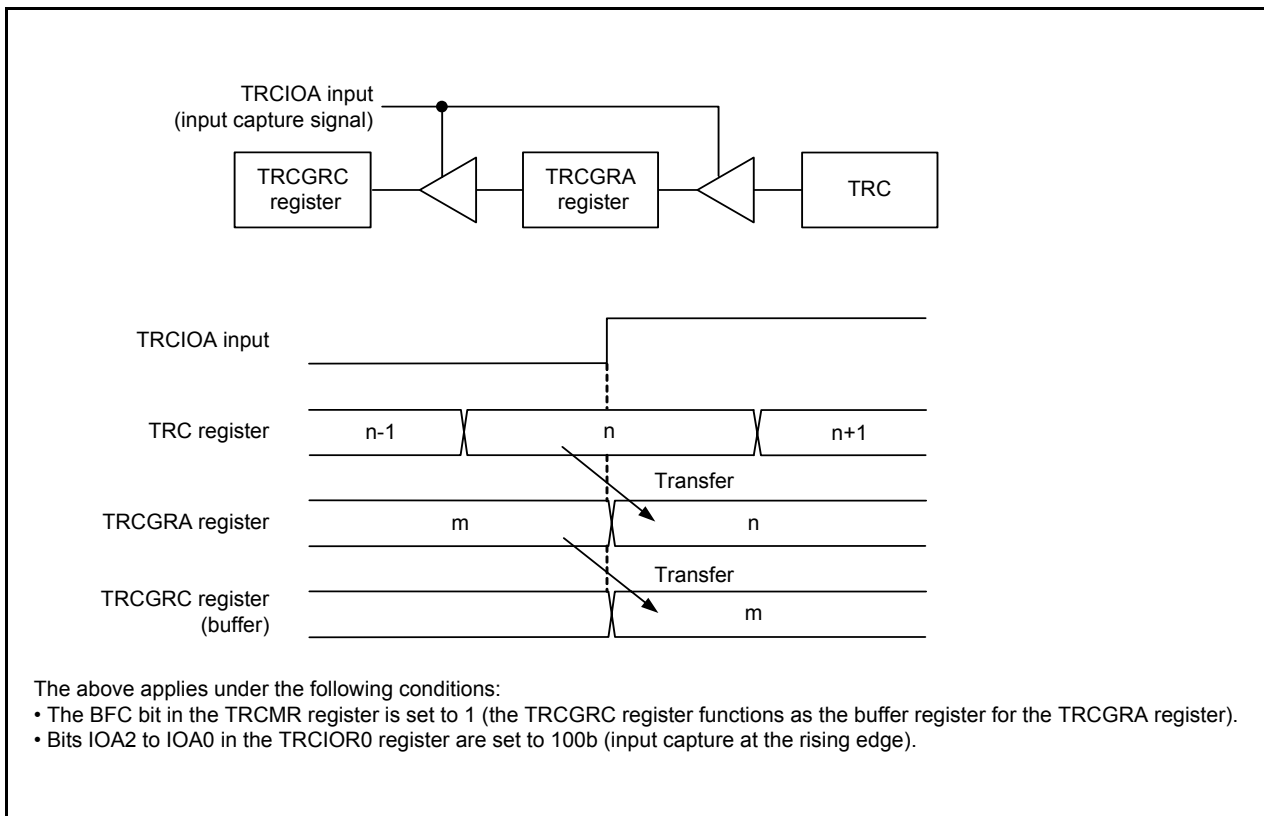


Figure 14.39 Buffer Operation for Input Capture Function

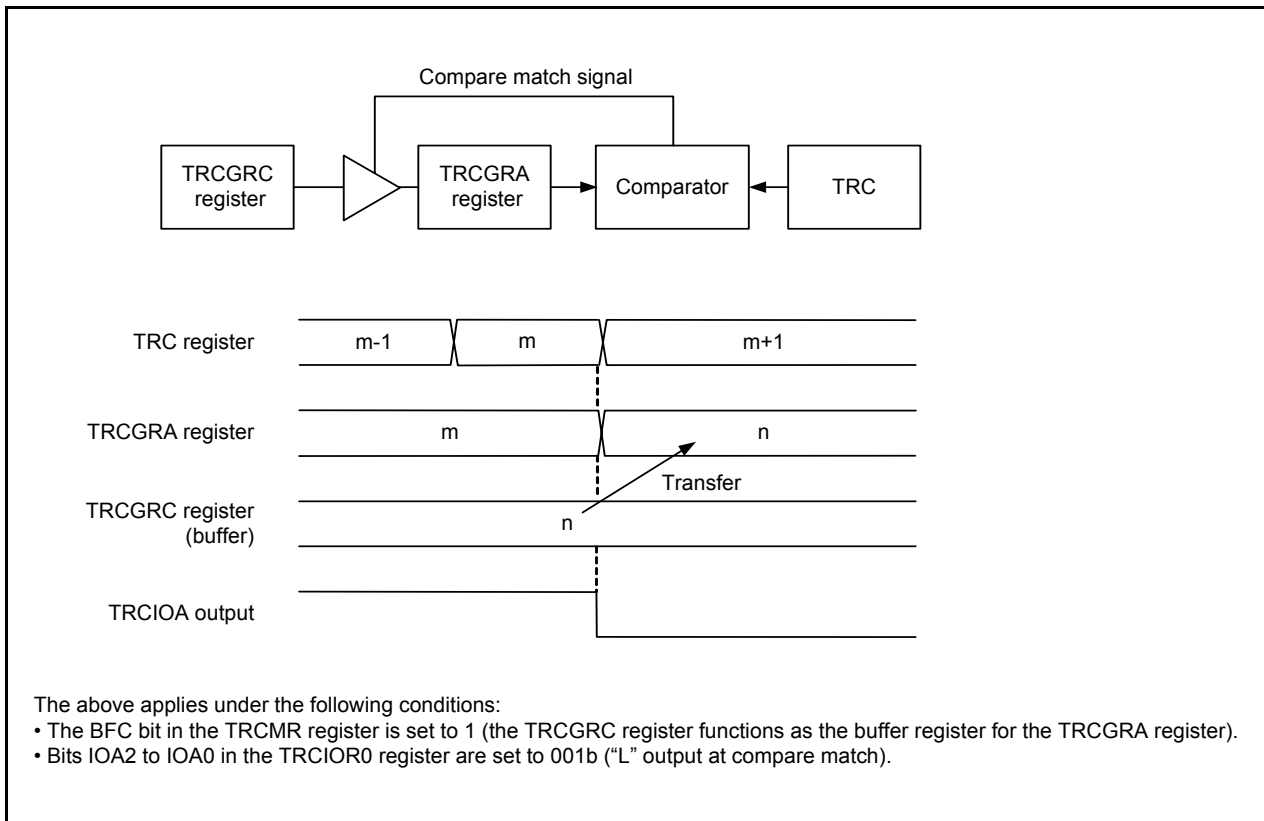


Figure 14.40 Buffer Operation for Output Compare Function

Make the following settings in timer mode.

- To use the TRCGRC register as the buffer register for the TRCGRA register:
Set the IOC2 bit in the TRCIOR1 register to the same value as the IOA2 bit in the TRCIOR0 register.
- To use the TRCGRD register as the buffer register for the TRCGRB register:
Set the IOD2 bit in the TRCIOR1 register to the same value as the IOB2 bit in the TRCIOR0 register.

The output compare function, PWM mode, or PWM2 mode, and the TRCGRC or TRCGRD register is functioning as a buffer register, the IMFC bit or IMFD bit in the TRCSR register is set to 1 when a compare match with the TRC register occurs.

The input capture function and the TRCGRC register or TRCGRD register is functioning as a buffer register, the IMFC bit or IMFD bit in the TRCSR register is set to 1 at the input edge of a signal input to the TRCIOC pin or TRCIOD pin.

14.3.3.3 Digital Filter

The input to TRCTR_j or TRCIO_j (j = A, B, C, or D) is sampled, and the level is considered to be determined when three matches occur. The digital filter function and sampling clock are selected using the TRCDF register. Figure 14.41 shows a Block Diagram of Digital Filter.

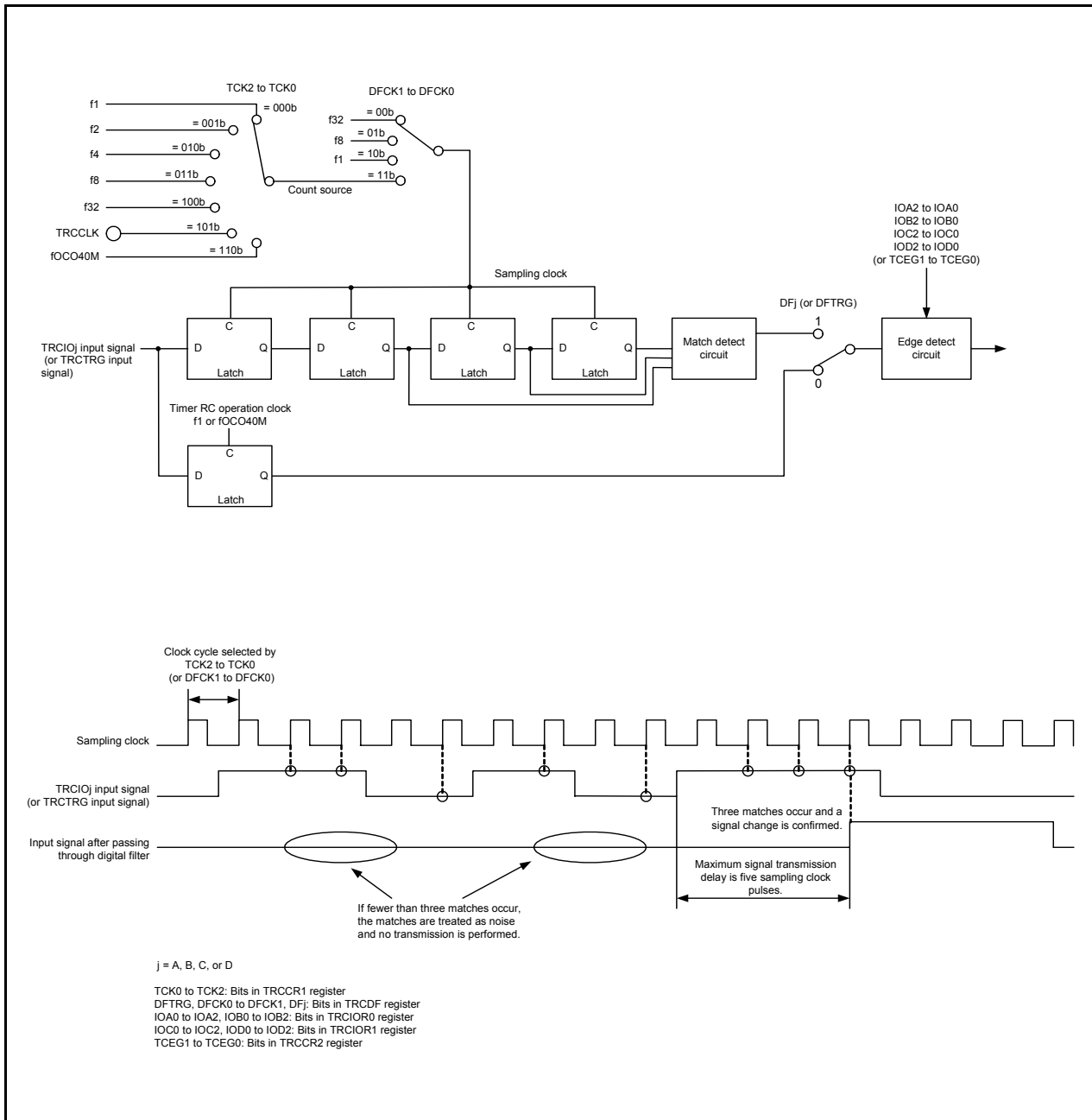


Figure 14.41 Block Diagram of Digital Filter

14.3.3.4 Forced Cutoff of Pulse Output

When using the timer mode's output compare function, the PWM mode, or the PWM2 mode, pulse output from the TRCIOj (j = A, B, C, or D) output pin can be forcibly cut off and the TRCIOj pin set to function as a programmable I/O port by means of input to the $\overline{\text{INT0}}$ pin.

A pin used for output by the timer mode's output compare function, the PWM mode, or the PWM2 mode can be set to function as the timer RC output pin by setting the Ej bit in the TRCOER register to 0 (timer RC output enabled). If "L" is input to the $\overline{\text{INT0}}$ pin while the PTO bit in the TRCOER register is set to 1 (pulse output forced cutoff signal input $\overline{\text{INT0}}$ enabled), bits EA, EB, EC, and ED in the TRCOER register are all set to 1 (timer RC output disabled, TRCIOj output pin functions as the programmable I/O port). When one or two cycles of the timer RC operation clock after "L" input to the $\overline{\text{INT0}}$ pin (refer to **Table 14.12 Timer RC Operation Clock**) has elapsed, the TRCIOj output pin becomes a programmable I/O port.

Make the following settings to use this function.

- Set the pin state following forced cutoff of pulse output (high impedance (input), "L" output, or "H" output). (Refer to **7. Programmable I/O Ports**.)
- Set the INT0EN bit to 1 ($\overline{\text{INT0}}$ input enabled) and the INT0PL bit to 0 (one edge) in the INTEN register.
- Set the PD4_5 bit in the PD4 register to 0 (input mode).
- Select the $\overline{\text{INT0}}$ digital filter by means of bits INT0F1 to INT0F0 in the INTF register.
- Set the PTO bit in the TRCOER register to 1 (pulse output forced cutoff signal input $\overline{\text{INT0}}$ enabled).

The IR bit in the INT0IC register is set to 1 (interrupt request) in accordance with the setting of the POL bit and a change in the $\overline{\text{INT0}}$ pin input (refer to **12.6 Notes on Interrupts**).

For details on interrupts, refer to **12. Interrupts**.

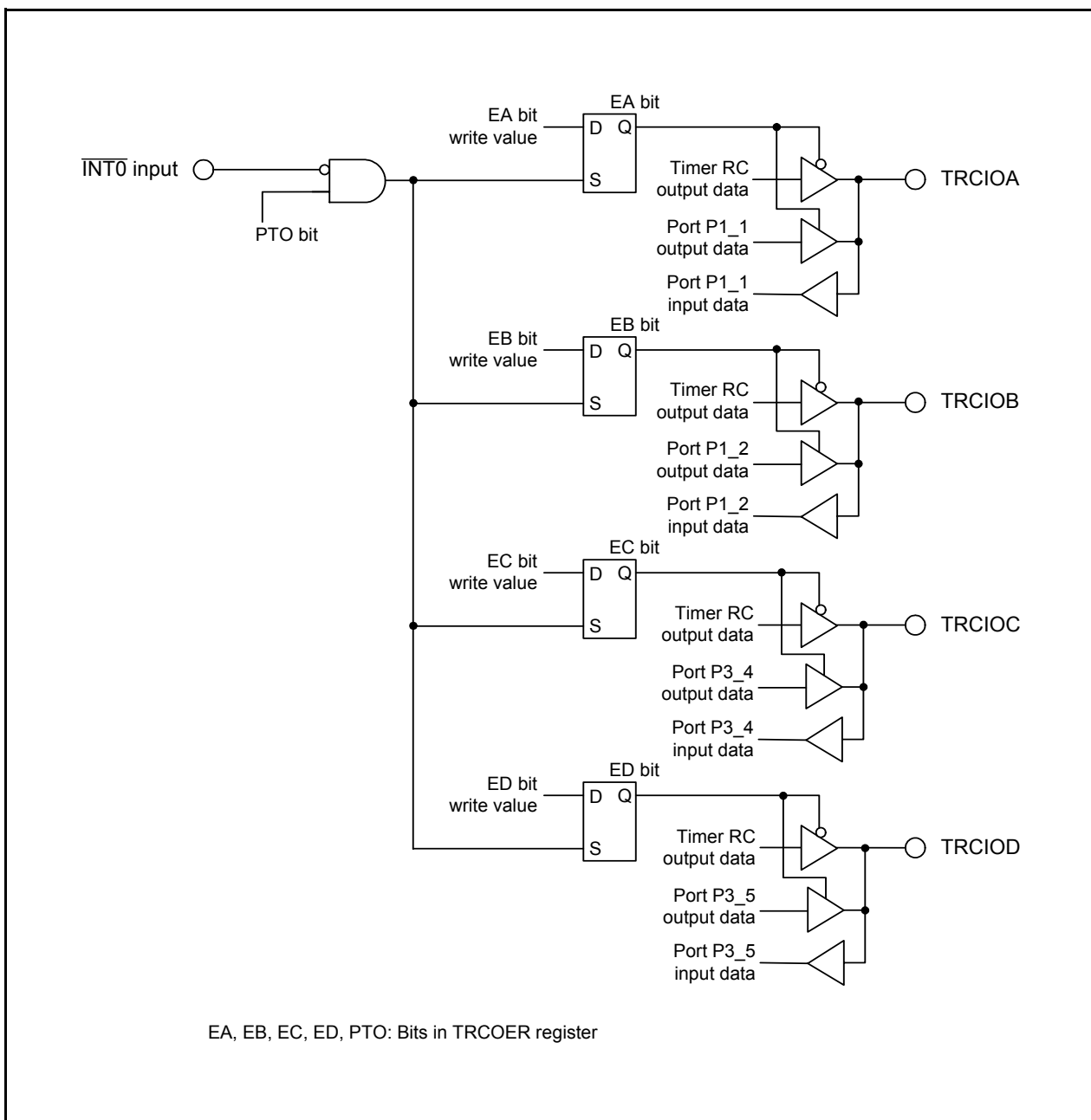


Figure 14.42 Forced Cutoff of Pulse Output

14.3.4 Timer Mode (Input Capture Function)

This function measures the width or period of an external signal. An external signal input to the TRCIO_j (j = A, B, C, or D) pin acts as a trigger for transferring the contents of the TRC register (counter) to the TRCGR_j register (input capture). The input capture function, or any other mode or function, can be selected for each individual pin.

The TRCGRA register can also select fOCO128 signal as input-capture trigger input.

Table 14.17 lists the Specifications of Input Capture Function, Figure 14.43 shows a Block Diagram of Input Capture Function, Figures 14.44 and 14.45 show registers associated with the input capture function, Table 14.18 lists the Functions of TRCGR_j Register when Using Input Capture Function, and Figure 14.46 shows an Operating Example of Input Capture Function.

Table 14.17 Specifications of Input Capture Function

| Item | Specification |
|--|--|
| Count source | f1, f2, f4, f8, f32, fOCO40M, or external signal (rising edge) input to TRCCLK pin |
| Count operation | Increment |
| Count period | $1/f_k \times 65,536$ f _k : Count source frequency |
| Count start condition | 1 (count starts) is written to the TSTART bit in the TRCMR register. |
| Count stop condition | 0 (count stops) is written to the TSTART bit in the TRCMR register. The TRC register retains a value before count stops. |
| Interrupt request generation timing | <ul style="list-style-type: none"> Input capture (valid edge of TRCIO_j input or fOCO128 signal edge) The TRC register overflows. |
| TRCIOA, TRCIOB, TRCIOC, and TRCIOD pin functions | Programmable I/O port or input capture input (selectable individually by pin) |
| INT0 pin function | Programmable I/O port or INT0 interrupt input |
| Read from timer | The count value can be read by reading TRC register. |
| Write to timer | The TRC register can be written to. |
| Select functions | <ul style="list-style-type: none"> Input capture input pin select One or more of pins TRCIOA, TRCIOB, TRCIOC, and TRCIOD Input capture input valid edge selected Rising edge, falling edge, or both rising and falling edges Buffer operation (Refer to 14.3.3.2 Buffer Operation.) Digital filter (Refer to 14.3.3.3 Digital Filter.) Input-capture trigger selected fOCO128 can be selected for input-capture trigger input of the TRCGRA register. |

j = A, B, C, or D

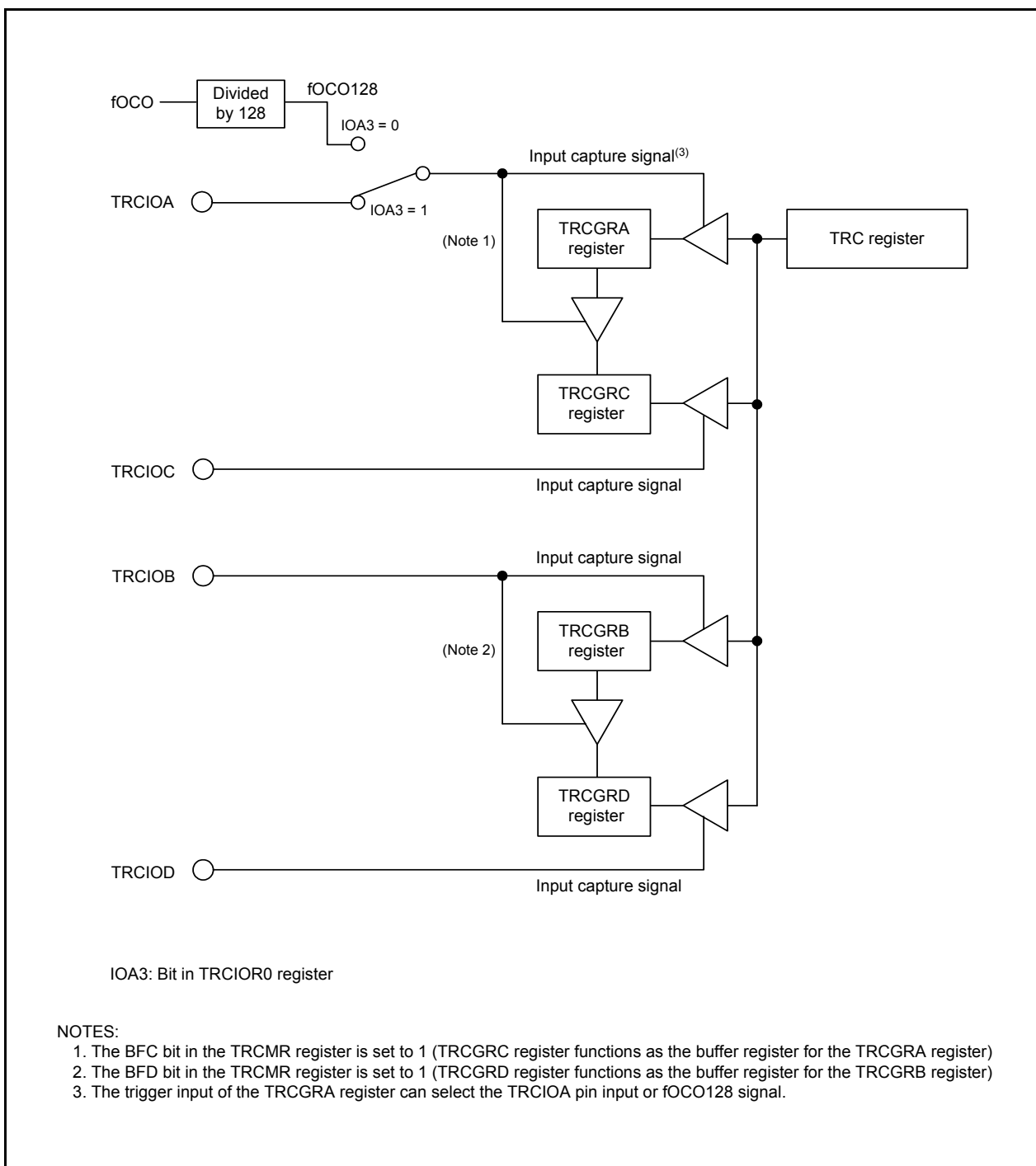


Figure 14.43 Block Diagram of Input Capture Function

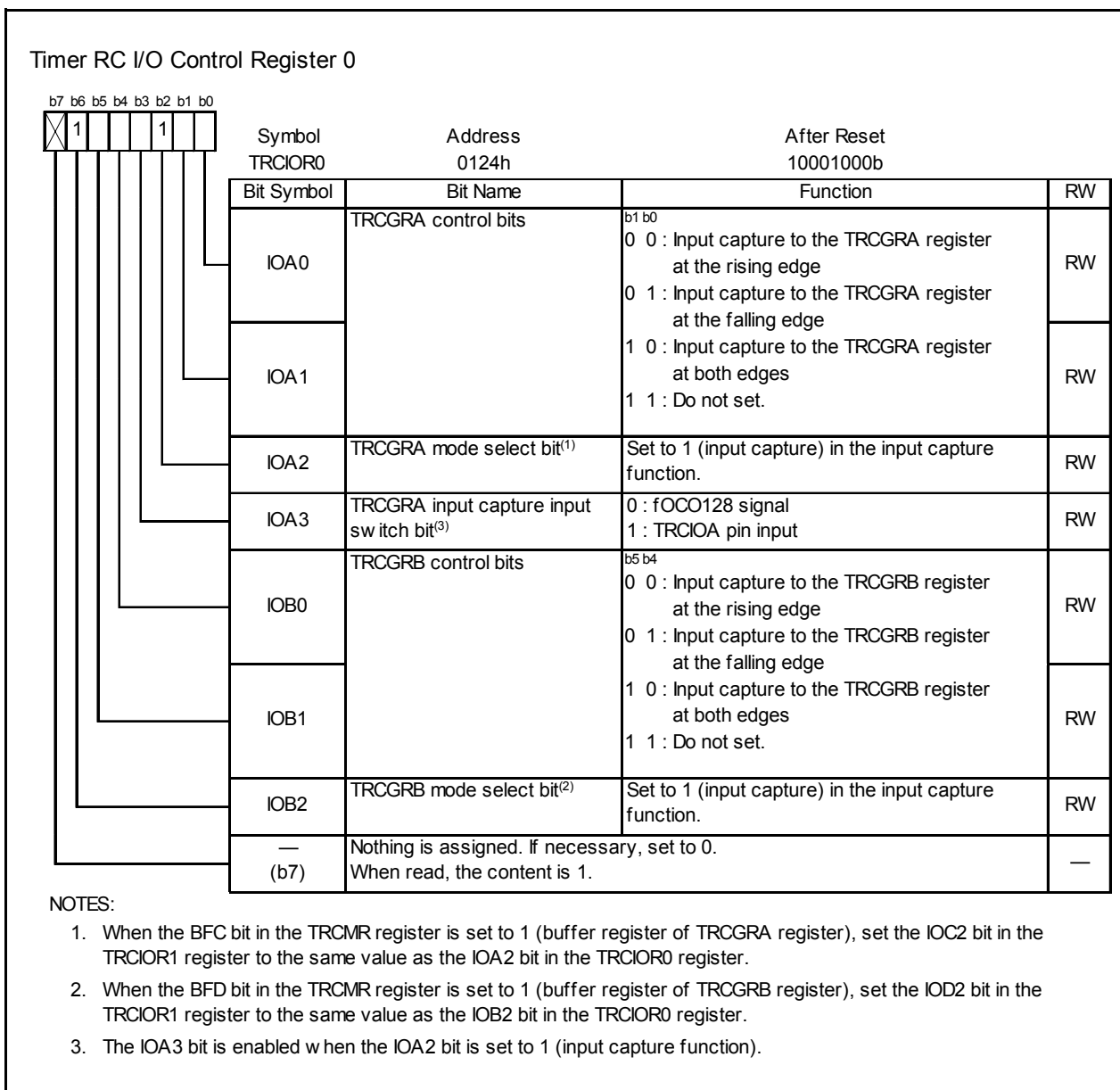


Figure 14.44 TRCIOR0 Register for Input Capture Function

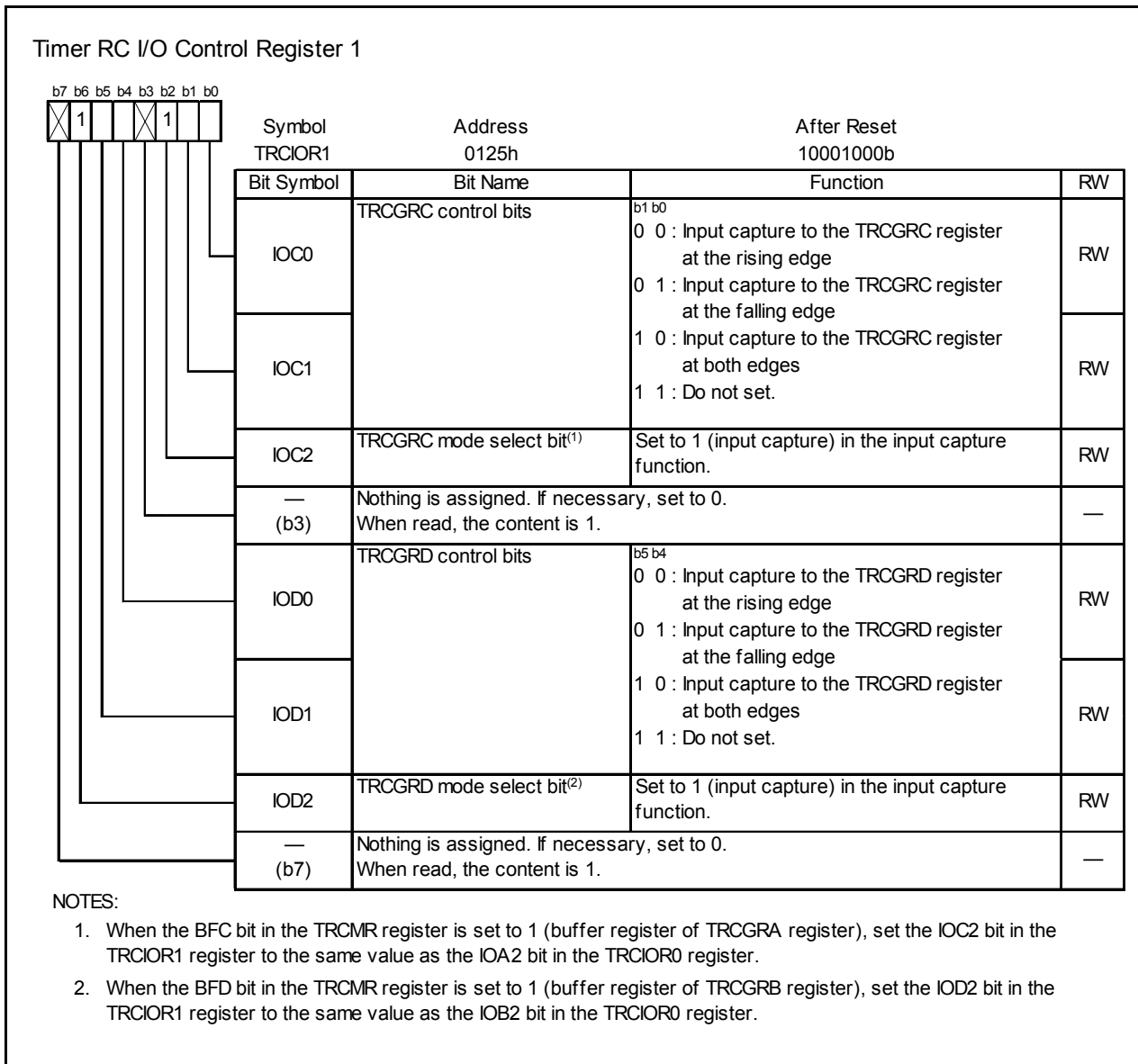


Figure 14.45 TRCIOR1 Register for Input Capture Function

Table 14.18 Functions of TRCGRj Register when Using Input Capture Function

| Register | Setting | Register Function | Input Capture Input Pin |
|----------|---------|--|-------------------------|
| TRCGRA | — | General register. Can be used to read the TRC register value at input capture. | TRCIOA |
| TRCGRB | | | TRCIOB |
| TRCGRC | BFC = 0 | General register. Can be used to read the TRC register value at input capture. | TRCIOC |
| TRCGRD | BFD = 0 | | TRCIOD |
| TRCGRC | BFC = 1 | Buffer registers. Can be used to hold transferred value from the general register. (Refer to 14.3.3.2 Buffer Operation.) | TRCIOA |
| TRCGRD | BFD = 1 | | TRCIOB |

j = A, B, C, or D

BFC, BFD: Bits in TRCMR register

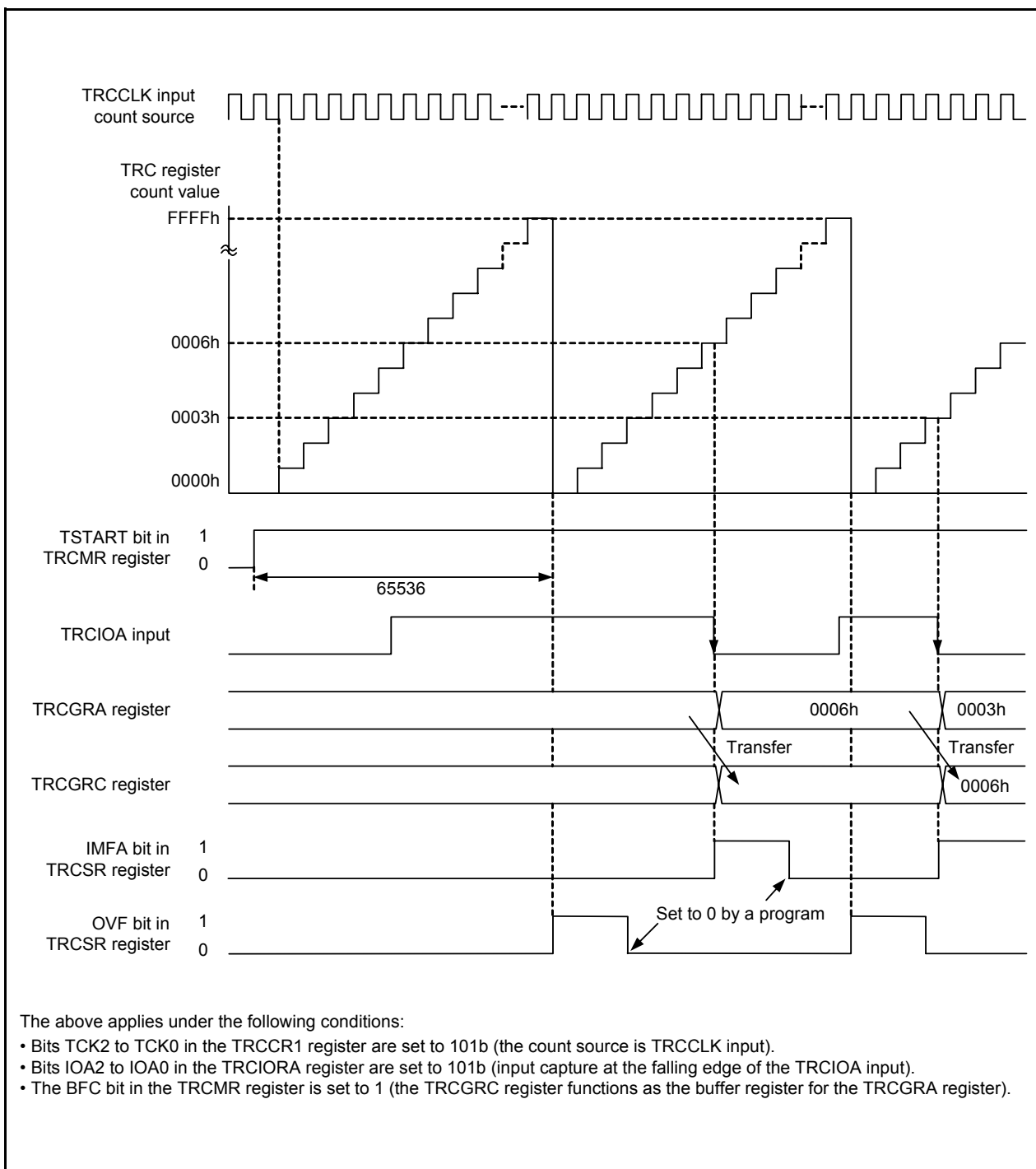


Figure 14.46 Operating Example of Input Capture Function

14.3.5 Timer Mode (Output Compare Function)

This function detects when the contents of the TRC register (counter) and the TRCGR_j register (j = A, B, C, or D) match (compare match). When a match occurs a signal is output from the TRCIO_j pin at a given level. The output compare function, or other mode or function, can be selected for each individual pin.

Table 14.19 lists the Specifications of Output Compare Function, Figure 14.47 shows a Block Diagram of Output Compare Function, Figures 14.48 to 14.50 show registers associated with the output compare function, Table 14.20 lists the Functions of TRCGR_j Register when Using Output Compare Function, and Figure 14.51 shows an Operating Example of Output Compare Function.

Table 14.19 Specifications of Output Compare Function

| Item | Specification |
|--|--|
| Count source | f1, f2, f4, f8, f32, fOCO40M, or external signal (rising edge) input to TRCCLK pin |
| Count operation | Increment |
| Count period | <ul style="list-style-type: none"> The CCLR bit in the TRCCR1 register is set to 0 (free running operation): $1/fk \times 65,536$ fk: Count source frequency The CCLR bit in the TRCCR1 register is set to 1 (TRC register set to 0000h at TRCGRA compare match): $1/fk \times (n + 1)$ n: TRCGRA register setting value |
| Waveform output timing | Compare match |
| Count start condition | 1 (count starts) is written to the TSTART bit in the TRCMR register. |
| Count stop condition | 0 (count stops) is written to the TSTART bit in the TRCMR register. The output compare output pin retains output level before count stops, the TRC register retains a value before count stops. |
| Interrupt request generation timing | <ul style="list-style-type: none"> Compare match (contents of registers TRC and TRCGR_j match) The TRC register overflows. |
| TRCIOA, TRCIOB, TRCIOC, and TRCIOD pin functions | Programmable I/O port or output compare output (selectable individually by pin) |
| INT0 pin function | Programmable I/O port, pulse output forced cutoff signal input, or INT0 interrupt input |
| Read from timer | The count value can be read by reading the TRC register. |
| Write to timer | The TRC register can be written to. |
| Select functions | <ul style="list-style-type: none"> Output compare output pin selected One or more of pins TRCIOA, TRCIOB, TRCIOC, and TRCIOD Compare match output level select “L” output, “H” output, or toggle output Initial output level select Sets output level for period from count start to compare match Timing for clearing the TRC register to 0000h Overflow or compare match with the TRCGRA register Buffer operation (Refer to 14.3.3.2 Buffer Operation.) Pulse output forced cutoff signal input (Refer to 14.3.3.4 Forced Cutoff of Pulse Output.) Can be used as an internal timer by disabling timer RC output |

j = A, B, C, or D

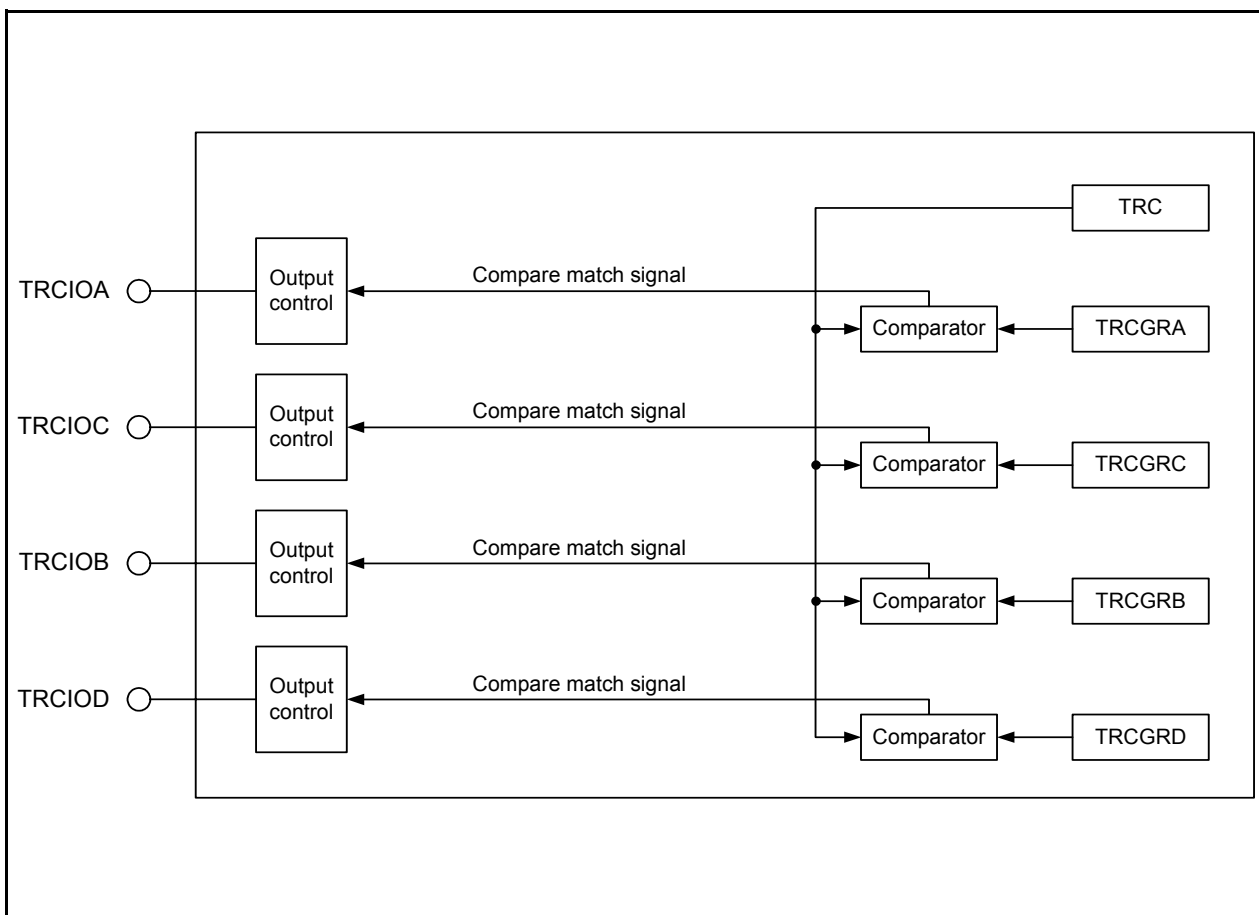


Figure 14.47 Block Diagram of Output Compare Function

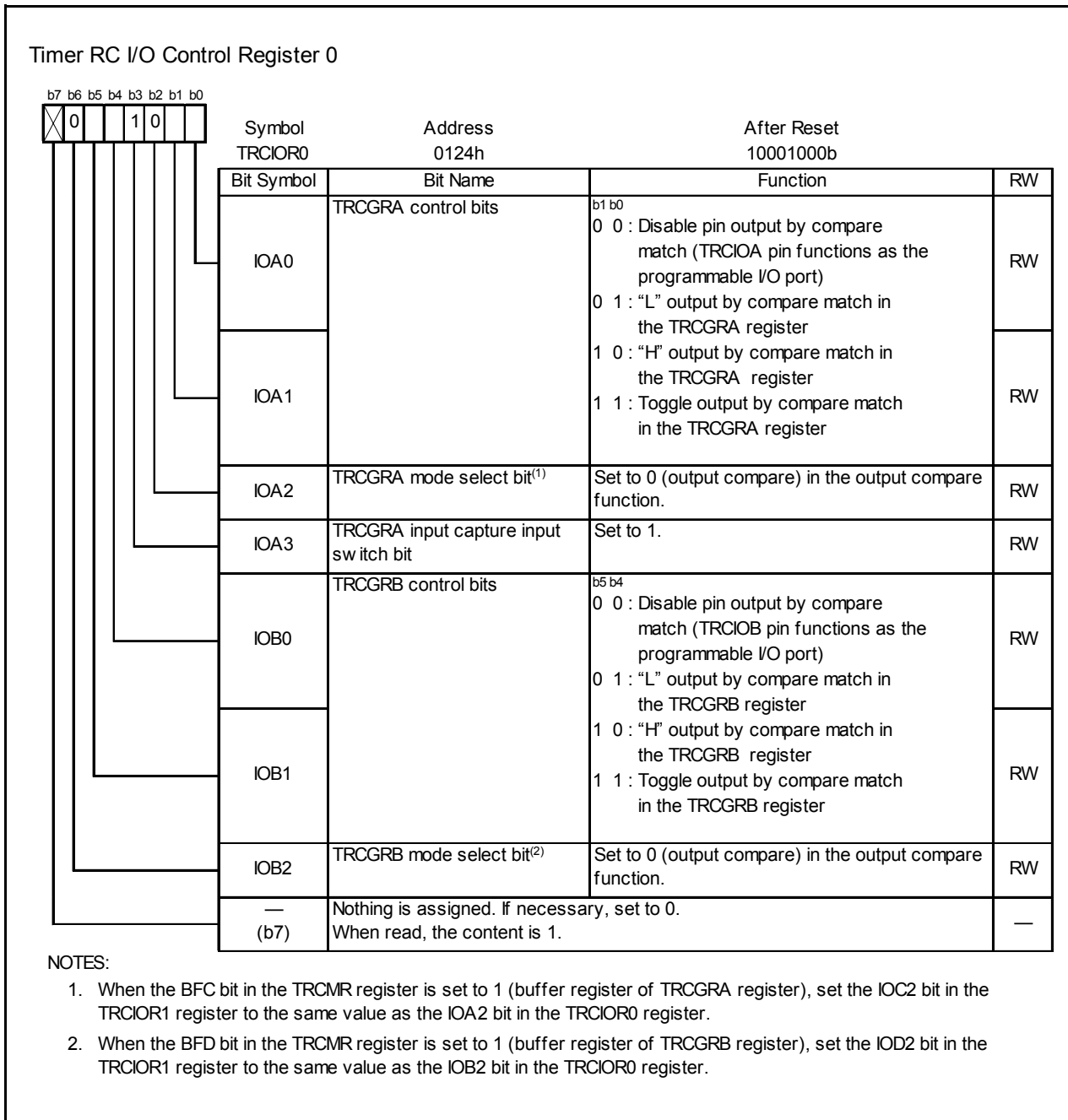


Figure 14.48 TRCIOR0 Register for Output Compare Function

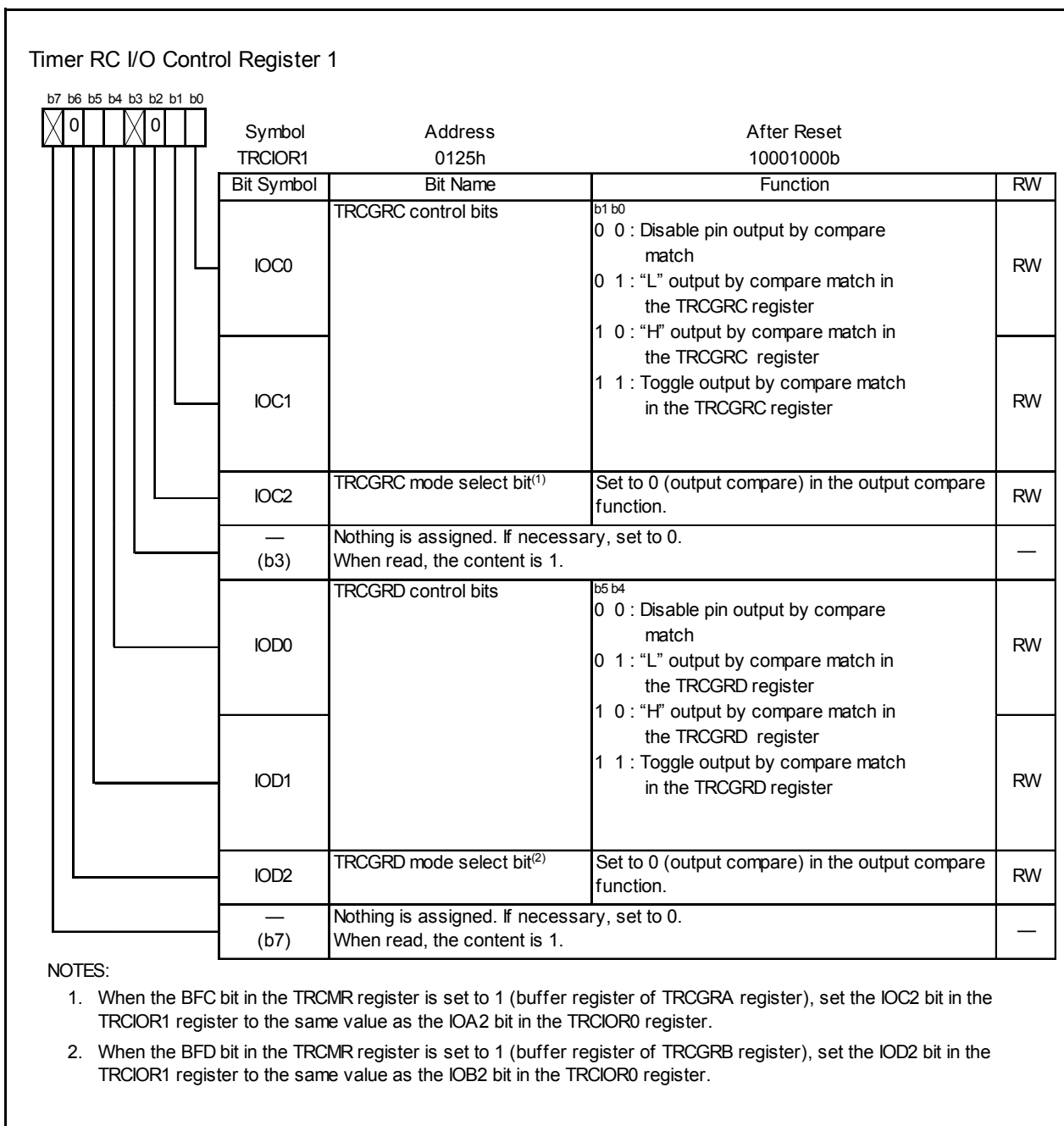


Figure 14.49 TRCIOR1 Register for Output Compare Function

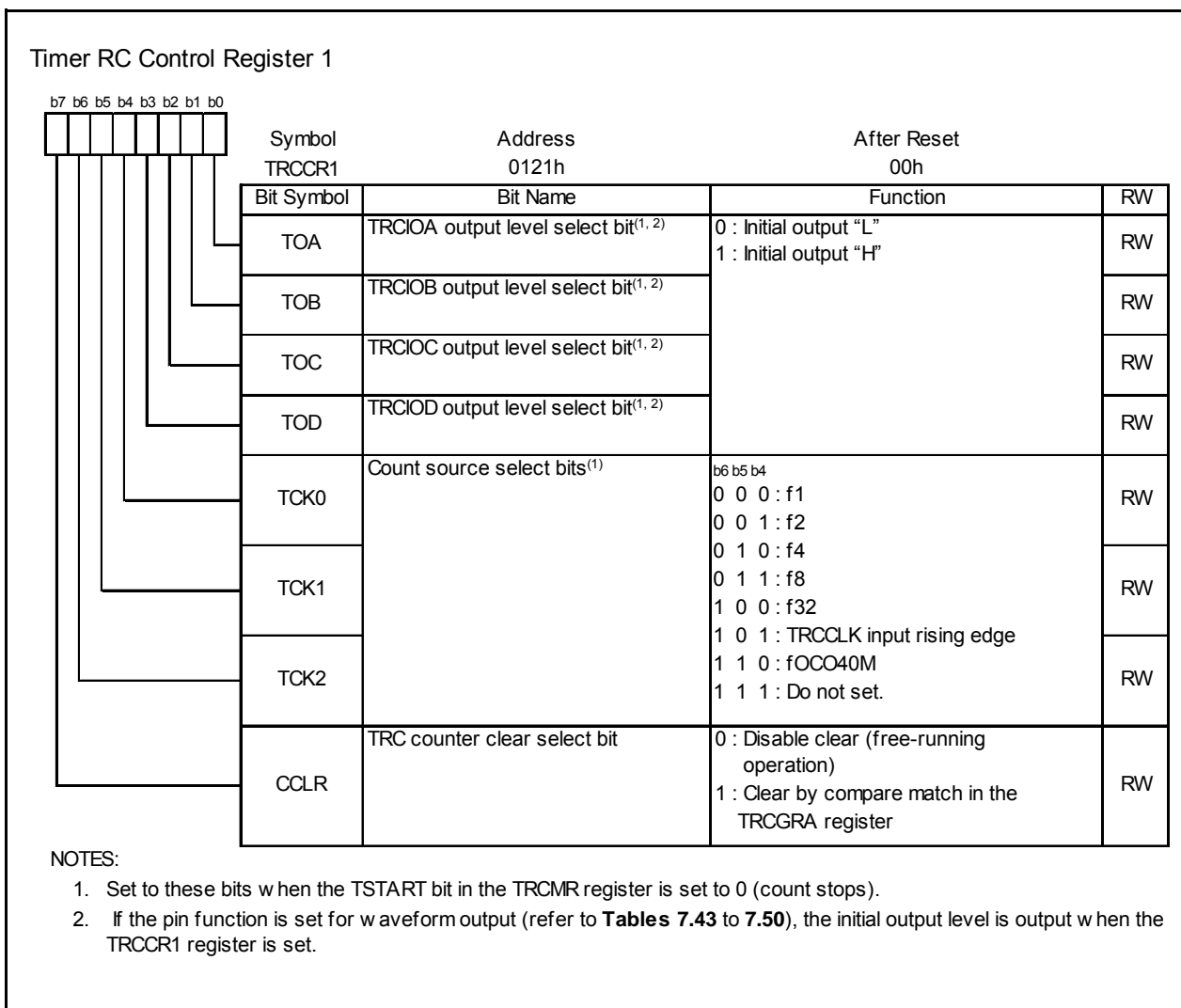


Figure 14.50 TRCCR1 Register for Output Compare Function

Table 14.20 Functions of TRCGRj Register when Using Output Compare Function

| Register | Setting | Register Function | Output Compare Output Pin |
|----------|---------|--|---------------------------|
| TRCGRA | - | General register. Write a compare value to one of these registers. | TRCIOA |
| TRCGRB | | | TRCIOB |
| TRCGRC | BFC = 0 | General register. Write a compare value to one of these registers. | TRCIOC |
| TRCGRD | BFD = 0 | | TRCIOD |
| TRCGRC | BFC = 1 | Buffer register. Write the next compare value to one of these registers. (Refer to 14.3.3.2 Buffer Operation .) | TRCIOA |
| TRCGRD | BFD = 1 | | TRCIOB |

j = A, B, C, or D

BFC, BFD: Bits in TRCMR register

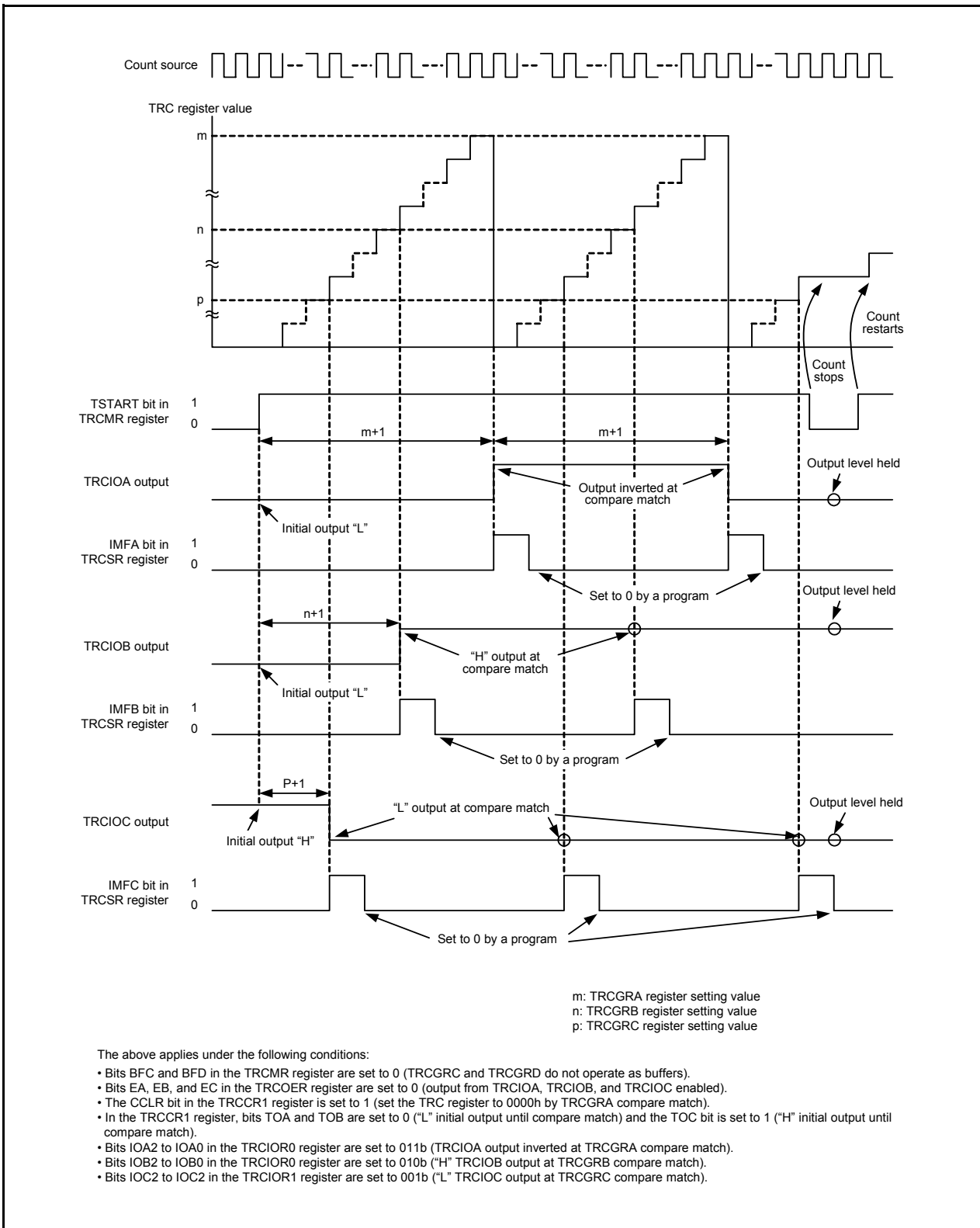


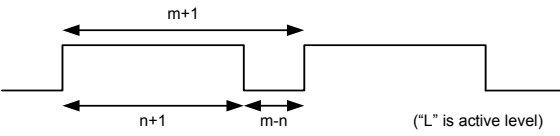
Figure 14.51 Operating Example of Output Compare Function

14.3.6 PWM Mode

This mode outputs PWM waveforms. A maximum of three PWM waveforms with the same period are output. The PWM mode, or the timer mode, can be selected for each individual pin. (However, since the TRCGRA register is used when using any pin for the PWM mode, the TRCGRA register cannot be used for the timer mode.)

Table 14.21 lists the Specifications of PWM Mode, Figure 14.52 shows a Block Diagram of PWM Mode, Figure 14.53 shows the register associated with the PWM mode, Table 14.22 lists the Functions of TRCGRj Register in PWM Mode, and Figures 14.54 and 14.55 show Operating Examples of PWM Mode.

Table 14.21 Specifications of PWM Mode

| Item | Specification |
|--|---|
| Count source | f1, f2, f4, f8, f32, fOCO40M, or external signal (rising edge) input to TRCCLK pin |
| Count operation | Increment |
| PWM waveform | PWM period: $1/f_k \times (m + 1)$ Active level width: $1/f_k \times (m - n)$ Inactive width: $1/f_k \times (n + 1)$ f _k : Count source frequency m: TRCGRA register setting value n: TRCGR _j register setting value  |
| Count start condition | 1 (count starts) is written to the TSTART bit in the TRCMR register. |
| Count stop condition | 0 (count stops) is written to the TSTART bit in the TRCMR register. PWM output pin retains output level before count stops, TRC register retains value before count stops. |
| Interrupt request generation timing | <ul style="list-style-type: none"> • Compare match (contents of registers TRC and TRCGR_j match) • The TRC register overflows. |
| TRCIOA pin function | Programmable I/O port |
| TRCIOB, TRCIOC, and TRCIOD pin functions | Programmable I/O port or PWM output (selectable individually by pin) |
| INT0 pin function | Programmable I/O port, pulse output forced cutoff signal input, or INT0 interrupt input |
| Read from timer | The count value can be read by reading the TRC register. |
| Write to timer | The TRC register can be written to. |
| Select functions | <ul style="list-style-type: none"> • One to three pins selectable as PWM output pins per channel One or more of pins TRCIOB, TRCIOC, and TRCIOD • Active level selectable by individual pin • Buffer operation (Refer to 14.3.3.2 Buffer Operation.) • Pulse output forced cutoff signal input (Refer to 14.3.3.4 Forced Cutoff of Pulse Output.) |

j = A, B, C, or D

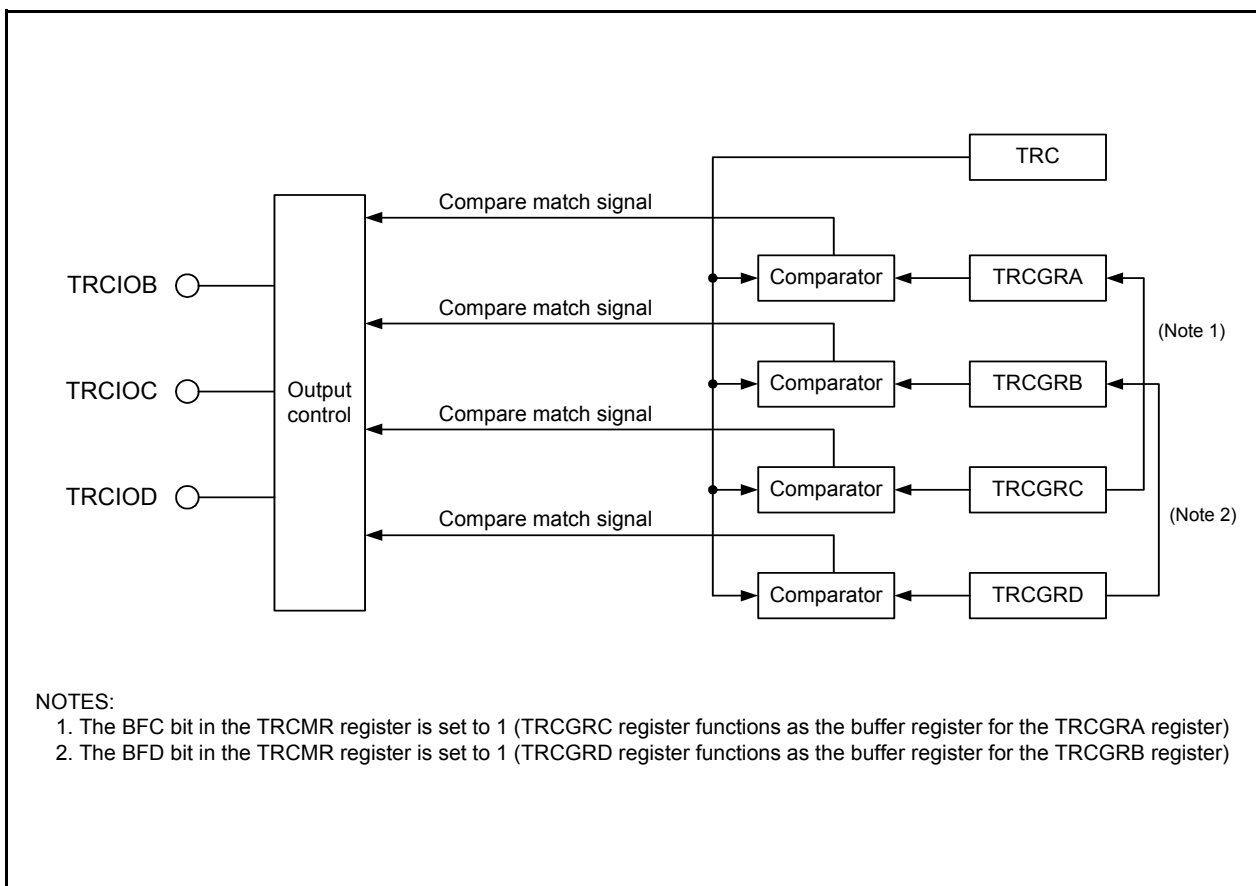


Figure 14.52 Block Diagram of PWM Mode

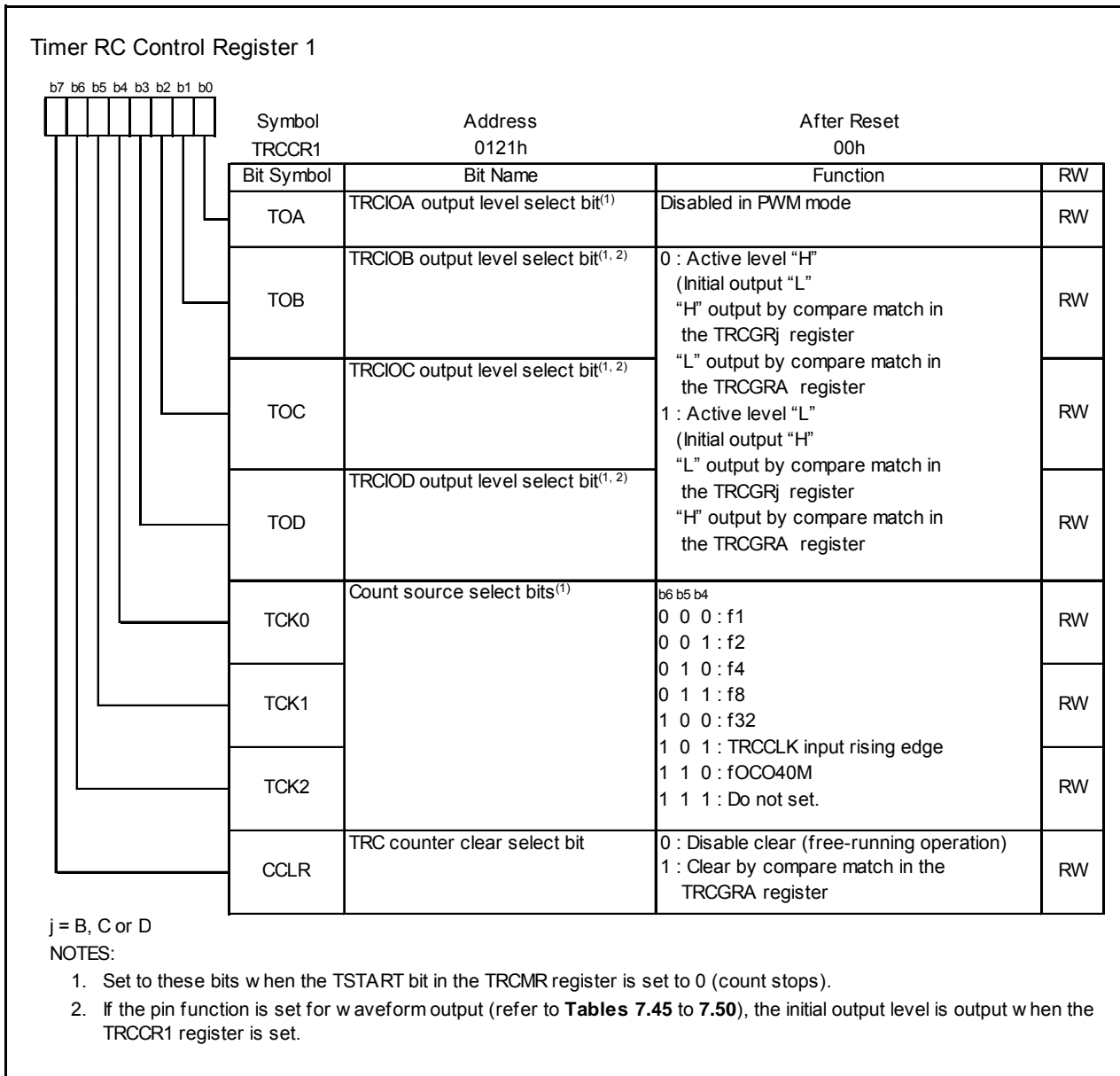


Figure 14.53 TRCCR1 Register in PWM Mode

Table 14.22 Functions of TRCGRj Register in PWM Mode

| Register | Setting | Register Function | PWM Output Pin |
|----------|---------|--|----------------|
| TRCGRA | – | General register. Set the PWM period. | – |
| TRCGRB | – | General register. Set the PWM output change point. | TRCIOB |
| TRCGRC | BFC = 0 | General register. Set the PWM output change point. | TRCIOC |
| TRCGRD | BFD = 0 | | TRCIOD |
| TRCGRC | BFC = 1 | Buffer register. Set the next PWM period. (Refer to 14.3.3.2 Buffer Operation.) | – |
| TRCGRD | BFD = 1 | Buffer register. Set the next PWM output change point. (Refer to 14.3.3.2 Buffer Operation.) | TRCIOB |

j = A, B, C, or D

BFC, BFD: Bits in TRCMR register

NOTE:

- The output level does not change even when a compare match occurs if the TRCGRA register value (PWM period) is the same as the TRCGRB, TRCGRC, or TRCGRD register value.

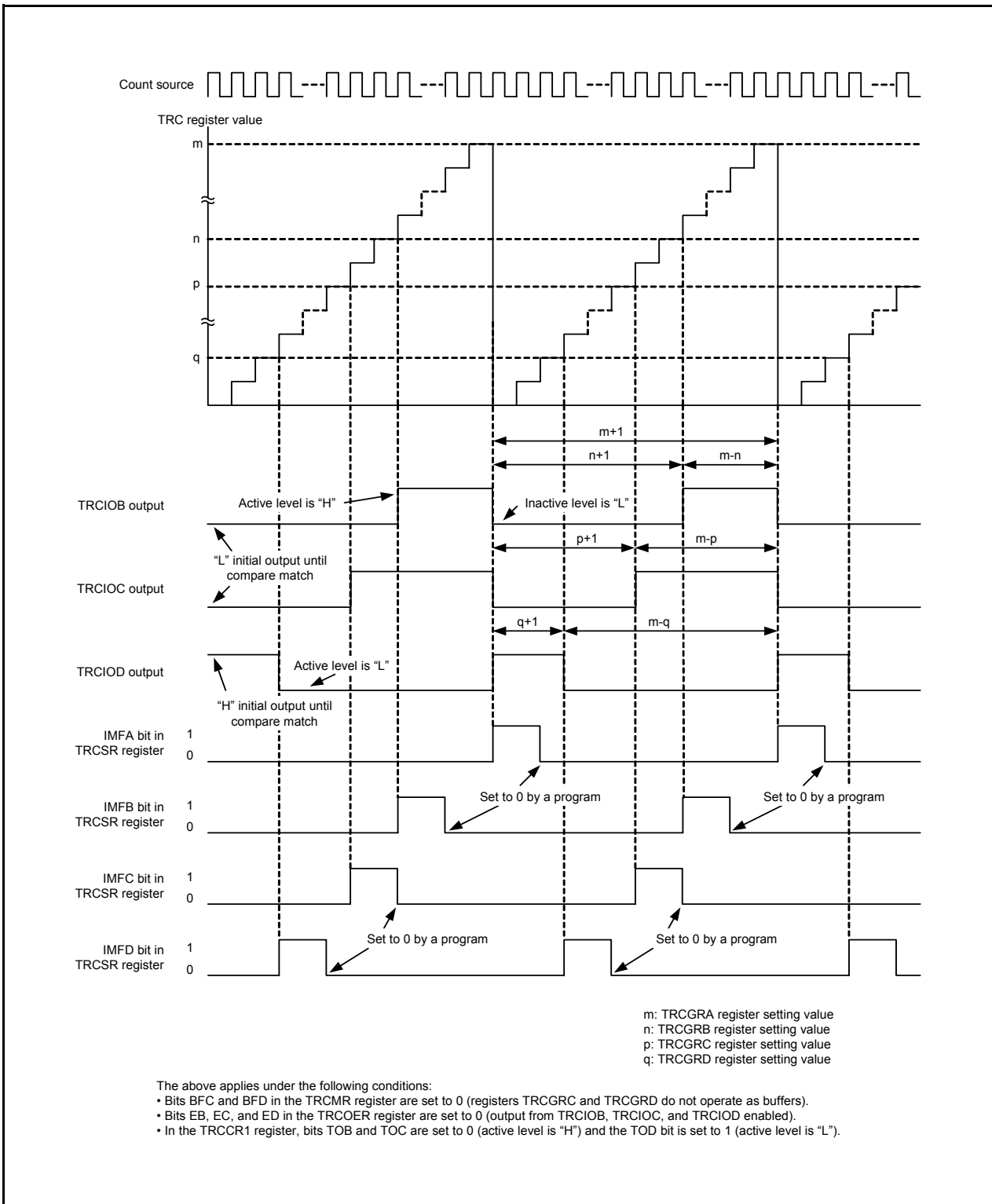


Figure 14.54 Operating Example of PWM Mode

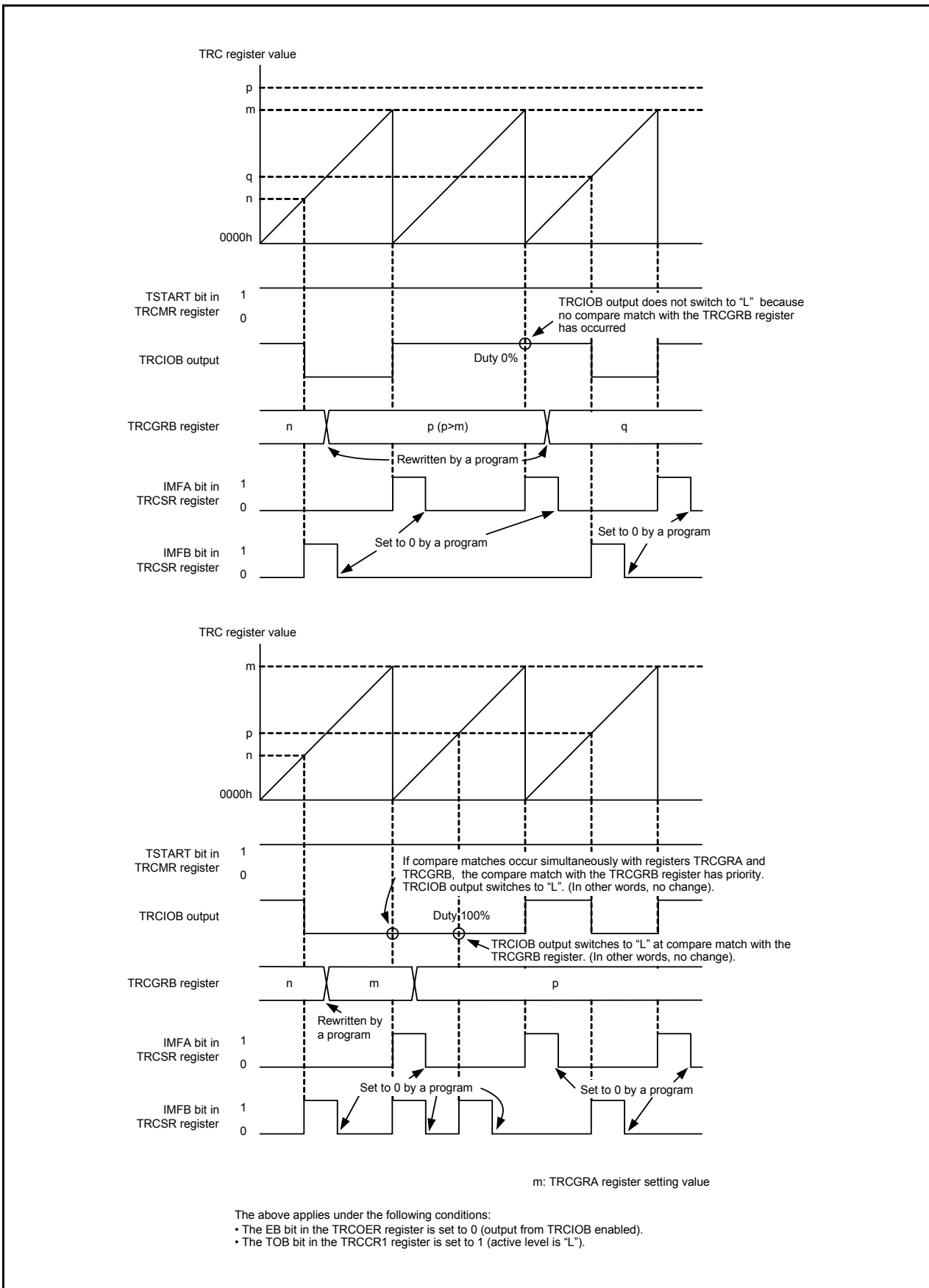


Figure 14.55 Operating Example of PWM Mode (Duty 0% and Duty 100%)

14.3.7 PWM2 Mode

This mode outputs a single PWM waveform. After a given wait duration has elapsed following the trigger, the pin output switches to active level. Then, after a given duration, the output switches back to inactive level. Furthermore, the counter stops at the same time the output returns to inactive level, making it possible to use PWM2 mode to output a programmable wait one-shot waveform.

Since timer RC uses multiple general registers in PWM2 mode, other modes cannot be used in conjunction with it.

Figure 14.56 shows a Block Diagram of PWM2 Mode, Table 14.23 lists the Specifications of PWM2 Mode, Figure 14.57 shows the register associated with PWM2 mode, Table 14.24 lists the Functions of TRCGRj Register in PWM2 Mode, and Figures 14.58 to 14.60 show Operating Examples of PWM2 Mode.

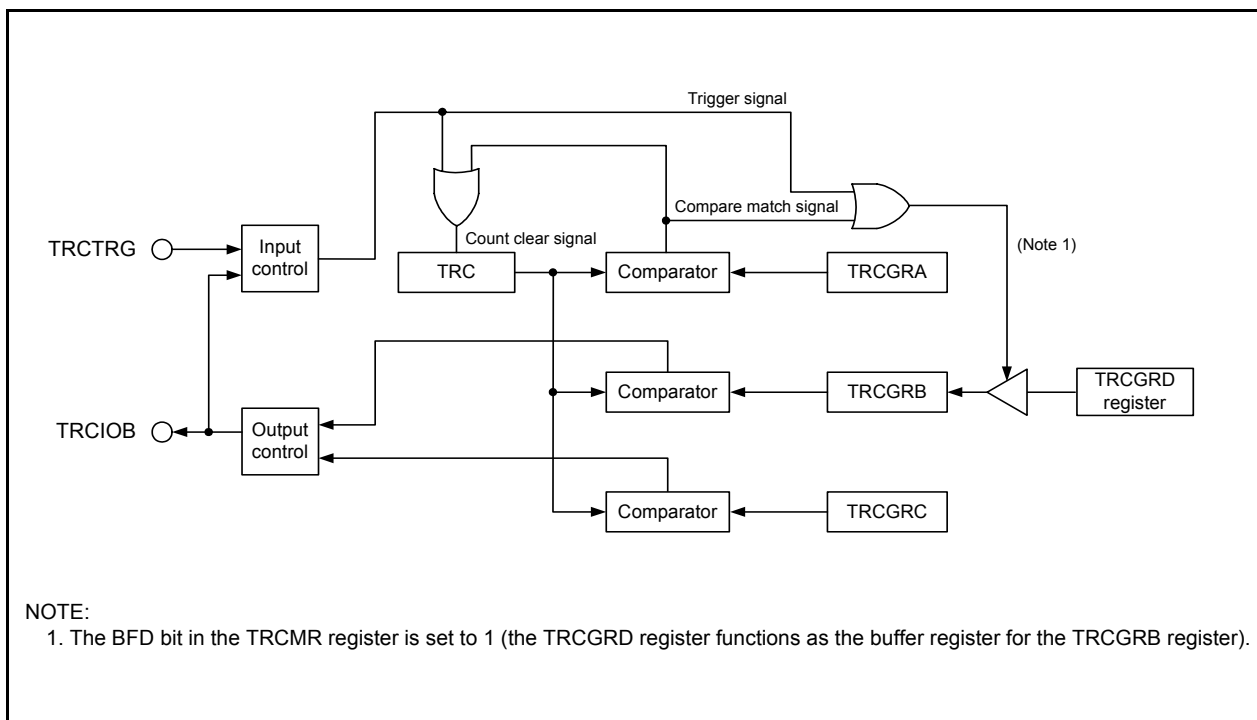


Figure 14.56 Block Diagram of PWM2 Mode

Table 14.23 Specifications of PWM2 Mode

| Item | Specification |
|-------------------------------------|--|
| Count source | f1, f2, f4, f8, f32, fOCO40M, or external signal (rising edge) input to TRCCLK pin |
| Count operation | Increase TRC register |
| PWM waveform | <p>PWM period: $1/f_k \times (m + 1)$ (no TRCTRIG input) Active level width: $1/f_k \times (n - p)$ Wait time from count start or trigger: $1/f_k \times (p + 1)$ f_k: Count source frequency m: TRCGRA register setting value n: TRCGRB register setting value p: TRCGRC register setting value</p> <p style="text-align: center;">(TRCTRIG: Rising edge, active level is "H")</p> |
| Count start conditions | <ul style="list-style-type: none"> • Bits TCEG1 to TCEG0 in the TRCCR2 register are set to 00b (TRCTRIG trigger disabled) or the CSEL bit in the TRCCR2 register is set to 0 (count continues). 1 (count starts) is written to the TSTART bit in the TRCMR register. • Bits TCEG1 to TCEG0 in the TRCCR2 register are set to 01b, 10b, or 11b (TRCTRIG trigger enabled) and the TSTART bit in the TRCMR register is set to 1 (count starts). A trigger is input to the TRCTRIG pin |
| Count stop conditions | <ul style="list-style-type: none"> • 0 (count stops) is written to the TSTART bit in the TRCMR register while the CSEL bit in the TRCCR2 register is set to 0 or 1. The TRCIOB pin outputs the initial level in accordance with the value of the TOB bit in the TRCCR1 register. The TRC register retains the value before count stops. • The count stops due to a compare match with TRCGRA while the CSEL bit in the TRCCR2 register is set to 1. The TRCIOB pin outputs the initial level. The TRC register retains the value before count stops if the CCLR bit in the TRCCR1 register is set to 0. The TRC register is set to 0000h if the CCLR bit in the TRCCR1 register is set to 1. |
| Interrupt request generation timing | <ul style="list-style-type: none"> • Compare match (contents of TRC and TRCGRj registers match) • The TRC register overflows |
| TRCIOA/TRCTRIG pin function | Programmable I/O port or TRCTRIG input |
| TRCIOB pin function | PWM output |
| TRCIOC and TRCIOD pin functions | Programmable I/O port |
| INT0 pin function | Programmable I/O port, pulse output forced cutoff signal input, or $\overline{\text{INT0}}$ interrupt input |
| Read from timer | The count value can be read by reading the TRC register. |
| Write to timer | The TRC register can be written to. |
| Select functions | <ul style="list-style-type: none"> • External trigger and valid edge selected The edge or edges of the signal input to the TRCTRIG pin can be used as the PWM output trigger: rising edge, falling edge, or both rising and falling edges • Buffer operation (Refer to 14.3.3.2 Buffer Operation.) • Pulse output forced cutoff signal input (Refer to 14.3.3.4 Forced Cutoff of Pulse Output.) • Digital filter (Refer to 14.3.3.3 Digital Filter.) |

j = A, B, C, or D

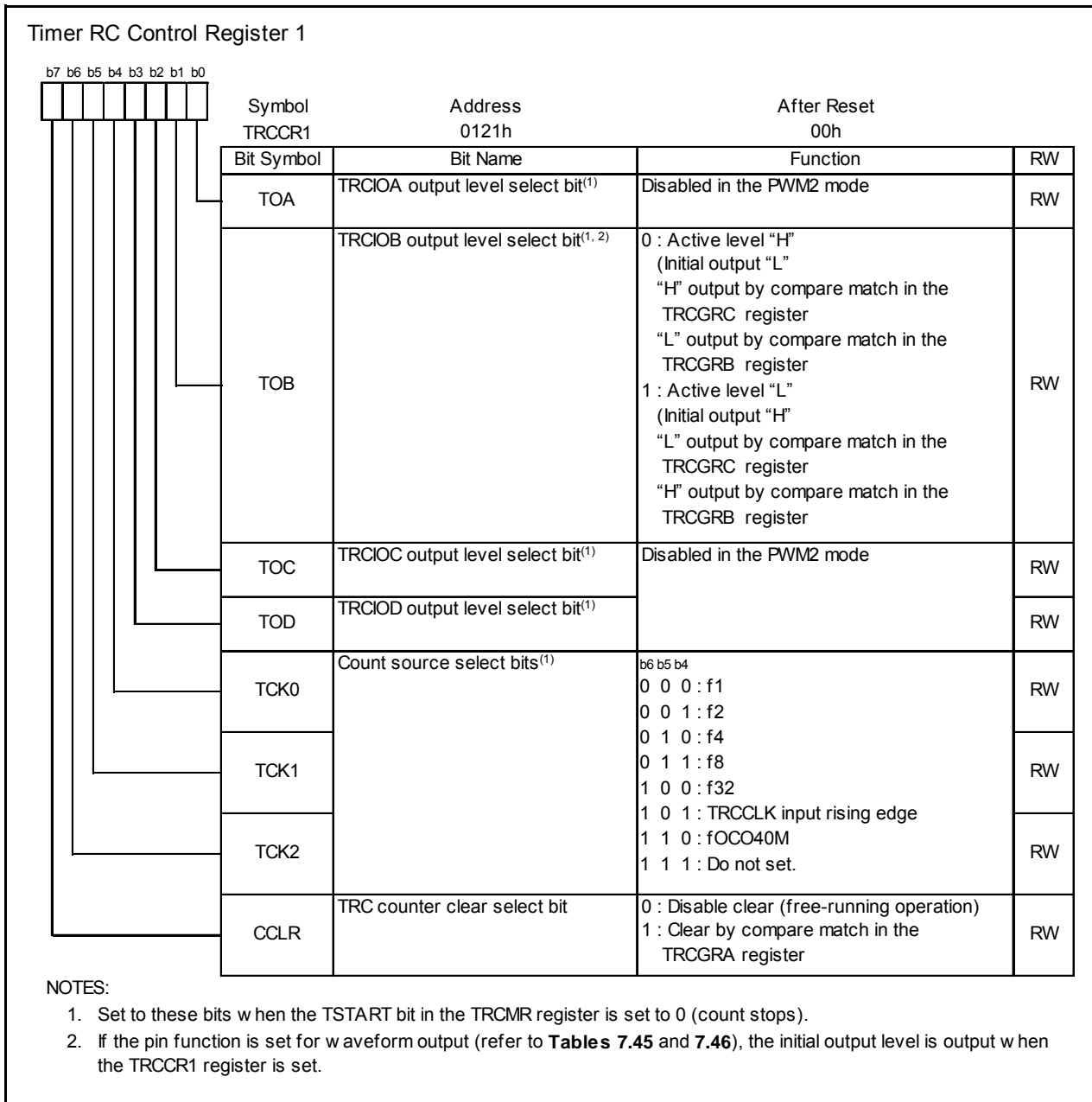


Figure 14.57 TRCCR1 Register in PWM2 Mode

Table 14.24 Functions of TRCGRj Register in PWM2 Mode

| Register | Setting | Register Function | PWM2 Output Pin |
|----------|---------|--|-----------------|
| TRCGRA | – | General register. Set the PWM period. | TRCIOB pin |
| TRCGRB | – | General register. Set the PWM output change point. | |
| TRCGRC | BFC = 0 | General register. Set the PWM output change point (wait time after trigger). | |
| TRCGRD | BFD = 0 | (Not used in PWM2 mode) | – |
| TRCGRD | BFD = 1 | Buffer register. Set the next PWM output change point. (Refer to 14.3.3.2 Buffer Operation .) | TRCIOB pin |

j = A, B, C, or D

BFC, BFD: Bits in TRCMR register

NOTE:

- Do not set the TRCGRB and TRCGRC registers to the same value.

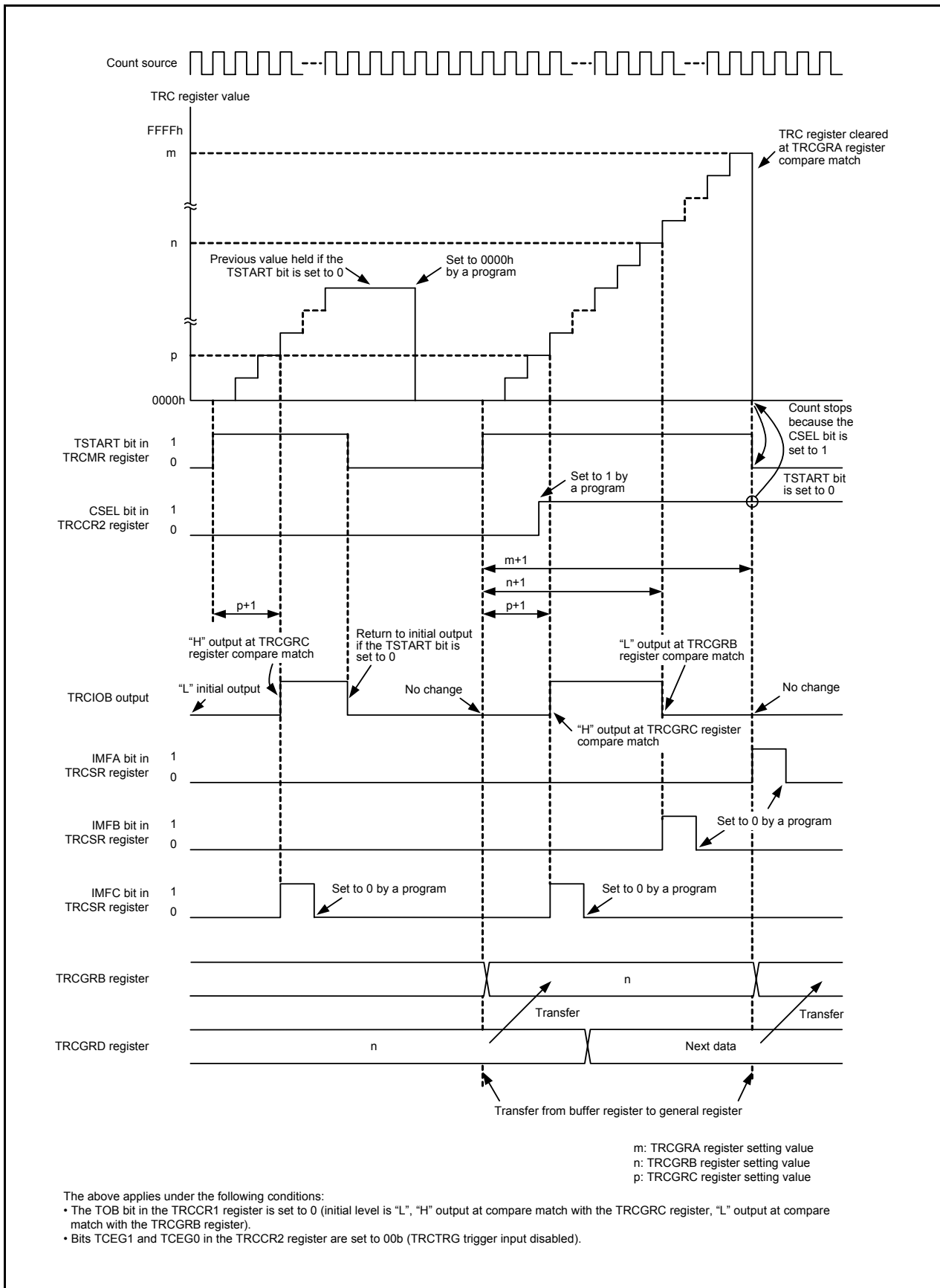


Figure 14.58 Operating Example of PWM2 Mode (TRCTRIG Trigger Input Disabled)

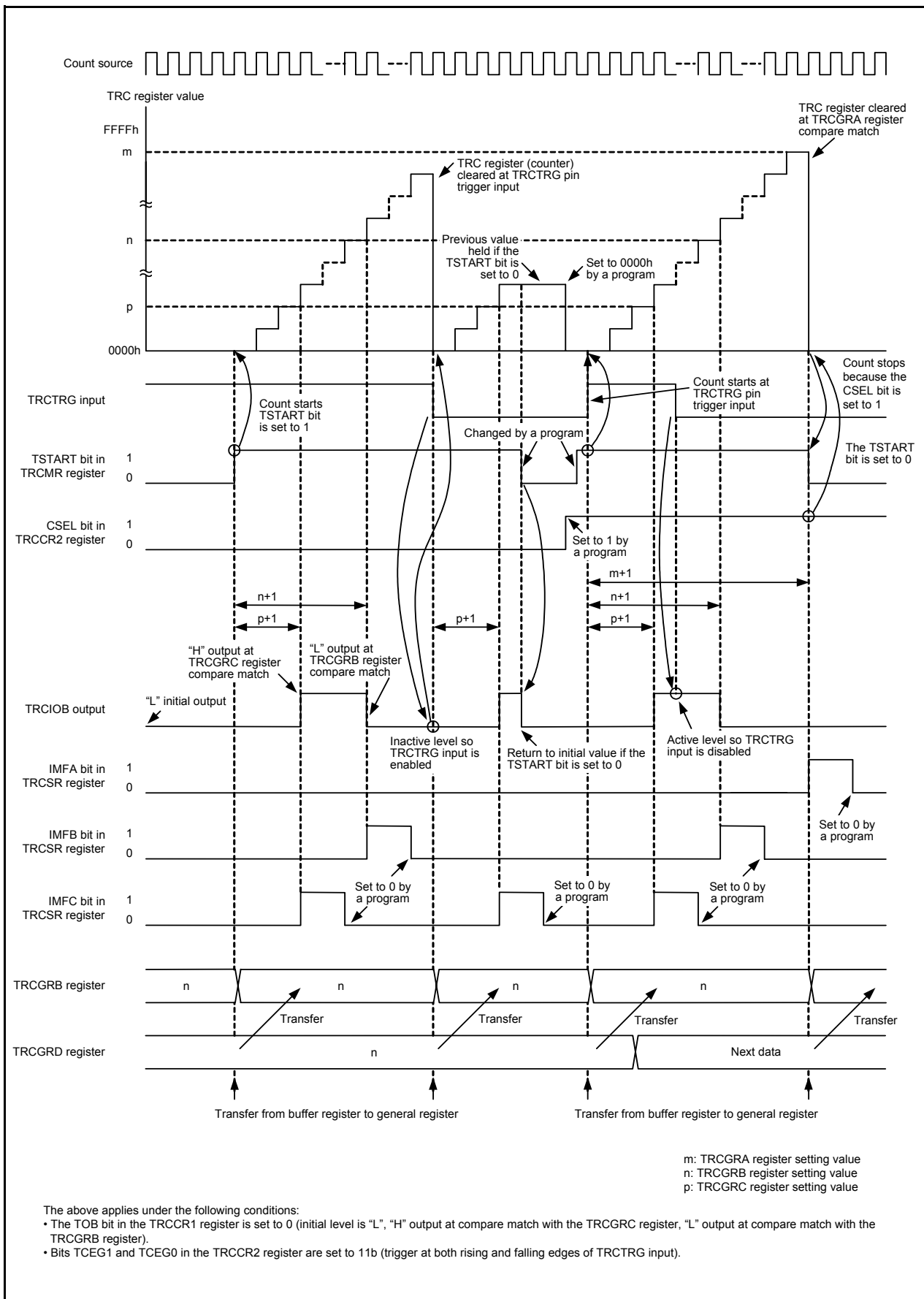


Figure 14.59 Operating Example of PWM2 Mode (TRCTRГ Trigger Input Enabled)

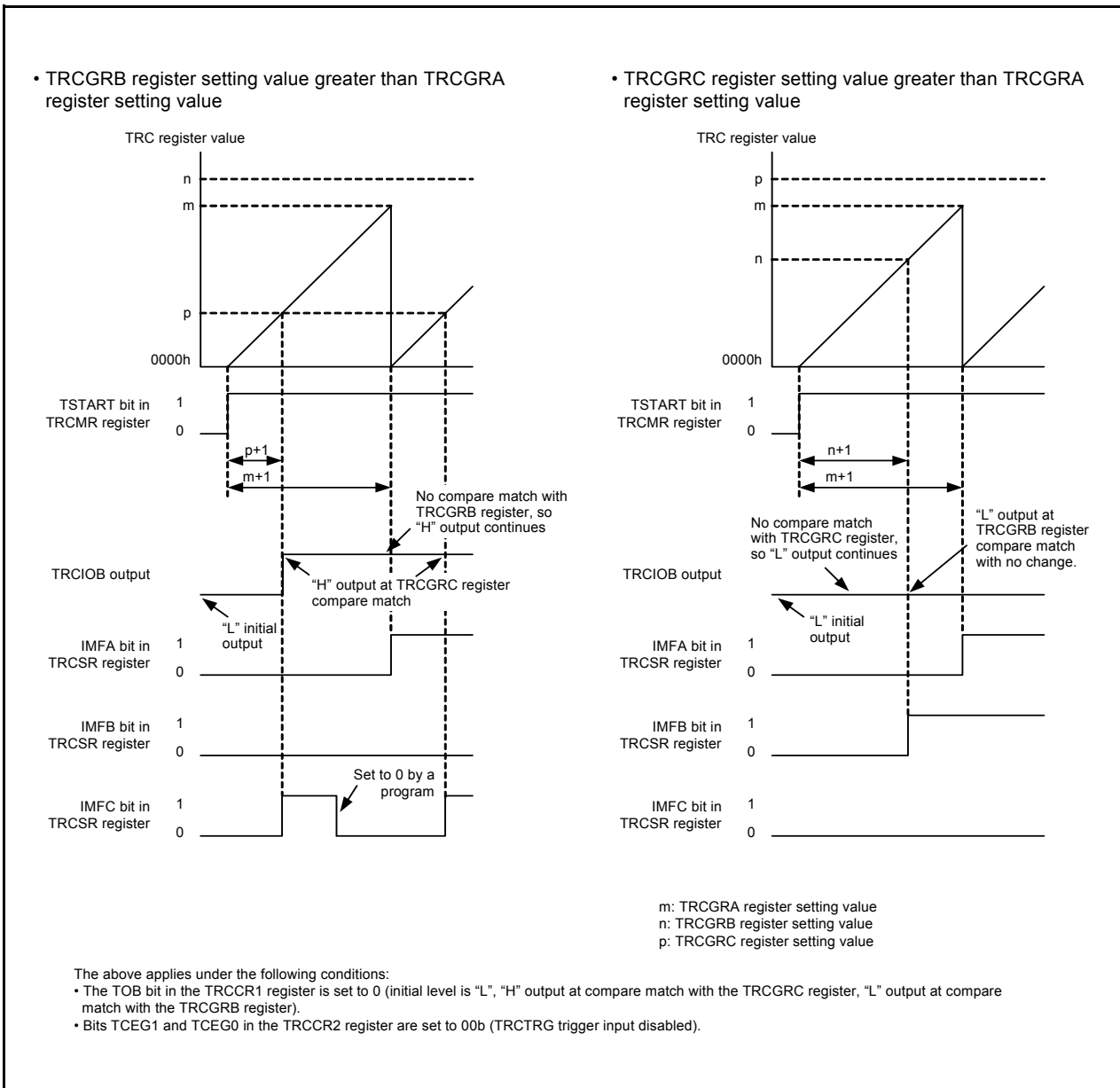


Figure 14.60 Operating Example of PWM2 Mode (Duty 0% and Duty 100%)

14.3.8 Timer RC Interrupt

Timer RC generates a timer RC interrupt request from five sources. The timer RC interrupt uses the single TRCIC register (bits IR and ILVL0 to ILVL2) and a single vector.

Table 14.25 lists the Registers Associated with Timer RC Interrupt, and Figure 14.61 is a Timer RC Interrupt Block Diagram.

Table 14.25 Registers Associated with Timer RC Interrupt

| Timer RC Status Register | Timer RC Interrupt Enable Register | Timer RC Interrupt Control Register |
|--------------------------|------------------------------------|-------------------------------------|
| TRCSR | TRCIER | TRCIC |

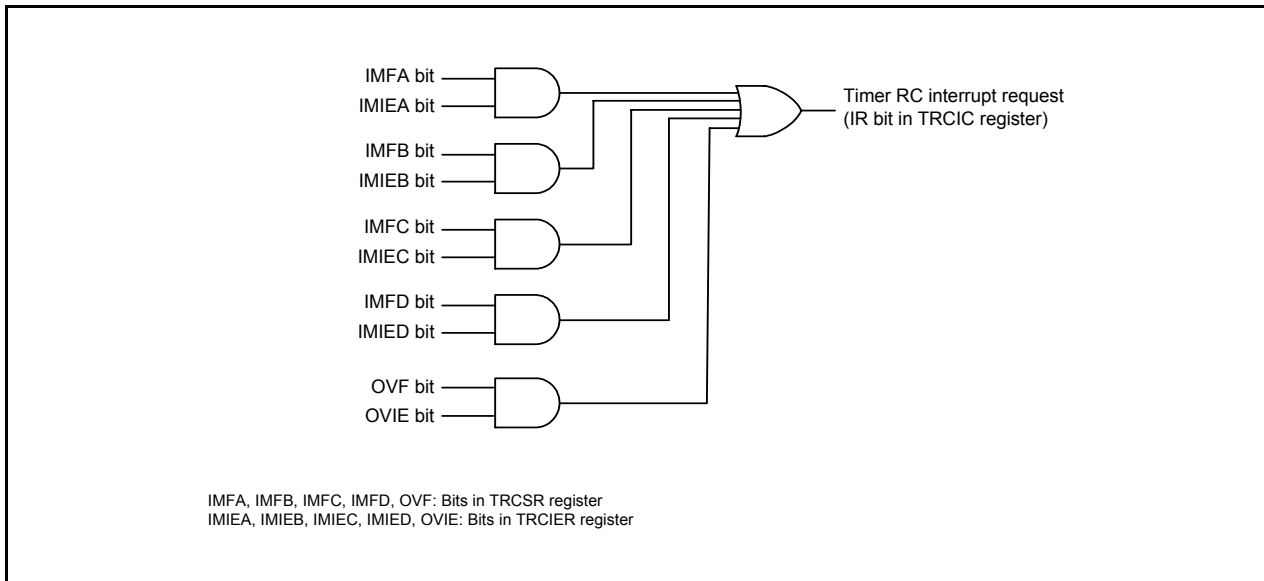


Figure 14.61 Timer RC Interrupt Block Diagram

Like other maskable interrupts, the timer RC interrupt is controlled by the combination of the I flag, IR bit, bits ILVL0 to ILVL2, and IPL. However, it differs from other maskable interrupts in the following respects because a single interrupt source (timer RC interrupt) is generated from multiple interrupt request sources.

- The IR bit in the TRCIC register is set to 1 (interrupt requested) when a bit in the TRCSR register is set to 1 and the corresponding bit in the TRCIER register is also set to 1 (interrupt enabled).
- The IR bit is set to 0 (no interrupt request) when the bit in the TRCSR register or the corresponding bit in the TRCIER register is set to 0, or both are set to 0. In other words, the interrupt request is not maintained if the IR bit is once set to 1 but the interrupt is not acknowledged.
- If after the IR bit is set to 1 another interrupt source is triggered, the IR bit remains set to 1 and does not change.
- If multiple bits in the TRCIER register are set to 1, use the TRCSR register to determine the source of the interrupt request.
- The bits in the TRCSR register are not automatically set to 0 when an interrupt is acknowledged. Set them to 0 within the interrupt routine. Refer to **Figure 14.31 TRCSR Register**, for the procedure for setting these bits to 0.

Refer to **Figure 14.30 TRCIER Register**, for details of the TRCIER register.

Refer to **12.1.6 Interrupt Control**, for details of the TRCIC register and **12.1.5.2 Relocatable Vector Tables**, for information on interrupt vectors.

14.3.9 Notes on Timer RC

14.3.9.1 TRC Register

- The following note applies when the CCLR bit in the TRCCR1 register is set to 1 (clear TRC register at compare match with TRCGRA register).

When using a program to write a value to the TRC register while the TSTART bit in the TRCMR register is set to 1 (count starts), ensure that the write does not overlap with the timing with which the TRC register is set to 0000h.

If the timing of the write to the TRC register and the setting of the TRC register to 0000h coincide, the write value will not be written to the TRC register and the TRC register will be set to 0000h.

- Reading from the TRC register immediately after writing to it can result in the value previous to the write being read out. To prevent this, execute the JMP.B instruction between the read and the write instructions.

```

Program Example      MOV.W      #XXXXh, TRC          ;Write
                   JMP.B      L1              ;JMP.B instruction
                   L1:         MOV.W      TRC,DATA        ;Read

```

14.3.9.2 TRCSR Register

Reading from the TRCSR register immediately after writing to it can result in the value previous to the write being read out. To prevent this, execute the JMP.B instruction between the read and the write instructions.

```

Program Example      MOV.B      #XXh, TRCSR        ;Write
                   JMP.B      L1              ;JMP.B instruction
                   L1:         MOV.B      TRCSR,DATA   ;Read

```

14.3.9.3 Count Source Switching

- Stop the count before switching the count source.

Switching procedure

- Set the TSTART bit in the TRCMR register to 0 (count stops).
- Change the settings of bits TCK2 to TCK0 in the TRCCR1 register.

- After switching the count source from fOCO40M to another clock, allow a minimum of two cycles of f1 to elapse after changing the clock setting before stopping fOCO40M.

Switching procedure

- Set the TSTART bit in the TRCMR register to 0 (count stops).
- Change the settings of bits TCK2 to TCK0 in the TRCCR1 register.
- Wait for a minimum of two cycles of f1.
- Set the FRA00 bit in the FRA0 register to 0 (high-speed on-chip oscillator off).

14.3.9.4 Input Capture Function

- The pulse width of the input capture signal should be three cycles or more of the timer RC operation clock (refer to **Table 14.12 Timer RC Operation Clock**).
- The value of the TRC register is transferred to the TRCGRj register one or two cycles of the timer RC operation clock after the input capture signal is input to the TRCIOj (j = A, B, C, or D) pin (when the digital filter function is not used).

14.3.9.5 TRCMR Register in PWM2 Mode

When the CSEL bit in the TRCCR2 register is set to 1 (count stops at compare match with the TRCGRA register), do not set the TRCMR register at compare match timing of registers TRC and TRCGRA.

14.4 Timer RD

Timer RD has 2 16-bit timers (channels 0 and 1). Each channel has 4 I/O pins.

The operation clock of timer RD is f1 or fOCO40M. Table 14.26 lists the Timer RD Operation Clocks.

Table 14.26 Timer RD Operation Clocks

| Condition | Operation Clock of Timer RD |
|---|-----------------------------|
| The count source is f1, f2, f4, f8, f32, or TRDCLK input (bits TCK2 to TCK0 in registers TRDCR0 and TRDCR1 are set to a value from 000b to 101b). | f1 |
| The count source is fOCO40M (bits TCK2 to TCK0 in registers TRDCR0 and TRDCR1 are set to 110b). | fOCO40M |

Figure 14.62 shows a Block Diagram of Timer RD. Timer RD has 5 modes:

- Timer mode
 - Input capture function Transfer the counter value to a register with an external signal as the trigger
 - Output compare function Detect register value matches with a counter (Pin output can be changed at detection)

The following 4 modes use the output compare function.

- PWM mode Output pulse of any width continuously
- Reset synchronous PWM mode Output three-phase waveforms (6) without sawtooth wave modulation and dead time
- Complementary PWM mode Output three-phase waveforms (6) with triangular wave modulation and dead time
- PWM3 mode Output PWM waveforms (2) with a fixed period

In the input capture function, output compare function, and PWM mode, channels 0 and 1 have the equivalent functions, and functions or modes can be selected individually for each pin. Also, a combination of these functions and modes can be used in 1 channel.

In reset synchronous PWM mode, complementary PWM mode, and PWM3 mode, a waveform is output with a combination of counters and registers in channels 0 and 1.

Tables 14.27 to 14.35 list the Pin Functions of timer RD.

Table 14.27 Pin Functions TRDIOA0/TRDCLK(P2_0)

| Register | TRDOER1 | TRDFCR | | | TRDIORA0 | | Function |
|---------------|------------------|--------|-------|------------|----------|------------|--|
| Bit | EA0 | PWM3 | STCLK | CMD1, CMD0 | IOA3 | IOA2_IOA0 | |
| Setting value | 0 | 0 | 0 | 00b | X | XXXb | PWM3 mode waveform output |
| | 0 | 1 | 0 | 00b | 1 | 001b, 01Xb | Timer mode waveform output (output compare function) |
| | X | 1 | 0 | 00b | X | 1XXb | Timer mode trigger input (input capture function) ⁽¹⁾ |
| | | 1 | 1 | XXb | X | 000b | External clock input (TRDCLK) ⁽¹⁾ |
| | Other than above | | | | | | I/O port |

X: can be 0 or 1, no change in outcome

NOTE:

1. Set the PD2_0 bit in the PD2 register to 0 (input mode) at timer mode trigger input (input capture function) and external clock input (TRDCLK).

Table 14.28 Pin Functions TRDIOB0(P2_1)

| Register | TRDOER1 | TRDFCR | | TRDPMR | TRDIORA0 | Function |
|---------------|------------------|--------|------------|--------|------------|--|
| Bit | EB0 | PWM3 | CMD1, CMD0 | PWMB0 | IOB2_IOB0 | |
| Setting value | 0 | X | 1Xb | X | XXXb | Complementary PWM mode waveform output |
| | 0 | X | 01b | X | XXXb | Reset synchronous PWM mode waveform output |
| | 0 | 0 | 00b | X | XXXb | PWM3 mode waveform output |
| | 0 | 1 | 00b | 1 | XXXb | PWM mode waveform output |
| | 0 | 1 | 00b | 0 | 001b, 01Xb | Timer mode waveform output (output compare function) |
| | X | 1 | 00b | 0 | 1XXb | Timer mode trigger input (input capture function) ⁽¹⁾ |
| | Other than above | | | | | I/O port |

X: can be 0 or 1, no change in outcome

NOTE:

1. Set the PD2_1 bit in the PD2 register to 0 (input mode) at timer mode trigger input (input capture function).

Table 14.29 Pin Functions TRDIOC0(P2_2)

| Register | TRDOER1 | TRDFCR | | TRDPMR | TRDIORC0 | Function |
|---------------|------------------|--------|------------|--------|------------|--|
| Bit | EC0 | PWM3 | CMD1, CMD0 | PWMC0 | IOC2_IOC0 | |
| Setting value | 0 | X | 1Xb | X | XXXb | Complementary PWM mode waveform output |
| | 0 | X | 01b | X | XXXb | Reset synchronous PWM mode waveform output |
| | 0 | 1 | 00b | 1 | XXXb | PWM mode waveform output |
| | 0 | 1 | 00b | 0 | 001b, 01Xb | Timer mode waveform output (output compare function) |
| | X | 1 | 00b | 0 | 1XXb | Timer mode trigger input (input capture function) ⁽¹⁾ |
| | Other than above | | | | | I/O port |

X: can be 0 or 1, no change in outcome

NOTE:

1. Set the PD2_2 bit in the PD2 register to 0 (input mode) at timer mode trigger input (input capture function).

Table 14.30 Pin Functions TRDIOD0(P2_3)

| Register | TRDOER1 | TRDFCR | | TRDPMR | TRDIORC0 | Function |
|------------------|---------|--------|------------|--------|------------|--|
| Bit | ED0 | PWM3 | CMD1, CMD0 | PWMD0 | IOD2_IOD0 | |
| Setting value | 0 | X | 1Xb | X | XXXb | Complementary PWM mode waveform output |
| | 0 | X | 01b | X | XXXb | Reset synchronous PWM mode waveform output |
| | 0 | 1 | 00b | 1 | XXXb | PWM mode waveform output |
| | 0 | 1 | 00b | 0 | 001b, 01Xb | Timer mode waveform output (output compare function) |
| | X | 1 | 00b | 0 | 1XXb | Timer mode trigger input (input capture function) ⁽¹⁾ |
| Other than above | | | | | | I/O port |

X: can be 0 or 1, no change in outcome

NOTE:

1. Set the PD2_3 bit in the PD2 register to 0 (input mode) at timer mode trigger input (input capture function).

Table 14.31 Pin Functions TRDIOA1(P2_4)

| Register | TRDOER1 | TRDFCR | | TRDIOA1 | Function |
|------------------|---------|--------|------------|------------|--|
| Bit | EA1 | PWM3 | CMD1, CMD0 | IOA2_IOA0 | |
| Setting value | 0 | X | 1Xb | XXXb | Complementary PWM mode waveform output |
| | 0 | X | 01b | XXXb | Reset synchronous PWM mode waveform output |
| | 0 | 1 | 00b | 001b, 01Xb | Timer mode waveform output (output compare function) |
| | X | 1 | 00b | 1XXb | Timer mode trigger input (input capture function) ⁽¹⁾ |
| Other than above | | | | | I/O port |

X: can be 0 or 1, no change in outcome

NOTE:

1. Set the PD2_4 bit in the PD2 register to 0 (input mode) at timer mode trigger input (input capture function).

Table 14.32 Pin Functions TRDIOB1(P2_5)

| Register | TRDOER1 | TRDFCR | | TRDPMR | TRDIOA1 | Function |
|------------------|---------|--------|------------|--------|------------|--|
| Bit | EB1 | PWM3 | CMD1, CMD0 | PWMB1 | IOB2_IOB0 | |
| Setting value | 0 | X | 1Xb | X | XXXb | Complementary PWM mode waveform output |
| | 0 | X | 01b | X | XXXb | Reset synchronous PWM mode waveform output |
| | 0 | 1 | 00b | 1 | XXXb | PWM mode waveform output |
| | 0 | 1 | 00b | 0 | 001b, 01Xb | Timer mode waveform output (output compare function) |
| | X | 1 | 00b | 0 | 1XXb | Timer mode trigger input (input capture function) ⁽¹⁾ |
| Other than above | | | | | | I/O port |

X: can be 0 or 1, no change in outcome

NOTE:

1. Set the PD2_5 bit in the PD2 register to 0 (input mode) at timer mode trigger input (input capture function).

Table 14.33 Pin Functions TRDIOC1(P2_6)

| Register | TRDOER1 | TRDFCR | | TRDPMR | TRDIORC1 | Function |
|------------------|---------|--------|------------|--------|------------|--|
| Bit | EC1 | PWM3 | CMD1, CMD0 | PWMC1 | IOC2_IOC0 | |
| Setting value | 0 | X | 1Xb | X | XXXb | Complementary PWM mode waveform output |
| | 0 | X | 01b | X | XXXb | Reset synchronous PWM mode waveform output |
| | 0 | 1 | 00b | 1 | XXXb | PWM mode waveform output |
| | 0 | 1 | 00b | 0 | 001b, 01Xb | Timer mode waveform output (output compare function) |
| | X | 1 | 00b | 0 | 1XXb | Timer mode trigger input (input capture function) ⁽¹⁾ |
| Other than above | | | | | | I/O port |

X: can be 0 or 1, no change in outcome

NOTE:

1. Set the PD2_6 bit in the PD2 register to 0 (input mode) at timer mode trigger input (input capture function).

Table 14.34 Pin Functions TRDIOD1(P2_7)

| Register | TRDOER1 | TRDFCR | | TRDPMR | TRDIORC1 | Function |
|------------------|---------|--------|------------|--------|------------|--|
| Bit | ED1 | PWM3 | CMD1, CMD0 | PWMD1 | IOD2_IOD0 | |
| Setting value | 0 | X | 1Xb | X | XXXb | Complementary PWM mode waveform output |
| | 0 | X | 01b | X | XXXb | Reset synchronous PWM mode waveform output |
| | 0 | 1 | 00b | 1 | XXXb | PWM mode waveform output |
| | 0 | 1 | 00b | 0 | 001b, 01Xb | Timer mode waveform output (output compare function) |
| | X | 1 | 00b | 0 | 1XXb | Timer mode trigger input (input capture function) ⁽¹⁾ |
| Other than above | | | | | | I/O port |

X: can be 0 or 1, no change in outcome

NOTE:

1. Set the PD2_7 bit in the PD2 register to 0 (input mode) at timer mode trigger input (input capture function).

Table 14.35 Pin Functions $\overline{\text{INT0}}$ (P4_5)

| Register | TRDOER2 | INTEN | | PD4 | Function |
|------------------|---------|--------|--------|-------|--|
| Bit | PTO | INT0PL | INT0EN | PD4_5 | |
| Setting value | 1 | 0 | 1 | 0 | Pulse output forced cutoff signal input |
| Other than above | | | | | I/O port or $\overline{\text{INT0}}$ interrupt input |

X: can be 0 or 1, no change in outcome

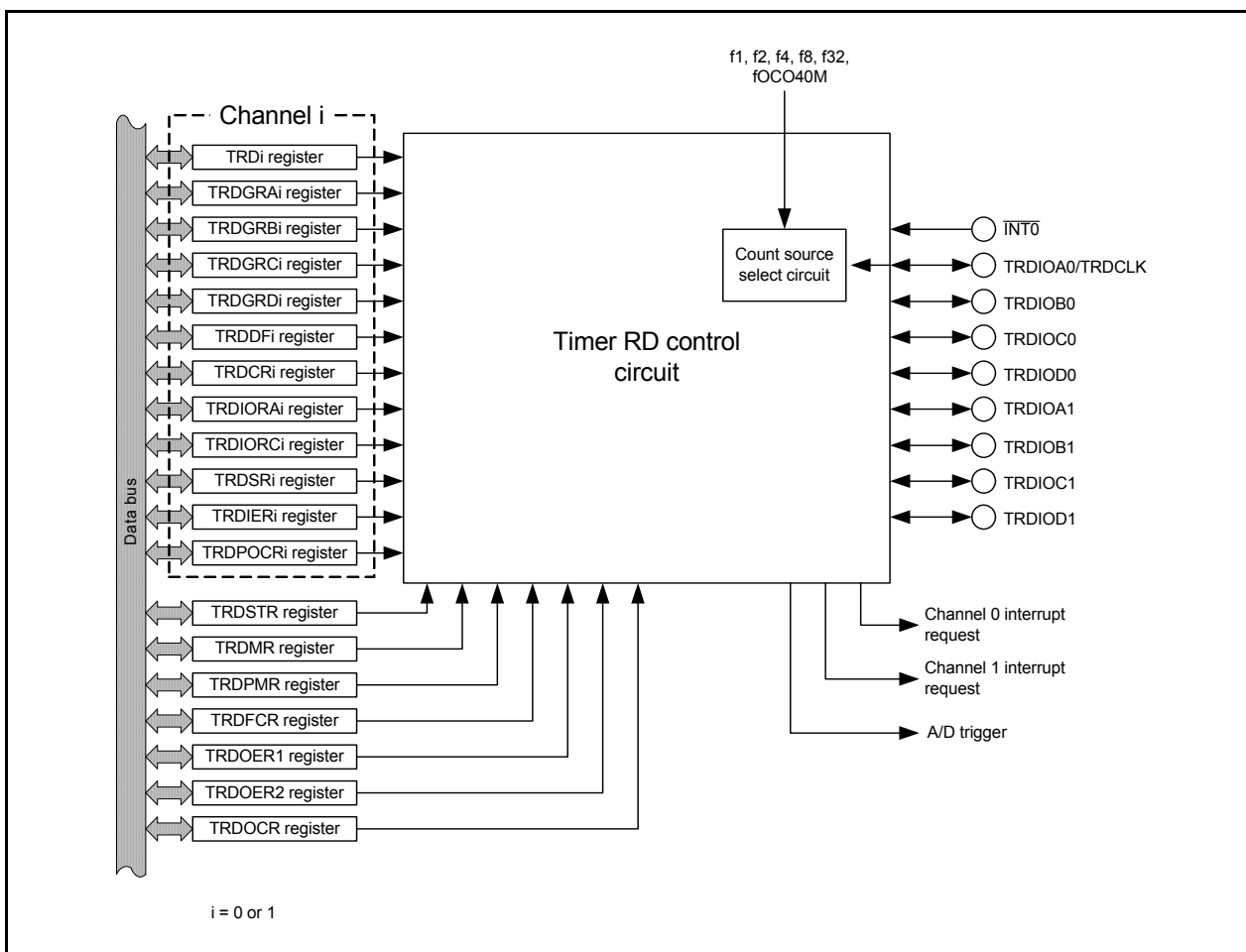


Figure 14.62 Block Diagram of Timer RD

14.4.1 Count Sources

The count source selection method is the same in all modes. However, in PWM3 mode, the external clock cannot be selected.

Table 14.36 Count Source Selection

| Count Source | Selection |
|-------------------------------------|--|
| f1, f2, f4, f8, f32 | The count source is selected by bits TCK2 to TCK0 in the TRDCR _i register. |
| fOCO40M ⁽¹⁾ | The FRA00 bit in the FRA0 register is set to 1 (high-speed on-chip oscillator frequency). Bits TCK2 to TCK0 in the TRDCR _i register is set to 110b (fOCO40M). |
| External signal input to TRDCLK pin | The STCLK bit in the TRDFCR register is set to 1 (external clock input enabled). Bits TCK2 to TCK0 in the TRDCR _i register are set to 101b (count source: external clock). The valid edge is selected by bits CKEG1 to CKEG0 in the TRDCR _i register. The PD2_0 bit in the PD2 register is set to 0 (input mode). |

i = 0 or 1

NOTE:

- The count source fOCO40M can be used with VCC = 3.0 to 5.5 V.

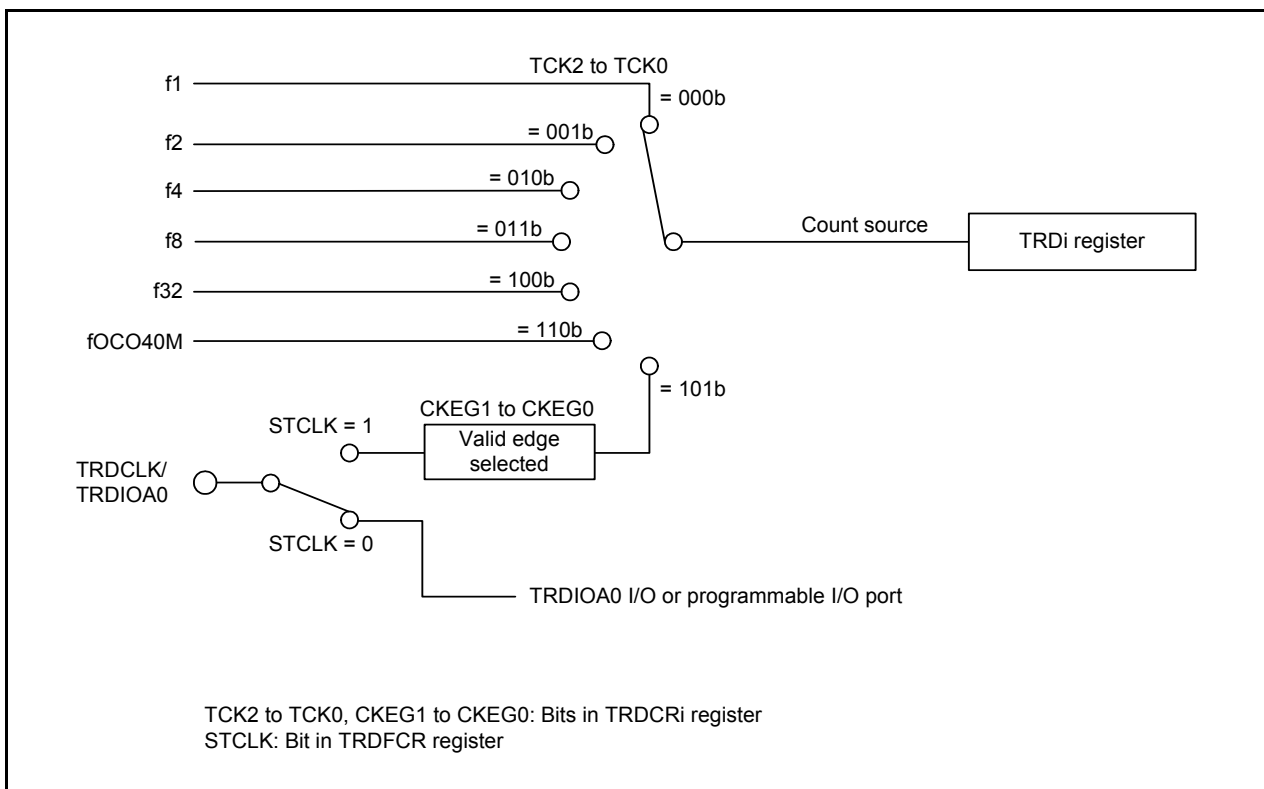


Figure 14.63 Block Diagram of Count Source

Set the pulse width of the external clock which inputs to the TRDCLK pin to 3 cycles or above of the operation clock of timer RD (refer to **Table 14.26 Timer RD Operation Clocks**).

When selecting fOCO40M for the count source, set the FRA00 bit in the FRA0 register to 1 (high-speed on-chip oscillator on) before setting bits TCK2 to TCK0 in the TRDCR_i register (i = 0 or 1) to 110b (fOCO40M).

14.4.2 Buffer Operation

The TRDGRC_i ($i = 0$ or 1) register can be used as the buffer register of the TRDGRA_i register, and the TRDGRD_i register can be used as the buffer register of the TRDGRB_i register by means of bits BFC_i and BFD_i in the TRDMR register.

- TRDGRA_i buffer register: TRDGRC_i register
- TRDGRB_i buffer register: TRDGRD_i register

Buffer operation depends on the mode. Table 14.37 lists the Buffer Operation in Each Mode.

Table 14.37 Buffer Operation in Each Mode

| Function and Mode | Transfer Timing | Transfer Register |
|----------------------------|---|--|
| Input capture function | Input capture signal input | Transfer content in TRDGRA _i (TRDGRB _i) register to buffer register |
| Output compare function | Compare match with TRD _i register and TRDGRA _i (TRDGRB _i) register | Transfer content in buffer register to TRDGRA _i (TRDGRB _i) register |
| PWM mode | | |
| Reset synchronous PWM mode | Compare match with TRD ₀ register and TRDGRA ₀ register | Transfer content in buffer register to TRDGRA _i (TRDGRB _i) register |
| Complementary PWM mode | <ul style="list-style-type: none"> • Compare match with TRD₀ register and TRDGRA₀ register • TRD₁ register underflow | Transfer content in buffer register to registers TRDGRB ₀ , TRDGRA ₁ , and TRDGRB ₁ |
| PWM3 mode | Compare match with TRD ₀ register and TRDGRA ₀ register | Transfer content in buffer register to registers TRDGRA ₀ , TRDGRB ₀ , TRDGRA ₁ , and TRDGRB ₁ |

$i = 0$ or 1

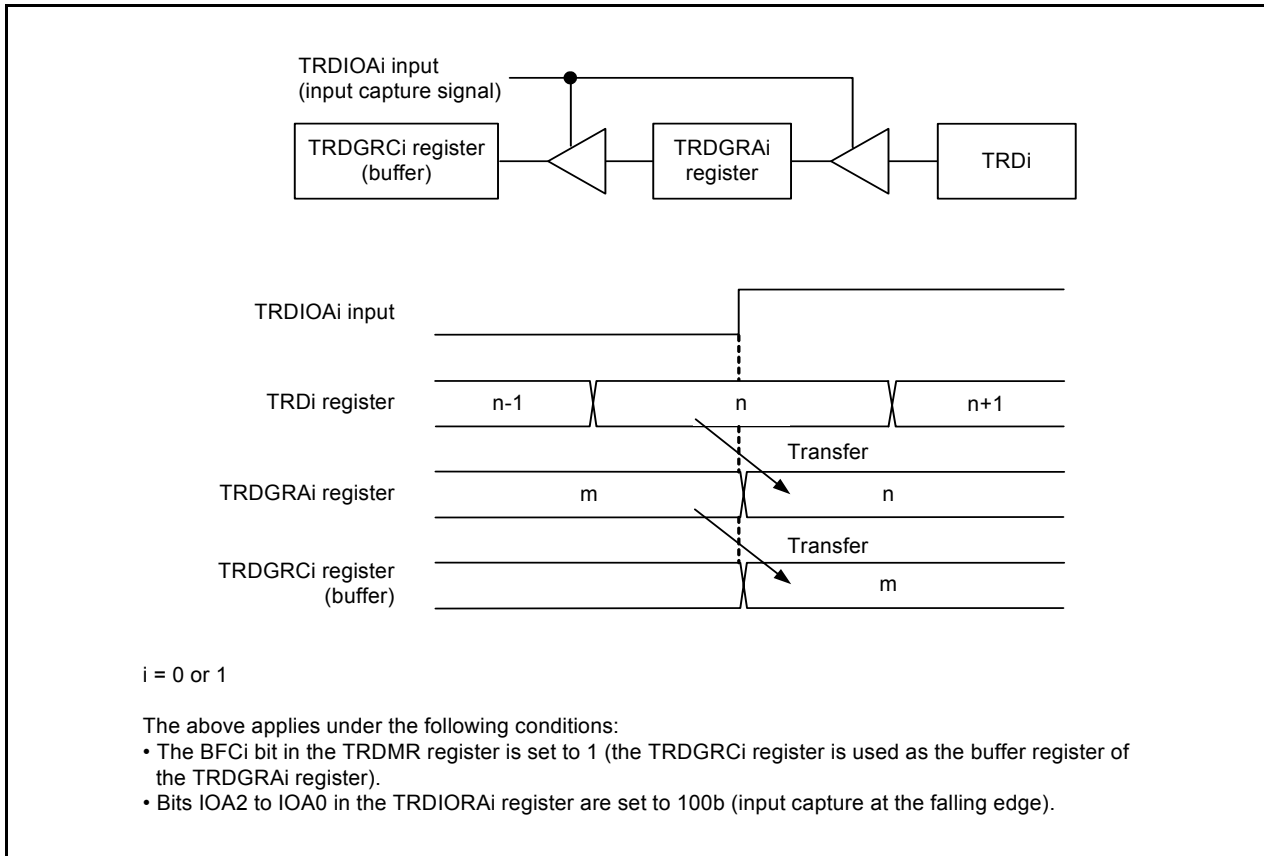


Figure 14.64 Buffer Operation in Input Capture Function

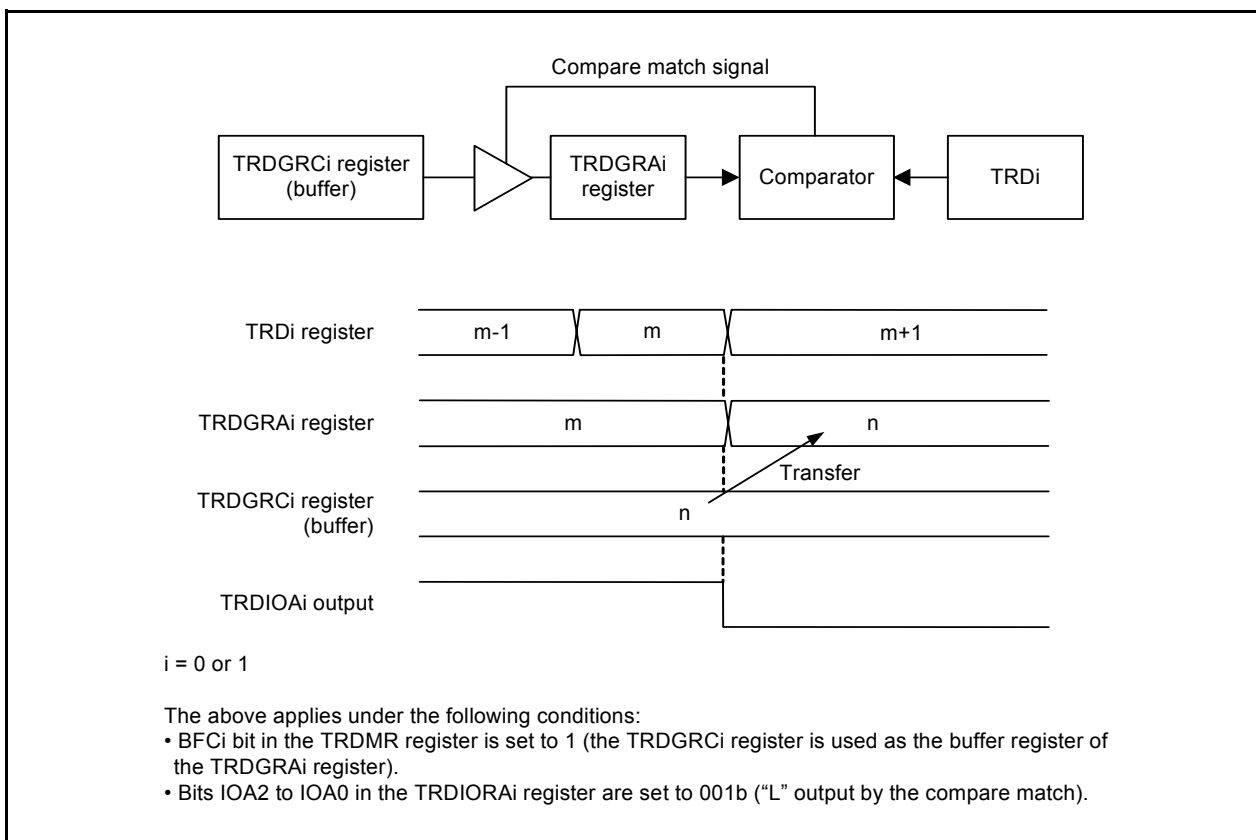


Figure 14.65 Buffer Operation in Output Compare Function

Perform the following for the timer mode (input capture and output compare functions).

When using the TRDGRC i ($i = 0 \text{ or } 1$) register as the buffer register of the TRDGRA i register

- Set the IOC3 bit in the TRDIORC i register to 1 (general register or buffer register).
- Set the IOC2 bit in the TRDIORC i register to the same value as the IOA2 bit in the TRDIOR A_i register.

When using the TRDGRD i register as the buffer register of the TRDGRB i register

- Set the IOD3 bit in the TRDIORD i register to 1 (general register or buffer register).
- Set the IOD2 bit in the TRDIORC i register to the same value as the IOB2 bit in the TRDIOR A_i register.

Bits IMFC and IMFD in the TRDSR i register are set to 1 at the input edge of the TRDIOC i pin when also using registers TRDGRC i and TRDGRD i as the buffer register in the input capture function.

When also using registers TRDGRC i and TRDGRD i as buffer registers for the output compare function, reset synchronous PWM mode, complementary PWM mode, and PWM3 mode, bits IMFC and IMFD in the TRDSR i register are set to 1 by a compare match with the TRDi register.

14.4.3 Synchronous Operation

The TRD1 register is synchronized with the TRD0 register.

- Synchronous preset

When the SYNC bit in the TRDMR register is set to 1 (synchronous operation), the data is written to both the TRD0 and TRD1 registers after writing to the TRDi register.

- Synchronous clear

When the SYNC bit in the TRDMR register is set to 1 and bits CCLR2 to CCLR0 in the TRDCRi register are set to 011b (synchronous clear), the TRD0 register is set to 0000h at the same time as the TRD1 register is set to 0000h.

Also, when the SYNC bit in the TRDMR register is set to 1 and bits CCLR2 to CCLR0 in the TRDCRi register are set to 011b (synchronous clear), the TRD1 register is set to 0000h at the same time as the TRD0 register is set to 0000h.

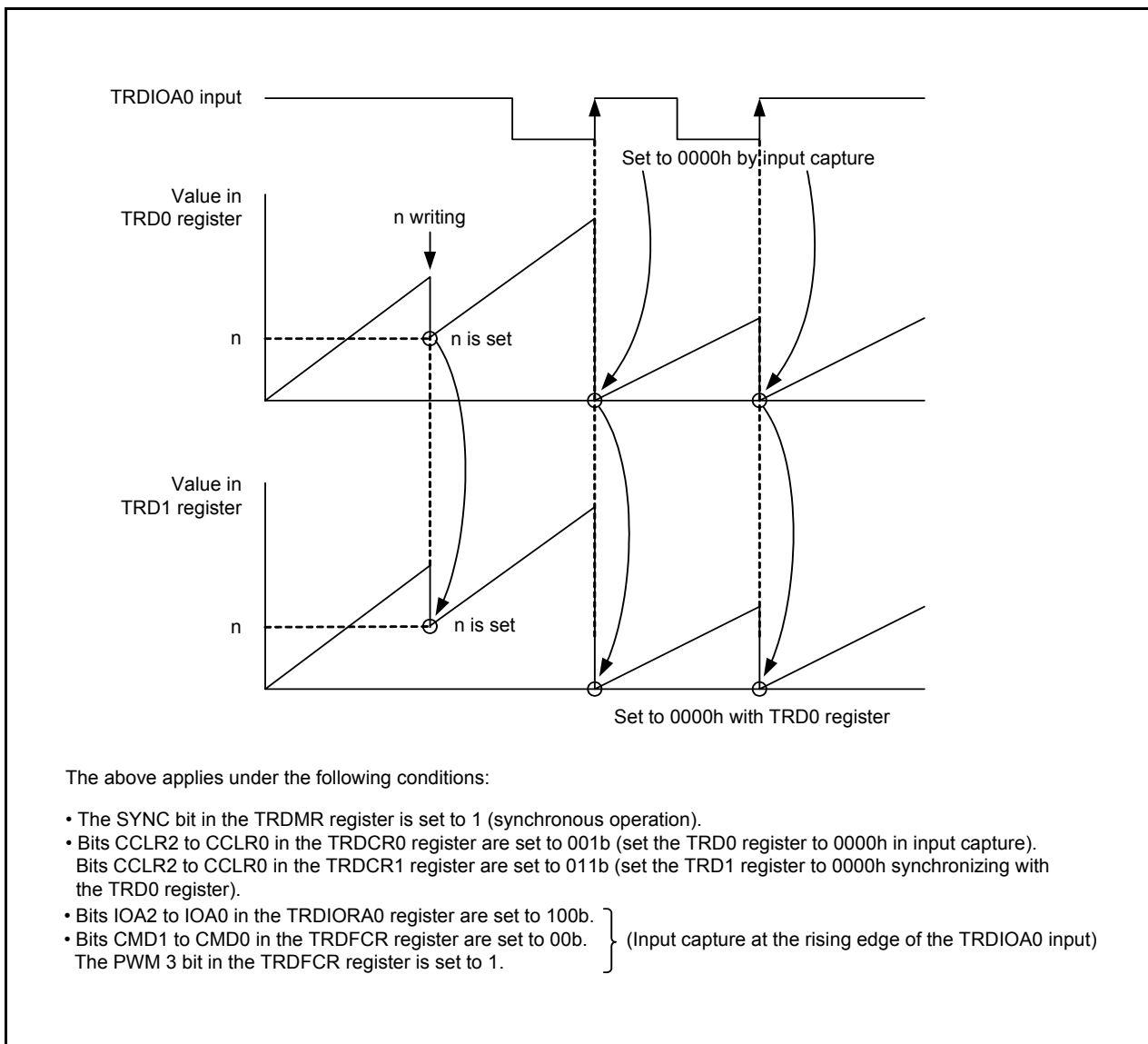


Figure 14.66 Synchronous Operation

14.4.4 Pulse Output Forced Cutoff

In the output compare function, PWM mode, reset synchronous PWM mode, complementary PWM mode, and PWM3 mode, the TRDIO_{ji} (i = 0 or 1, j = either A, B, C, or D) output pin can be forcibly set to a programmable I/O port by the $\overline{\text{INT0}}$ pin input, and pulse output can be cut off.

The pins used for output in these functions or modes can function as the output pin of timer RD when the applicable bit in the TRDOER1 register is set to 0 (enable timer RD output). When the PTO bit in the TRDOER2 register is set to 1 ($\overline{\text{INT0}}$ of pulse output forced cutoff signal input enabled), all bits in the TRDOER1 register are set to 1 (disable timer RD output, the TRDIO_{ji} output pin is used as the programmable I/O port) after “L” is applied to the $\overline{\text{INT0}}$ pin. The TRDIO_{ji} output pin is set to the programmable I/O port after “L” is applied to the $\overline{\text{INT0}}$ pin and waiting for 1 to 2 cycles of the timer RD operation clock (refer to **Table 14.26 Timer RD Operation Clocks**).

Set as below when using this function:

- Set the pin status (high impedance, “L” or “H” output) to pulse output forced cutoff by registers P2 and PD2.
- Set the INT0EN bit in the INTEN register to 1 (enable $\overline{\text{INT0}}$ input) and the INT0PL bit to 0 (one edge).
- Set the PD4_5 bit in the PD4 register to 0 (input mode).
- Set the $\overline{\text{INT0}}$ digital filter by bits INT0F1 to INT0F0 in the INTF register.
- Set the PTO bit in the TRDOER2 register to 1 (enable pulse output forced cutoff signal input $\overline{\text{INT0}}$).

According to the selection of the POL bit in the INT0IC register and change of the $\overline{\text{INT0}}$ pin input, the IR bit in the INT0IC register is set to 1 (interrupt request). Refer to **12. Interrupts** for details of interrupts.

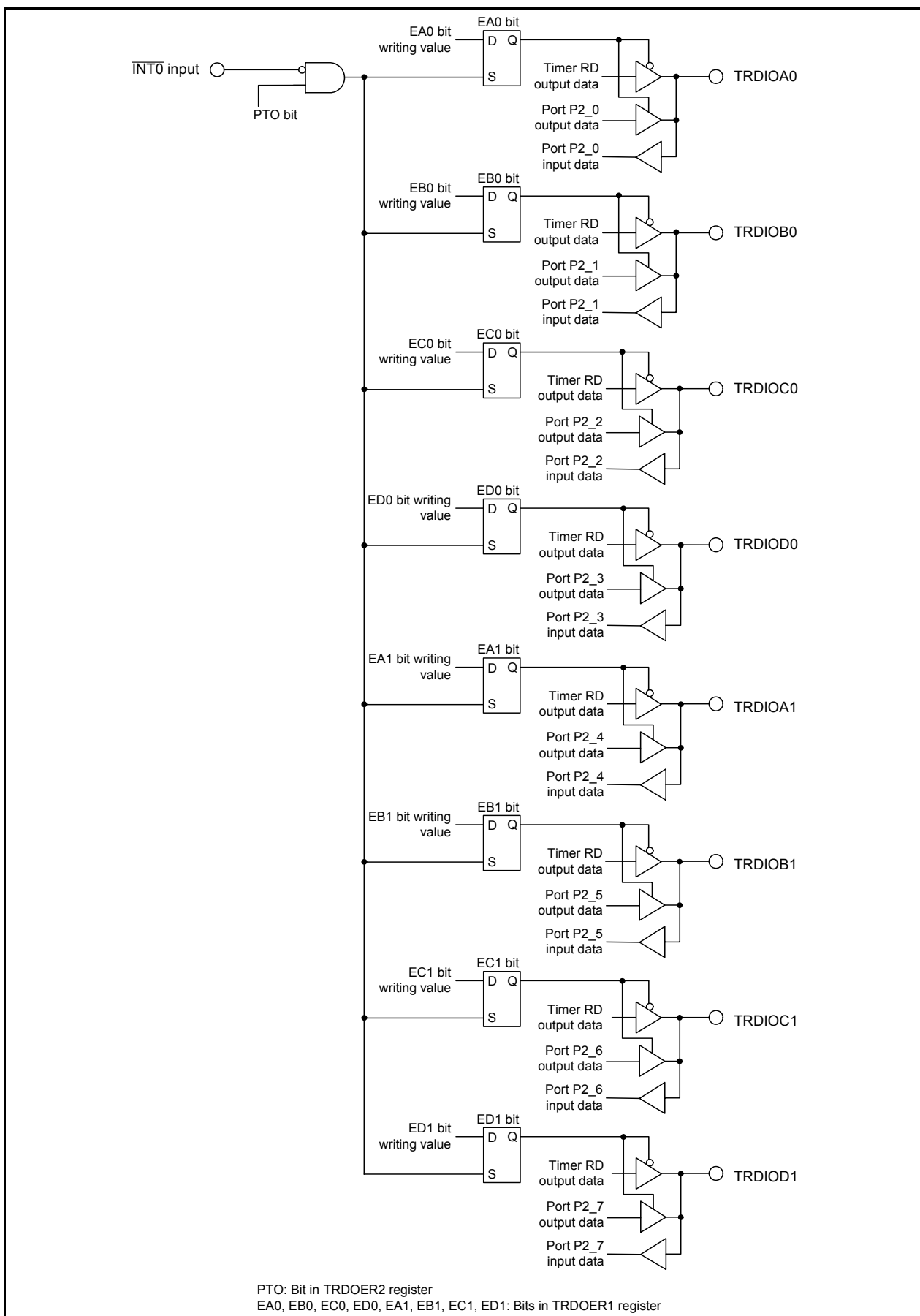


Figure 14.67 Pulse Output Forced Cutoff

14.4.5 Input Capture Function

The input capture function measures the external signal width and period. The content of the TRDi register (counter) is transferred to the TRDGRji register as a trigger of the TRDIOji ($i = 0$ or 1 , $j = \text{either A, B, C, or D}$) pin external signal (input capture). Since this function is enabled with a combination of the TRDIOji pin and TRDGRji register, the input capture function, or any other mode or function, can be selected for each individual pin.

The TRDGRA0 register can also select fOCO128 signal as input-capture trigger input.

Figure 14.68 shows a Block Diagram of Input Capture Function, Table 14.38 lists the Input Capture Function Specifications. Figures 14.69 to 14.80 show the Registers Associated with Input Capture Function, and Figure 14.81 shows an Operating Example of Input Capture Function.

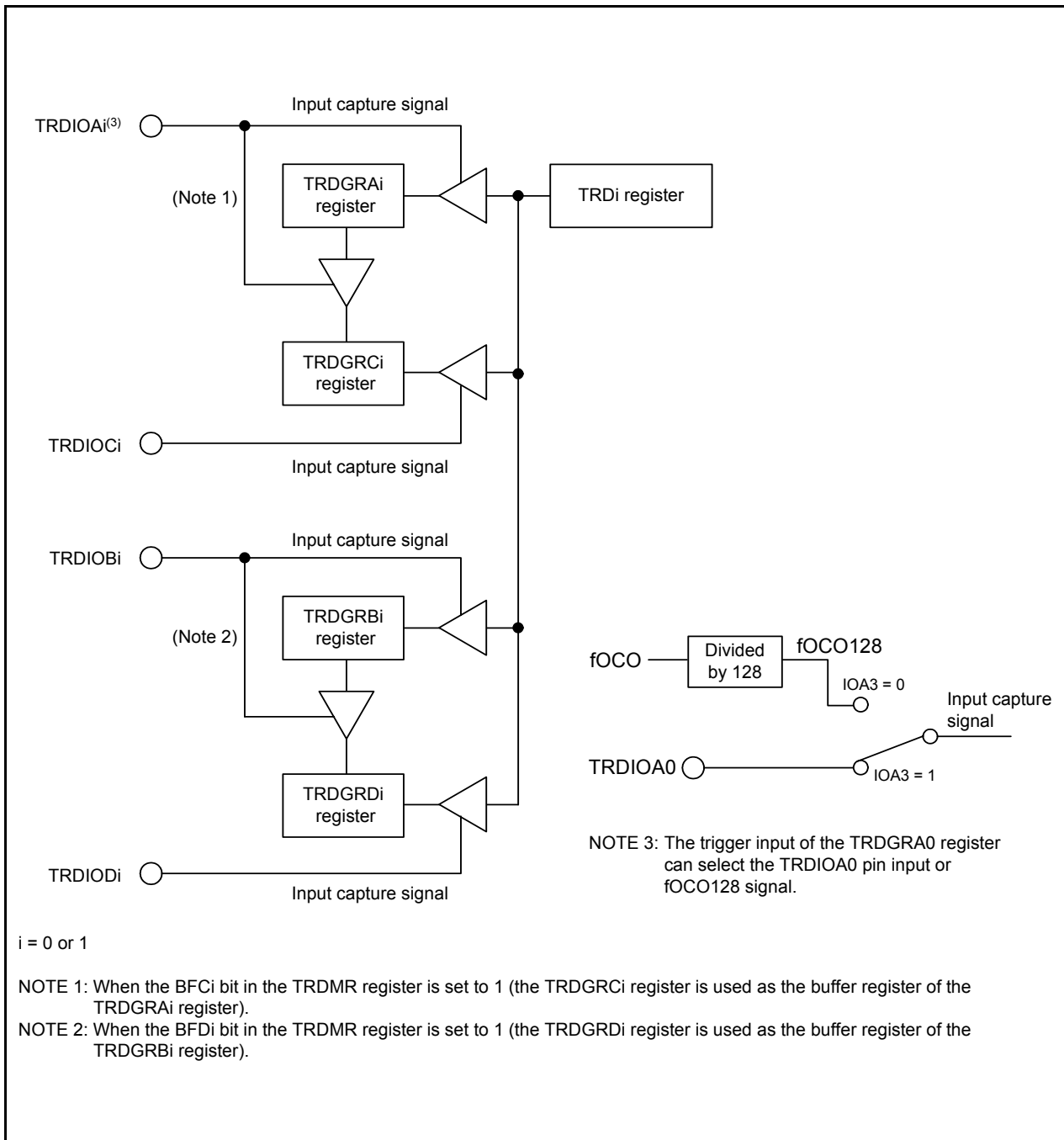


Figure 14.68 Block Diagram of Input Capture Function

Table 14.38 Input Capture Function Specifications

| Item | Specification |
|---|---|
| Count sources | f1, f2, f4, f8, f32, fOCO40M External signal input to the TRDCLK pin (valid edge selected by a program) |
| Count operations | Increment |
| Count period | When bits CCLR2 to CCLR0 in the TRDCRi register are set to 000b (free-running operation). $1/fk \times 65536$ fk: Frequency of count source |
| Count start condition | 1 (count starts) is written to the TSTARTi bit in the TRDSTR register. |
| Count stop condition | 0 (count stops) is written to the TSTARTi bit in the TRDSTR register when the CSELi bit in the TRDSTR register is set to 1. |
| Interrupt request generation timing | <ul style="list-style-type: none"> Input capture (valid edge of TRDIOji input or fOCO128 signal edge) TRDi register overflows |
| TRDIOA0 pin function | Programmable I/O port, input-capture input, or TRDCLK (external clock) input |
| TRDIOB0, TRDIOC0, TRDIOD0, TRDIOA1 to TRDIOD1 pin functions | Programmable I/O port, or input-capture input (selectable by pin) |
| INT0 pin function | Programmable I/O port or INT0 interrupt input |
| Read from timer | The count value can be read by reading the TRDi register. |
| Write to timer | <ul style="list-style-type: none"> When the SYNC bit in the TRDMR register is set to 0 (channels 0 and 1 operate independently). Data can be written to the TRDi register. When the SYNC bit in the TRDMR register is set to 1 (channels 0 and 1 operate synchronously). Data can be written to both the TRD0 and TRD1 registers by writing to the TRDi register. |
| Select functions | <ul style="list-style-type: none"> Input-capture input pin selected Either 1 pin or multiple pins among TRDIOAi, TRDIOBi, TRDIOCi, or TRDIODi. Input-capture input valid edge selected The rising edge, falling edge, or both the rising and falling edges The timing when the TRDi register is set to 0000h At overflow or input capture Buffer operation (Refer to 14.4.2 Buffer Operation.) Synchronous operation (Refer to 14.4.3 Synchronous Operation.) Digital filter The TRDIOji input is sampled, and when the sampled input level match as 3 times, the level is determined. Input-capture trigger selected fOCO128 can be selected for input-capture trigger input of the TRDGRA0 register. |

i = 0 or 1, j = either A, B, C, or D

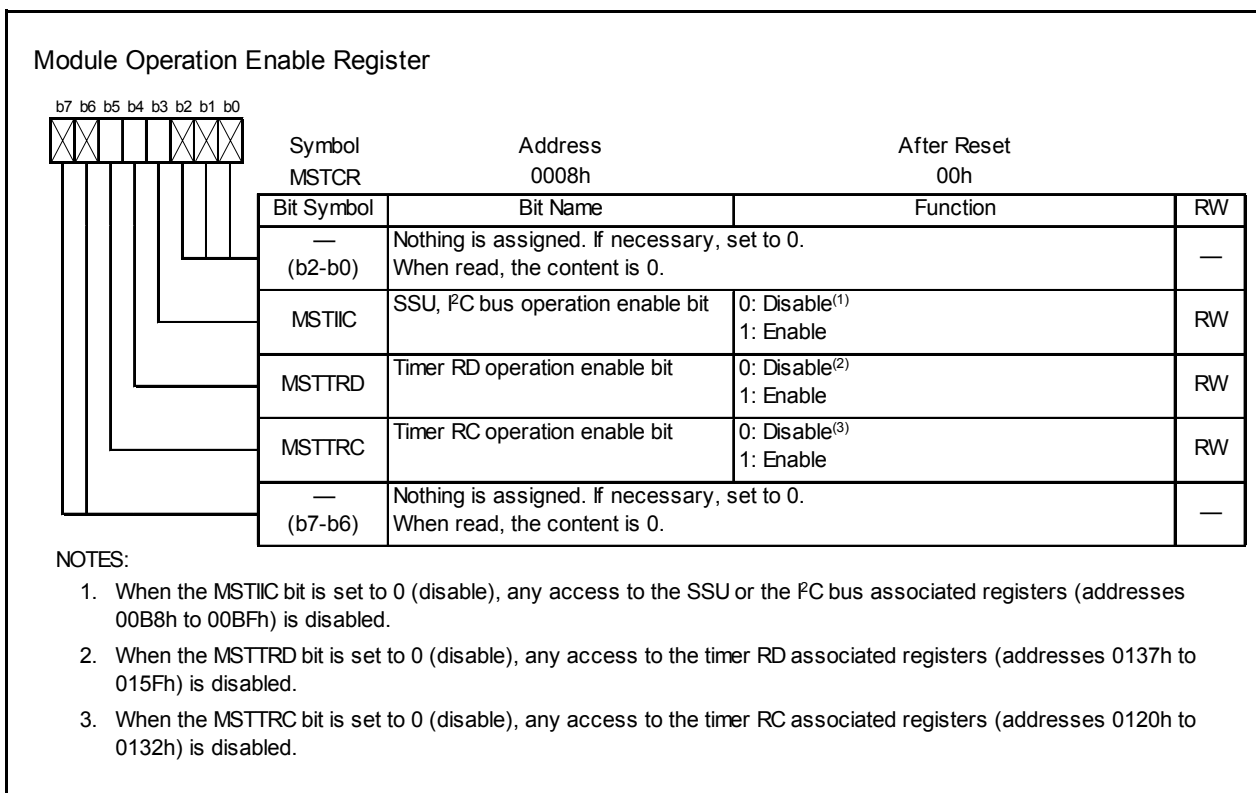


Figure 14.69 MSTCR Register

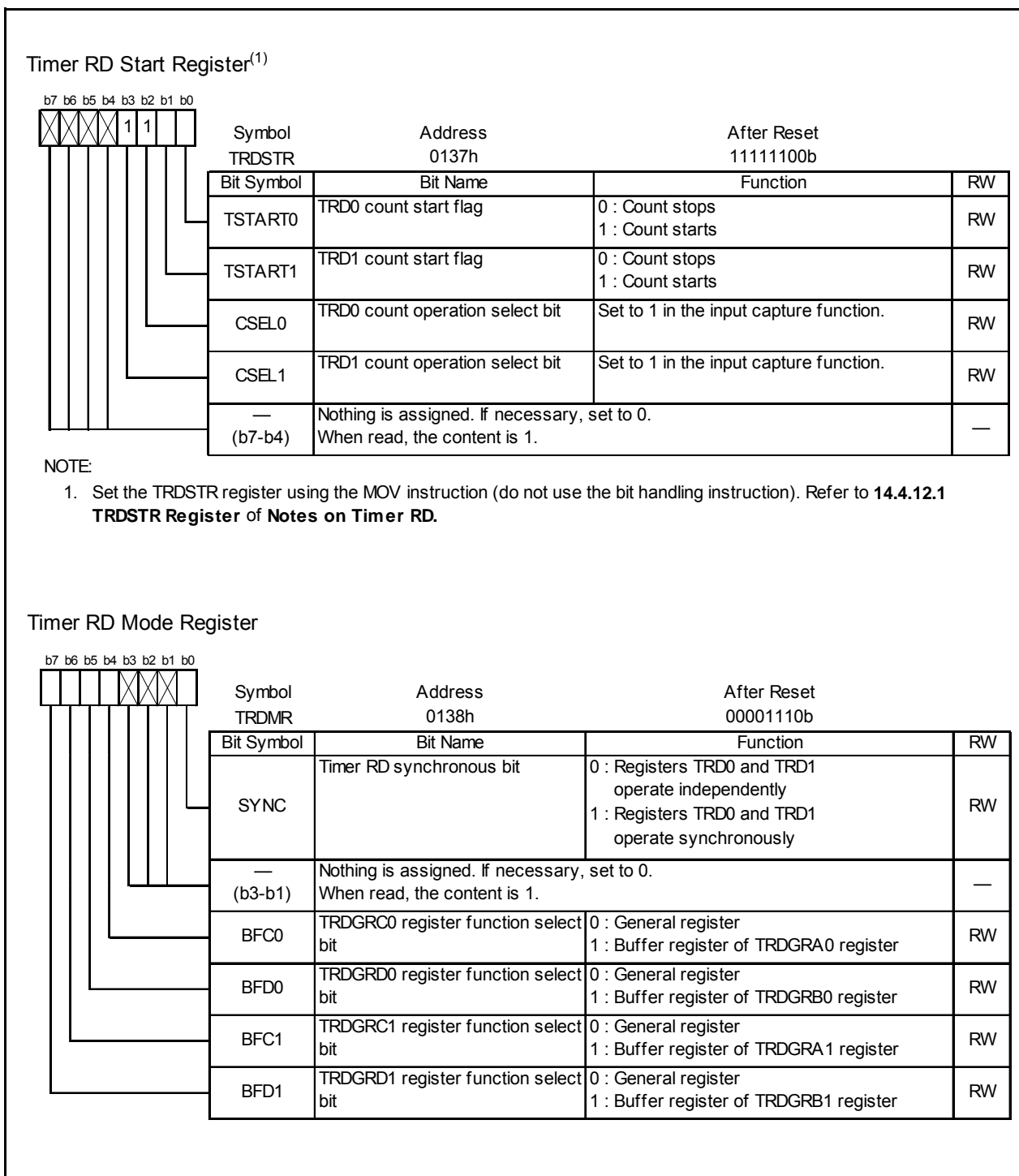


Figure 14.70 Registers TRDSTR and TRDMR in Input Capture Function

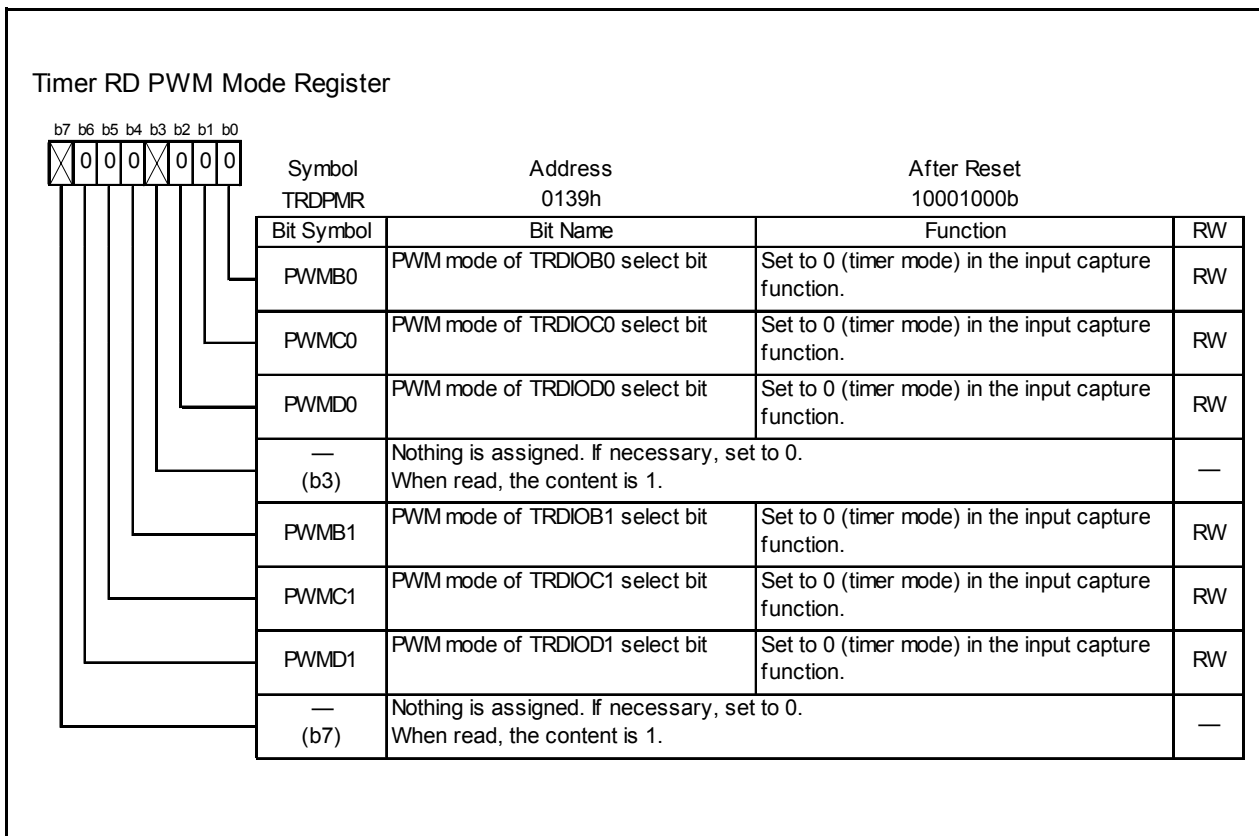


Figure 14.71 TRDPMR Register in Input Capture Function

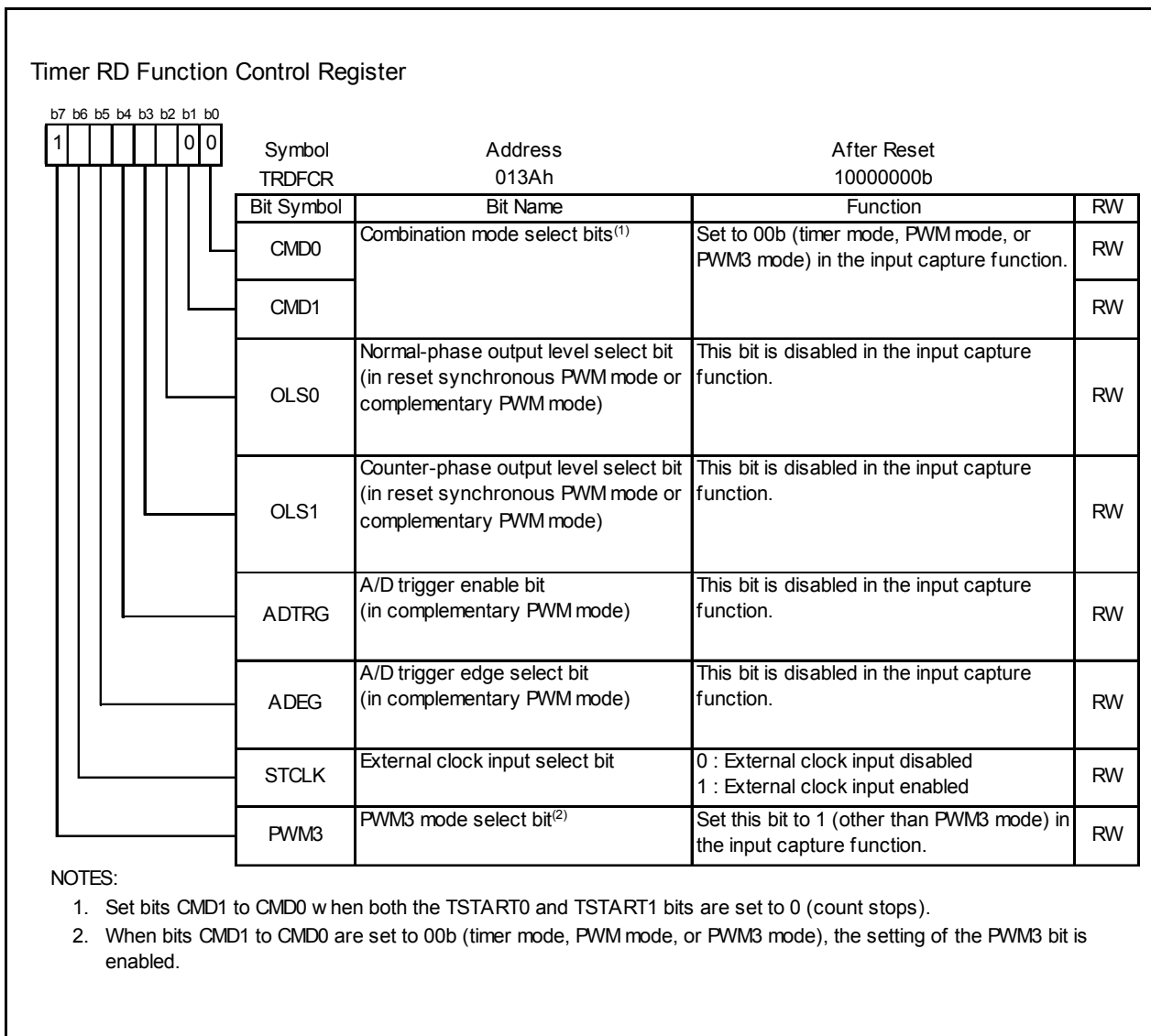


Figure 14.72 TRDFCR Register in Input Capture Function

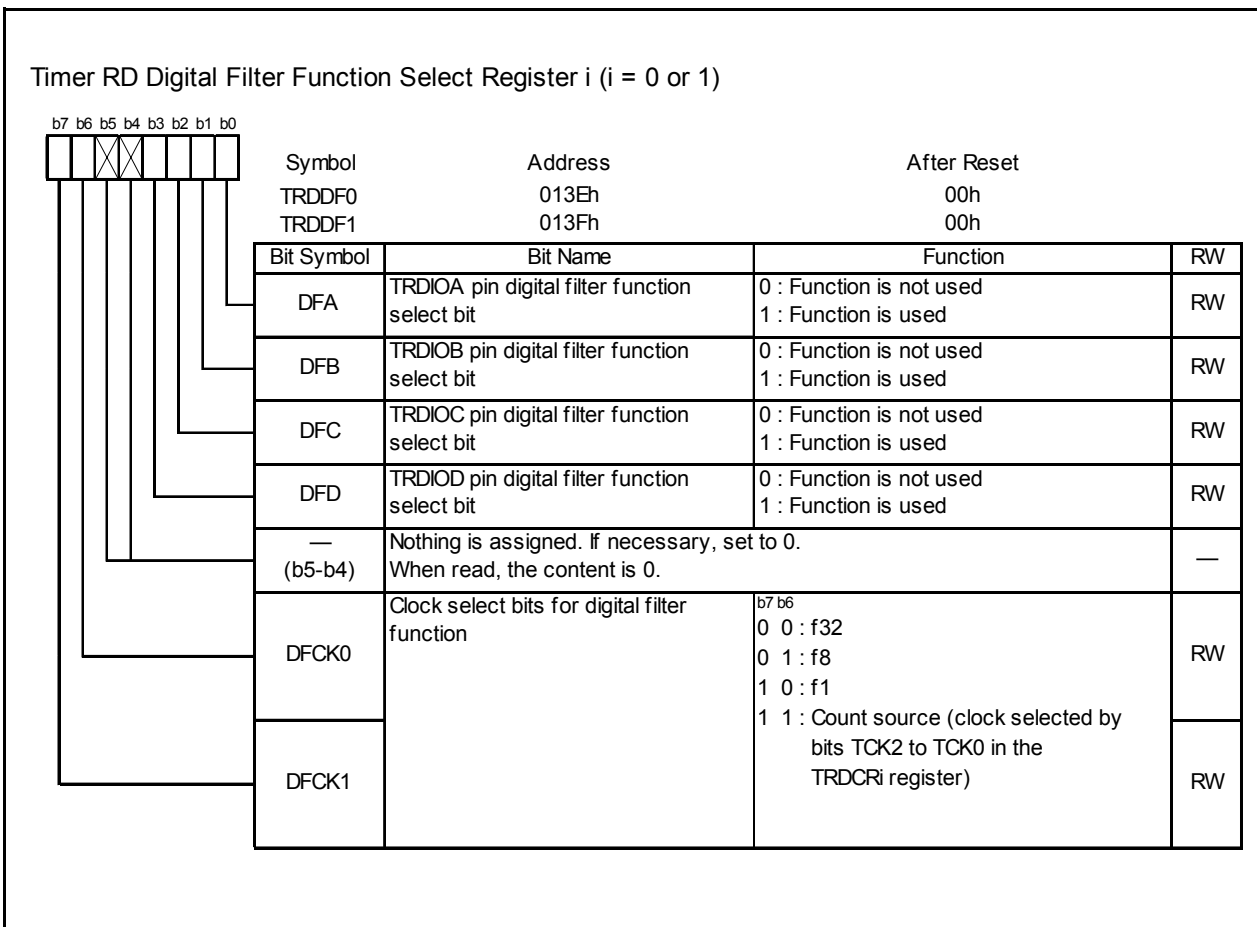


Figure 14.73 Registers TRDDF0 to TRDDF1 in Input Capture Function

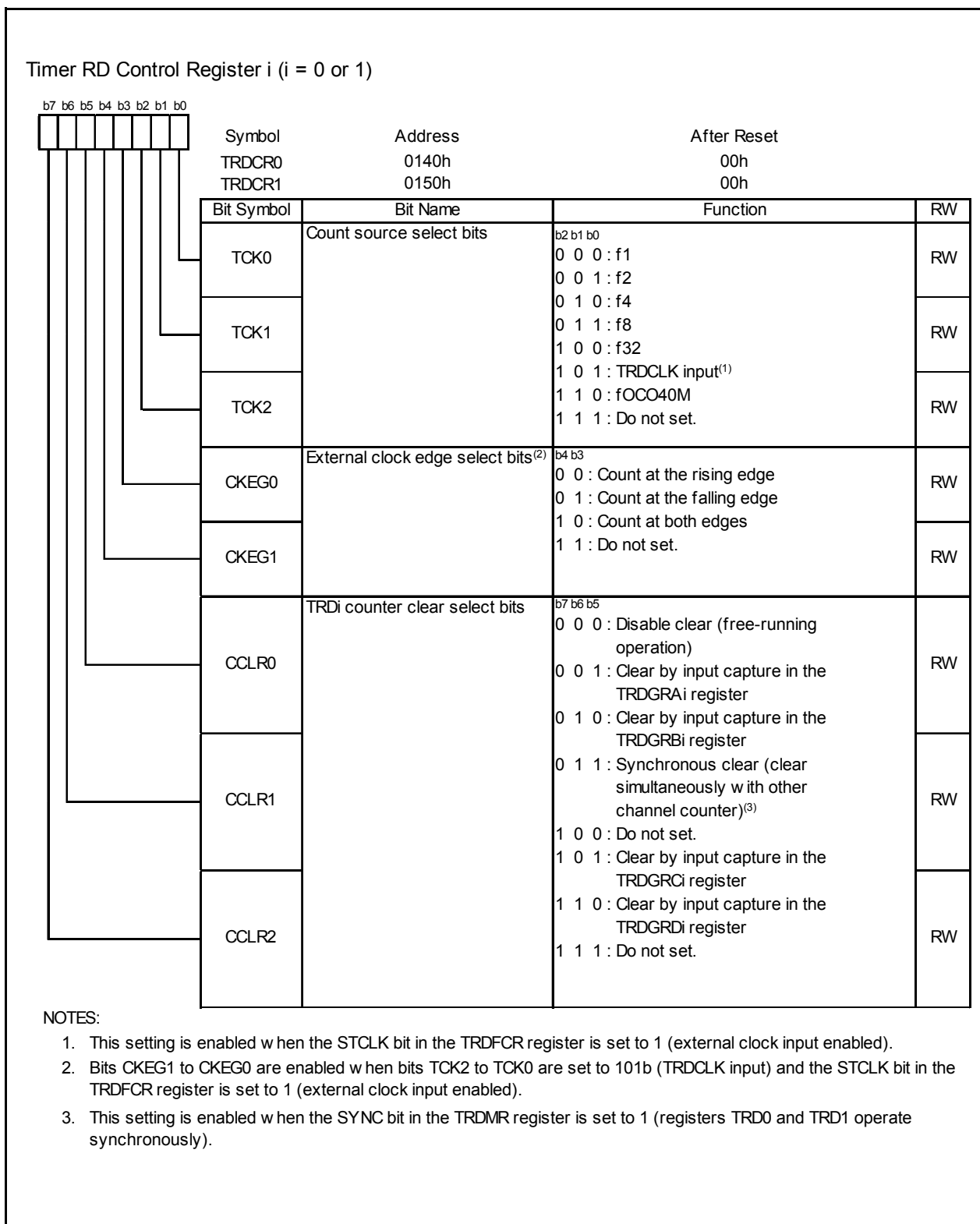


Figure 14.74 Registers TRDCR0 to TRDCR1 in Input Capture Function

Timer RD I/O Control Register Ai (i = 0 or 1)

| Symbol | Address | After Reset |
|----------|---------|-------------|
| TRDIORA0 | 0141h | 10001000b |
| TRDIORA1 | 0151h | 10001000b |

| Bit Symbol | Bit Name | Function | RW |
|------------|--|---|----|
| IOA0 | TRDGRA control bits | b1 b0 0 0 : Input capture to the TRDGRAi register at the rising edge | RW |
| | | 0 1 : Input capture to the TRDGRAi register at the falling edge | |
| IOA1 | TRDGRA control bits | 1 0 : Input capture to the TRDGRAi register at both edges | RW |
| | | 1 1 : Do not set. | |
| IOA2 | TRDGRA mode select bit ⁽¹⁾ | Set to 1 (input capture) in the input capture function. | RW |
| IOA3 | Input capture input switch bit ^(3, 4) | 0 : fOCO128 Signal 1 : TRDIOA0 pin input | RW |
| IOB0 | TRDGRB control bits | b5 b4 0 0 : Input capture to the TRDGRBi register at the rising edge | RW |
| | | 0 1 : Input capture to the TRDGRBi register at the falling edge | |
| IOB1 | TRDGRB control bits | 1 0 : Input capture to the TRDGRBi register at both edges | RW |
| | | 1 1 : Do not set. | |
| IOB2 | TRDGRB mode select bit ⁽²⁾ | Set to 1 (input capture) in the input capture function. | RW |
| — (b7) | Nothing is assigned. If necessary, set to 0. When read, the content is 1. | | — |

NOTES:

- To select 1 (the TRDGRCi register is used as a buffer register of the TRDGRAi register) for this bit by the BFCi bit in the TRDMR register, set the IOC2 bit in the TRDIORCi register to the same value as the IOA2 bit in the TRDIORAi register.
- To select 1 (the TRDGRDi register is used as a buffer register of the TRDGRBi register) for this bit by the BFDi bit in the TRDMR register, set the IOD2 bit in the TRDIORCi register to the same value as the IOB2 bit in the TRDIORAi register.
- The IOA3 bit is enabled in the TRDIORA0 register only. Set to the IOA3 bit in TRDIORA1 to 1.
- The IOA3 bit is enabled when the IOA2 bit is set to 1 (input capture function).

Figure 14.75 Registers TRDIORA0 to TRDIORA1 in Input Capture Function

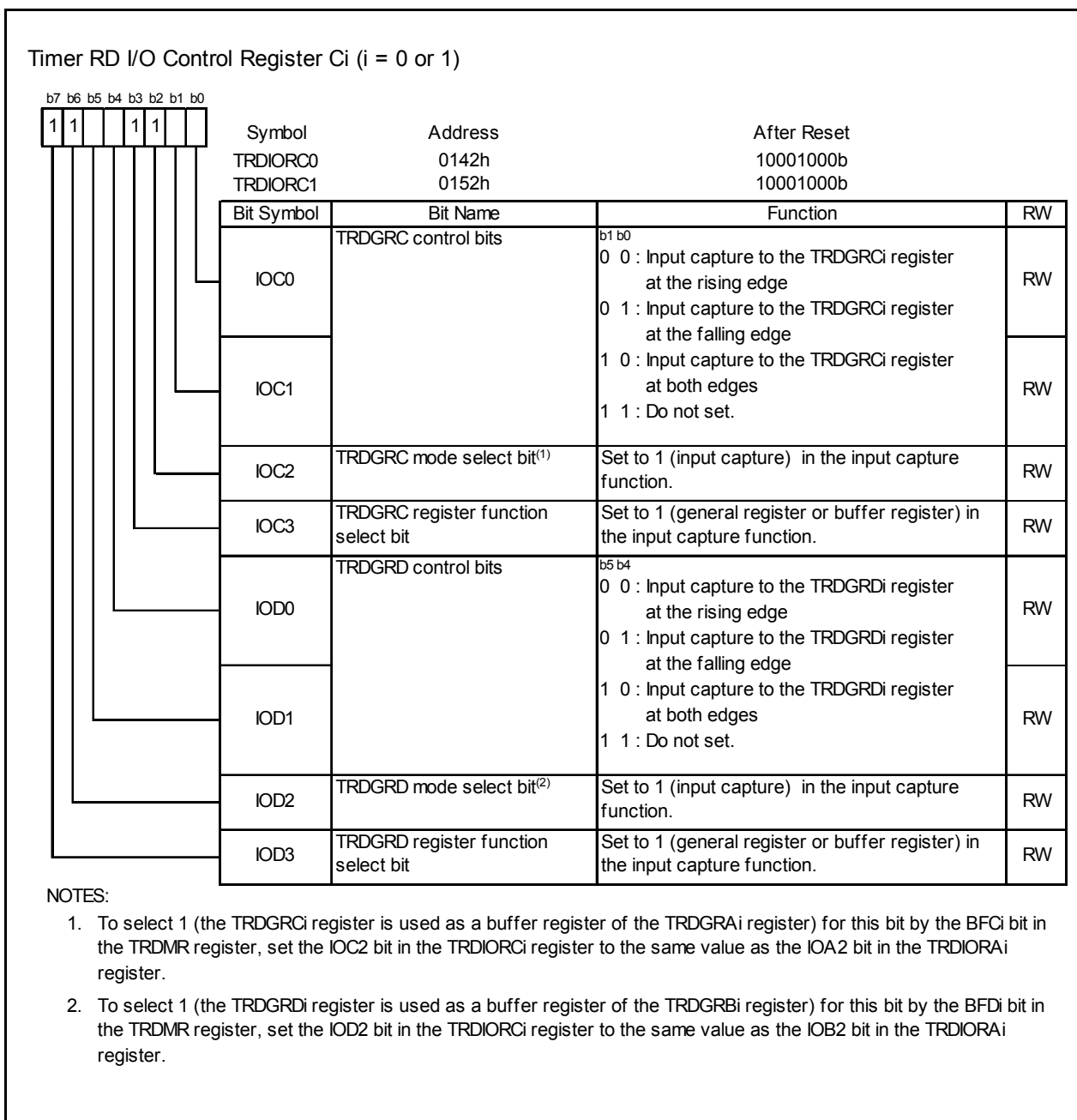


Figure 14.76 Registers TRDIORC0 to TRDIORC1 in Input Capture Function

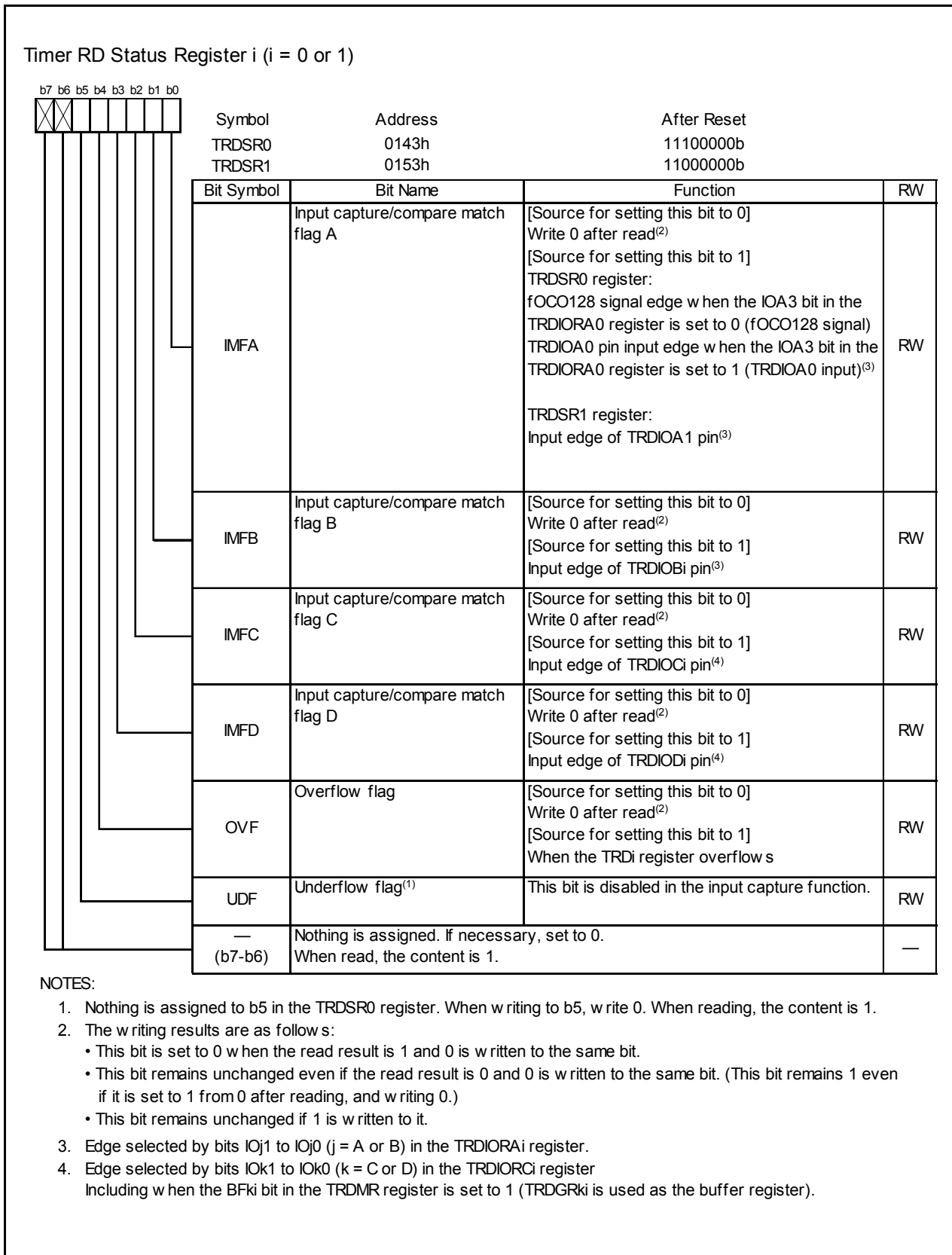


Figure 14.77 Registers TRDSR0 to TRDSR1 in Input Capture Function

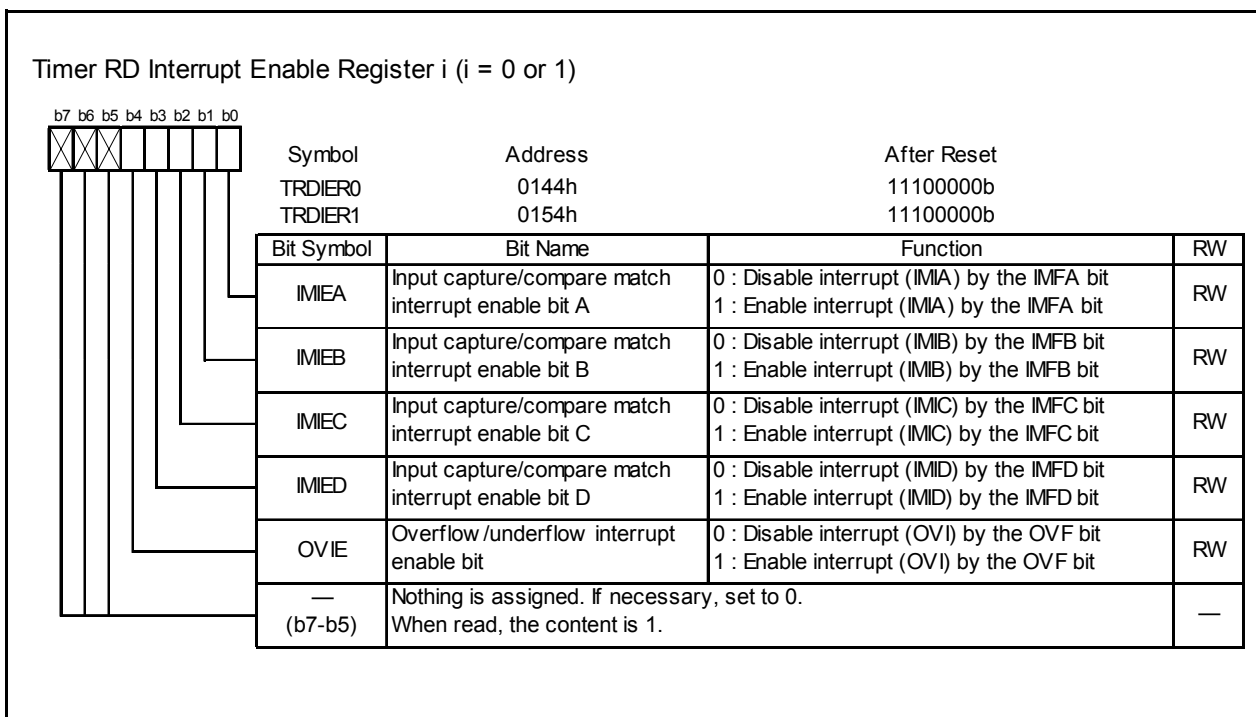


Figure 14.78 Registers TRDIER0 to TRDIER1 in Input Capture Function

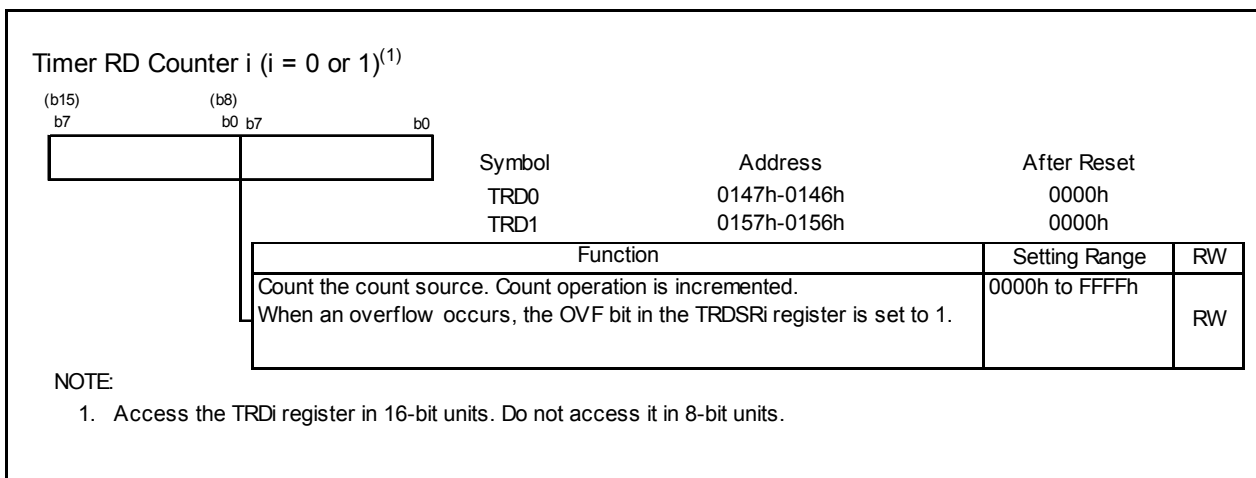


Figure 14.79 Registers TRD0 to TRD1 in Input Capture Function

| Timer RD General Registers Ai, Bi, Ci, and Di (i = 0 or 1) ⁽¹⁾ | | | | |
|---|------------|---------|-------------|-------------|
| (b15) b7 | (b8) b0 | Symbol | Address | After Reset |
| | | TRDGRA0 | 0149h-0148h | FFFFh |
| | | TRDGRB0 | 014Bh-014Ah | FFFFh |
| | | TRDGRC0 | 014Dh-014Ch | FFFFh |
| | | TRDGRD0 | 014Fh-014Eh | FFFFh |
| | | TRDGRA1 | 0159h-0158h | FFFFh |
| | | TRDGRB1 | 015Bh-015Ah | FFFFh |
| | | TRDGRC1 | 015Dh-015Ch | FFFFh |
| | | TRDGRD1 | 015Fh-015Eh | FFFFh |
| Function | | | | RW |
| Refer to Table 14.39 TRDGRji Register Functions in Input Capture Function | | | | RW |

NOTE:

- Access registers TRDGRAi to TRDGRDi in 16-bit units. Do not access them in 8-bit units.

Figure 14.80 Registers TRDGRAi, TRDGRBi, TRDGRCi, and TRDGRDi in Input Capture Function

The following registers are disabled in the input capture function: TRDOER1, TRDOER2, TRDOCR, TRDPOCR0, and TRDPOCR1.

Table 14.39 TRDGRji Register Functions in Input Capture Function

| Register | Setting | Register Function | Input-Capture Input Pin |
|----------|----------|--|-------------------------|
| TRDGRAi | – | General register The value in the TRDi register can be read at input capture. | TRDIOAi |
| TRDGRBi | | | TRDIOBi |
| TRDGRCi | BFCi = 0 | General register The value in the TRDi register can be read at input capture. | TRDIOCi |
| TRDGRDi | BFDi = 0 | | TRDIODi |
| TRDGRCi | BFCi = 1 | Buffer register The value in the TRDi register can be read at input capture. (Refer to 14.4.2 Buffer Operation) | TRDIOAi |
| TRDGRDi | BFDi = 1 | | TRDIOBi |

i = 0 or 1, j = either A, B, C, or D

BFCi, BFDi: Bits in TRDMR register

Set the pulse width of the input capture signal applied to the TRDIOji pin to 3 cycles or more of the timer RD operation clock (refer to **Table 14.26 Timer RD Operation Clocks**) for no digital filter (the DFj bit in the TRDDFi register set to 0).

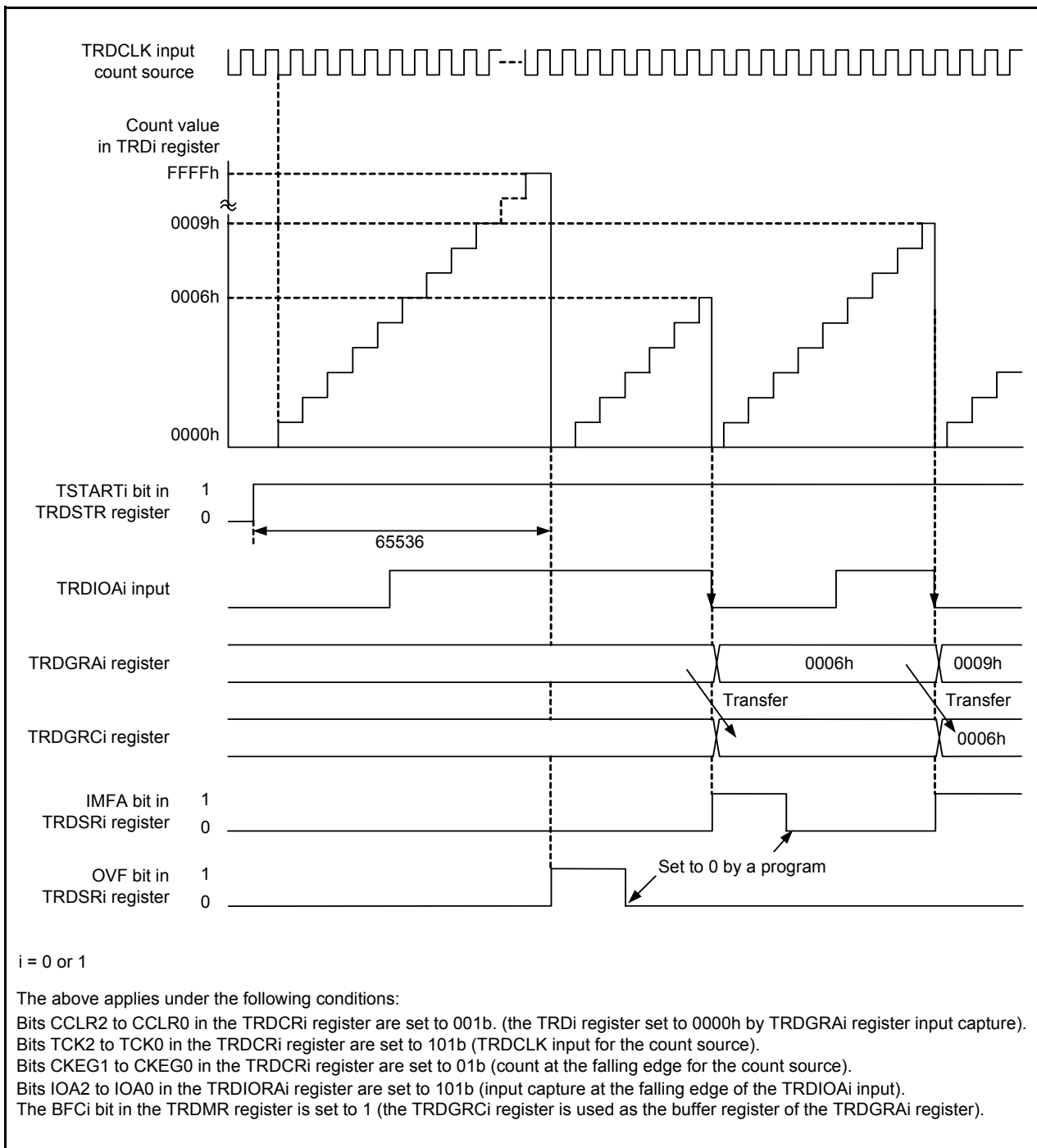


Figure 14.81 Operating Example of Input Capture Function

14.4.5.1 Digital Filter

The TRDIO_j input is sampled, and when the sampled input level matches 3 times, its level is determined. Select the digital filter function and sampling clock by the TRDDFi register.

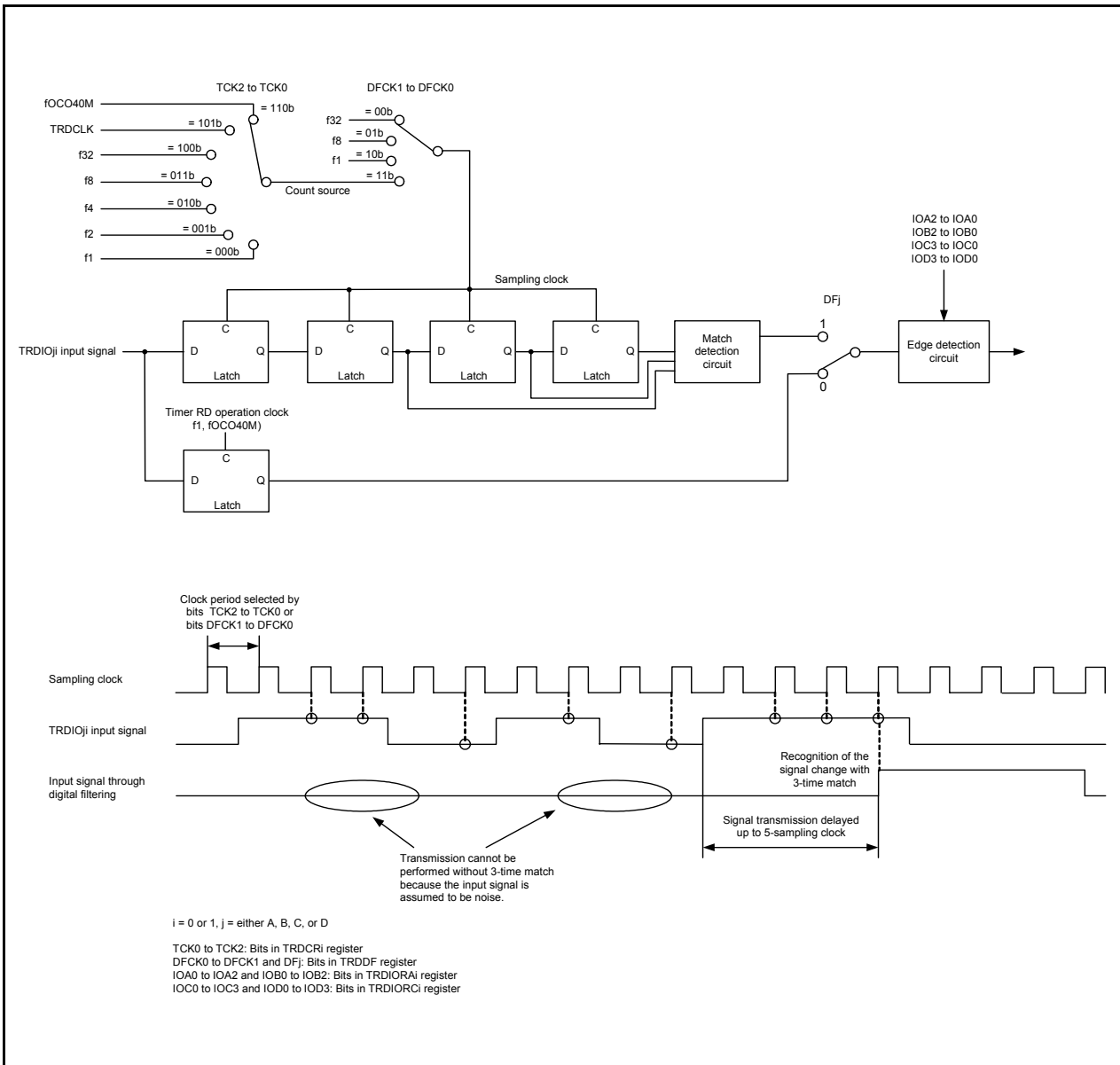


Figure 14.82 Block Diagram of Digital Filter

14.4.6 Output Compare Function

This function detects matches (compare match) between the content of the TRDGR_{ji} (j = either A, B, C, or D) register and the content of the TRD_i (i = 0 or 1) register. When the content matches, a user-set level is output from the TRDIO_{ji} pin. Since this function is enabled with a combination of the TRDIO_{ji} pin and TRDGR_{ji} register, the output compare function, or any other mode or function, can be selected for each individual pin. Figure 14.83 shows a Block Diagram of Output Compare Function, Table 14.40 lists the Output Compare Function Specifications. Figures 14.84 to 14.96 list the registers associated with output compare function, and Figure 14.97 shows an Operating Example of Output Compare Function.

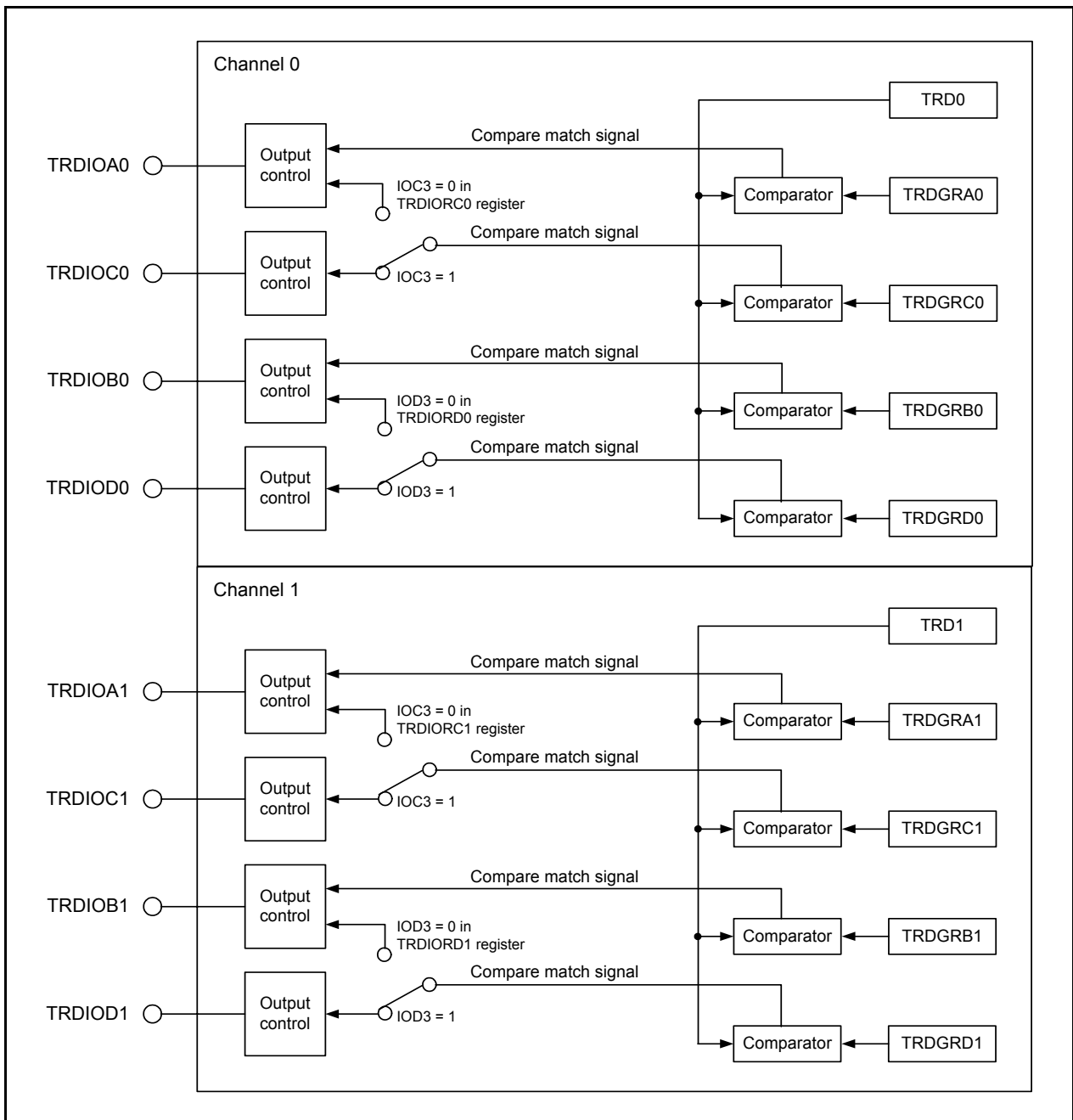


Figure 14.83 Block Diagram of Output Compare Function

Table 14.40 Output Compare Function Specifications

| Item | Specification |
|---|---|
| Count sources | f1, f2, f4, f8, f32, fOCO40M External signal input to the TRDCLK pin (valid edge selected by a program) |
| Count operations | Increment |
| Count period | <ul style="list-style-type: none"> When bits CCLR2 to CCLR0 in the TRDCRi register are set to 000b (free-running operation) $1/fk \times 65536$ fk: Frequency of count source Bits CCLR1 to CCLR0 in the TRDCRi register are set to 01b or 10b (set the TRDi register to 0000h at the compare match in the TRDGRji register). Frequency of count source x (n+1) n: Setting value in the TRDGRji register |
| Waveform output timing | Compare match |
| Count start condition | 1 (count starts) is written to the TSTARTi bit in the TRDSTR register. |
| Count stop conditions | <ul style="list-style-type: none"> 0 (count stops) is written to the TSTARTi bit in the TRDSTR register when the CSELi bit in the TRDSTR register is set to 1. The output compare output pin holds output level before the count stops. When the CSELi bit in the TRDSTR register is set to 0, the count stops at the compare match in the TRDGRAi register. The output compare output pin holds level after output change by the compare match. |
| Interrupt request generation timing | <ul style="list-style-type: none"> Compare match (content of the TRDi register matches content of the TRDGRji register.) TRDi register overflows |
| TRDIOA0 pin function | Programmable I/O port, output-compare output, or TRDCLK (external clock) input |
| TRDIOB0, TRDIOC0, TRDIOD0, TRDIOA1 to TRDIOD1 pin functions | Programmable I/O port or output-compare output (Selectable by pin) |
| $\overline{\text{INT0}}$ pin function | Programmable I/O port, pulse output forced cutoff signal input, or $\overline{\text{INT0}}$ interrupt input |
| Read from timer | The count value can be read by reading the TRDi register. |
| Write to timer | <ul style="list-style-type: none"> When the SYNC bit in the TRDMR register is set to 0 (channels 0 and 1 operate independently). Data can be written to the TRDi register. When the SYNC bit in the TRDMR register is set to 1 (channels 0 and 1 operate synchronously). Data can be written to both the TRD0 and TRD1 registers by writing to the TRDi register. |
| Select functions | <ul style="list-style-type: none"> Output-compare output pin selected Either 1 pin or multiple pins among TRDIOAi, TRDIOBi, TRDIOCi, or TRDIODi. Output level at the compare match selected "L" output, "H" output, or output level inversed Initial output level selected Set the level at period from the count start to the compare match. Timing to set the TRDi register to 0000h Overflow or compare match in the TRDGRAi register Buffer operation (Refer to 14.4.2 Buffer Operation.) Synchronous operation (Refer to 14.4.3 Synchronous Operation.) Output pin in registers TRDGRCi and TRDGRDi changed The TRDGRCi register can be used as output control of the TRDIOAi pin and the TRDGRDi register can be used as output control of the TRDIOBi pin. Pulse output forced cutoff signal input (Refer to 14.4.4 Pulse Output Forced Cutoff.) Timer RD can be used as the internal timer without output. |

i = 0 or 1, j = either A, B, C, or D

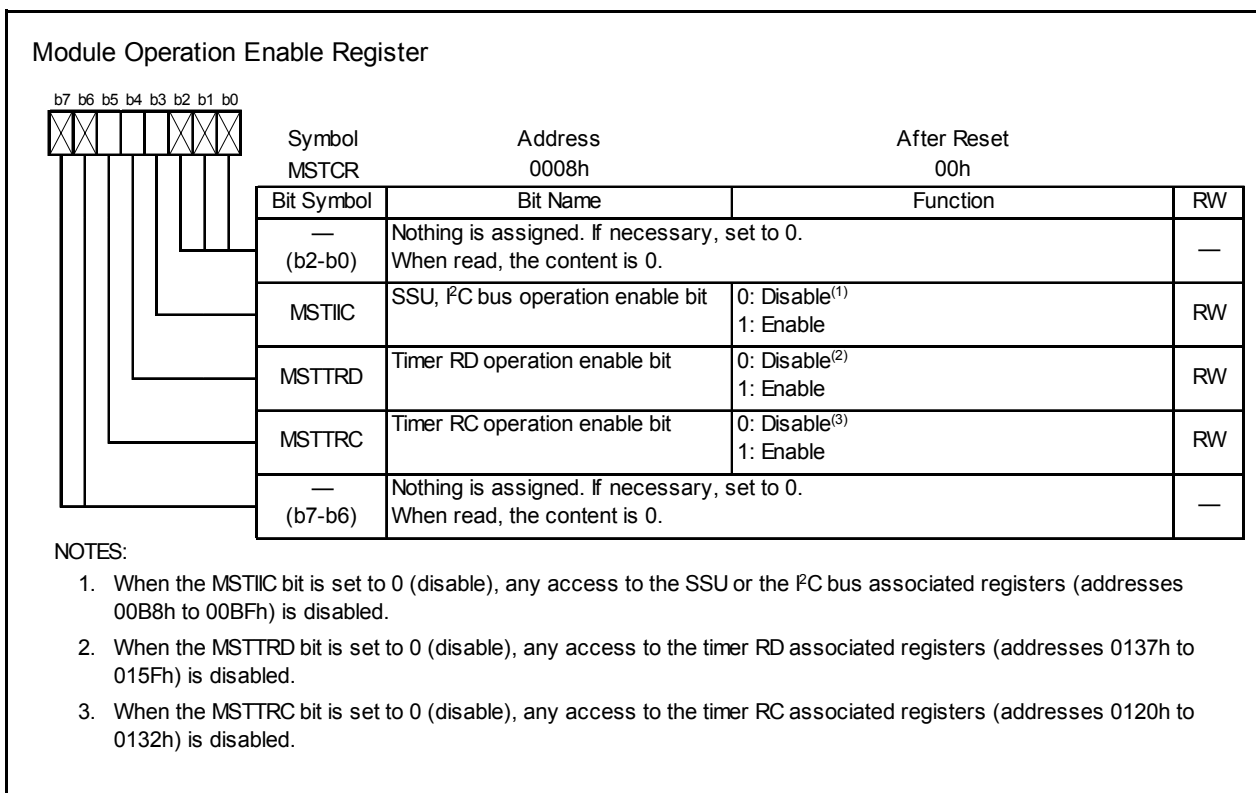


Figure 14.84 MSTCR Register

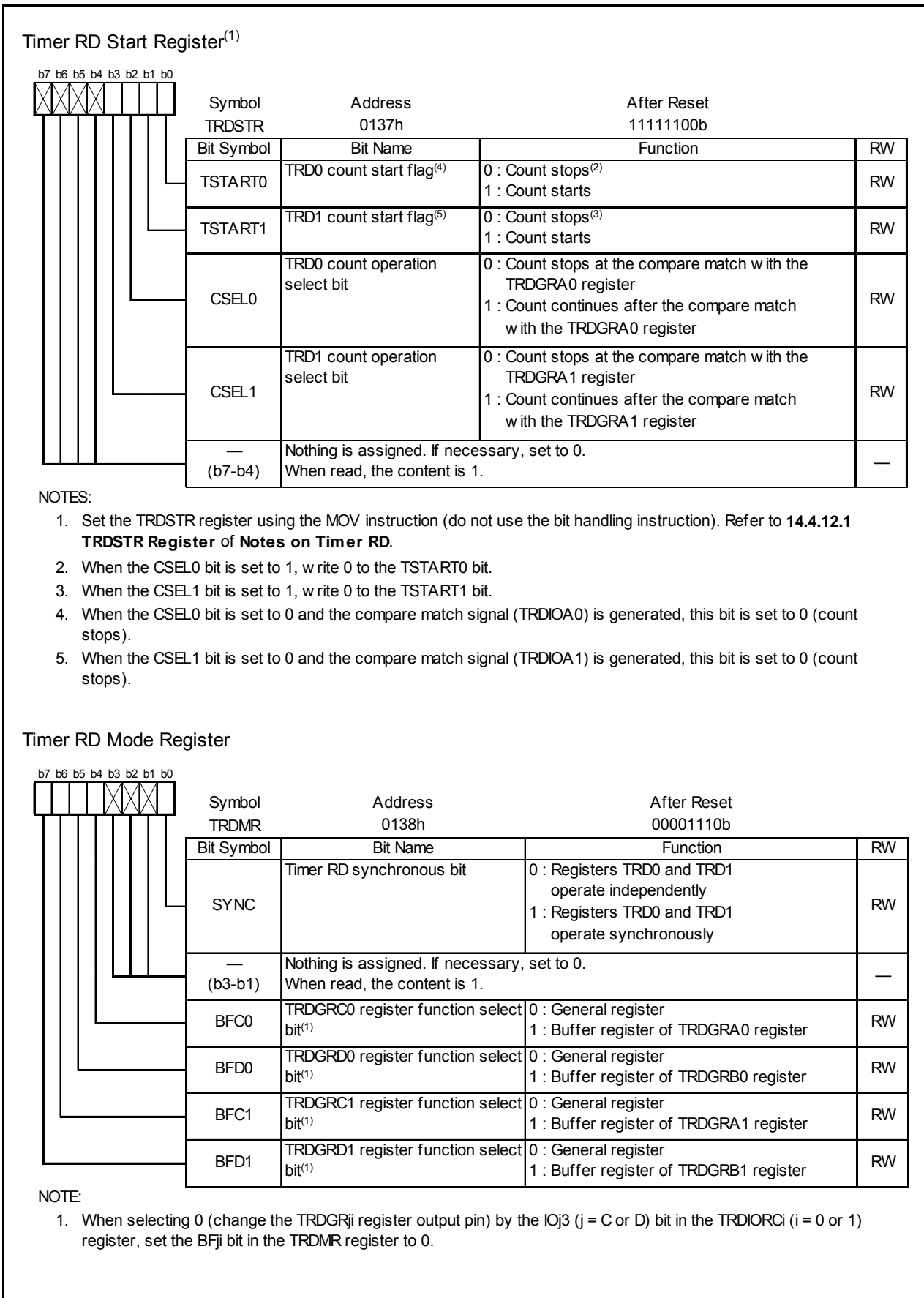


Figure 14.85 Registers TRDSTR and TRDMR in Output Compare Function

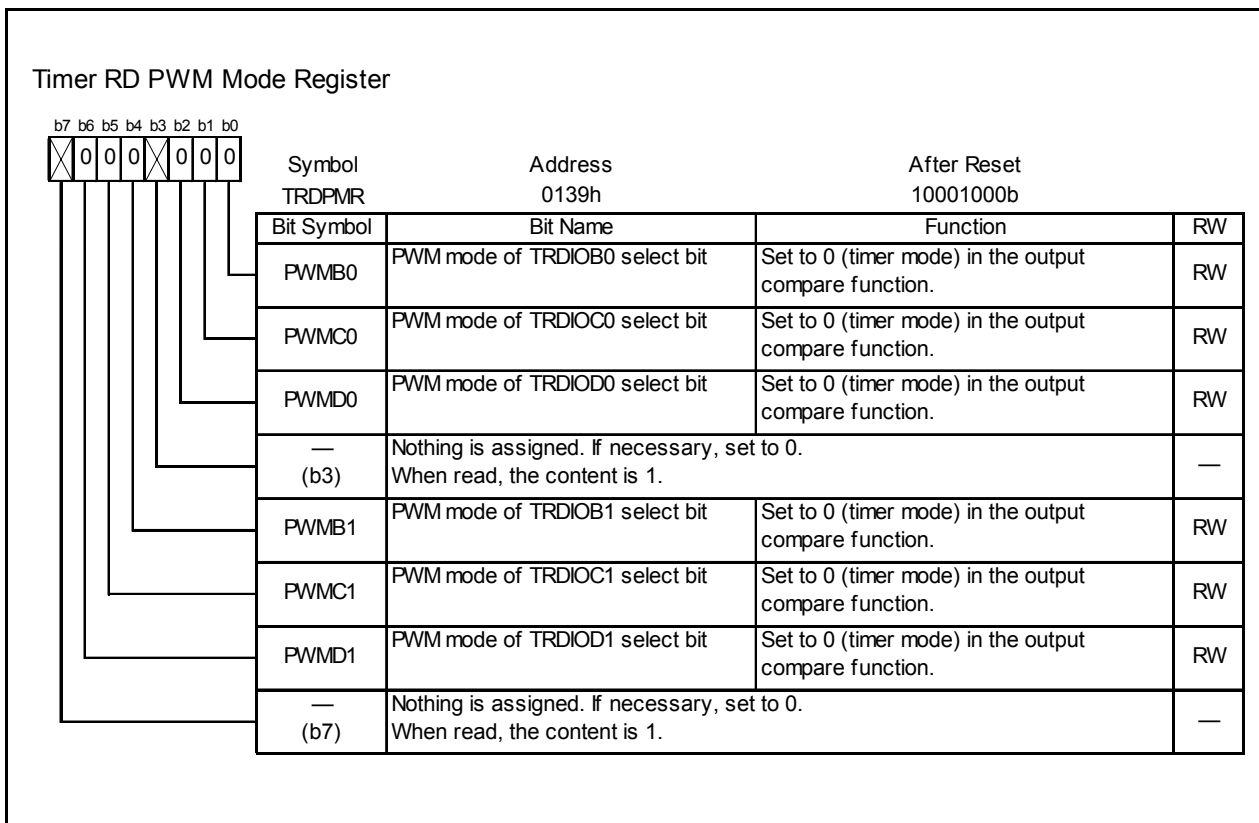


Figure 14.86 TRDPMR Register in Output Compare Function

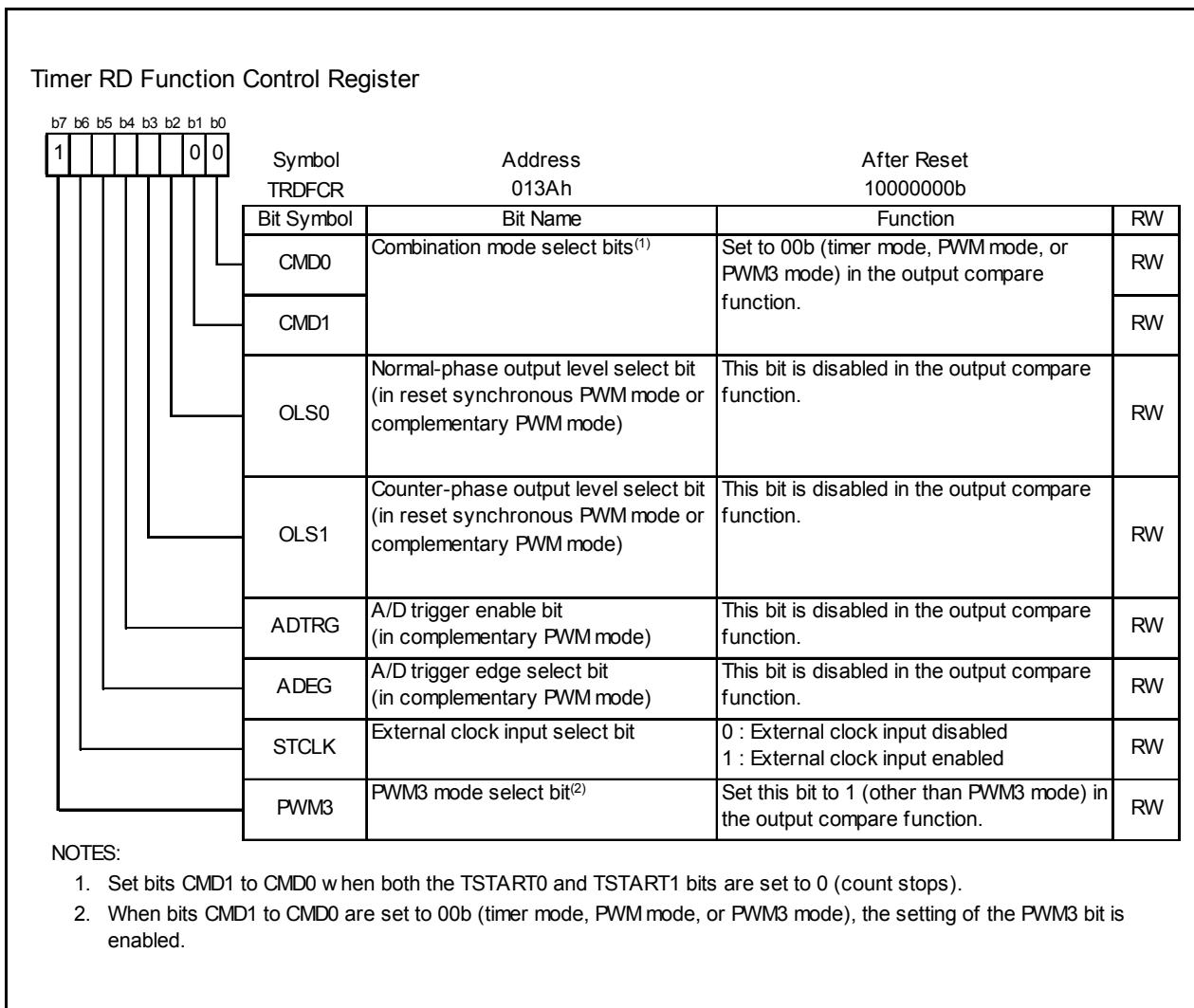


Figure 14.87 TRDFCR Register in Output Compare Function

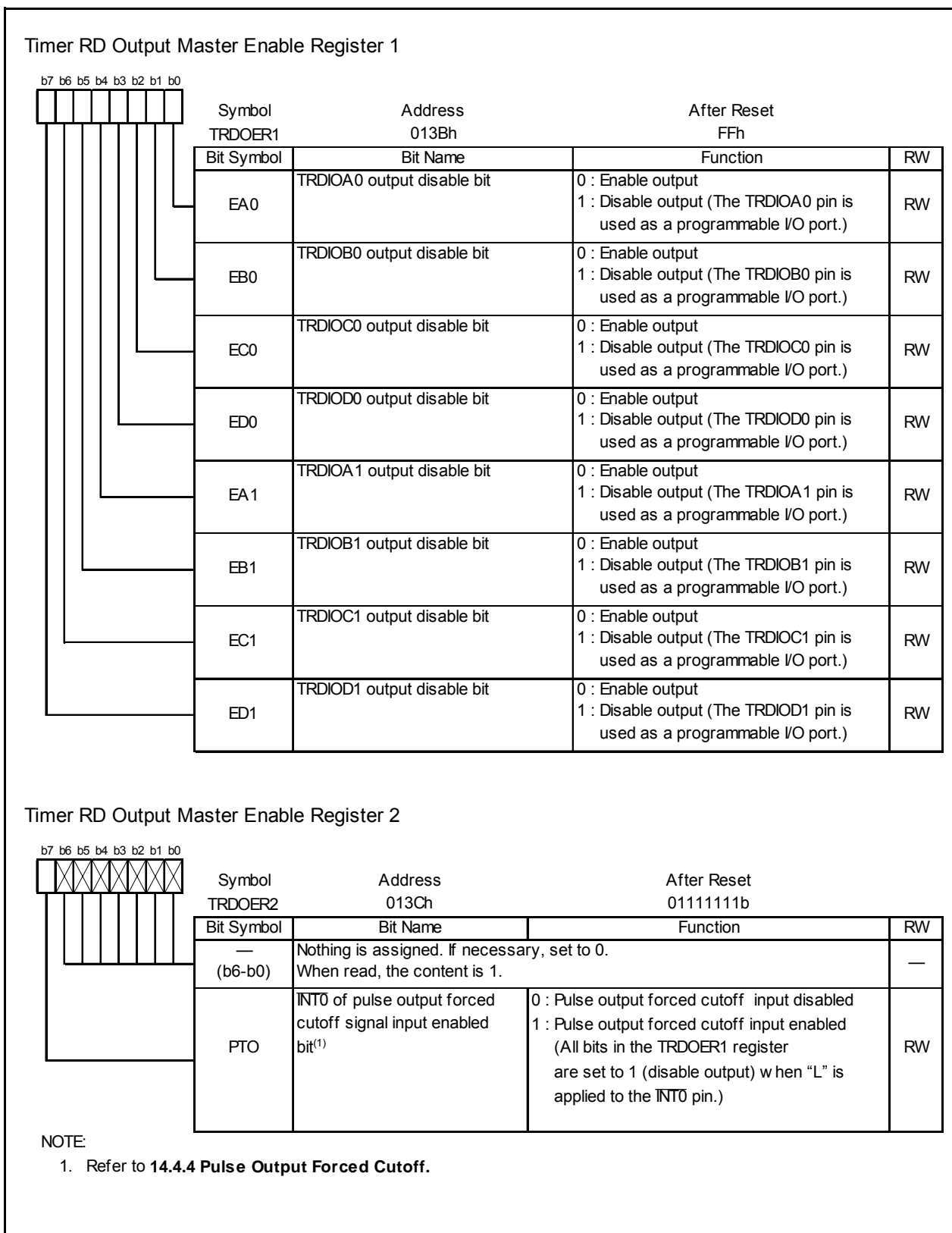


Figure 14.88 Registers TRDOER1 to TRDOER2 in Output Compare Function

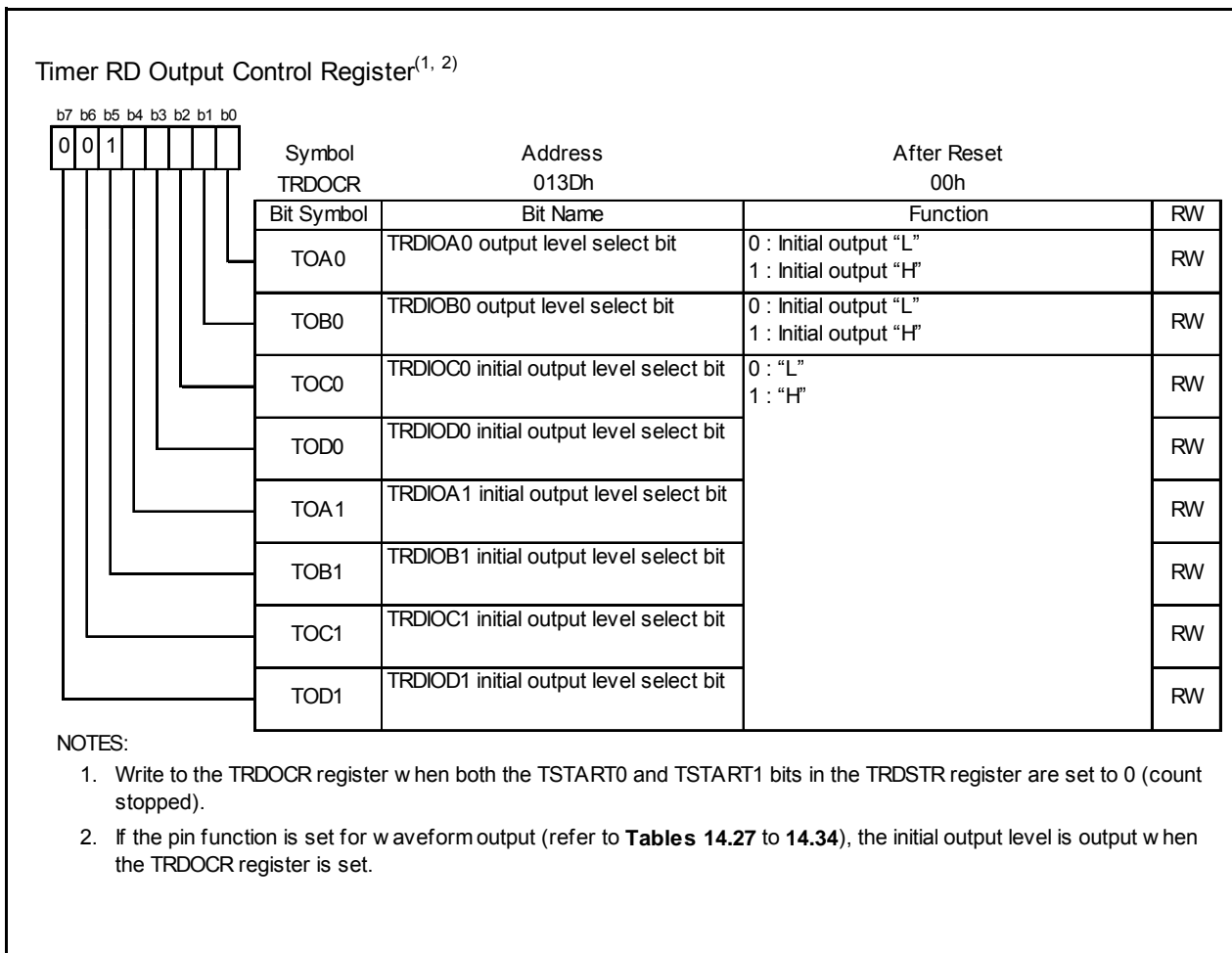


Figure 14.89 TRDOCR Register in Output Compare Function

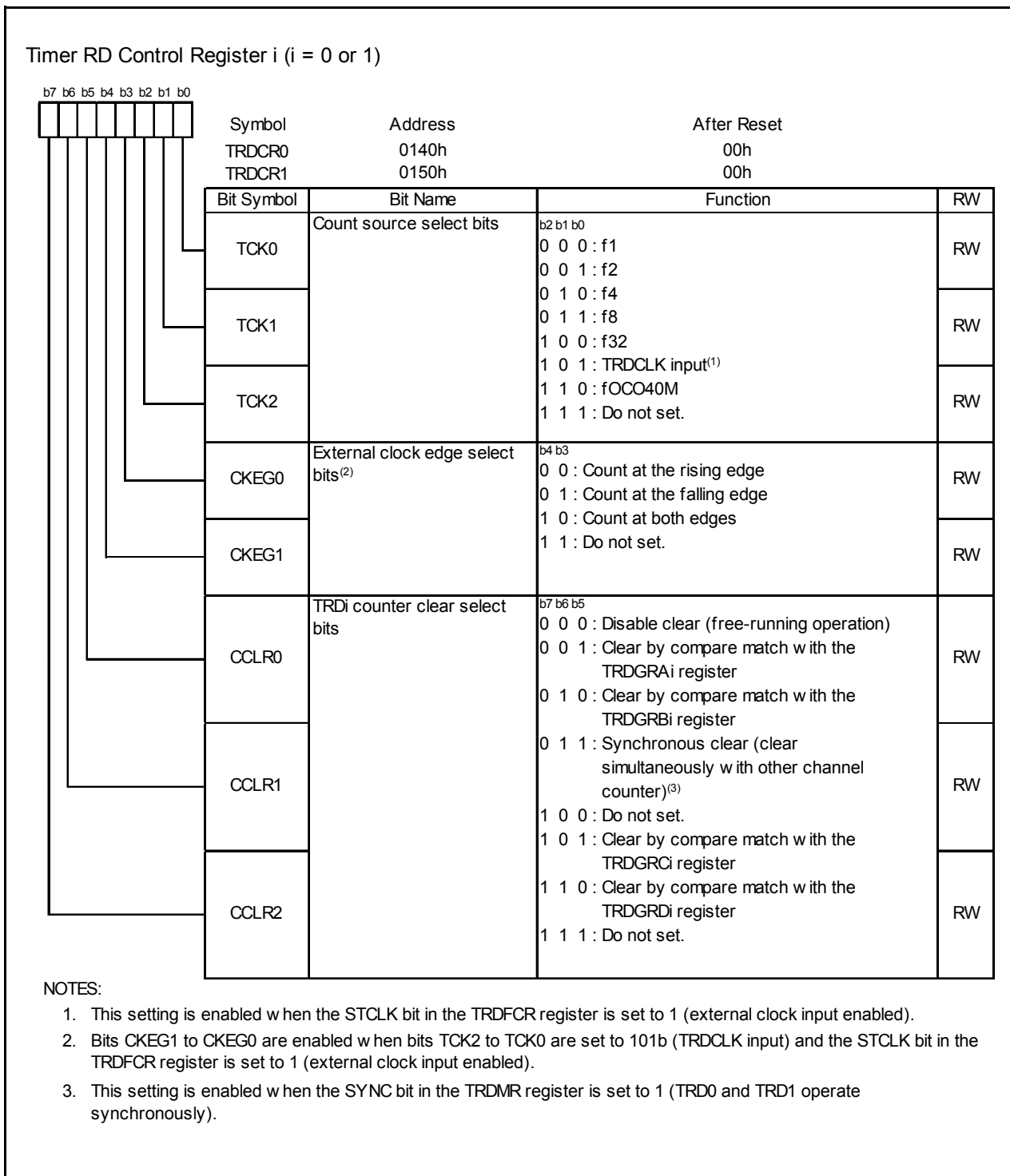


Figure 14.90 Registers TRDCR0 to TRDCR1 in Output Compare Function

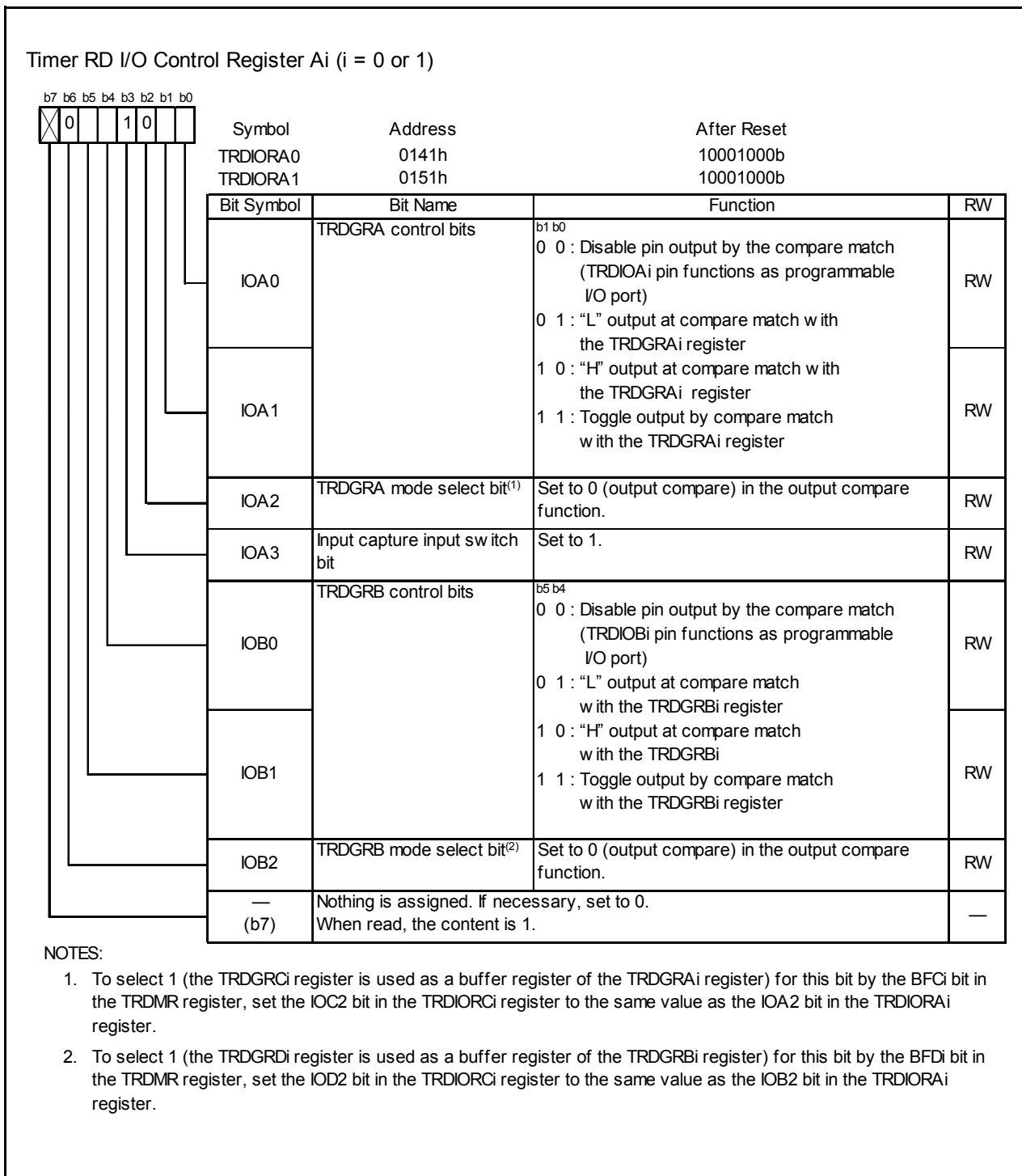


Figure 14.91 Registers TRDIORA0 to TRDIORA1 in Output Compare Function

Timer RD I/O Control Register Ci (i = 0 or 1)

| Bit | Symbol | Address | After Reset |
|-----|----------|---------|-------------|
| b7 | 0 | | |
| b6 | | | |
| b5 | | | |
| b4 | | | |
| b3 | | | |
| b2 | 0 | | |
| b1 | | | |
| b0 | | | |
| | Symbol | Address | After Reset |
| | TRDIORC0 | 0142h | 10001000b |
| | TRDIORC1 | 0152h | 10001000b |

| Bit Symbol | Bit Name | Function | RW |
|------------|---------------------------------------|--|----|
| IOC0 | TRDGRC control bits | b1 b0 0 0 : Disable pin output by compare match 0 1 : "L" output at compare match with the TRDGRCi register 1 0 : "H" output at compare match with the TRDGRCi register 1 1 : Toggle output by compare match with the TRDGRCi register | RW |
| | | | RW |
| IOC2 | TRDGRC mode select bit ⁽¹⁾ | Set to 0 (output compare) in the output compare function. | RW |
| IOC3 | TRDGRC register function select bit | 0 : TRDIOA output register (Refer to 14.4.6.1 Changing Output Pins in Registers TRDGRCi (i = 0 or 1) and TRDGRDi.) 1 : General register or buffer register | RW |
| IOD0 | TRDGRD control bits | b5 b4 0 0 : Disable pin output by compare match 0 1 : "L" output at compare match with the TRDGRDi register 1 0 : "H" output at compare match with the TRDGRDi register 1 1 : Toggle output by compare match with the TRDGRDi register | RW |
| | | | RW |
| IOD2 | TRDGRD mode select bit ⁽²⁾ | Set to 0 (output compare) in the output compare function. | RW |
| IOD3 | TRDGRD register function select bit | 0 : TRDIOB output register (Refer to 14.4.6.1 Changing Output Pins in Registers TRDGRCi (i = 0 or 1) and TRDGRDi.) 1 : General register or buffer register | RW |

NOTES:

- To select 1 (the TRDGRCi register is used as a buffer register of the TRDGRAi register) for this bit by the BFCi bit in the TRDMR register, set the IOC2 bit in the TRDIORCi register to the same value as the IOA2 bit in the TRDIORAi register.
- To select 1 (the TRDGRDi register is used as a buffer register of the TRDGRBi register) for this bit by the BFDi bit in the TRDMR register, set the IOD2 bit in the TRDIORCi register to the same value as the IOB2 bit in the TRDIORAi register.

Figure 14.92 Registers TRDIORC0 to TRDIORC1 in Output Compare Function

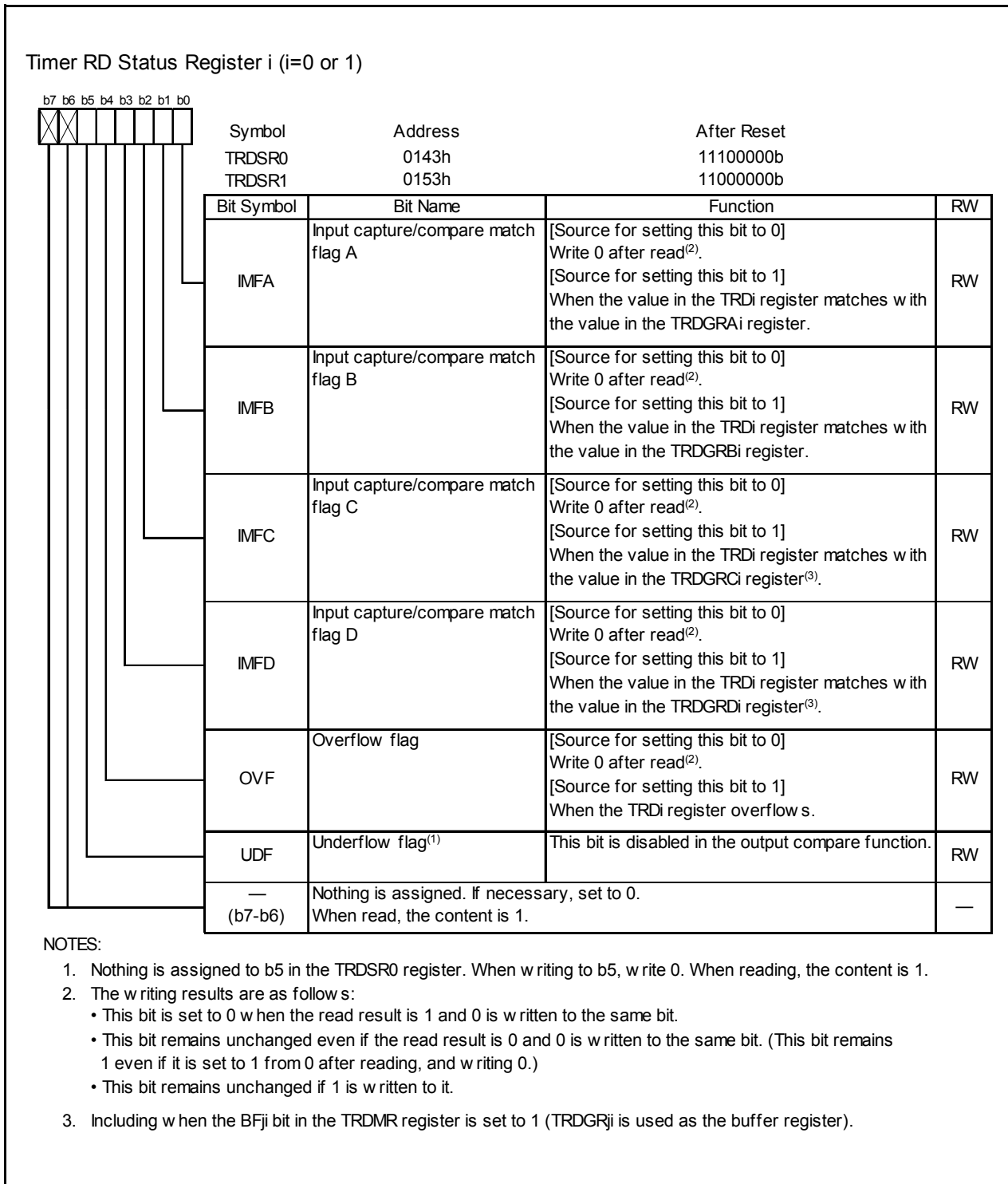


Figure 14.93 Registers TRDSR0 to TRDSR1 in Output Compare Function

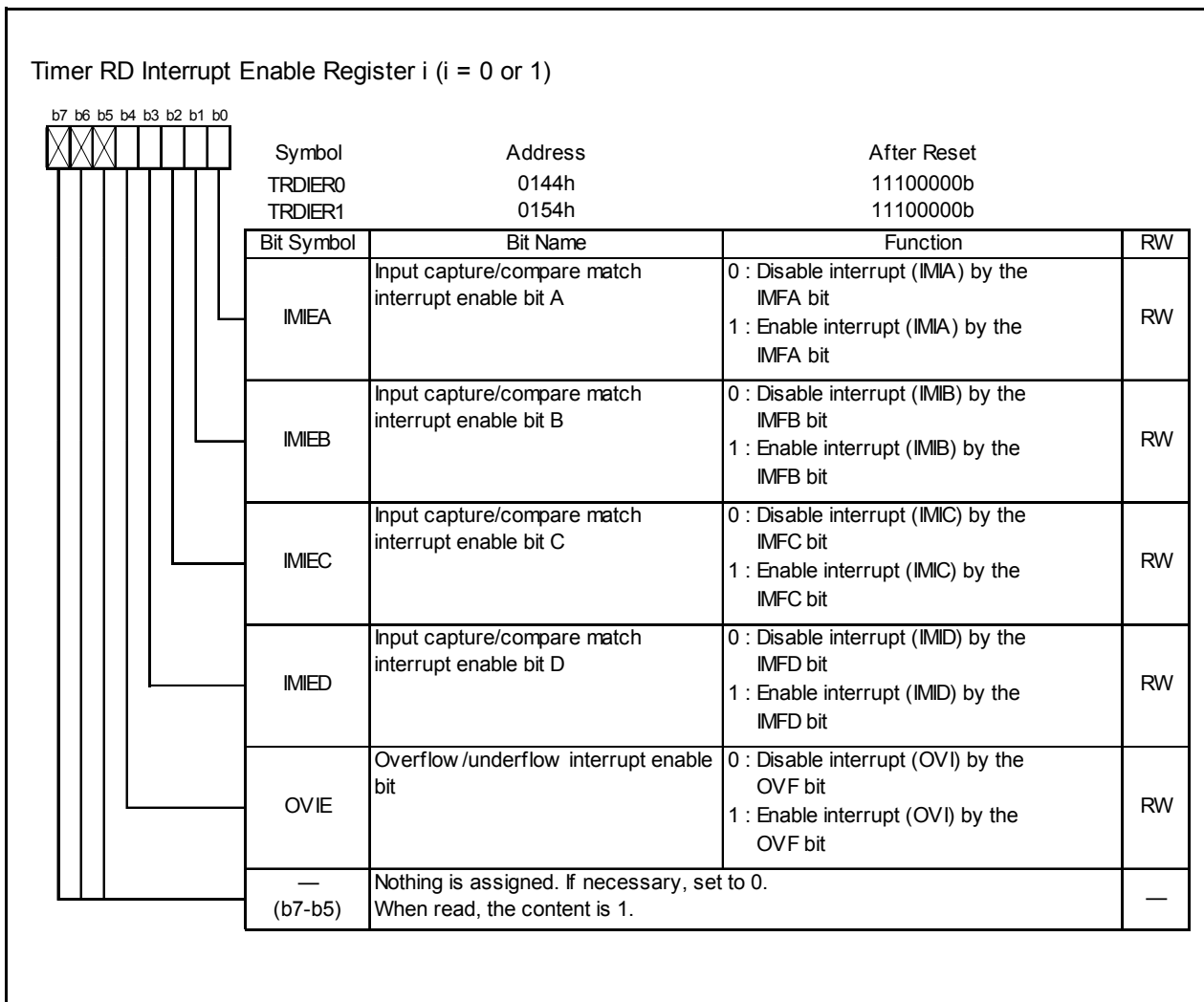


Figure 14.94 Registers TRDIER0 to TRDIER1 in Output Compare Function

| Timer RD Counter i ($i = 0$ or 1) ⁽¹⁾ | | Symbol | Address | After Reset | |
|---|------------|--------|-------------|----------------|----|
| (b15) b7 | (b8) b0 | | | | |
| | | TRD0 | 0147h-0146h | 0000h | |
| | | TRD1 | 0157h-0156h | 0000h | |
| Function | | | | Setting Range | RW |
| Count a count source. Count operation is incremented. | | | | 0000h to FFFFh | RW |
| When an overflow occurs, the OVF bit in the TRDSR i register is set to 1. | | | | | RW |

NOTE:

- Access the TRD i register in 16-bit units. Do not access it in 8-bit units.

Figure 14.95 Registers TRD0 to TRD1 in Output Compare Function

| Timer RD General Register A_i , B_i , C_i and D_i ($i = 0$ or 1) ⁽¹⁾ | | Symbol | Address | After Reset |
|---|------------|---------|-------------|-------------|
| (b15) b7 | (b8) b0 | | | |
| | | TRDGRA0 | 0149h-0148h | FFFFh |
| | | TRDGRB0 | 014Bh-014Ah | FFFFh |
| | | TRDGRC0 | 014Dh-014Ch | FFFFh |
| | | TRDGRD0 | 014Fh-014Eh | FFFFh |
| | | TRDGRA1 | 0159h-0158h | FFFFh |
| | | TRDGRB1 | 015Bh-015Ah | FFFFh |
| | | TRDGRC1 | 015Dh-015Ch | FFFFh |
| | | TRDGRD1 | 015Fh-015Eh | FFFFh |
| Function | | | | RW |
| Refer to Table 14.41 TRDGR j_i Register Function in Output Compare Function | | | | RW |

NOTE:

- Access registers TRDGRA i to TRDGRD i in 16-bit units. Do not access them in 8-bit units.

Figure 14.96 Registers TRDGRA i , TRDGRB i , TRDGRC i , and TRDGRD i in Output Compare Function

The following registers are disabled in the output compare function: TRDDF0, TRDDF1, TRDPOCR0, and TRDPOCR1.

Table 14.41 TRDGR j_i Register Function in Output Compare Function

| Register | Setting | | Register Function | Output-Compare Output Pin |
|------------|----------|----------|--|---------------------------|
| | BF j_i | IO j_3 | | |
| TRDGRA i | – | – | General register. Write the compare value. | TRDIOA i |
| TRDGRB i | | | | TRDIOB i |
| TRDGRC i | 0 | 1 | General register. Write the compare value. | TRDIOC i |
| TRDGRD i | | | | TRDIOD i |
| TRDGRC i | 1 | 1 | Buffer register. Write the next compare value (Refer to 14.4.2 Buffer Operation.) | TRDIOA i |
| TRDGRD i | | | | TRDIOB i |
| TRDGRC i | 0 | 0 | TRDIOA i output control (Refer to 14.4.6.1 Changing Output Pins in Registers TRDGRC i ($i = 0$ or 1) and TRDGRD i .) | TRDIOA i |
| TRDGRD i | | | | TRDIOB i |

$i = 0$ or 1 , $j =$ either A, B, C, or D

BF j_i : Bit in TRDMR register

IO j_3 : Bit in TRDIORC i register

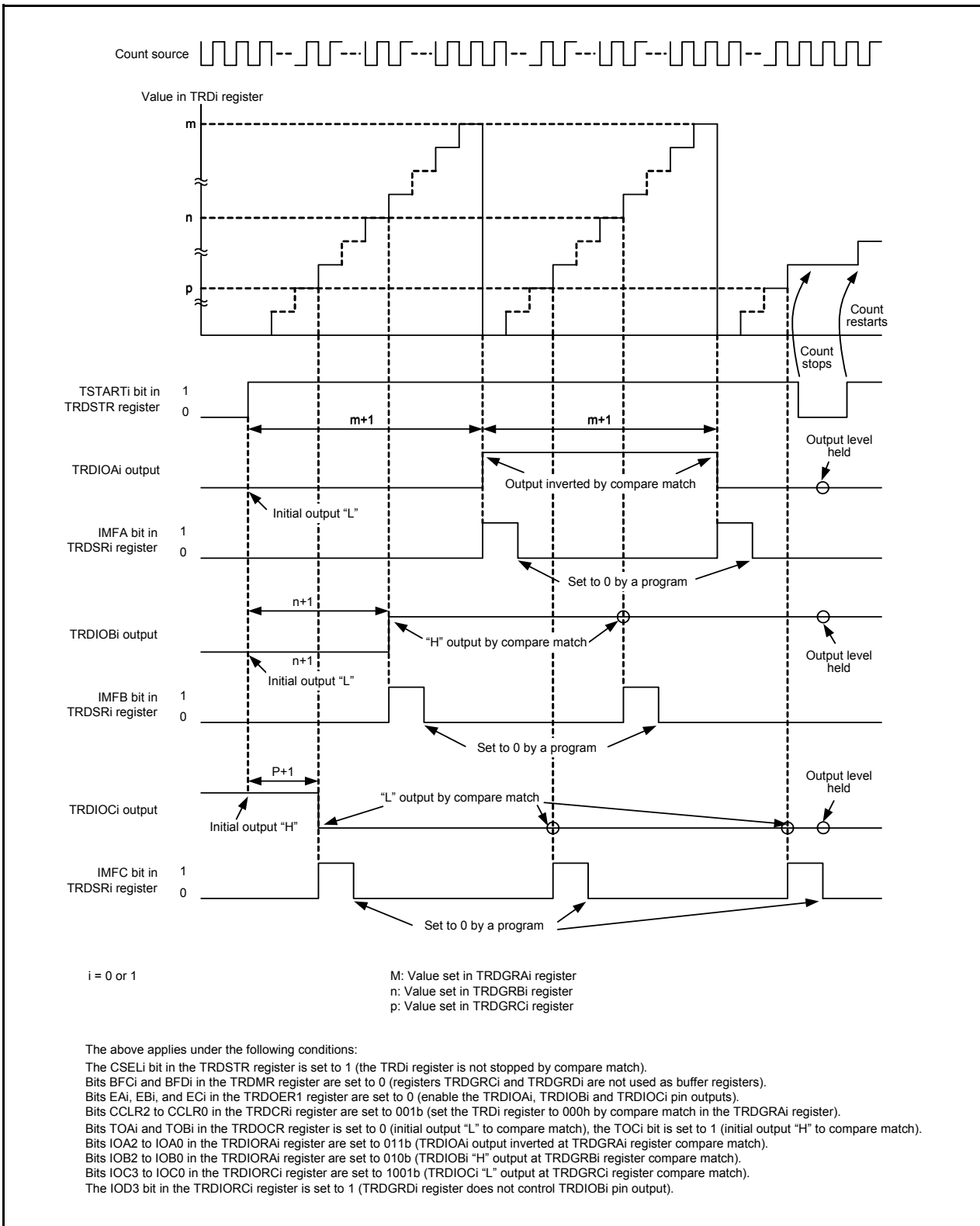


Figure 14.97 Operating Example of Output Compare Function

14.4.6.1 Changing Output Pins in Registers TRDGRCi (i = 0 or 1) and TRDGRDi

The TRDGRCi register can be used for output control of the TRDIOAi pin, and the TRDGRDi register can be used for output control of the TRDIOBi pin. Therefore, each pin output can be controlled as follows:

- TRDIOAi output is controlled by the values in registers TRDGRAi and TRDGRCi.
- TRDIOBi output is controlled by the values in registers TRDGRBi and TRDGRDi.

Change output pins in registers TRDGRCi and TRDGRDi as follows:

- Select 0 (change TRDGRji register output pin) by the IOj3 (j = C or D) bit in the TRDIORCi register.
- Set the BFji bit in the TRDMR register to 0 (general register).
- Set different values in registers TRDGRCi and TRDGRAi. Also, set different values in registers TRDGRDi and TRDGRBi.

Figure 14.99 shows an Operating Example When TRDGRCi Register is Used for Output Control of TRDIOAi Pin and TRDGRDi Register is Used for Output Control of TRDIOBi Pin.

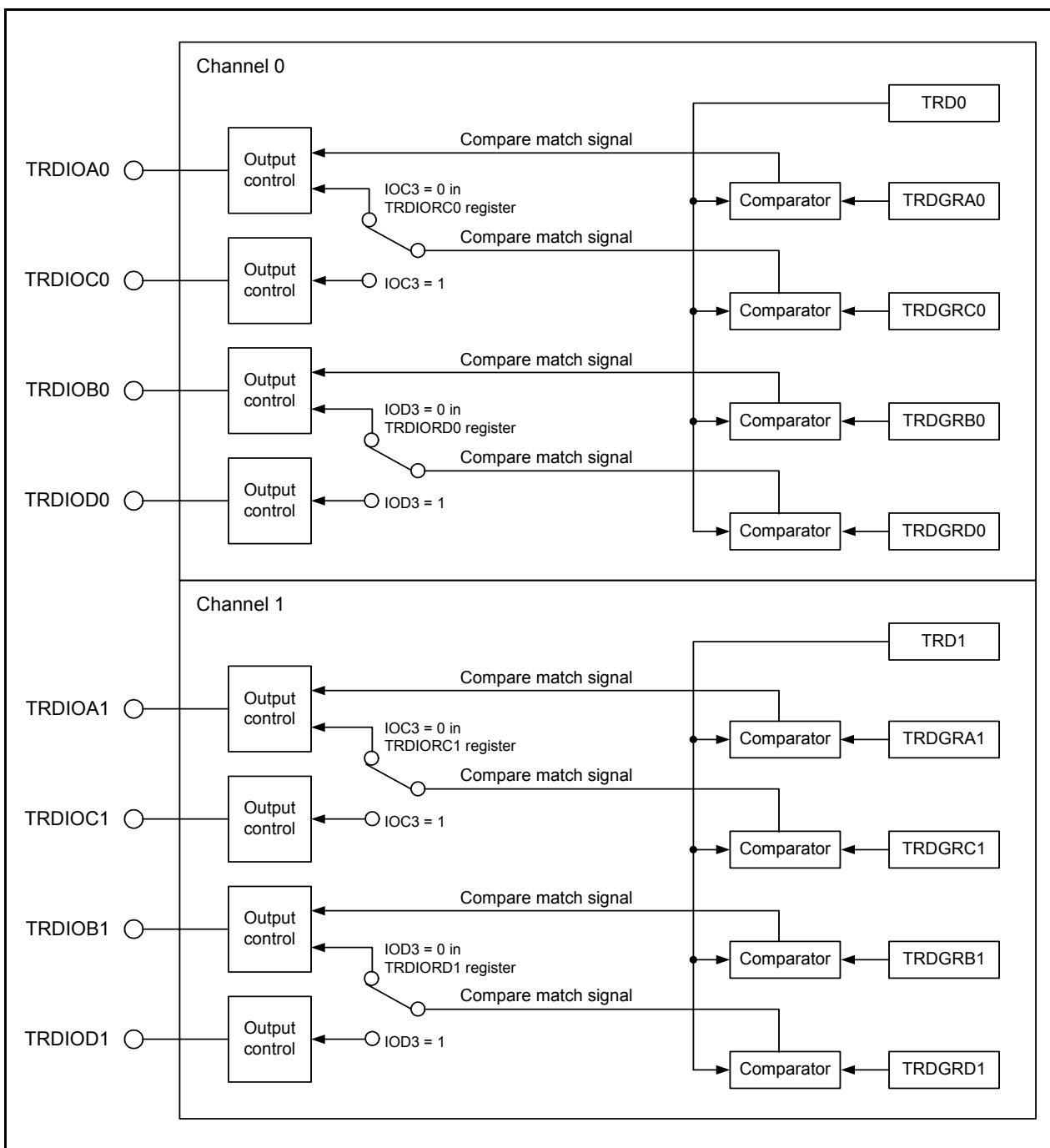


Figure 14.98 Changing Output Pins in Registers TRDGRCi and TRDGRDi

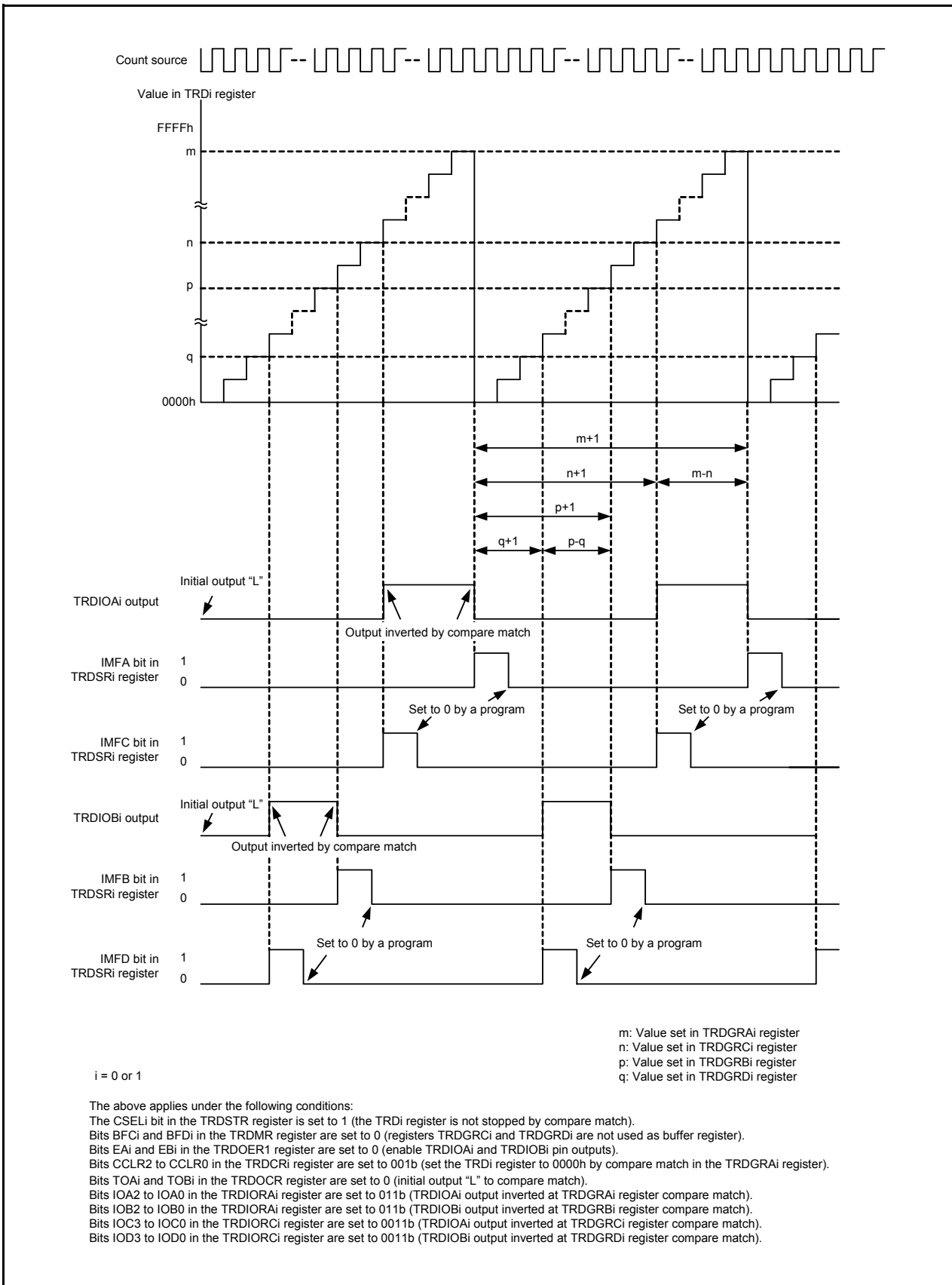


Figure 14.99 Operating Example When TRDGRGi Register is Used for Output Control of TRDIOAi Pin and TRDGRDi Register is Used for Output Control of TRDIOBi Pin

14.4.7 PWM Mode

In PWM mode, a PWM waveform is output. Up to 3 PWM waveforms with the same period can be output by 1 channel. Also, up to 6 PWM waveforms with the same period can be output by synchronizing channels 0 and 1. Since this mode functions by a combination of the TRDIO_{ji} ($i = 0$ or 1 , $j = B, C,$ or D) pin and TRDGR_{ji} register, the PWM mode, or any other mode or function, can be selected for each individual pin. (However, since the TRDGRA_i register is used when using any pin for PWM mode, the TRDGRA_i register cannot be used for other modes.)

Figure 14.100 shows a Block Diagram of PWM Mode, and Table 14.42 lists the PWM Mode Specifications. Figures 14.101 to 14.111 show the registers associated with PWM mode, and Figures 14.112 and 14.113 show the Operations of PWM Mode.

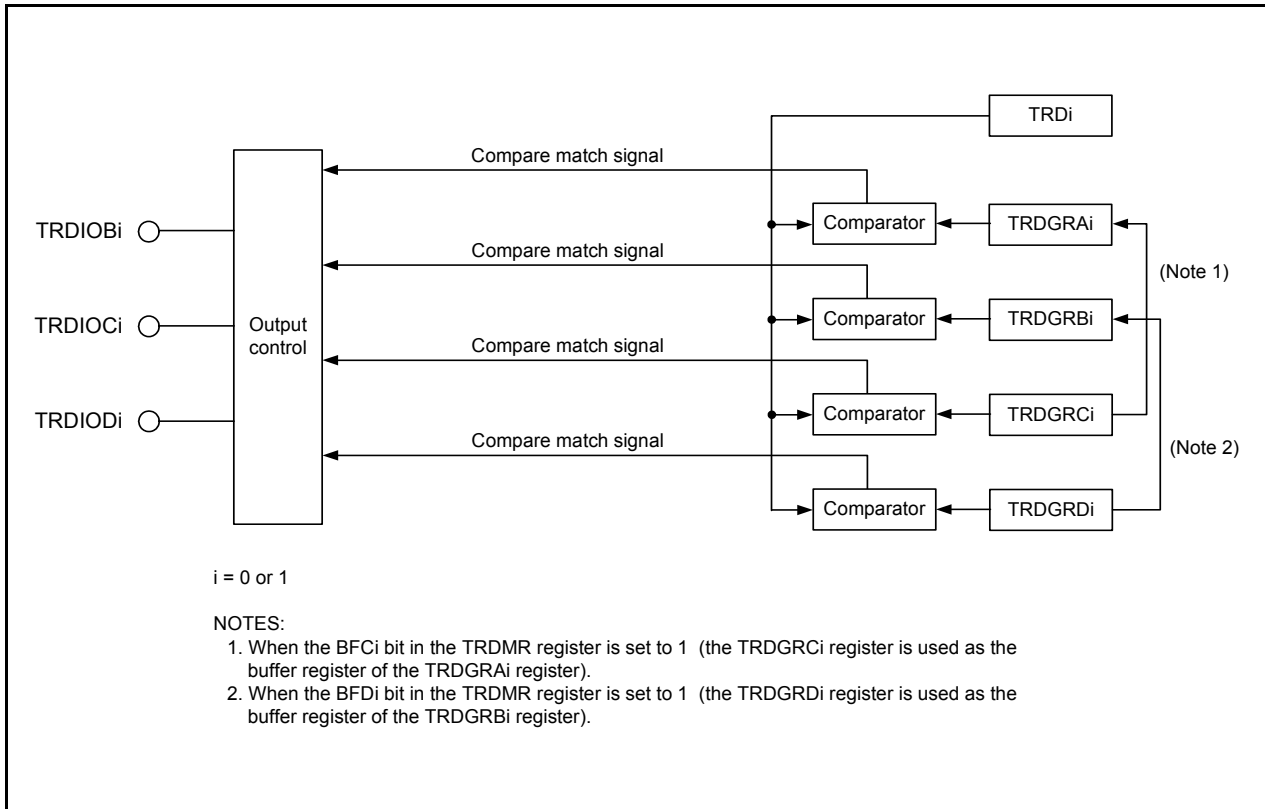
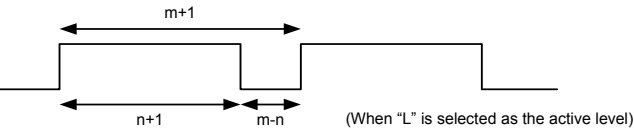


Figure 14.100 Block Diagram of PWM Mode

Table 14.42 PWM Mode Specifications

| Item | Specification |
|--|--|
| Count sources | f1, f2, f4, f8, f32, fOCO40M External signal input to the TRDCLK pin (valid edge selected by a program) |
| Count operations | Increment |
| PWM waveform | <p>PWM period: $1/f_k \times (m+1)$ Active level width: $1/f_k \times (m-n)$ Inactive level width: $1/f_k \times (n+1)$ f_k: Frequency of count source m: Value set in the TRDGRA$_i$ ($i = 0$ or 1) register n: Value set in the TRDGR$_{ji}$ ($j = B, C, \text{ or } D$) register</p>  <p style="text-align: center;">(When "L" is selected as the active level)</p> |
| Count start condition | 1 (count starts) is written to the TSTART $_i$ bit in the TRDSTR register. |
| Count stop conditions | <ul style="list-style-type: none"> • 0 (count stops) is written to the TSTART$_i$ bit in the TRDSTR register when the CSEL$_i$ bit in the TRDSTR register is set to 1. The PWM output pin holds output level before the count stops. • When the CSEL$_i$ bit in the TRDSTR register is set to 0, the count stops at the compare match in the TRDGRA$_i$ register. The PWM output pin holds level after output change by compare match. |
| Interrupt request generation timing | <ul style="list-style-type: none"> • Compare match (The content of the TRD$_i$ register matches content of the TRDGR$_{ji}$ register.) • TRD$_i$ register overflows |
| TRDIOA0 pin function | Programmable I/O port or TRDCLK (external clock) input |
| TRDIOA1 pin function | Programmable I/O port |
| TRDIOB0, TRDIOC0, TRDIOD0, TRDIOB1, TRDIOC1, TRDIOD1 pin functions | Programmable I/O port or pulse output (selectable by pin) |
| $\overline{\text{INT0}}$ pin function | Programmable I/O port, pulse output forced cutoff signal input, or $\overline{\text{INT0}}$ interrupt input |
| Read from timer | The count value can be read by reading the TRD $_i$ register. |
| Write to timer | The value can be written to the TRD $_i$ register. |
| Select functions | <ul style="list-style-type: none"> • 1 to 3 PWM output pins selected per 1 channel Either 1 pin or multiple pins of the TRDIOB$_i$, TRDIOC$_i$ or TRDIOD$_i$ pin. • The active level selected by pin. • Initial output level selected by pin. • Synchronous operation (Refer to 14.4.3 Synchronous Operation.) • Buffer operation (Refer to 14.4.2 Buffer Operation.) • Pulse output forced cutoff signal input (Refer to 14.4.4 Pulse Output Forced Cutoff.) |

i = 0 or 1

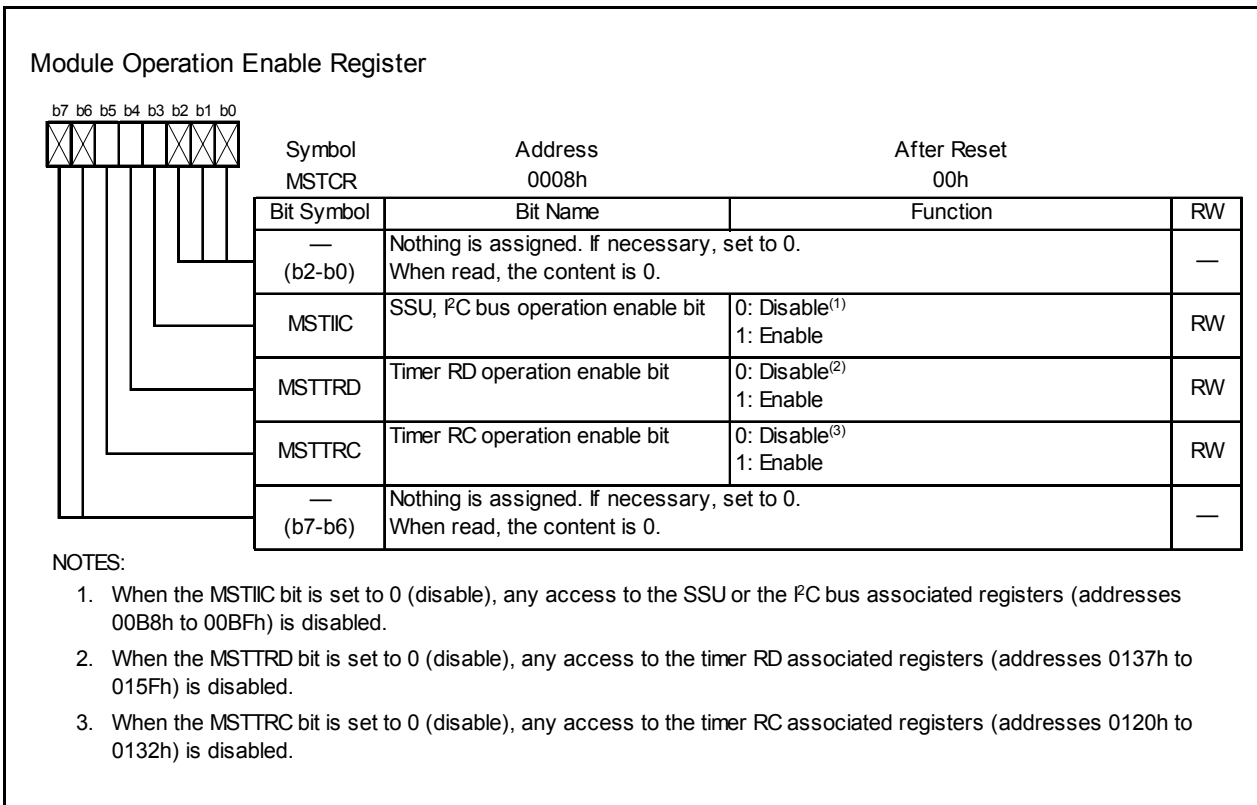


Figure 14.101 MSTCR Register

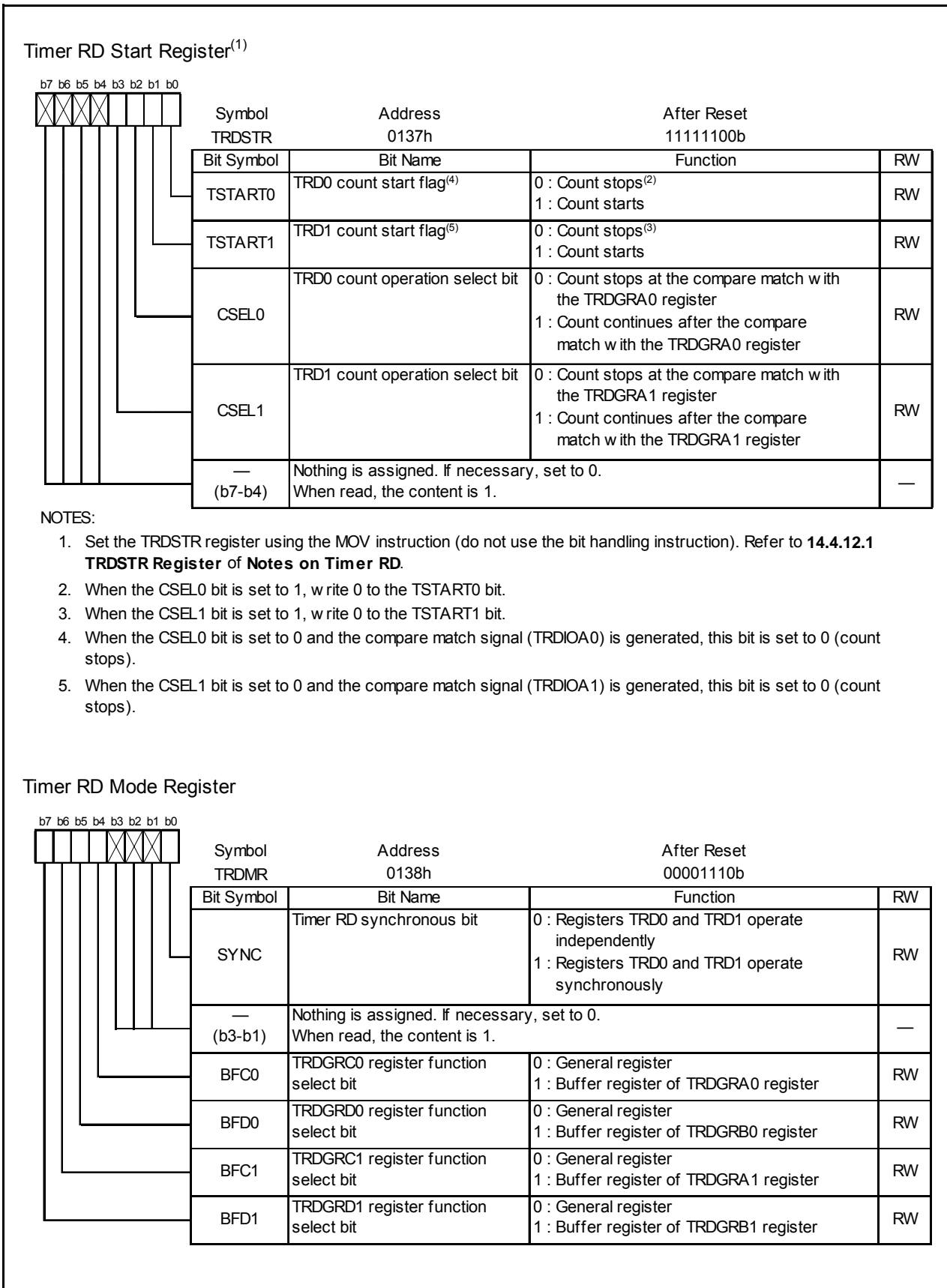


Figure 14.102 Registers TRDSTR and TRDMR in PWM Mode

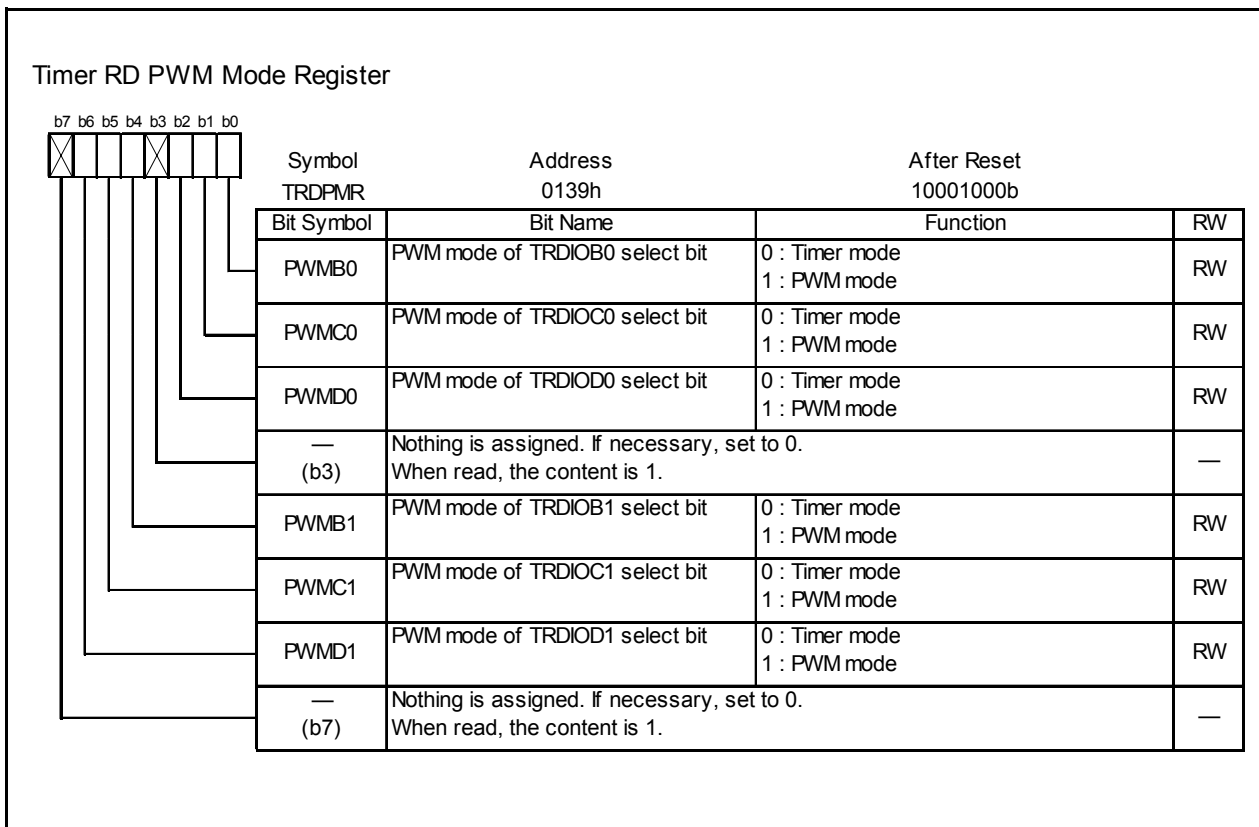


Figure 14.103 TRDPMR Register in PWM Mode

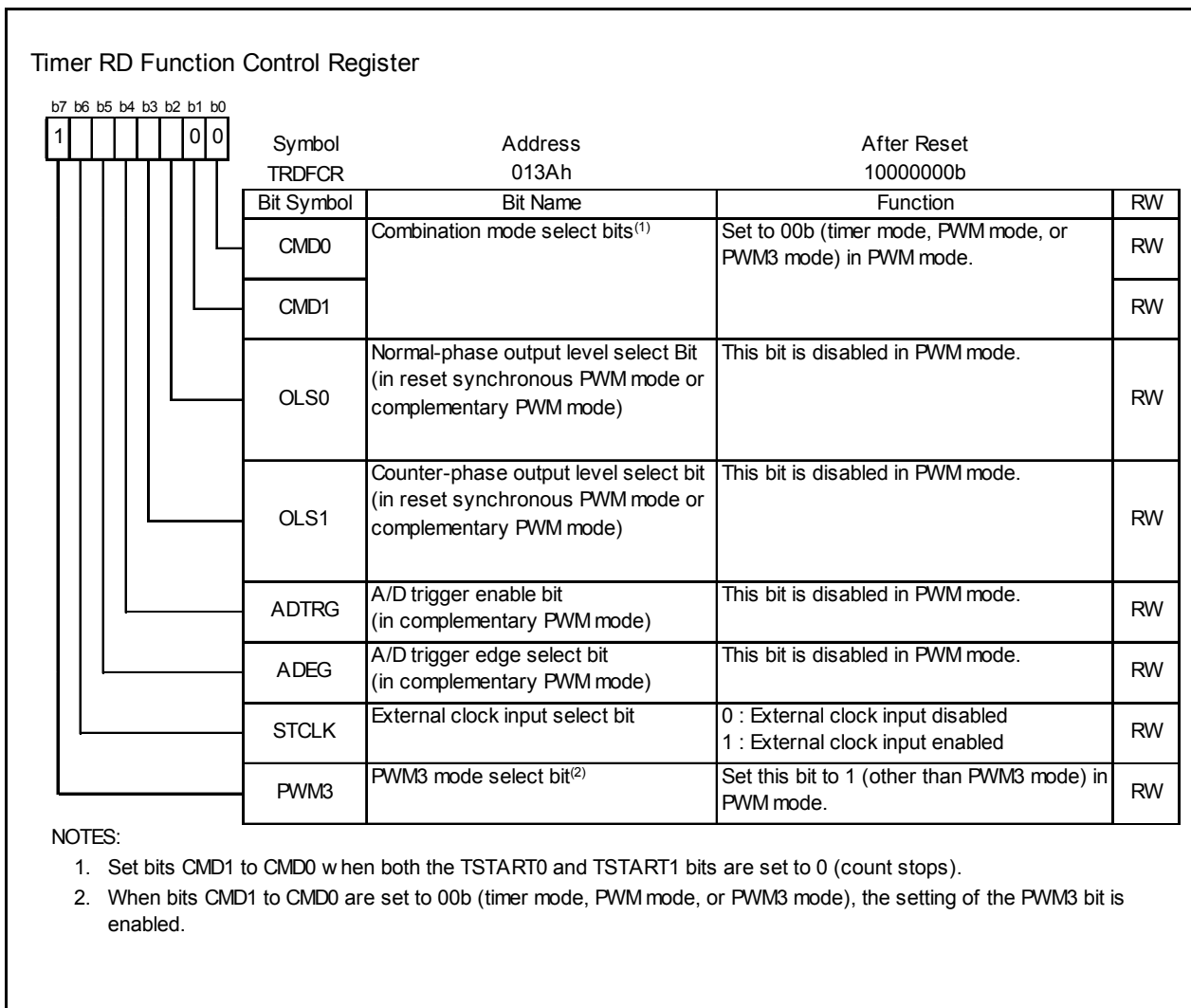


Figure 14.104 TRDFCR Register in PWM Mode

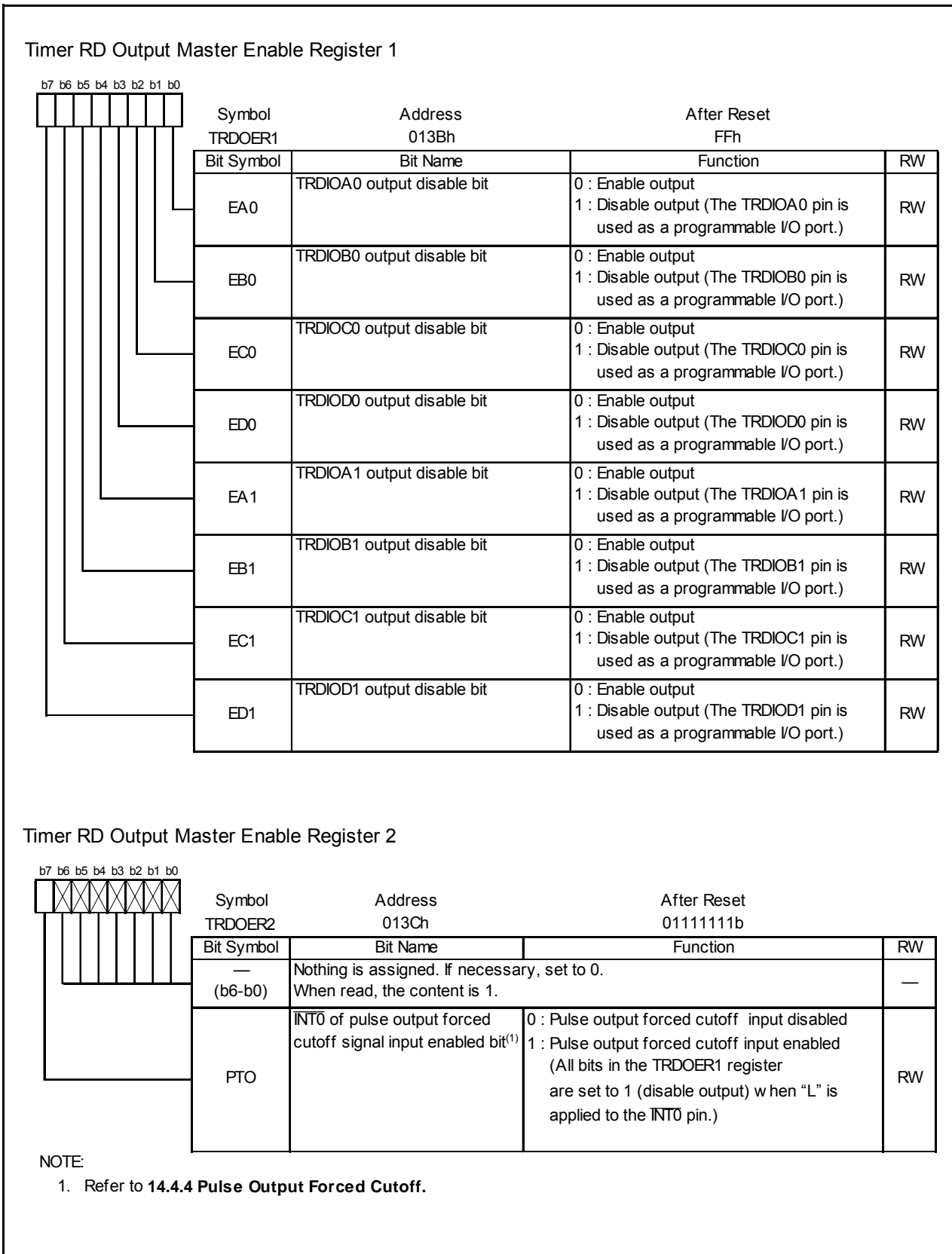


Figure 14.105 Registers TRDOER1 to TRDOER2 in PWM Mode

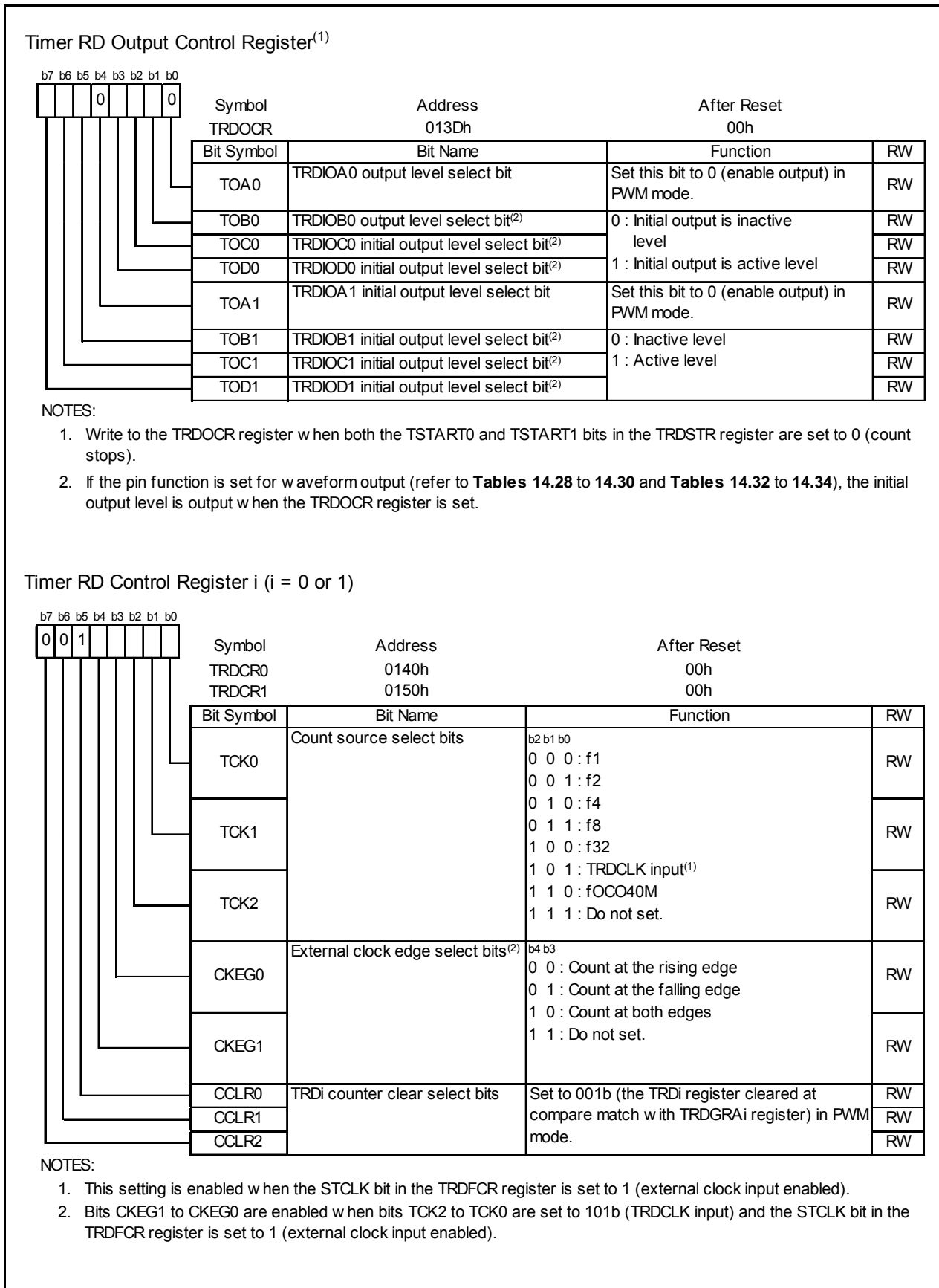


Figure 14.106 Registers TRDOCR and TRDCR0 to TRDCR1 in PWM Mode

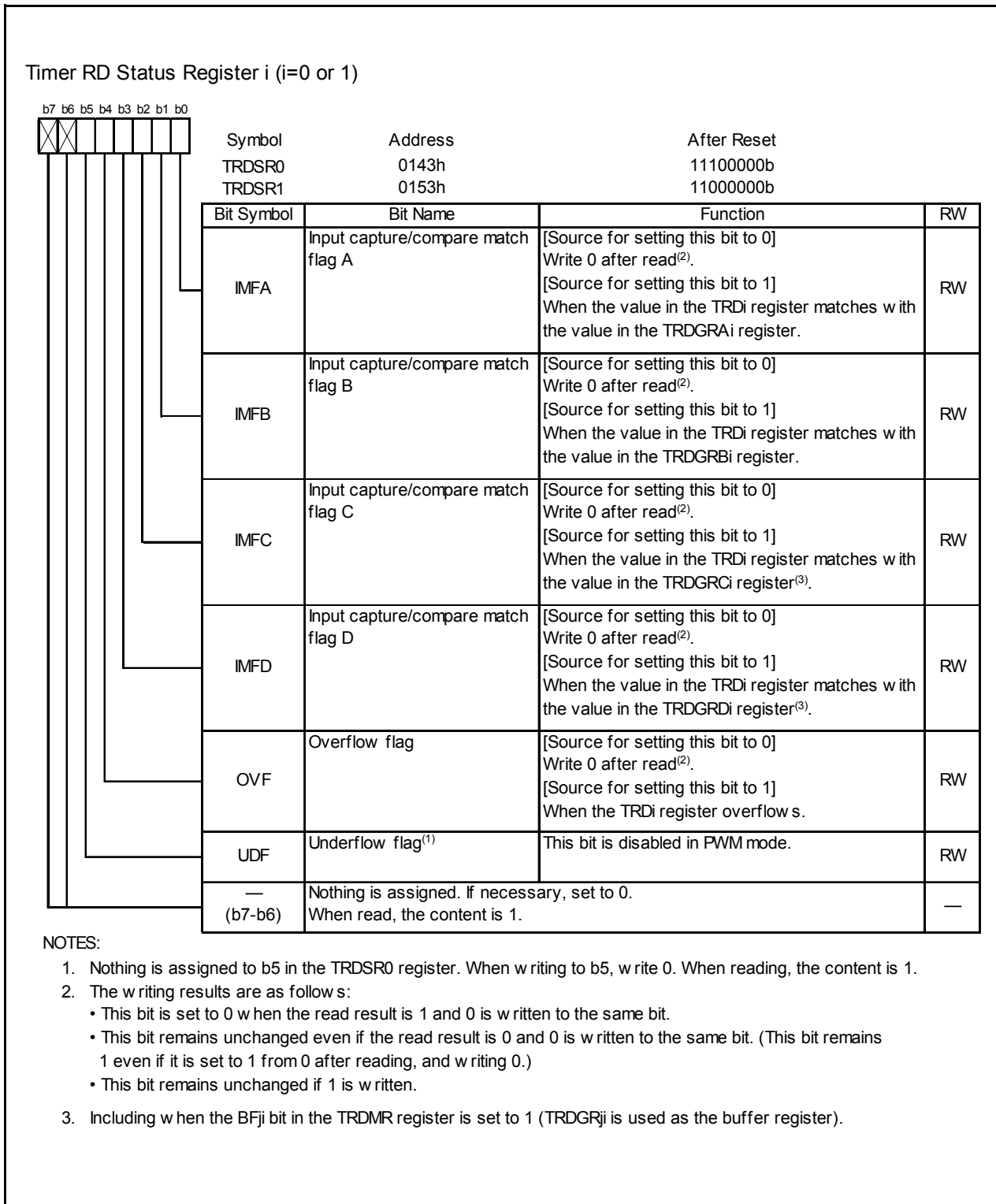


Figure 14.107 Registers TRDSR0 to TRDSR1 in PWM Mode

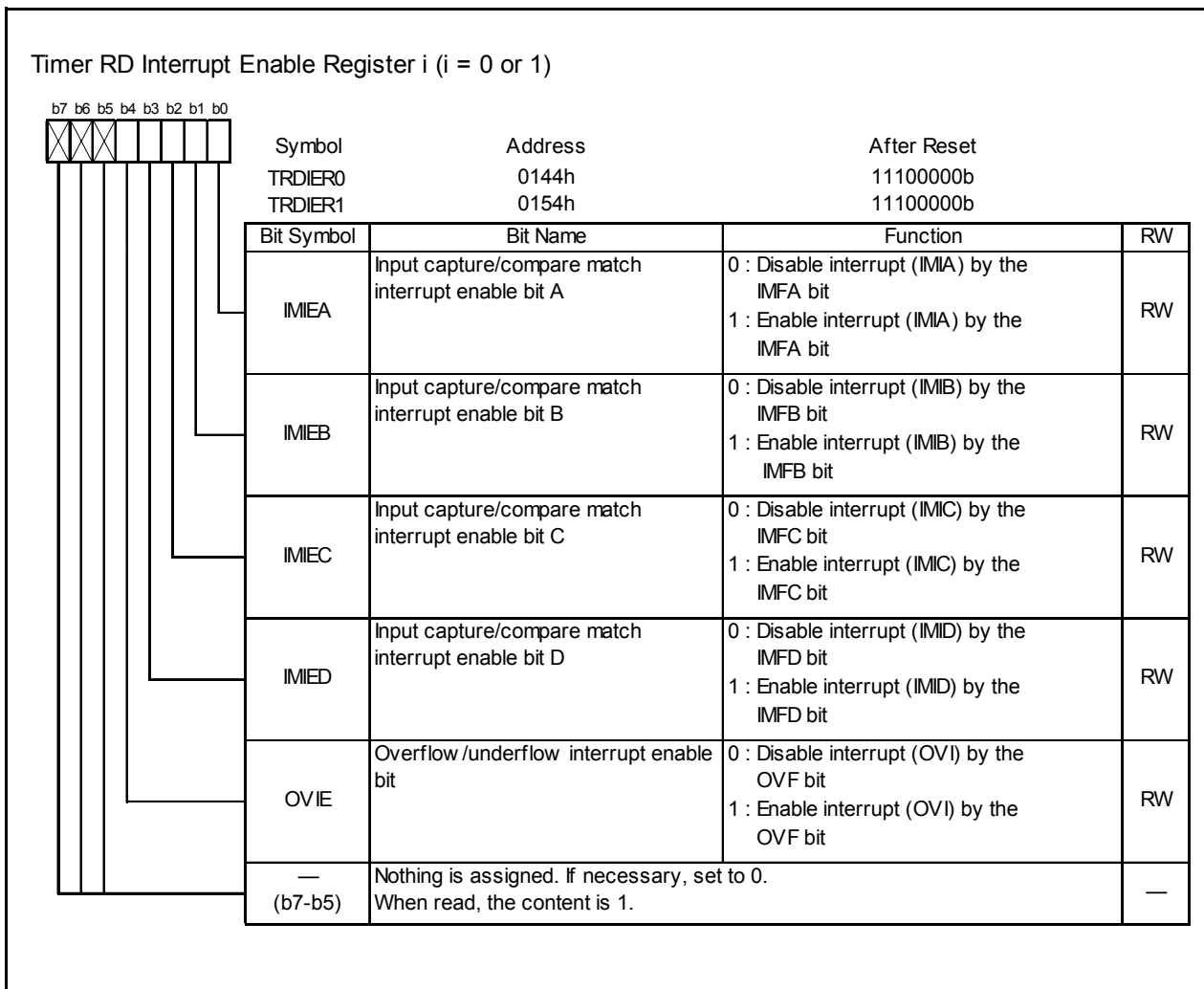


Figure 14.108 Registers TRDIER0 to TRDIER1 in PWM Mode

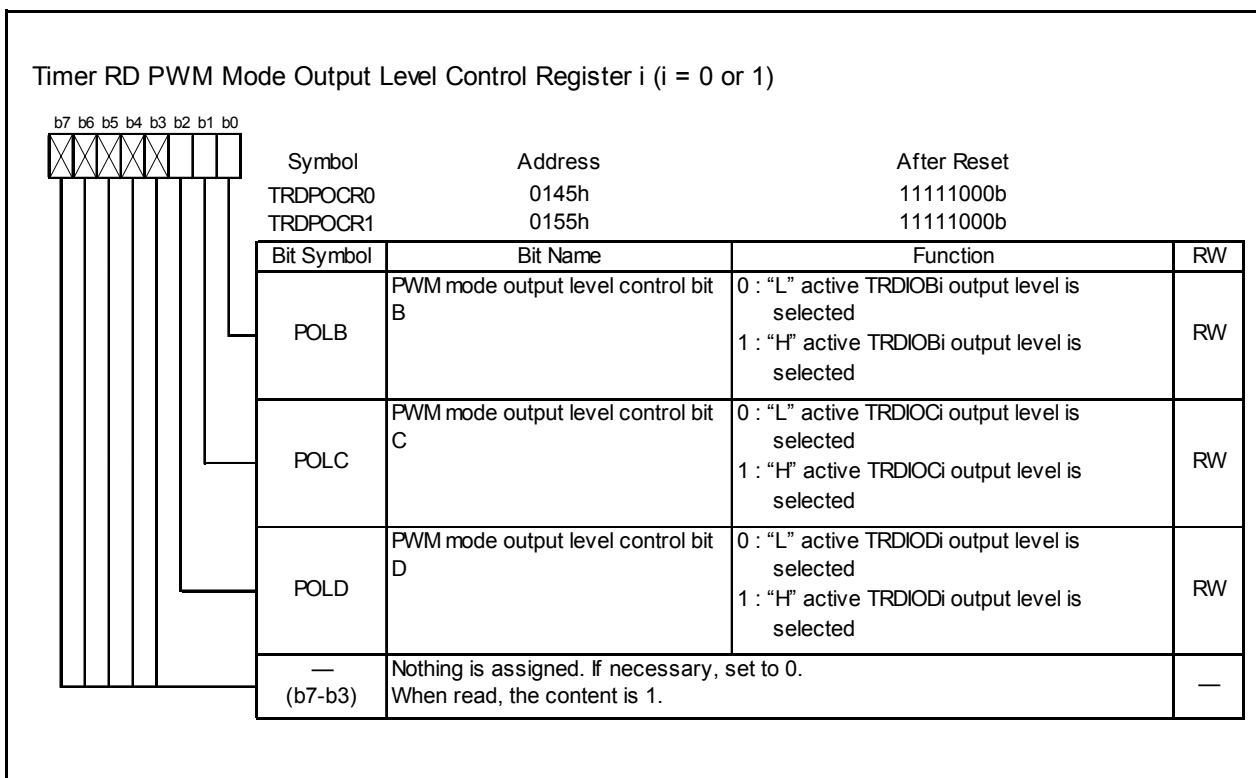


Figure 14.109 Registers TRDPOCR0 to TRDPOCR1 in PWM Mode

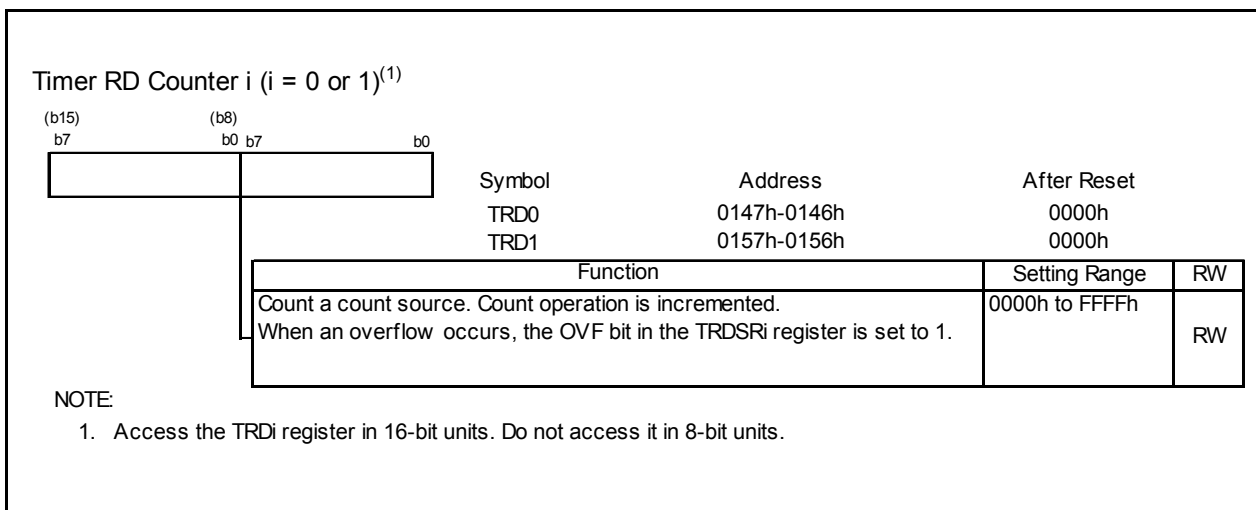


Figure 14.110 Registers TRD0 to TRD1 in PWM Mode

Timer RD General Registers Ai, Bi, Ci, and Di (i = 0 or 1)⁽¹⁾

| (b15) b7 | (b8) b0 | Symbol | Address | After Reset |
|--|------------|---------|-------------|-------------|
| | | TRDGRA0 | 0149h-0148h | FFFFh |
| | | TRDGRB0 | 014Bh-014Ah | FFFFh |
| | | TRDGRC0 | 014Dh-014Ch | FFFFh |
| | | TRDGRD0 | 014Fh-014Eh | FFFFh |
| | | TRDGRA1 | 0159h-0158h | FFFFh |
| | | TRDGRB1 | 015Bh-015Ah | FFFFh |
| | | TRDGRC1 | 015Dh-015Ch | FFFFh |
| | | TRDGRD1 | 015Fh-015Eh | FFFFh |
| Function | | | | RW |
| Refer to Table 14.43 TRDGRji Register Functions in PWM Mode. | | | | RW |

NOTE:

- Access registers TRDGRAi to TRDGRDi in 16-bit units. Do not access them in 8-bit units.

Figure 14.111 Registers TRDGRAi, TRDGRBi, TRDGRCi, and TRDGRDi in PWM Mode

The following registers are disabled in the PWM mode: TRDDF0, TRDDF1, TRDIORA0, TRDIORC0, TRDIORA1, and TRDIORC1.

Table 14.43 TRDGRji Register Functions in PWM Mode

| Register | Setting | Register Function | PWM Output Pin |
|----------|----------|---|----------------|
| TRDGRAi | – | General register. Set the PWM period | – |
| TRDGRBi | – | General register. Set the changing point of PWM output | TRDIOBi |
| TRDGRCi | BFCi = 0 | General register. Set the changing point of PWM output | TRDIOCi |
| TRDGRDi | BFDi = 0 | | TRDIODi |
| TRDGRCi | BFCi = 1 | Buffer register. Set the next PWM period (Refer to 14.4.2 Buffer Operation.) | – |
| TRDGRDi | BFDi = 1 | Buffer register. Set the changing point of the next PWM output (Refer to 14.4.2 Buffer Operation.) | TRDIOBi |

i = 0 or 1

BFCi, BFDi: Bits in TRDMR register

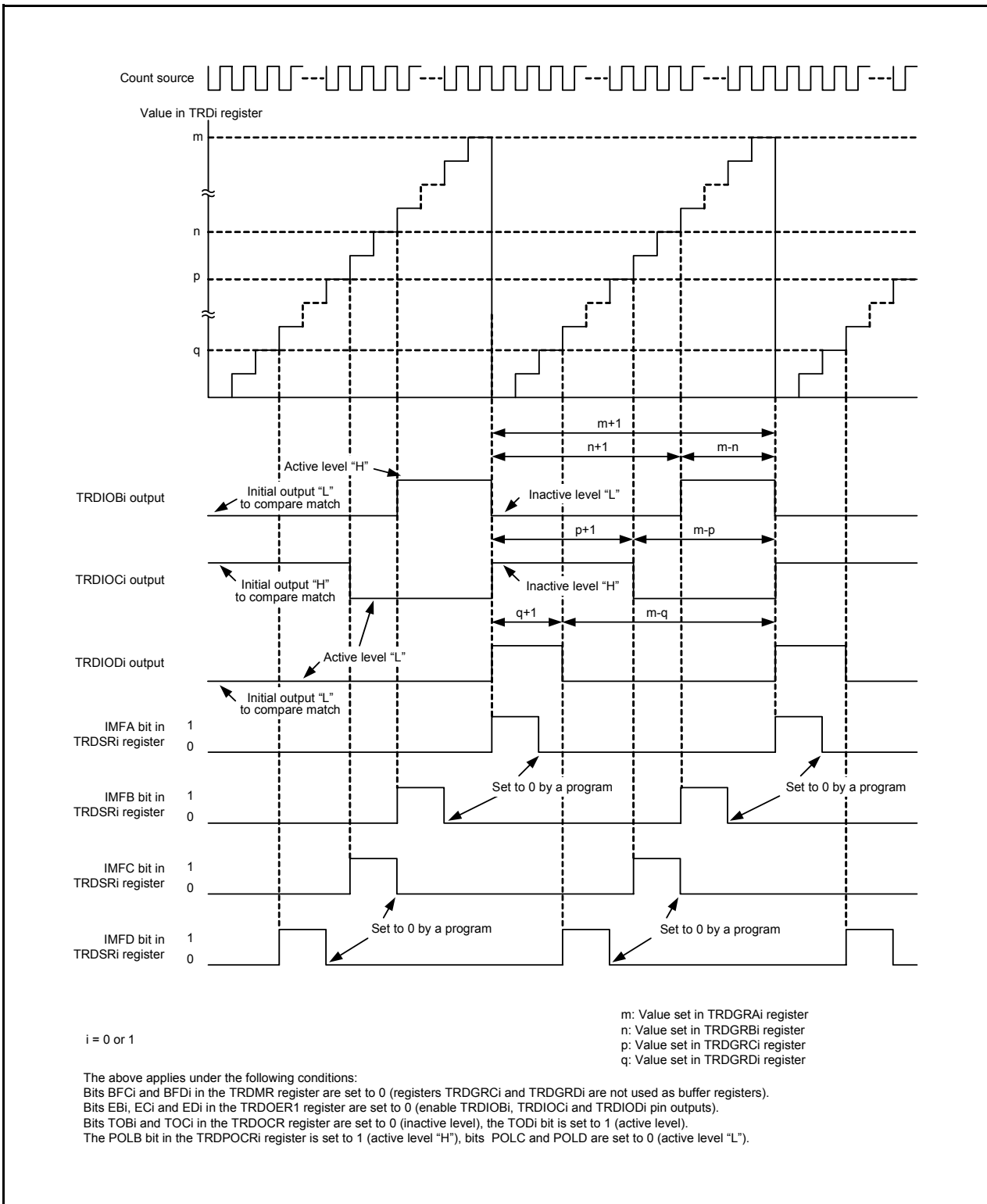


Figure 14.112 Operating Example of PWM Mode

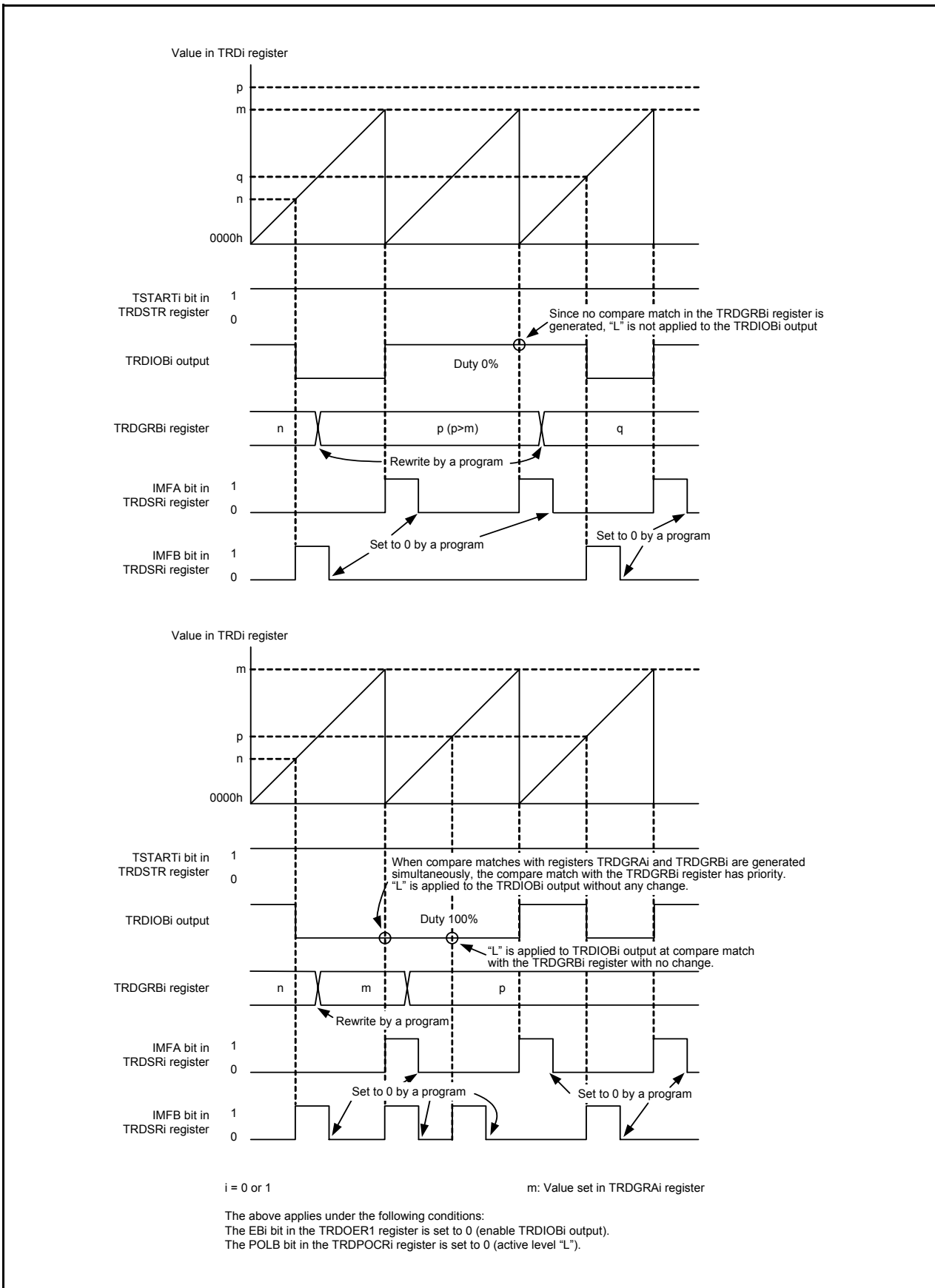


Figure 14.113 Operating Example of PWM Mode (Duty 0%, Duty 100%)

14.4.8 Reset Synchronous PWM Mode

In this mode, 3 normal-phases and 3 counter-phases of the PWM waveform are output with the same period (three-phase, sawtooth wave modulation, and no dead time).

Figure 14.114 shows a Block Diagram of Reset Synchronous PWM Mode, and Table 14.44 lists the Reset Synchronous PWM Mode Specifications. Figures 14.115 to 14.123 show the registers associated with reset synchronous PWM mode and Figure 14.124 shows an Operating Example of Reset Synchronous PWM Mode. Refer to **Figure 14.113 Operating Example of PWM Mode (Duty 0%, Duty 100%)** for an operating example of PWM Mode with duty 0% and duty 100%.

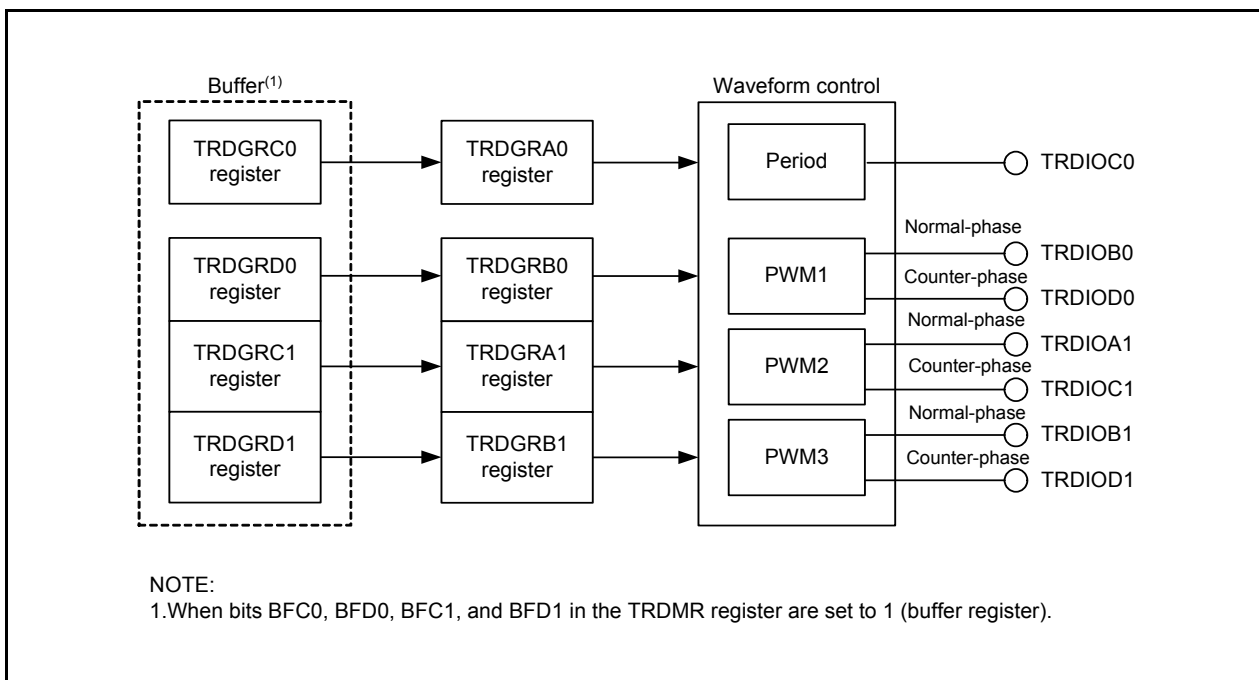


Figure 14.114 Block Diagram of Reset Synchronous PWM Mode

Table 14.44 Reset Synchronous PWM Mode Specifications

| Item | Specification |
|-------------------------------------|--|
| Count sources | f1, f2, f4, f8, f32, fOCO40M External signal input to the TRDCLK pin (valid edge selected by a program) |
| Count operations | The TRD0 register is incremented (the TRD1 register is not used). |
| PWM waveform | <p>PWM period : $1/f_k \times (m+1)$ Active level width of normal-phase : $1/f_k \times (m-n)$ Active level width of counter-phase: $1/f_k \times (n+1)$ f_k: Frequency of count source m: Value set in the TRDGRA0 register n: Value set in the TRDGRB0 register (PWM1 output), Value set in the TRDGRA1 register (PWM2 output), Value set in the TRDGRB1 register (PWM3 output)</p> <p>(When "L" is selected as the active level)</p> |
| Count start condition | 1 (count starts) is written to the TSTART0 bit in the TRDSTR register. |
| Count stop conditions | <ul style="list-style-type: none"> • 0 (count stops) is written to the TSTART0 bit in the TRDSTR register when the CSEL0 bit in the TRDSTR register is set to 1. The PWM output pin holds output level before the count stops • When the CSEL0 bit in the TRDSTR register is set to 0, the count stops at the compare match in the TRDGRA0 register. The PWM output pin holds level after output change at compare match. |
| Interrupt request generation timing | <ul style="list-style-type: none"> • Compare match (the content of the TRD0 register matches content of registers TRDGRj0, TRDGRA1, and TRDGRB1). • The TRD0 register overflows |
| TRDIOA0 pin function | Programmable I/O port or TRDCLK (external clock) input |
| TRDIOB0 pin function | PWM1 output normal-phase output |
| TRDIOD0 pin function | PWM1 output counter-phase output |
| TRDIOA1 pin function | PWM2 output normal-phase output |
| TRDIOC1 pin function | PWM2 output counter-phase output |
| TRDIOB1 pin function | PWM3 output normal-phase output |
| TRDIOD1 pin function | PWM3 output counter-phase output |
| TRDIOC0 pin function | Output inverted every PWM period |
| INT0 pin function | Programmable I/O port, pulse output forced cutoff signal input, or INT0 interrupt input |
| Read from timer | The count value can be read by reading the TRD0 register. |
| Write to timer | The value can be written to the TRD0 register. |
| Select functions | <ul style="list-style-type: none"> • The active level of normal-phase and counter-phase and initial output level selected individually. • Buffer operation (Refer to 14.4.2 Buffer Operation.) • Pulse output forced cutoff signal input (Refer to 14.4.4 Pulse Output Forced Cutoff.) |

j = either A, B, C, or D

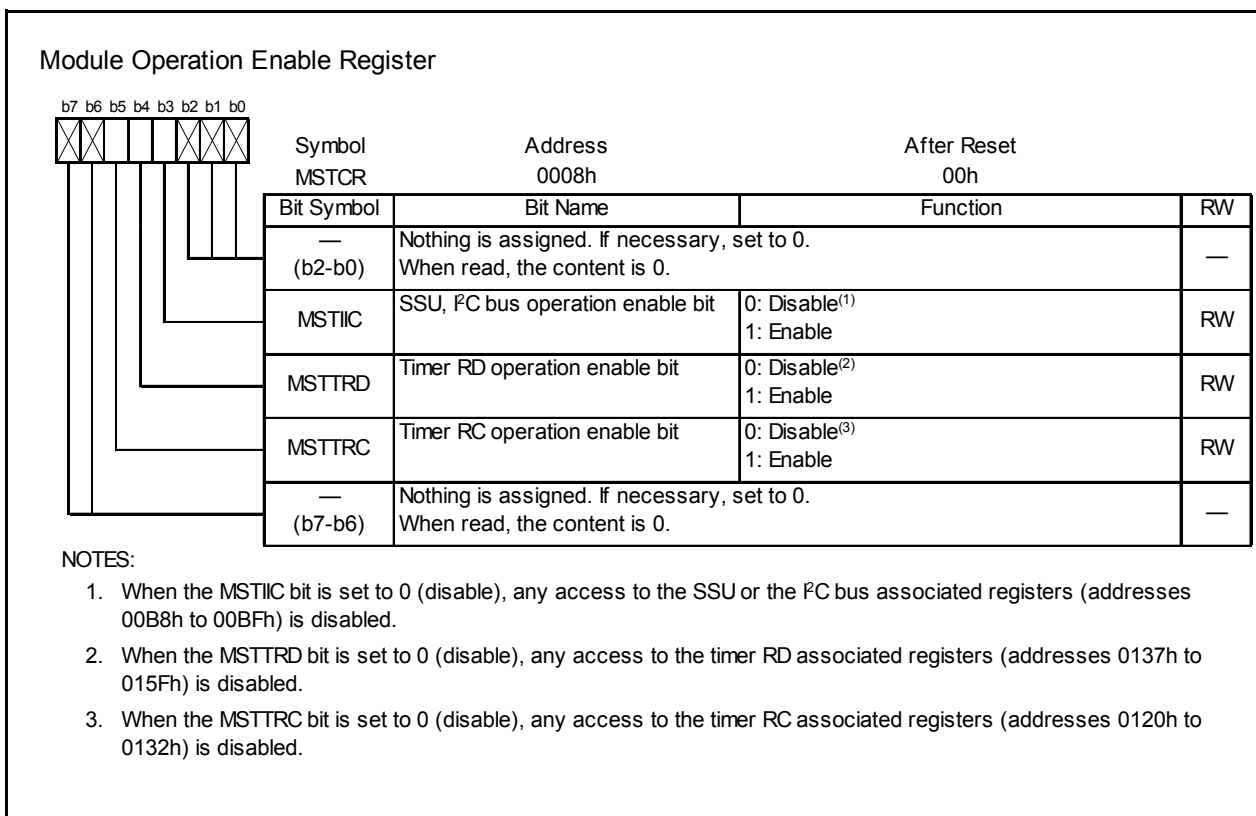


Figure 14.115 MSTCR Register

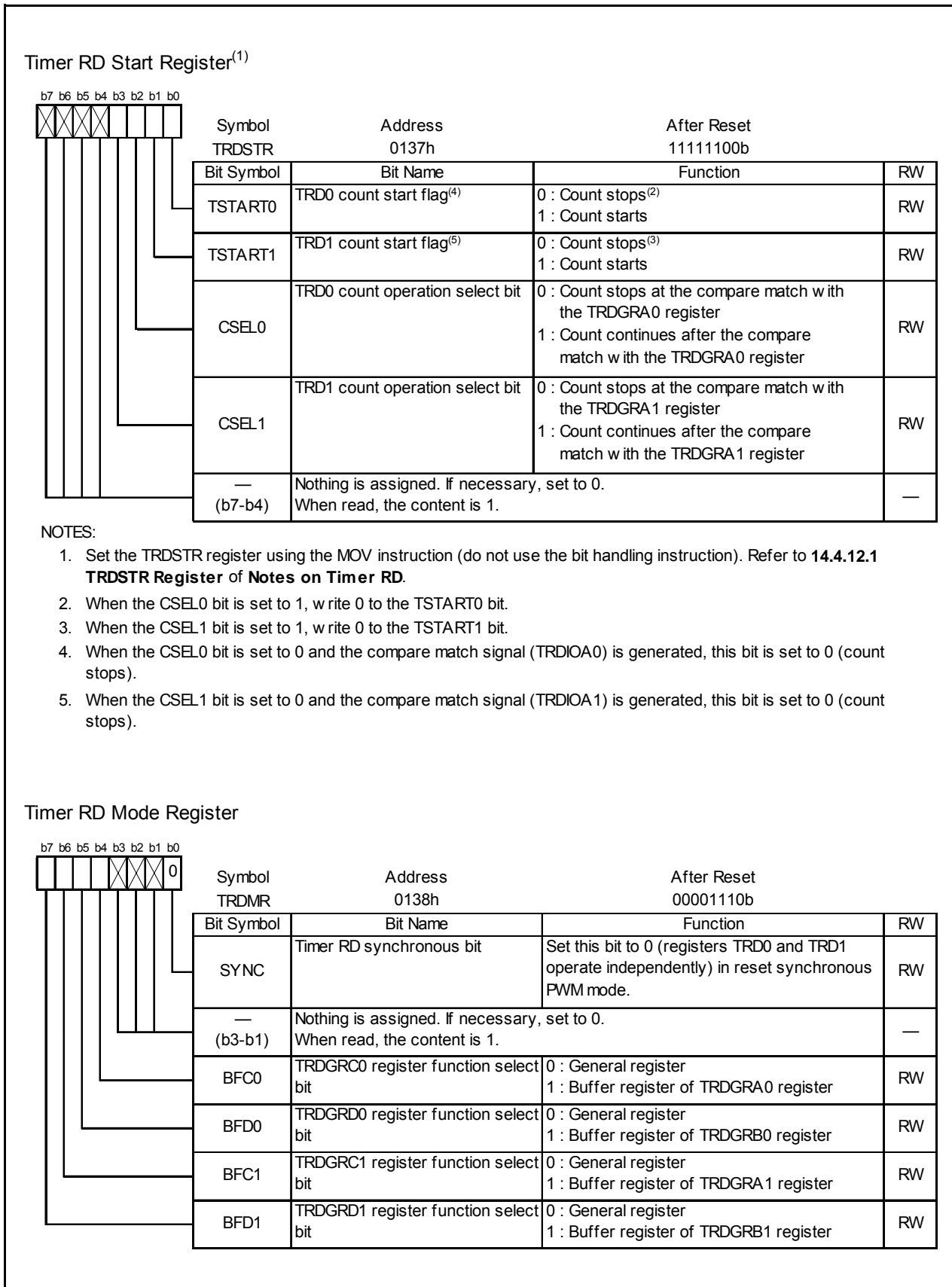


Figure 14.116 Registers TRDSTR and TRDMR in Reset Synchronous PWM Mode

Timer RD Function Control Register

| Symbol | Address | After Reset | |
|------------|---|--|----|
| TRDFCR | 013Ah | 1000000b | |
| Bit Symbol | Bit Name | Function | RW |
| CMD0 | Combination mode select bits ^(1, 2) | Set to 01b (reset synchronous PWM mode) in reset synchronous PWM mode. | RW |
| CMD1 | | | RW |
| OLS0 | Normal-phase output level select bit (in reset synchronous PWM mode or complementary PWM mode) | 0 : Initial output "H" Active level "L" 1 : Initial output "L" Active level "H" | RW |
| OLS1 | Counter-phase output level select bit (in reset synchronous PWM mode or complementary PWM mode) | 0 : Initial output "H" Active level "L" 1 : Initial output "L" Active level "H" | RW |
| ADTRG | A/D trigger enable bit (in complementary PWM mode) | This bit is disabled in reset synchronous PWM mode. | RW |
| ADEG | A/D trigger edge select bit (in complementary PWM mode) | This bit is disabled in reset synchronous PWM mode. | RW |
| STCLK | External clock input select bit | 0 : External clock input disabled 1 : External clock input enabled | RW |
| PWM3 | PWM3 mode select bit ⁽³⁾ | This bit is disabled in reset synchronous PWM mode. | RW |

NOTES:

- When bits CMD1 to CMD0 are set to 01b, 10b, or 11b, the MCU enters reset synchronous PWM mode or complementary PWM mode in spite of the setting of the TRDPMR register.
- Set bits CMD1 to CMD0 when both the TSTART0 and TSTART1 bits are set to 0 (count stops).
- When bits CMD1 to CMD0 are set to 00b (timer mode, PWM mode, or PWM3 mode), the setting of the PWM3 bit is enabled.

Figure 14.117 TRDFCR Register in Reset Synchronous PWM Mode

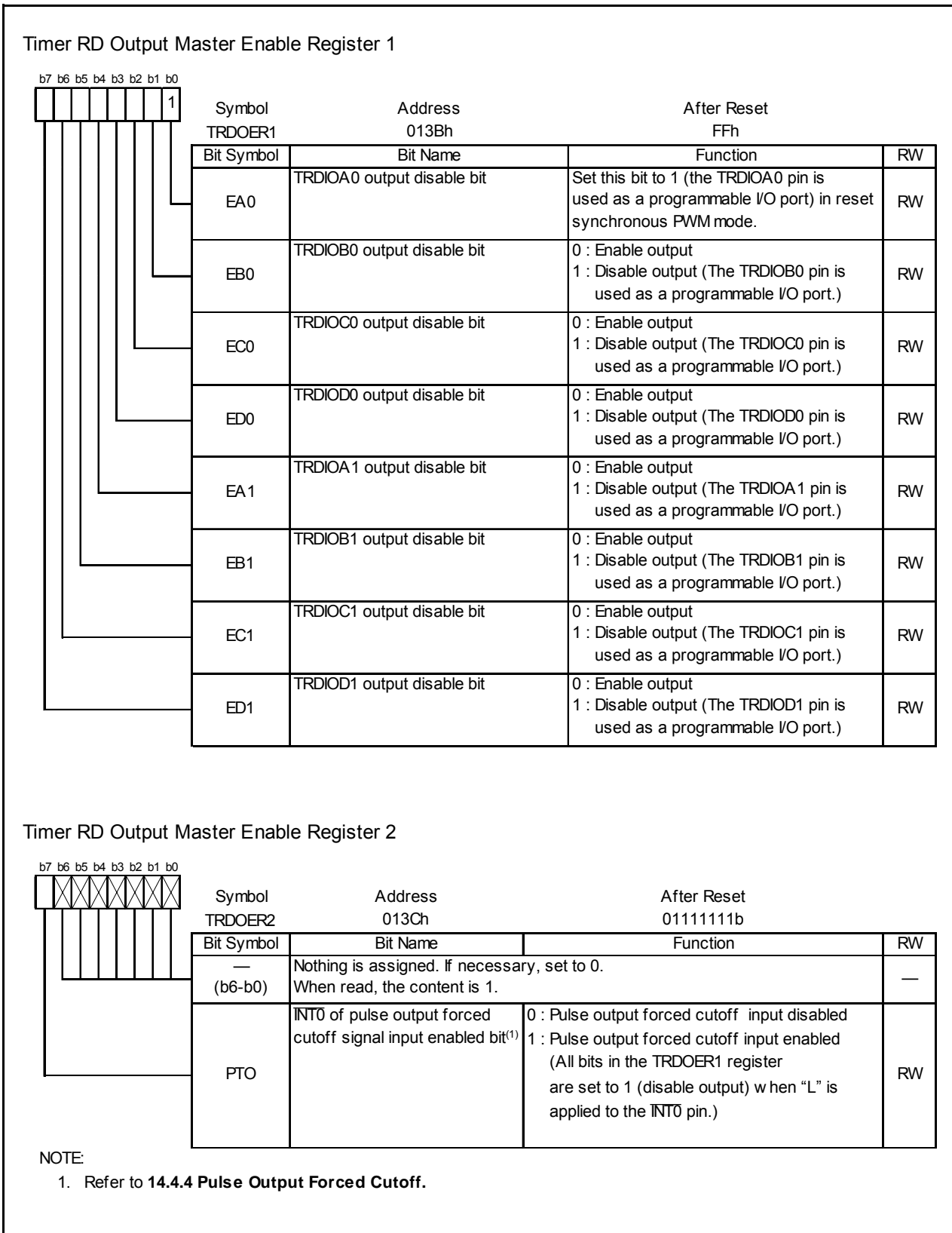


Figure 14.118 Registers TRDOER1 to TRDOER2 in Reset Synchronous PWM Mode

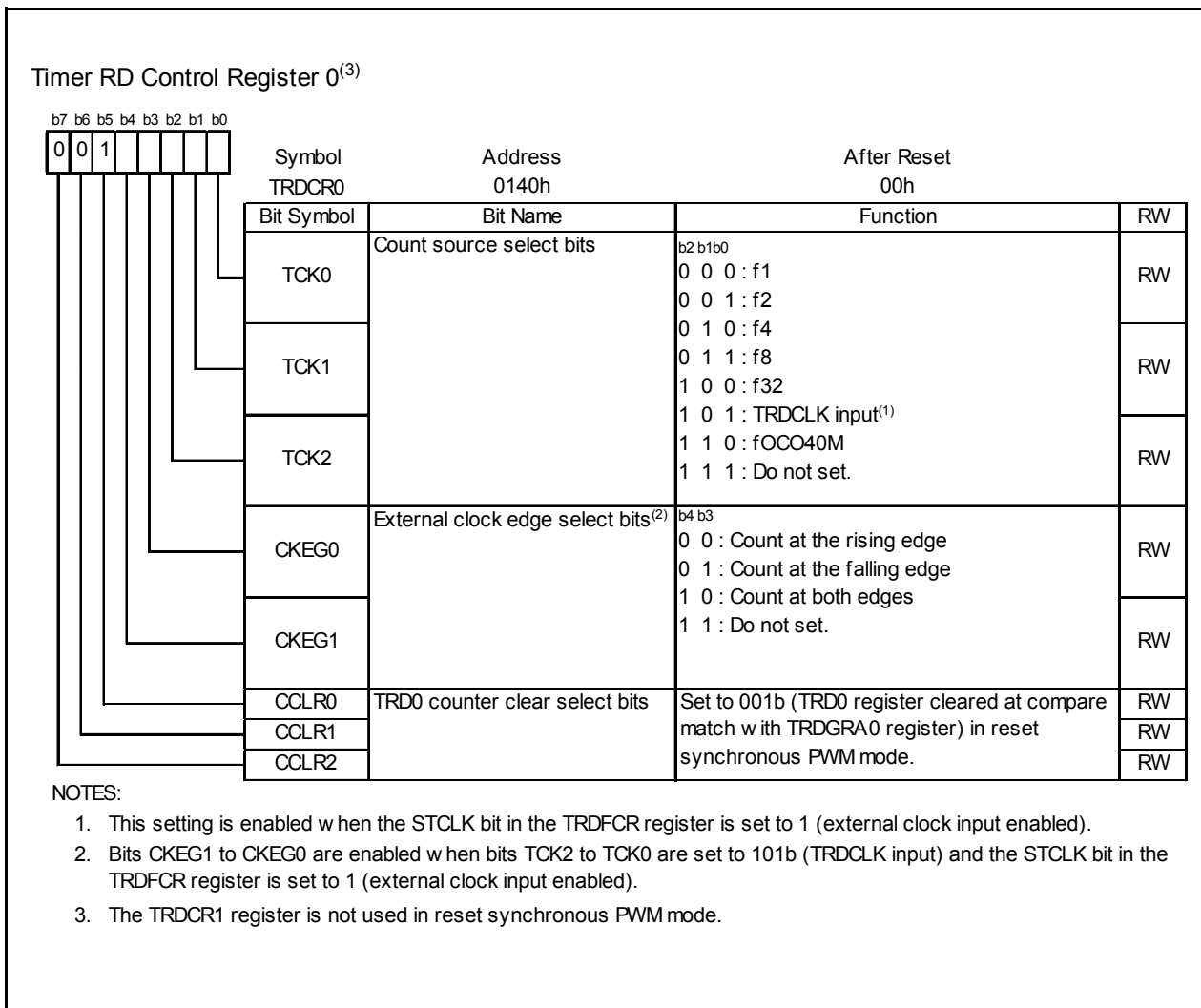


Figure 14.119 TRDCR0 Register in Reset Synchronous PWM Mode

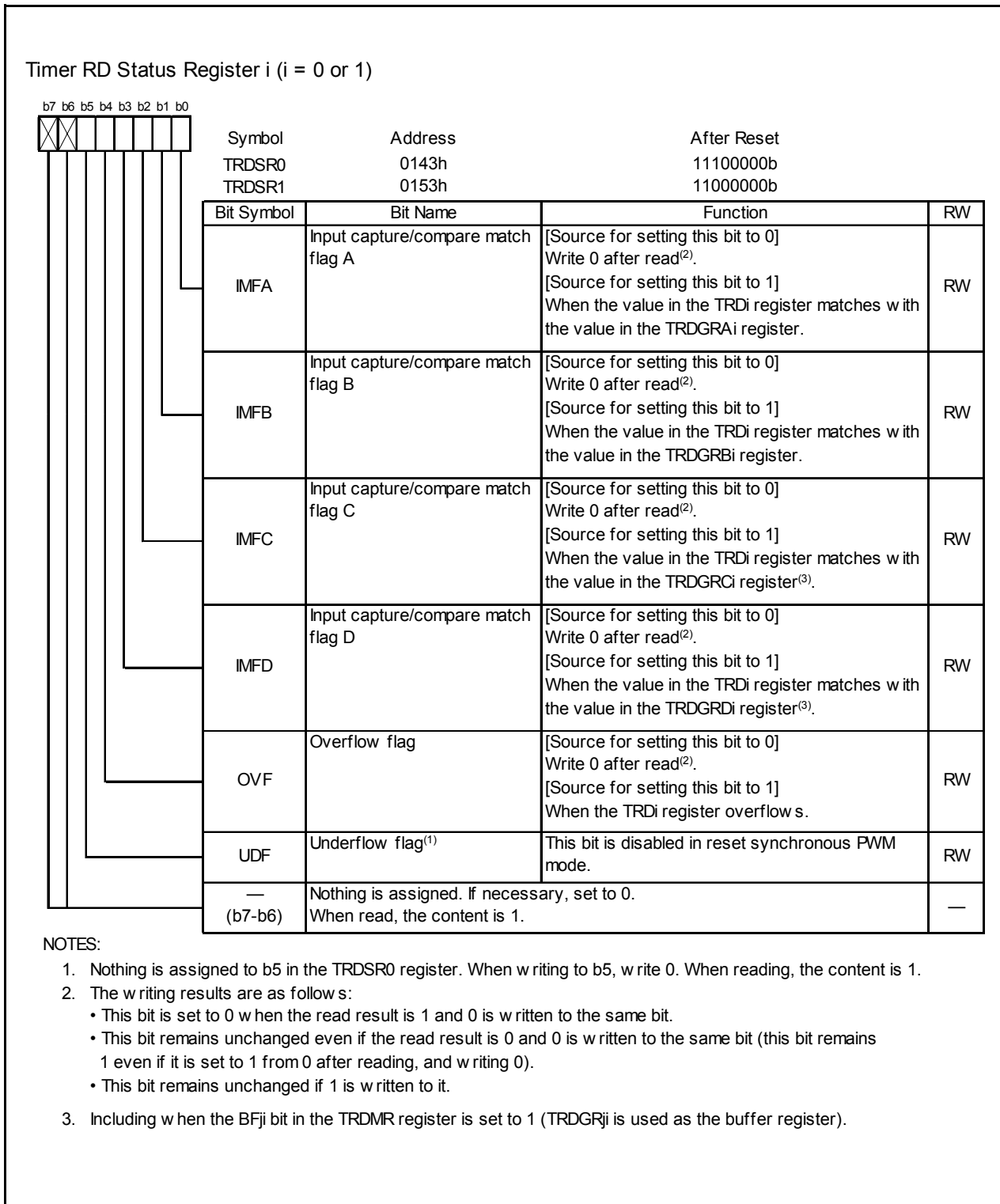


Figure 14.120 Registers TRDSR0 to TRDSR1 in Reset Synchronous PWM Mode

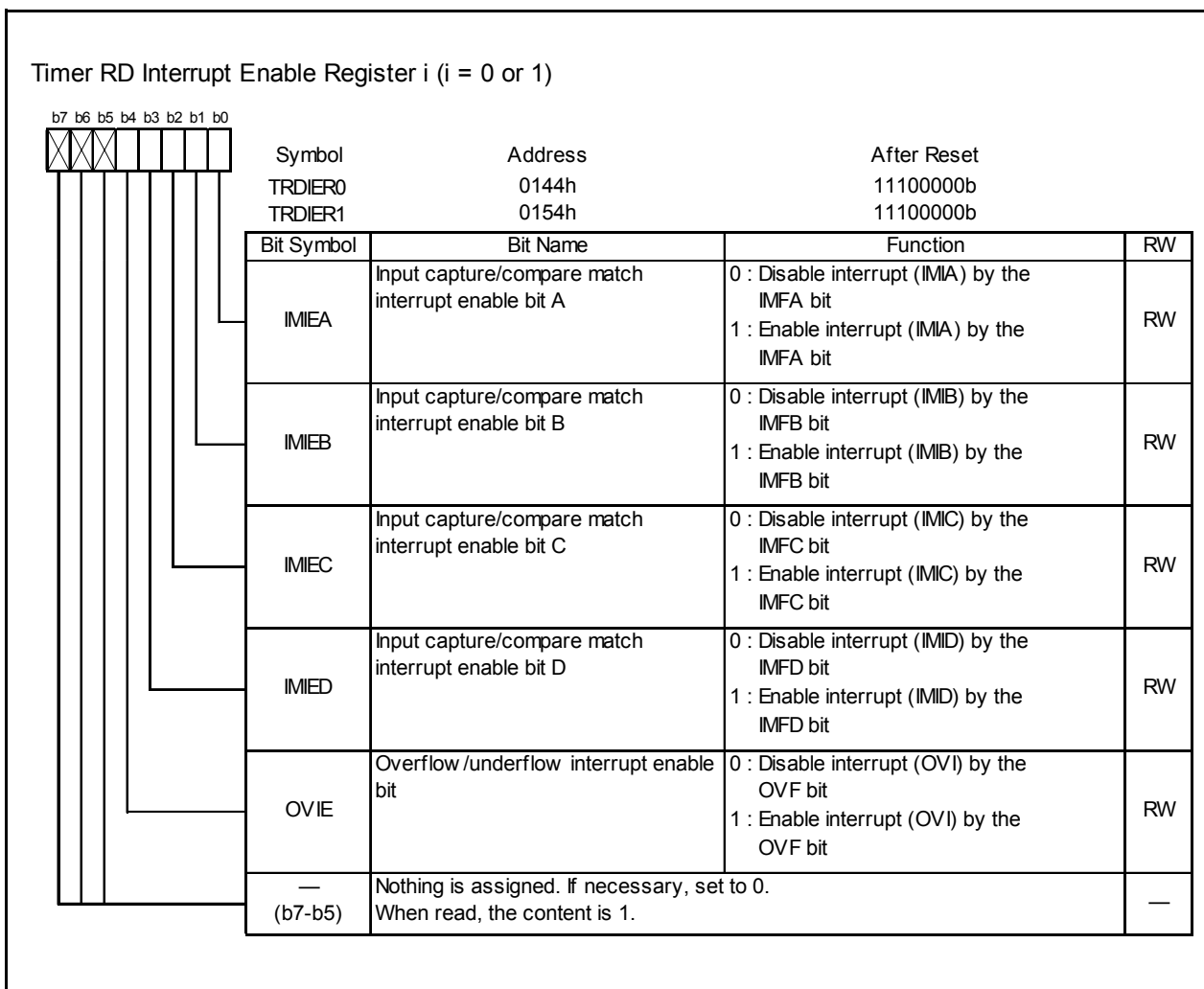


Figure 14.121 Registers TRDIER0 to TRDIER1 in Reset Synchronous PWM Mode

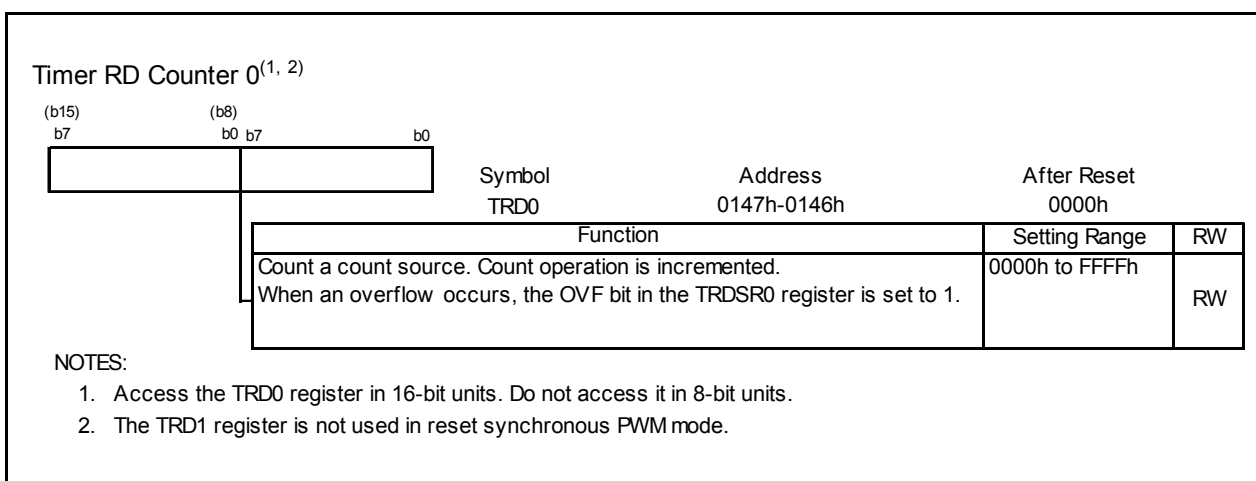


Figure 14.122 TRD0 Registrar in Reset Synchronous PWM Mode

| Timer RD General Registers Ai, Bi, Ci, and Di (i = 0 or 1) ⁽¹⁾ | | Symbol | Address | After Reset |
|--|------------------|---------|-------------|-------------|
| (b15) b7 | (b8) b0 b7 b0 | | | |
| | | TRDGRA0 | 0149h-0148h | FFFFh |
| | | TRDGRB0 | 014Bh-014Ah | FFFFh |
| | | TRDGRC0 | 014Dh-014Ch | FFFFh |
| | | TRDGRD0 | 014Fh-014Eh | FFFFh |
| | | TRDGRA1 | 0159h-0158h | FFFFh |
| | | TRDGRB1 | 015Bh-015Ah | FFFFh |
| | | TRDGRC1 | 015Dh-015Ch | FFFFh |
| | | TRDGRD1 | 015Fh-015Eh | FFFFh |
| Function | | | | RW |
| Refer to Table 14.45 TRDGRji Register Functions in Reset Synchronous PWM Mode. | | | | RW |

NOTE:
1. Access registers TRDGRAi to TRDGRDi in 16-bit units. Do not access them in 8-bit units.

Figure 14.123 Registers TRDGRAi, TRDGRBi, TRDGRCi, and TRDGRDi in Reset Synchronous PWM Mode

The following registers are disabled in the reset synchronous PWM mode: TRDPMR, TRDOCR, TRDDF0, TRDDF1, TRDIOA0, TRDIORC0, TRDPOCR0, TRDIOA1, TRDIORC1, and TRDPOCR1.

Table 14.45 TRDGRji Register Functions in Reset Synchronous PWM Mode

| Register | Setting | Register Function | PWM Output Pin |
|----------|----------|--|--|
| TRDGRA0 | – | General register. Set the PWM period. | (Output inverted every PWM period and TRDIOC0 pin) |
| TRDGRB0 | – | General register. Set the changing point of PWM1 output. | TRDIOB0 TRDIOD0 |
| TRDGRC0 | BFC0 = 0 | (These registers are not used in reset synchronous PWM mode.) | – |
| TRDGRD0 | BFD0 = 0 | | |
| TRDGRA1 | – | General register. Set the changing point of PWM2 output. | TRDIOA1 TRDIOC1 |
| TRDGRB1 | – | General register. Set the changing point of PWM3 output. | TRDIOB1 TRDIOD1 |
| TRDGRC1 | BFC1 = 0 | (These points are not used in reset synchronous PWM mode.) | – |
| TRDGRD1 | BFD1 = 0 | | |
| TRDGRC0 | BFC0 = 1 | Buffer register. Set the next PWM period. (Refer to 14.4.2 Buffer Operation.) | (Output inverted every PWM period and TRDIOC0 pin) |
| TRDGRD0 | BFD0 = 1 | Buffer register. Set the changing point of the next PWM1 output. (Refer to 14.4.2 Buffer Operation.) | TRDIOB0 TRDIOD0 |
| TRDGRC1 | BFC1 = 1 | Buffer register. Set the changing point of the next PWM2 output. (Refer to 14.4.2 Buffer Operation.) | TRDIOA1 TRDIOC1 |
| TRDGRD1 | BFD1 = 1 | Buffer register. Set the changing point of the next PWM3 output. (Refer to 14.4.2 Buffer Operation.) | TRDIOB1 TRDIOD1 |

BFC0, BFD0, BFC1, BFD1: Bits in TRDMR register

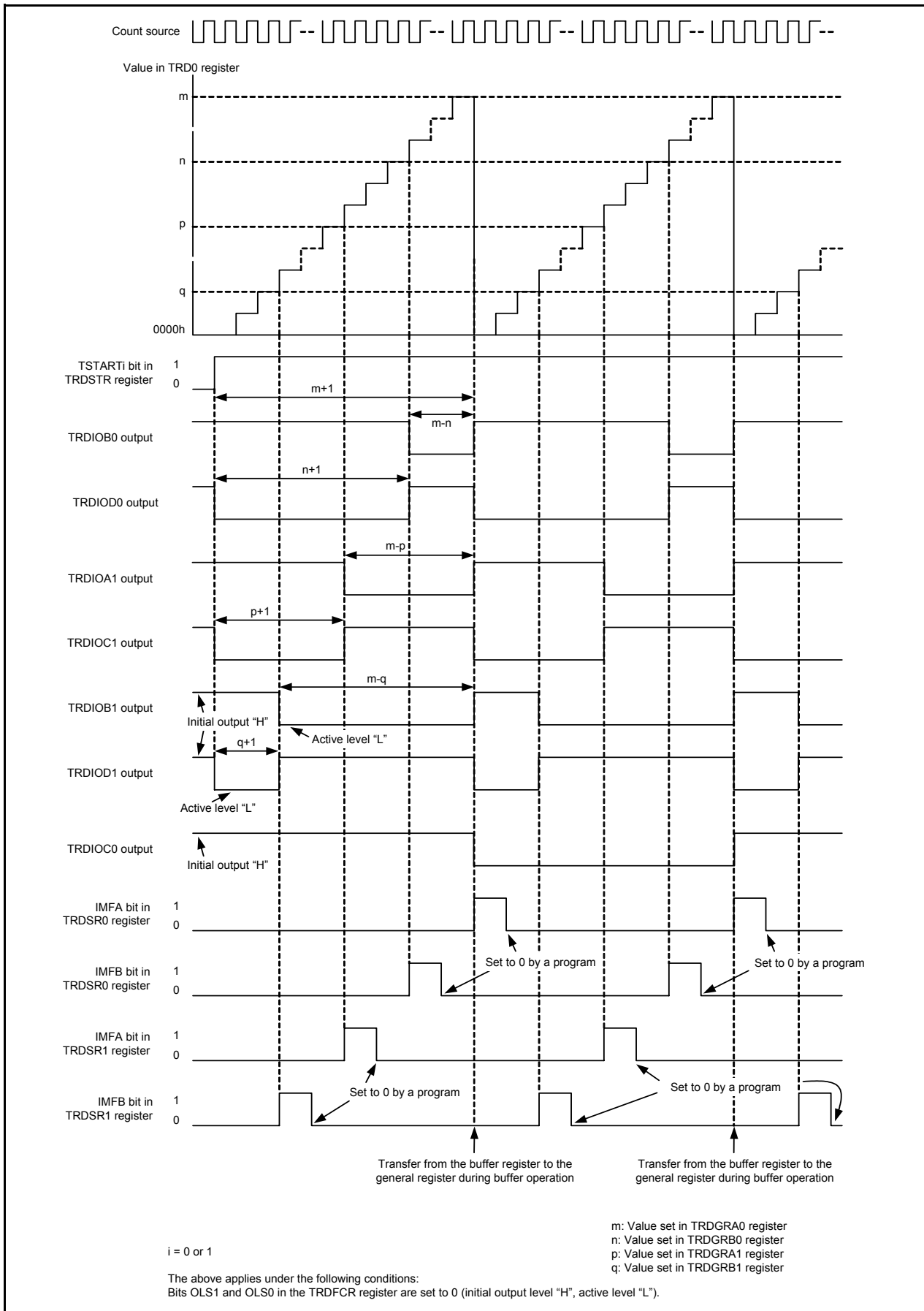


Figure 14.124 Operating Example of Reset Synchronous PWM Mode

14.4.9 Complementary PWM Mode

In this mode, 3 normal-phases and 3 counter-phases of the PWM waveform are output with the same period (three-phase, triangular wave modulation, and with dead time).

Figure 14.125 shows a Block Diagram of Complementary PWM Mode, and Table 14.46 lists the Complementary PWM Mode Specifications. Figures 14.126 to 14.135 show the Registers Associated with Complementary PWM Mode, Figure 14.136 shows Output Model of Complementary PWM Mode and Figure 14.137 shows Operating Example of Complementary PWM Mode.

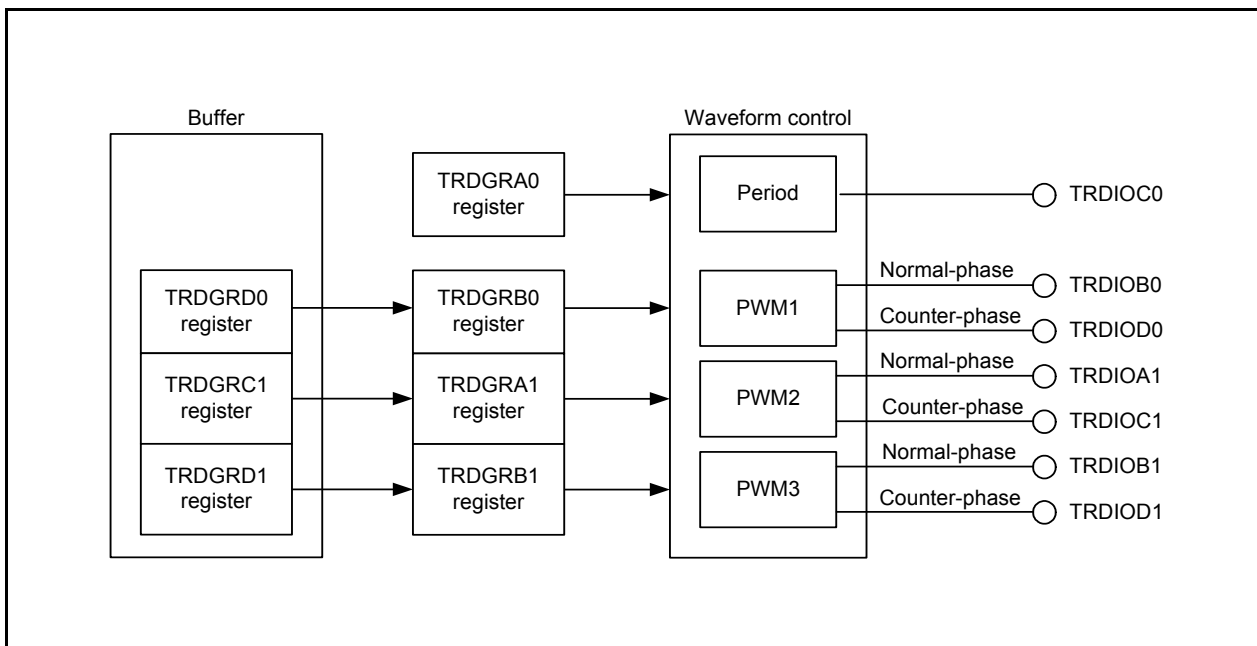


Figure 14.125 Block Diagram of Complementary PWM Mode

Table 14.46 Complementary PWM Mode Specifications

| Item | Specification |
|-------------------------------------|---|
| Count sources | f1, f2, f4, f8, f32, fOCO40M External signal input to the TRDCLK pin (valid edge selected by a program) Set bits TCK2 to TCK0 in the TRDCR1 register to the same value (same count source) as bits TCK2 to TCK0 in the TRDCR0 register. |
| Count operations | Increment or decrement Registers TRD0 and TRD1 are decremented with the compare match in registers TRD0 and TRDGRA0 during increment operation. The TRD1 register value is changed from 0000h to FFFFh during decrement operation, and registers TRD0 and TRD1 are incremented. |
| PWM operations | PWM period: $1/fk \times (m+2-p) \times 2^{(1)}$ Dead time: p Active level width of normal-phase: $1/fk \times (m-n-p+1) \times 2$ Active level width of counter-phase: $1/fk \times (n+1-p) \times 2$ fk: Frequency of count source m: Value set in the TRDGRA0 register n: Value set in the TRDGRB0 register (PWM1 output) Value set in the TRDGRA1 register (PWM2 output) Value set in the TRDGRB1 register (PWM3 output) p: Value set in the TRD0 register |
| Count start condition | 1 (count starts) is written to bits TSTART0 and TSTART1 in the TRDSTR register. |
| Count stop conditions | 0 (count stops) is written to bits TSTART0 and TSTART1 in the TRDSTR register when the CSEL0 bit in the TRDSTR register is set to 1. (The PWM output pin holds output level before the count stops.) |
| Interrupt request generation timing | <ul style="list-style-type: none"> Compare match (The content of the TRDi register matches content of the TRDGRji register.) The TRD1 register underflows |
| TRDIOA0 pin function | Programmable I/O port or TRDCLK (external clock) input |
| TRDIOB0 pin function | PWM1 output normal-phase output |
| TRDIOD0 pin function | PWM1 output counter-phase output |
| TRDIOA1 pin function | PWM2 output normal-phase output |
| TRDIOC1 pin function | PWM2 output counter-phase output |
| TRDIOB1 pin function | PWM3 output normal-phase output |
| TRDIOD1 pin function | PWM3 output counter-phase output |
| TRDIOC0 pin function | Output inverted every 1/2 period of PWM |
| INT0 pin function | Programmable I/O port, pulse output forced cutoff signal input or INT0 interrupt input |
| Read from timer | The count value can be read by reading the TRDi register. |
| Write to timer | The value can be written to the TRDi register. |
| Select functions | <ul style="list-style-type: none"> Pulse output forced cutoff signal input (Refer to 14.4.4 Pulse Output Forced Cutoff.) The active level of normal-phase and counter-phase and initial output level selected individually Transfer timing from the buffer register selected A/D trigger generated |

i = 0 or 1, j = either A, B, C, or D

NOTE:

- After a count starts, the PWM period is fixed.

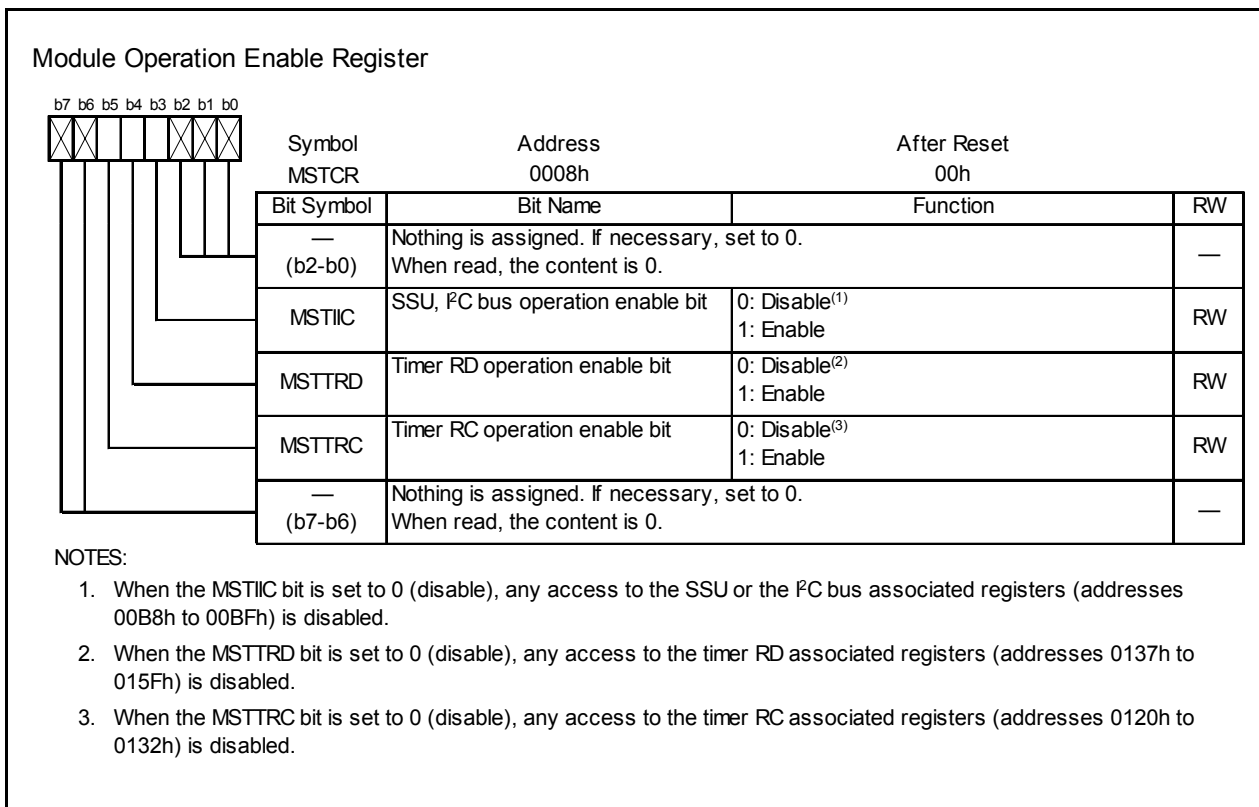


Figure 14.126 MSTCR Register

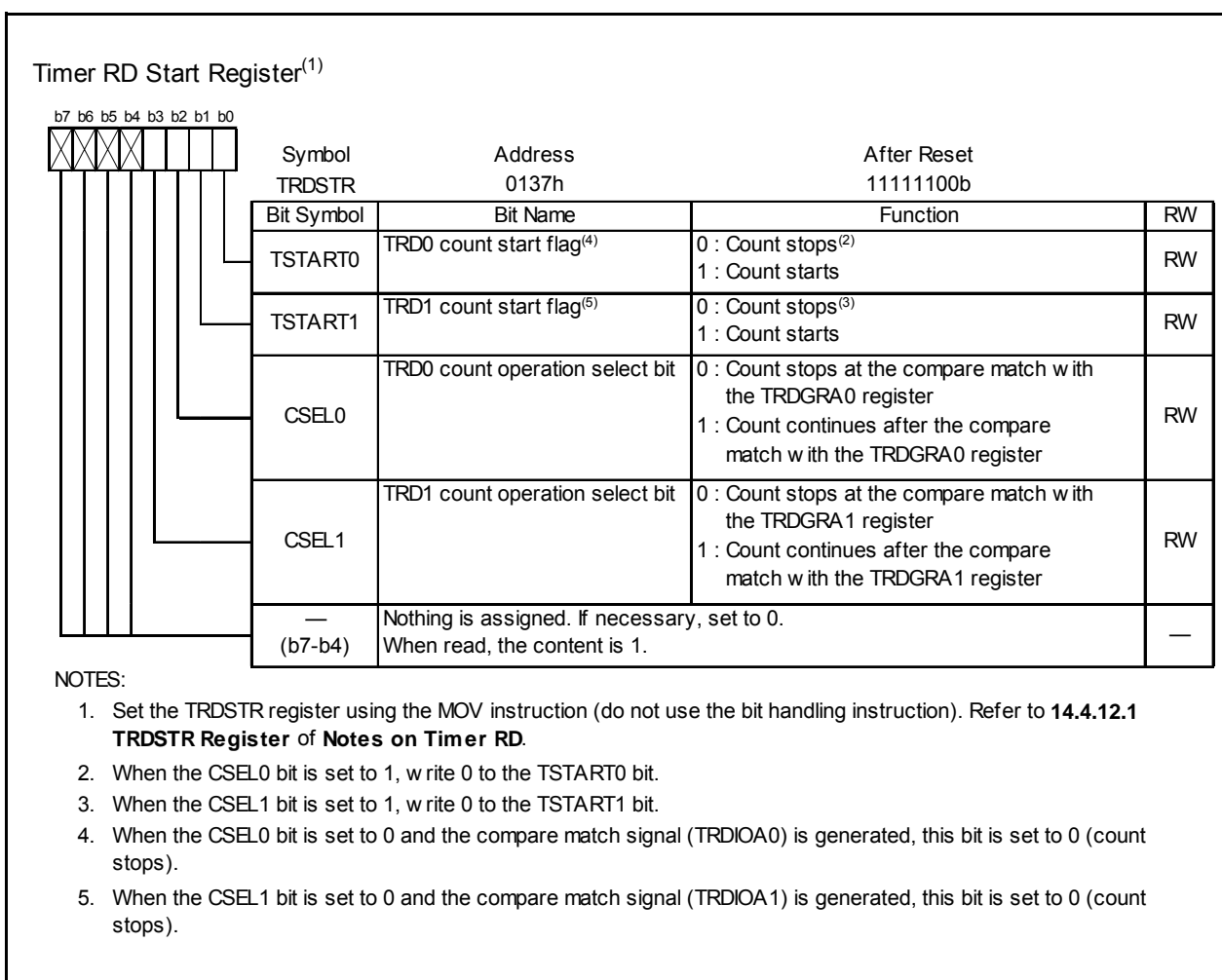


Figure 14.127 TRDSTR Register in Complementary PWM Mode

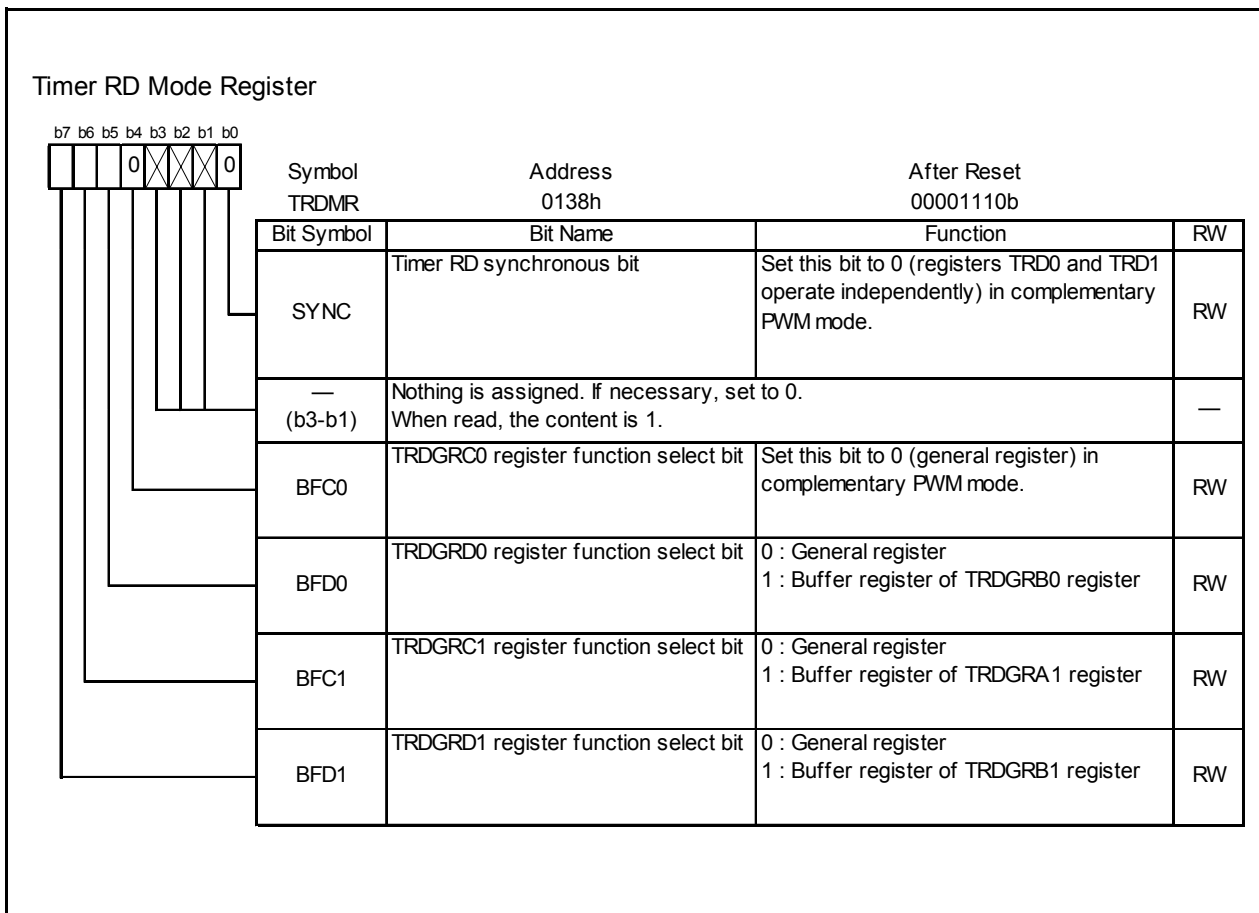


Figure 14.128 TRDMR Register in Complementary PWM Mode

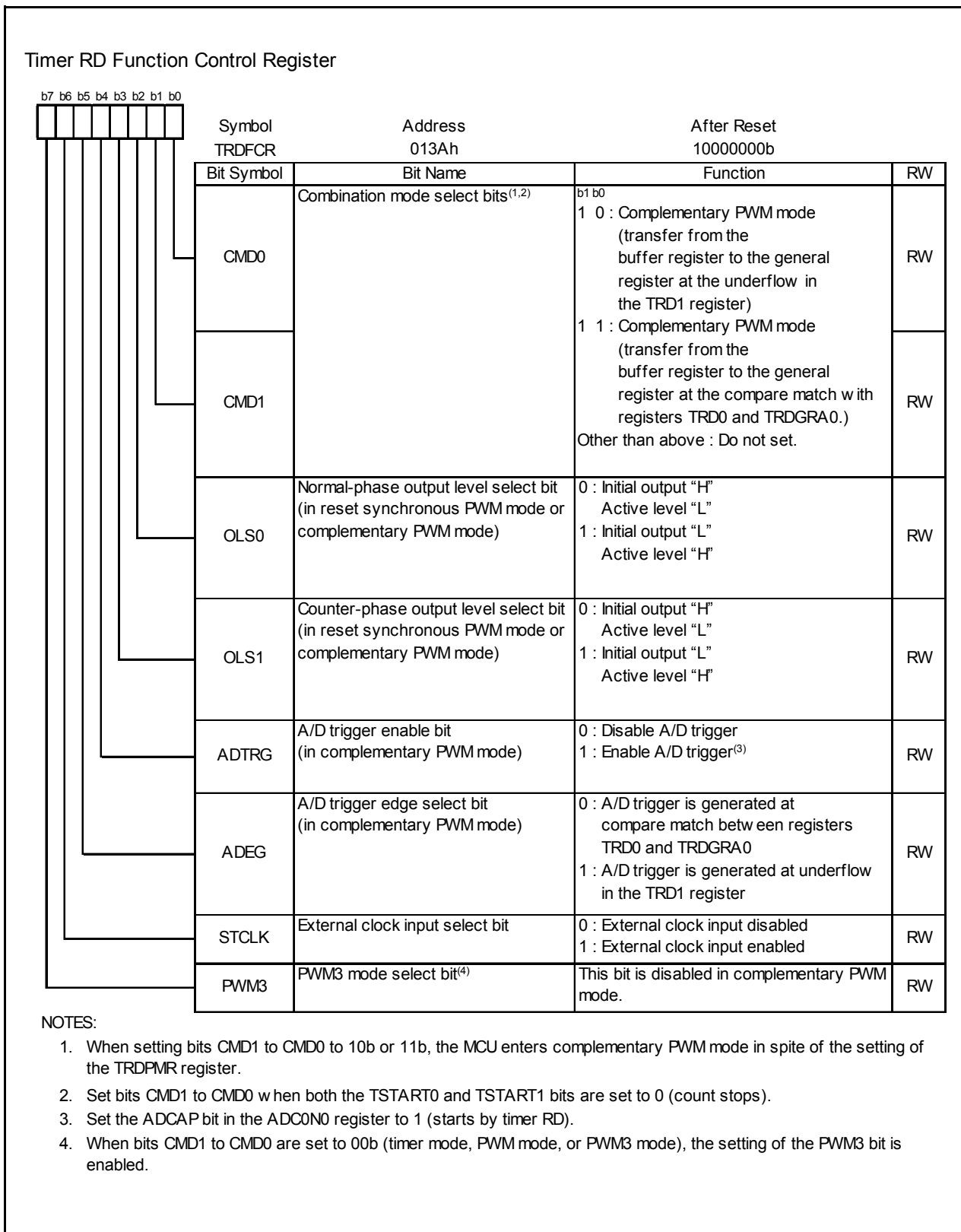


Figure 14.129 TRDFCR Register in Complementary PWM Mode

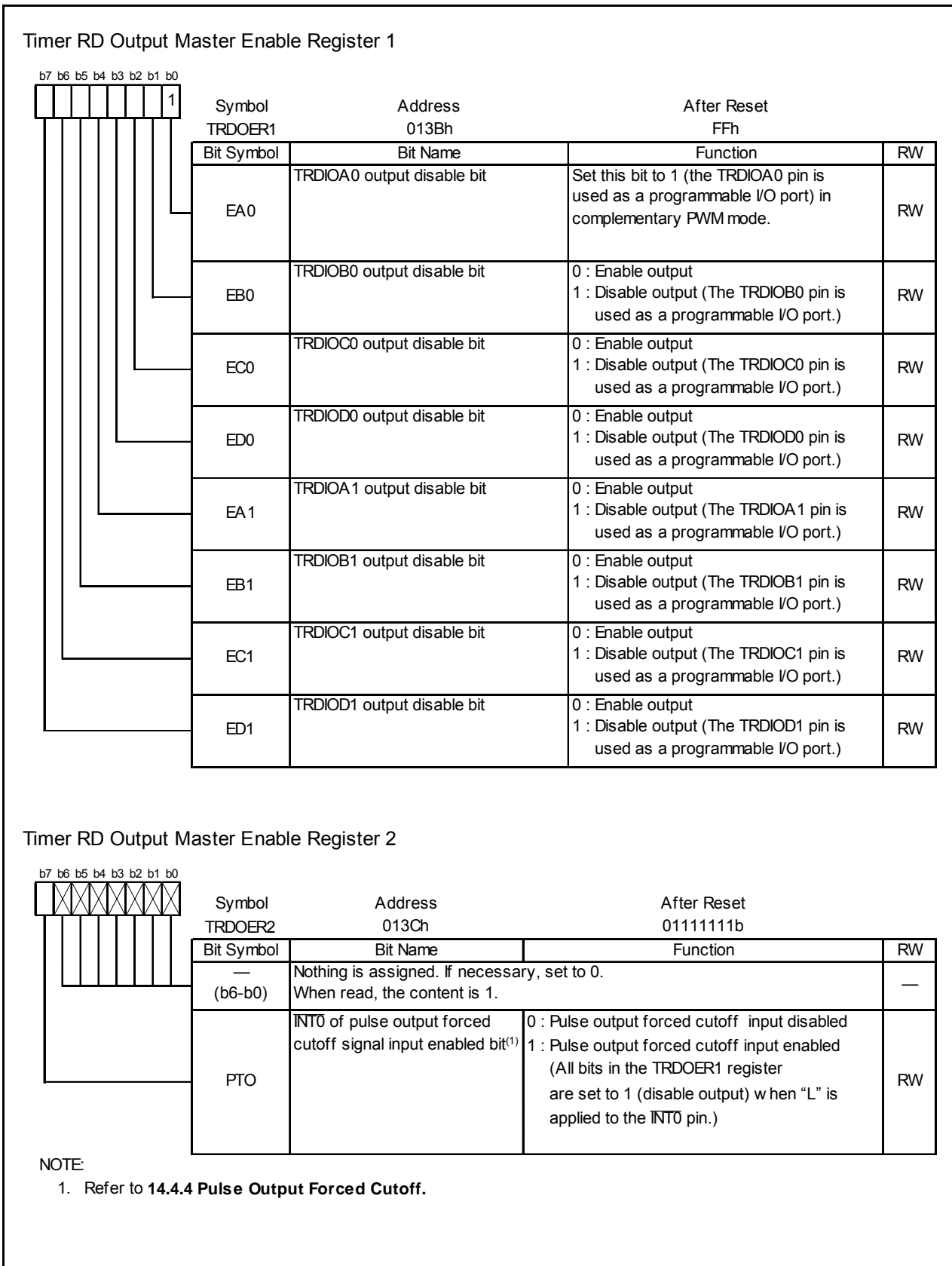


Figure 14.130 Registers TRDOER1 to TRDOER2 in Complementary PWM Mode

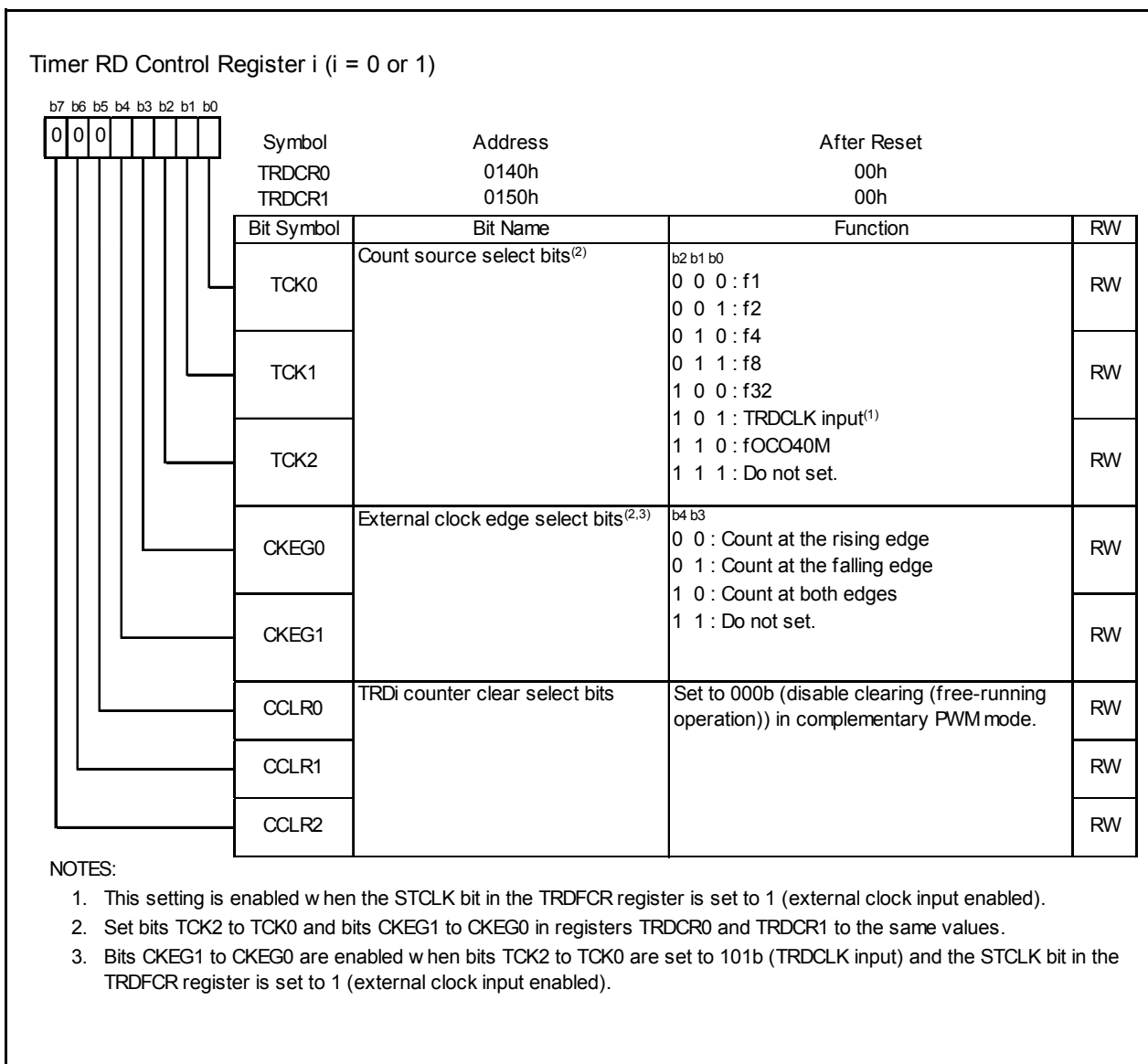


Figure 14.131 Registers TRDCR0 to TRDCR1 in Complementary PWM Mode

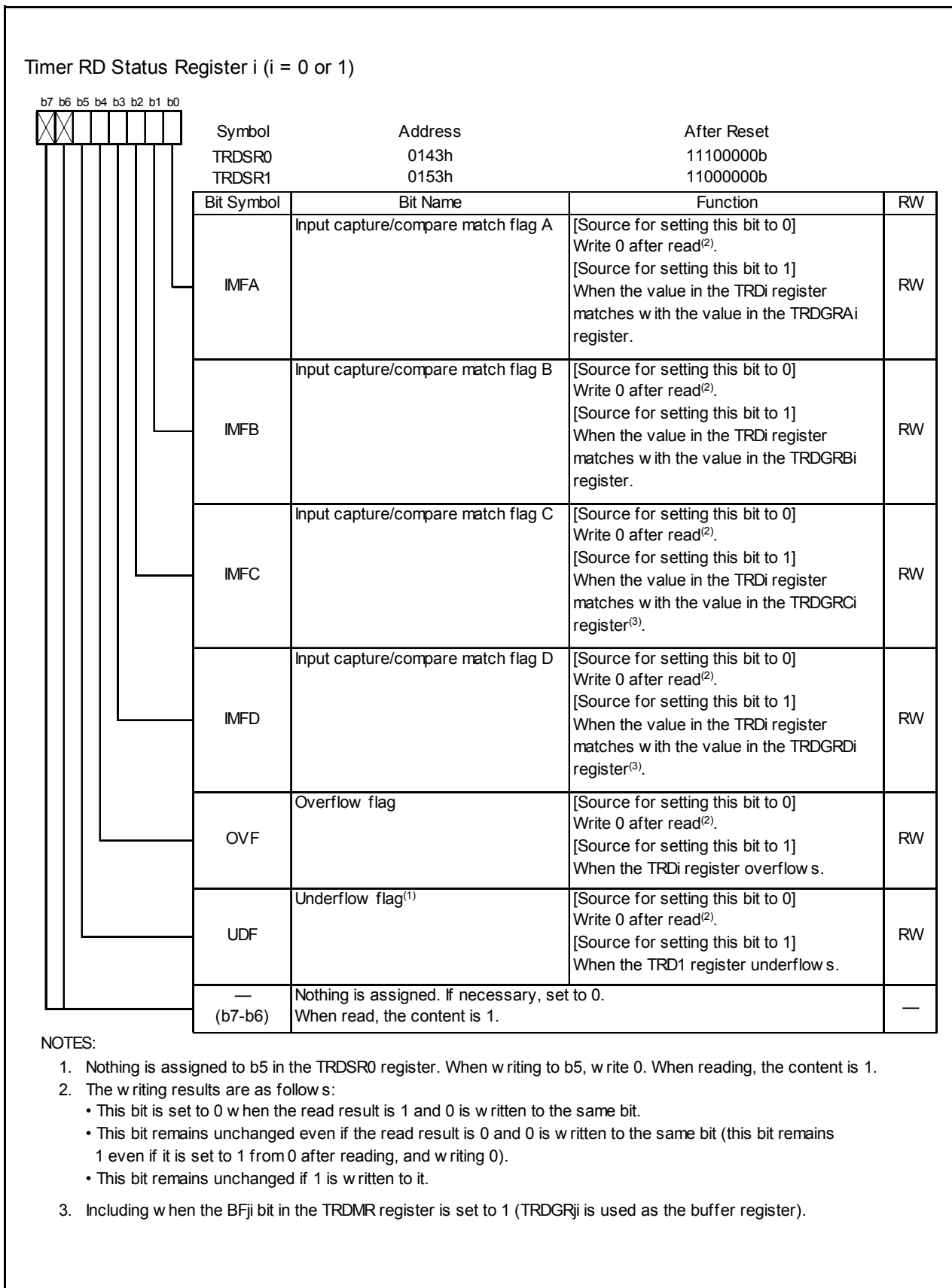


Figure 14.132 Registers TRDSR0 to TRDSR1 in Complementary PWM Mode

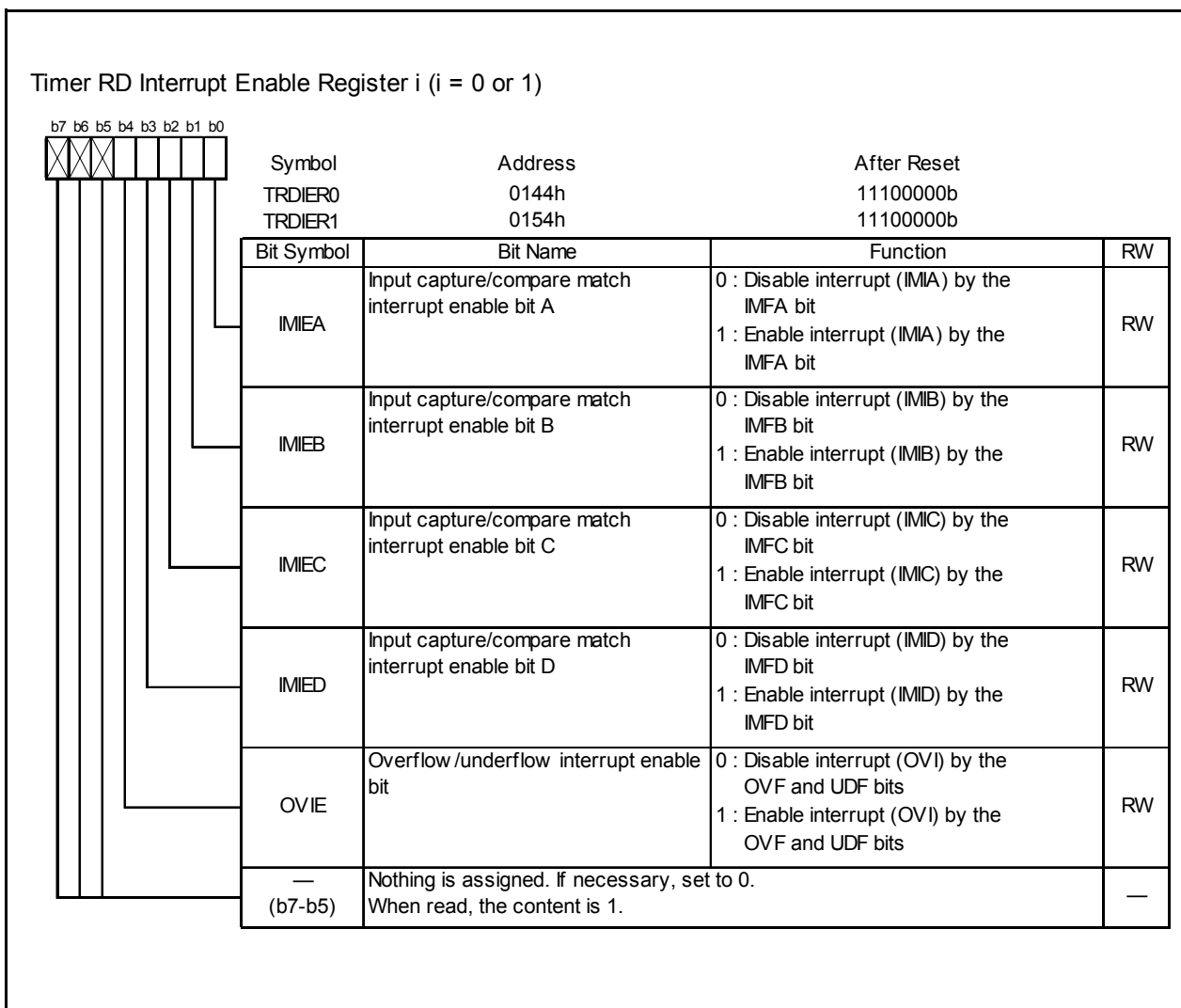


Figure 14.133 Registers TRDIER0 to TRDIER1 in Complementary PWM Mode

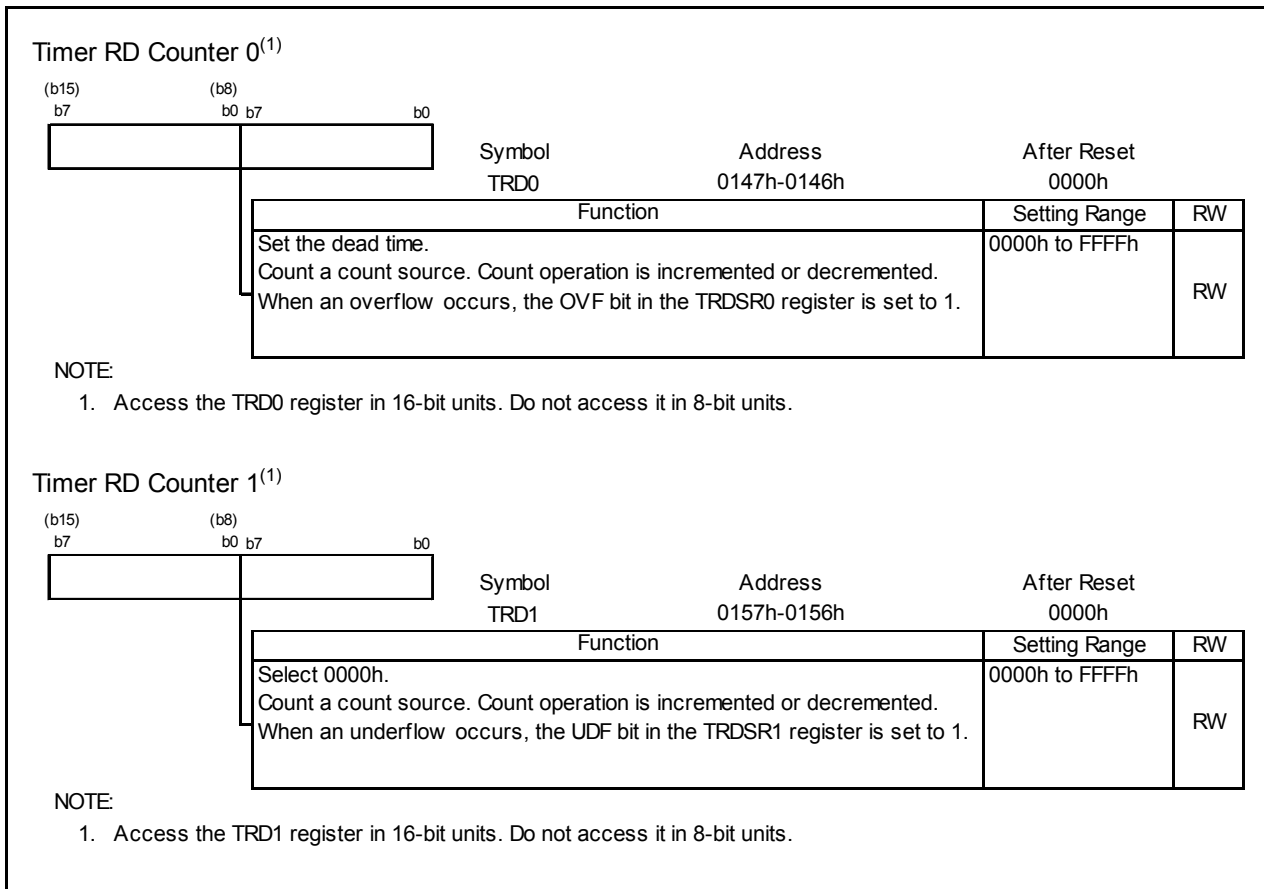


Figure 14.134 Registers TRD0 to TRD1 in Complementary PWM Mode

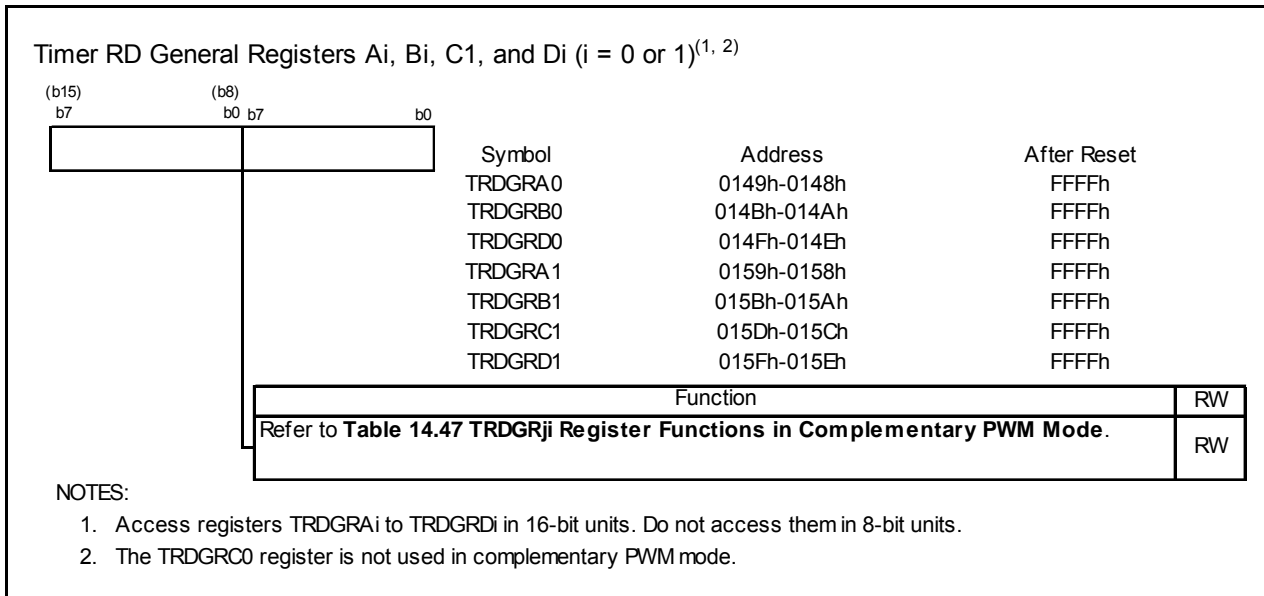


Figure 14.135 Registers TRDGRA_i, TRDGRB_i, TRDGRC1, and TRDGRD_i in Complementary PWM Mode

The following registers are disabled in the complementary PWM mode: TRDPMR, TRDOCR, TRDDF0, TRDDF1, TRDIORA0, TRDIORC0, TRDPOCR0, TRDIORA1, TRDIORC1, and TRDPOCR1.

Table 14.47 TRDGR*j*i Register Functions in Complementary PWM Mode

| Register | Setting | Register Function | PWM Output Pin |
|----------|----------|--|--|
| TRDGRA0 | – | General register. Set the PWM period at initialization. Setting range: Setting value or above in TRD0 register FFFFh - TRD0 register setting value or below Do not write to this register when the TSTART0 and TSTART1 bits in the TRDSTR register are set to 1 (count starts). | (Output inverted every half period of TRDIOC0 pin) |
| TRDGRB0 | – | General register. Set the changing point of PWM1 output at initialization. Setting range: Setting value or above in TRD0 register TRDGRA0 register - TRD0 register setting value or below Do not write to this register when the TSTART0 and TSTART1 bits in the TRDSTR register are set to 1 (count starts). | TRDIOB0 TRDIOD0 |
| TRDGRA1 | – | General register. Set the changing point of PWM2 output at initialization. Setting range: Setting value or above in TRD0 register TRDGRA0 register - TRD0 register setting value or below Do not write to this register when the TSTART0 and TSTART1 bits in the TRDSTR register are set to 1 (count starts). | TRDIOA1 TRDIOC1 |
| TRDGRB1 | – | General register. Set the changing point of PWM3 output at initialization. Setting range: Setting value or above in TRD0 register TRDGRA0 register - TRD0 register setting value or below Do not write to this register when the TSTART0 and TSTART1 bits in the TRDSTR register are set to 1 (count starts). | TRDIOB1 TRDIOD1 |
| TRDGRC0 | – | This register is not used in complementary PWM mode. | – |
| TRDGRD0 | BFD0 = 1 | Buffer register. Set the changing point of next PWM1 output. (Refer to 14.4.2 Buffer Operation .) Setting range: Setting value or above in TRD0 register TRDGRA0 register - TRD0 register setting value or below Set this register to the same value as the TRDGRB0 register for initialization. | TRDIOB0 TRDIOD0 |
| TRDGRC1 | BFC1 = 1 | Buffer register. Set the changing point of next PWM2 output. (Refer to 14.4.2 Buffer Operation .) Setting range: Setting value or above in TRD0 register TRDGRA0 register - TRD0 register setting value or below Set this register to the same value as the TRDGRA1 register for initialization. | TRDIOA1 TRDIOC1 |
| TRDGRD1 | BFD1 = 1 | Buffer register. Set the changing point of next PWM3 output. (Refer to 14.4.2 Buffer Operation .) Setting range: Setting value or above in TRD0 register TRDGRA0 register - TRD0 register setting value or below Set this register to the same value as the TRDGRB1 register for initialization. | TRDIOB1 TRDIOD1 |

BFC0, BFD0, BFC1, BFD1: Bits in TRDMR register

Since values cannot be written to the TRDGRB0, TRDGRA1, or TRDGRB1 register directly after count operation starts (prohibited item), use the TRDGRD0, TRDGRC1, or TRDGRD1 register as a buffer register. However, to write data to the TRDGRD0, TRDGRC1, or TRDGRD1 register, set bits BFD0, BFC1, and BFD1 to 0 (general register). After this, bits BFD0, BFC1, and BFD1 may be set to 1 (buffer register).

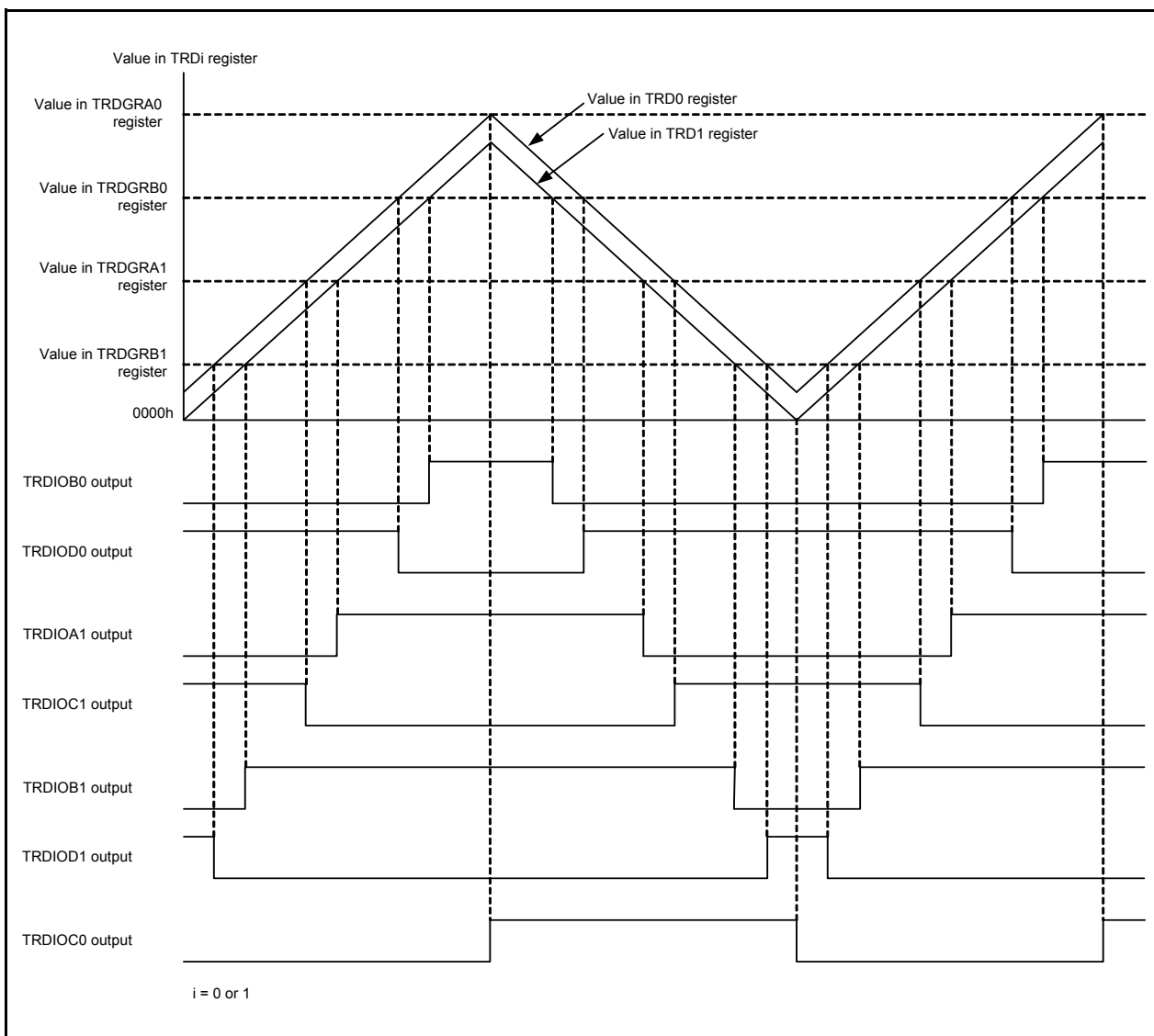


Figure 14.136 Output Model of Complementary PWM Mode

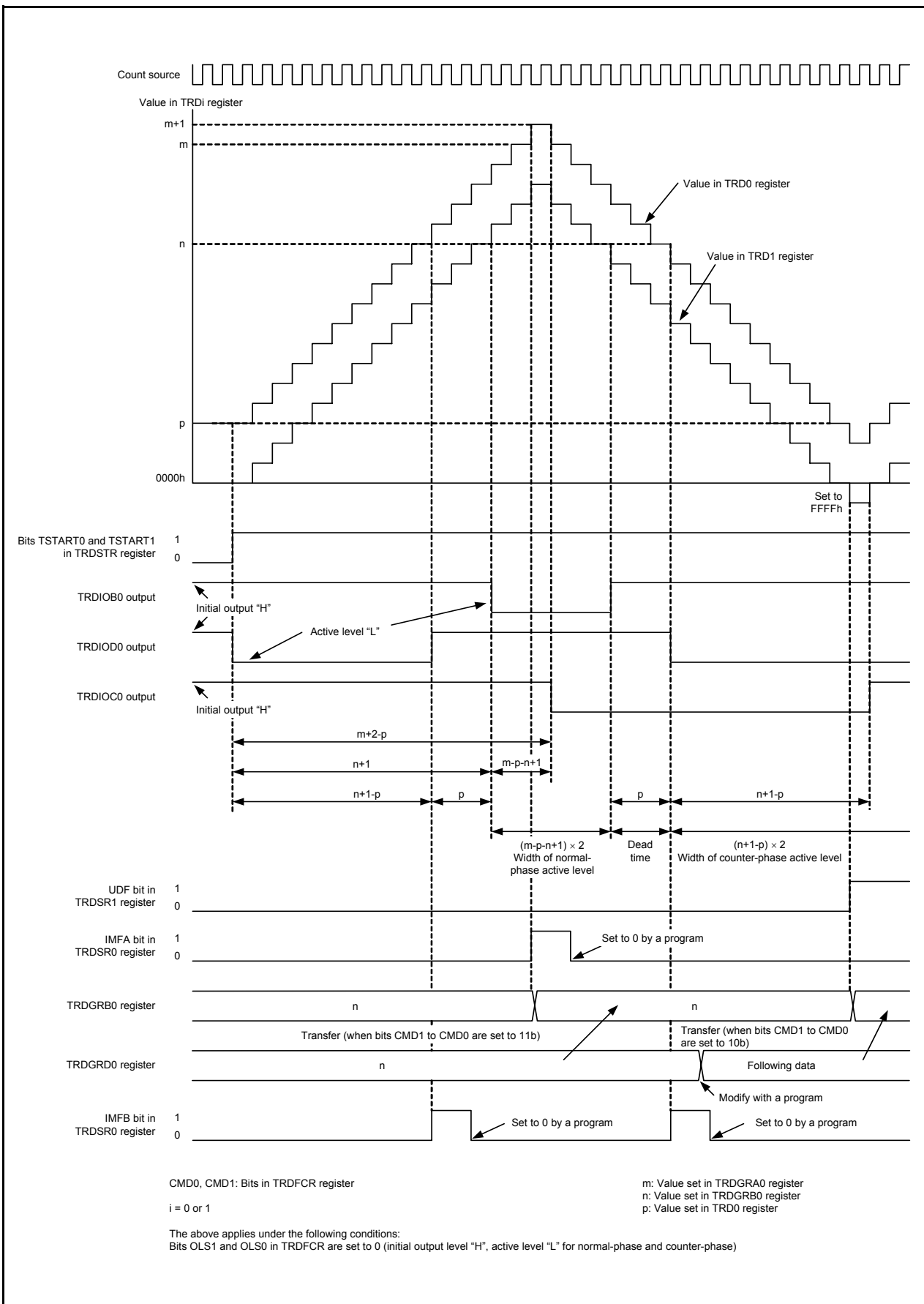


Figure 14.137 Operating Example of Complementary PWM Mode

14.4.9.1 Transfer Timing from Buffer Register

- Transfer from the TRDGRD0, TRDGRC1, or TRDGRD1 register to the TRDGRB0, TRDGRA1, or TRDGRB1 register.

When bits CMD1 to CMD0 in the TRDFCR register are set to 10b, the content is transferred when the TRD1 register underflows.

When bits CMD1 to CMD0 are set to 11b, the content is transferred at compare match between registers TRD0 and TRDGRA0.

14.4.9.2 A/D Trigger Generation

Compare match between registers TRD0 and TRDGRA0 and TRD1 underflow can be used as the conversion start trigger of the A/D converter. The trigger is selected by bits ADEG and ADTRG in the TRDFCR register. Also, set the ADCAP bit in the ADCON0 register to 1 (starts by timer RD).

14.4.10 PWM3 Mode

In this mode, 2 PWM waveforms are output with the same period.
 Figure 14.138 shows a Block Diagram of PWM3 Mode, and Table 14.48 lists the PWM3 Mode Specifications.
 Figures 14.139 to 14.149 show the registers associated with PWM3 mode, and Figure 14.150 shows an Operating Example of PWM3 Mode.

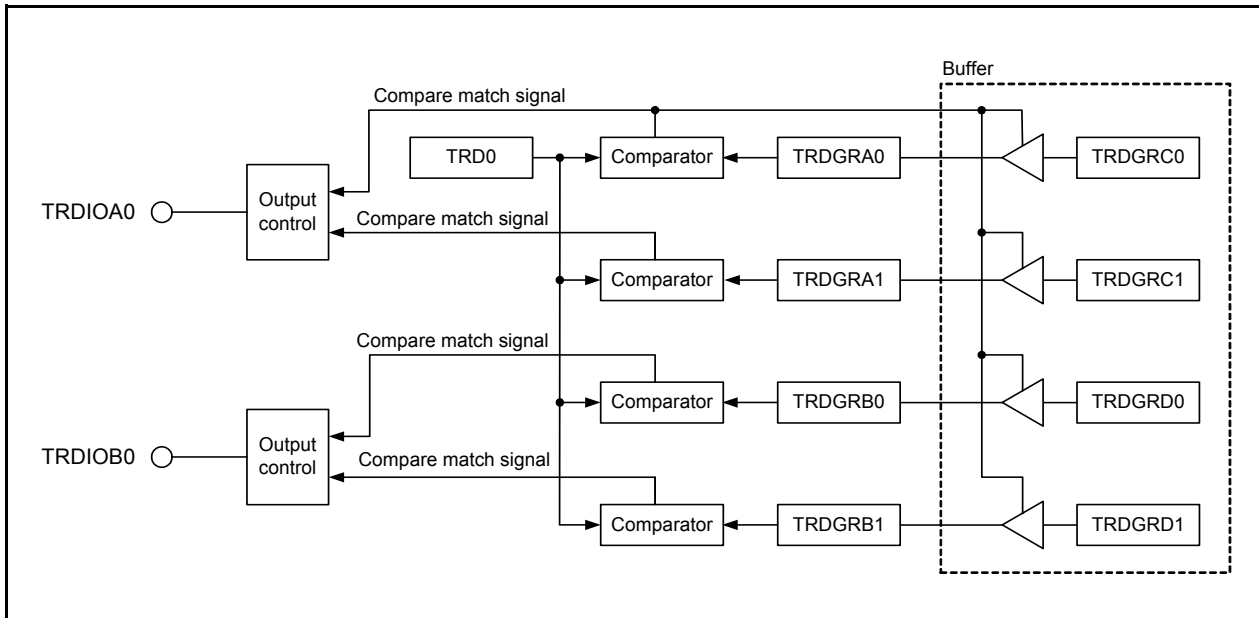


Figure 14.138 Block Diagram of PWM3 Mode

Table 14.48 PWM3 Mode Specifications

| Item | Specification |
|--|---|
| Count sources | f1, f2, f4, f8, f32, fOCO40M |
| Count operations | The TRD0 register is incremented (the TRD1 is not used). |
| PWM waveform | <p>PWM period: $1/f_k \times (m+1)$ Active level width of TRDIOA0 output: $1/f_k \times (m-n)$ Active level width of TRDIOB0 output: $1/f_k \times (p-q)$ f_k: Frequency of count source m: Value set in the TRDGRA0 register n: Value set in the TRDGRA1 register p: Value set in the TRDGRB0 register q: Value set in the TRDGRB1 register</p> <p>(When "H" is selected as the active level)</p> |
| Count start condition | 1 (count starts) is written to the TSTART0 bit in the TRDSTR register. |
| Count stop conditions | <ul style="list-style-type: none"> • 0 (count stops) is written to the TSTART0 bit in the TRDSTR register when the CSEL0 bit in the TRDSTR register is set to 1. The PWM output pin holds output level before the count stops • When the CSEL0 bit in the TRDSTR register is set to 0, the count stops at compare match with the TRDGRA0 register. The PWM output pin holds level after output change by compare match. |
| Interrupt request generation timing | <ul style="list-style-type: none"> • Compare match (The content of the TRDi register matches content of the TRDGRji register.) • The TRD0 register overflows |
| TRDIOA0, TRDIOB0 pin functions | PWM output |
| TRDIOC0, TRDIOD0, TRDIOA1 to TRDIOD1 pin functions | Programmable I/O port |
| INT0 pin function | Programmable I/O port, pulse output forced cutoff signal input, or INT0 interrupt input |
| Read from timer | The count value can be read by reading the TRD0 register. |
| Write to timer | The value can be written to the TRD0 register. |
| Select functions | <ul style="list-style-type: none"> • Pulse output forced cutoff signal input (Refer to 14.4.4 Pulse Output Forced Cutoff.) • Buffer Operation (Refer to 14.4.2 Buffer Operation.) • Active level selectable by pin |

$i = 0$ or 1 , $j =$ either A, B, C, or D

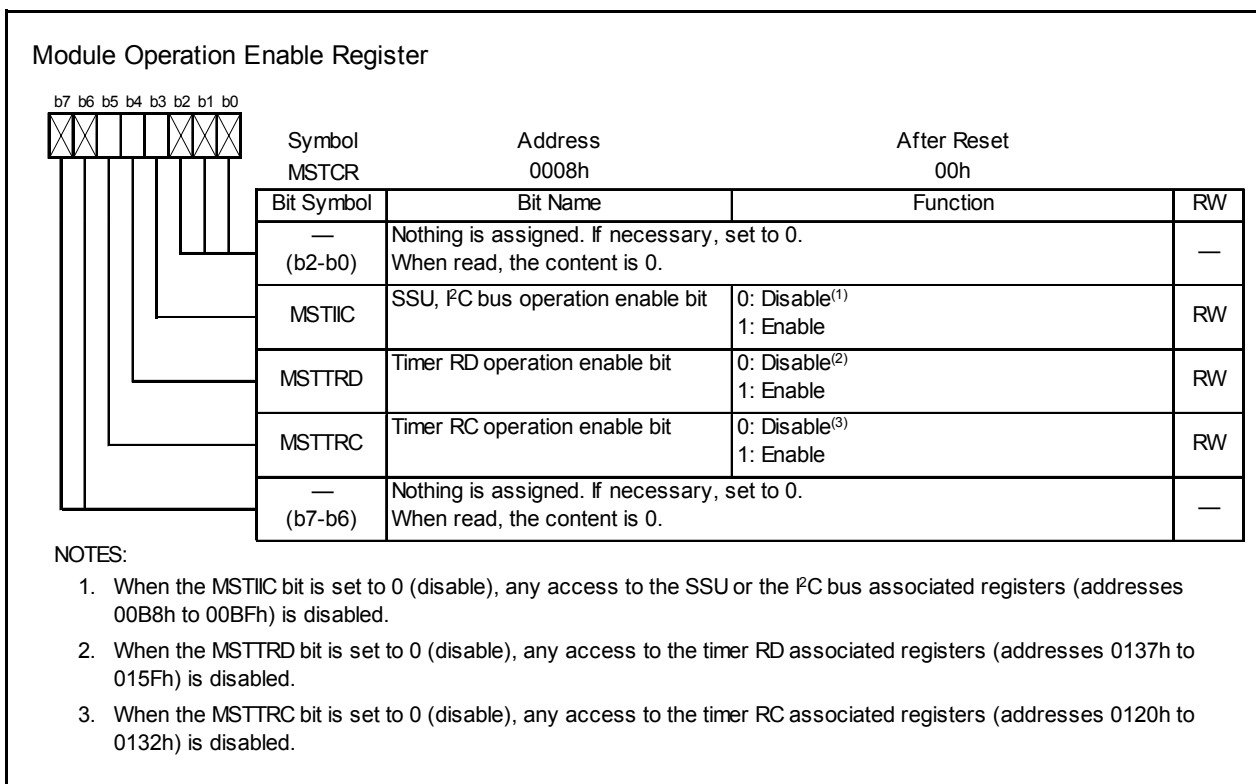


Figure 14.139 MSTCR Register

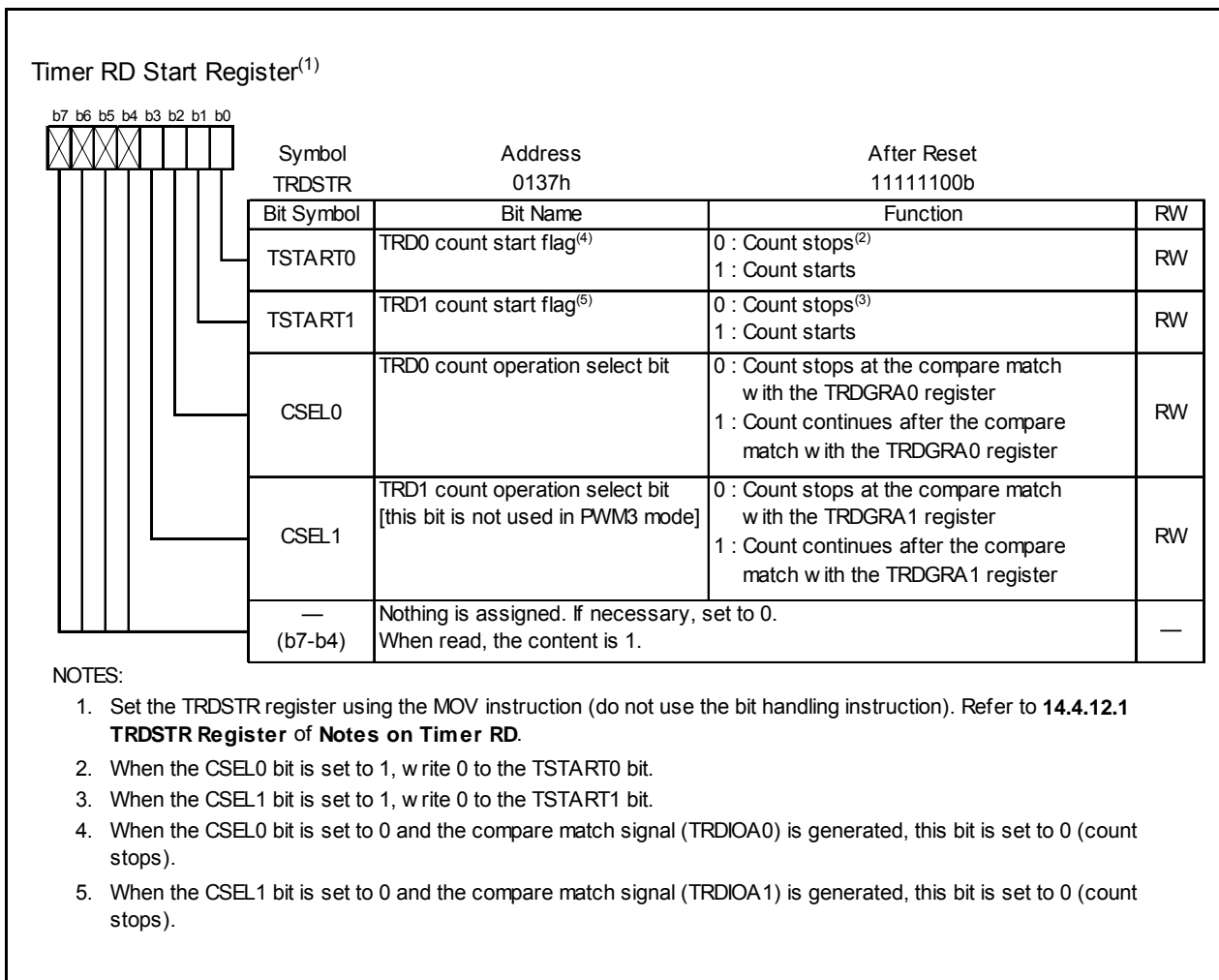


Figure 14.140 TRDSTR Register in PWM3 Mode

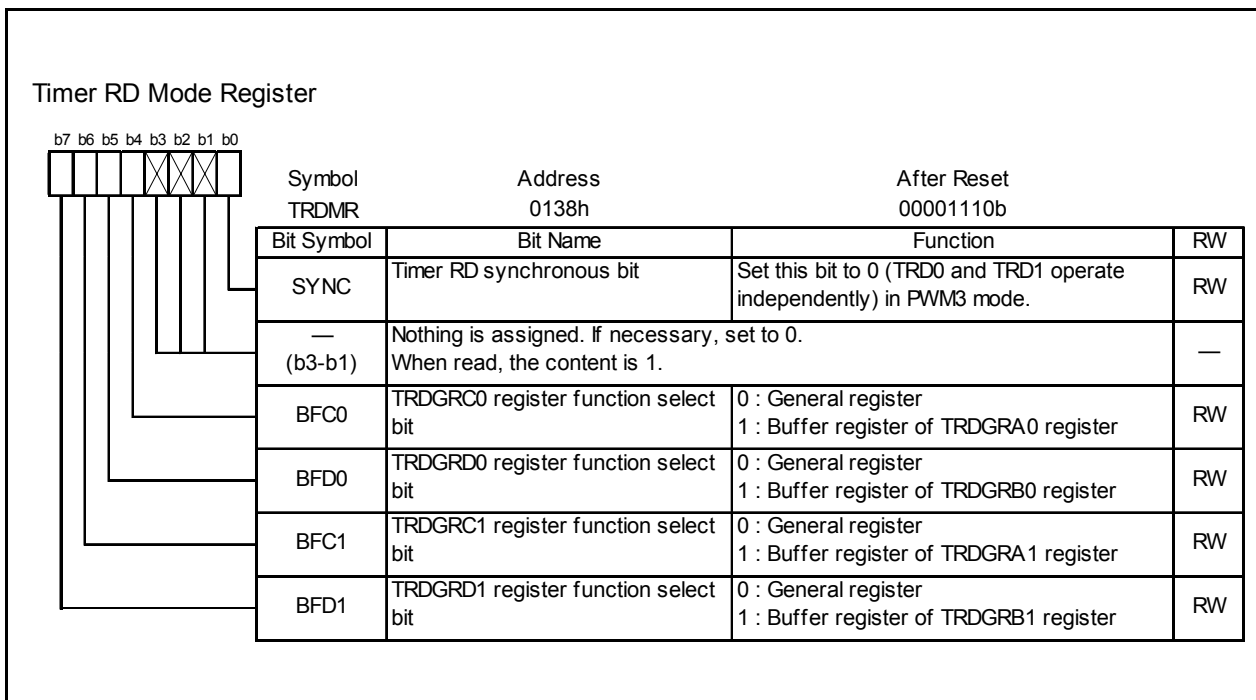


Figure 14.141 TRDMR Register in PWM3 Mode

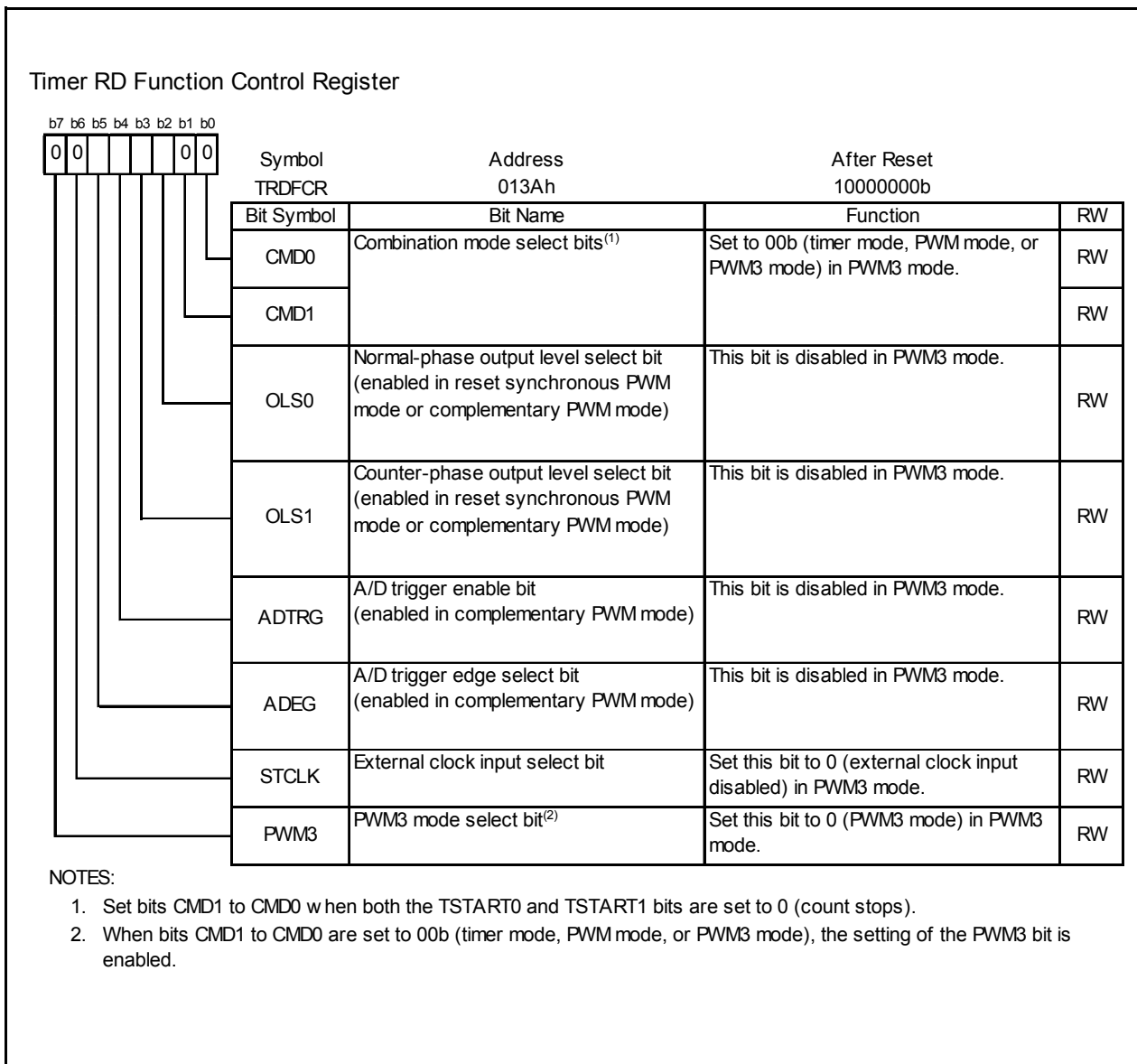


Figure 14.142 TRDFCR Register in PWM3 Mode

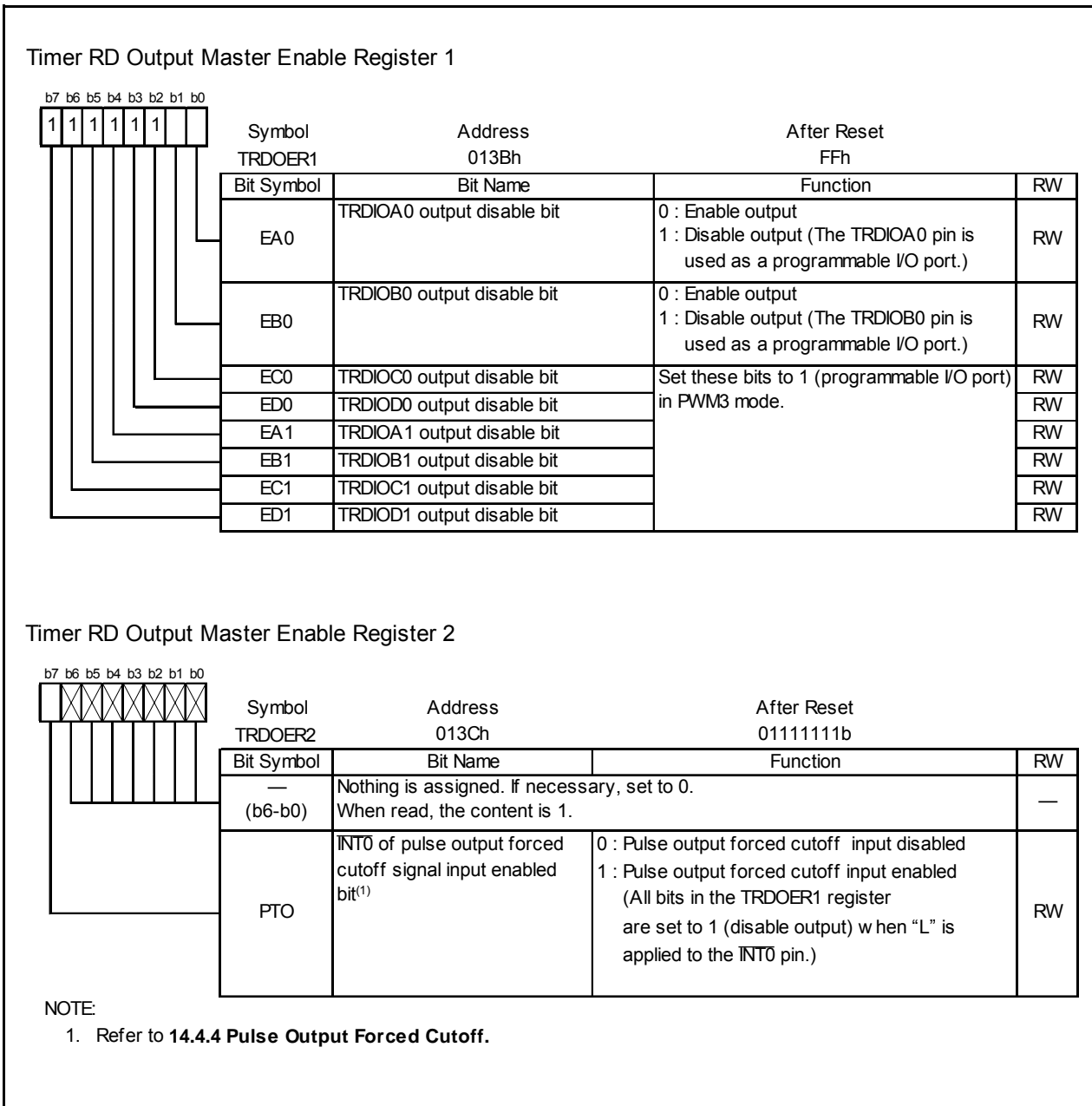


Figure 14.143 Registers TRDOER1 to TRDOER2 in PWM3 Mode

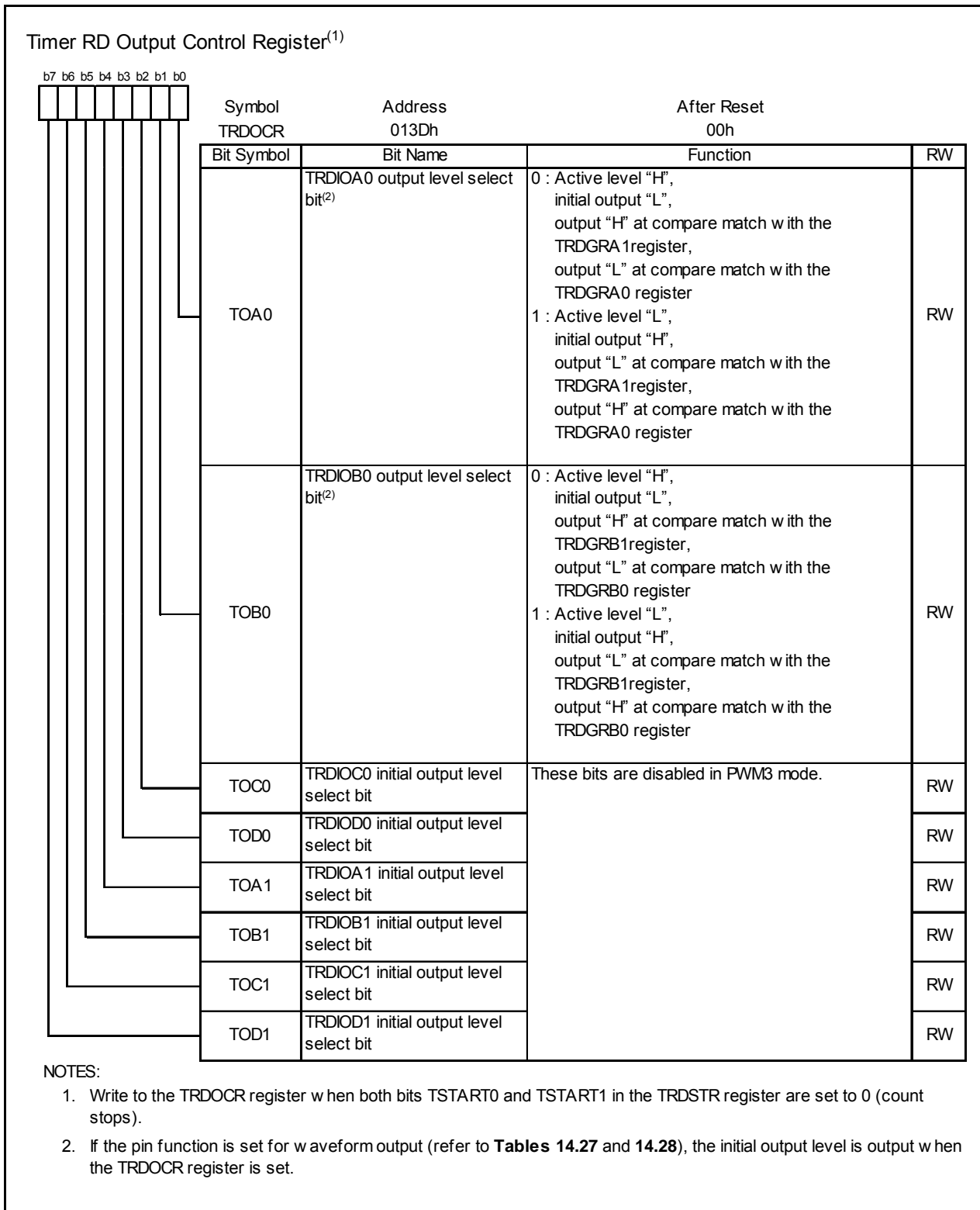


Figure 14.144 TRDOCR Register in PWM3 Mode

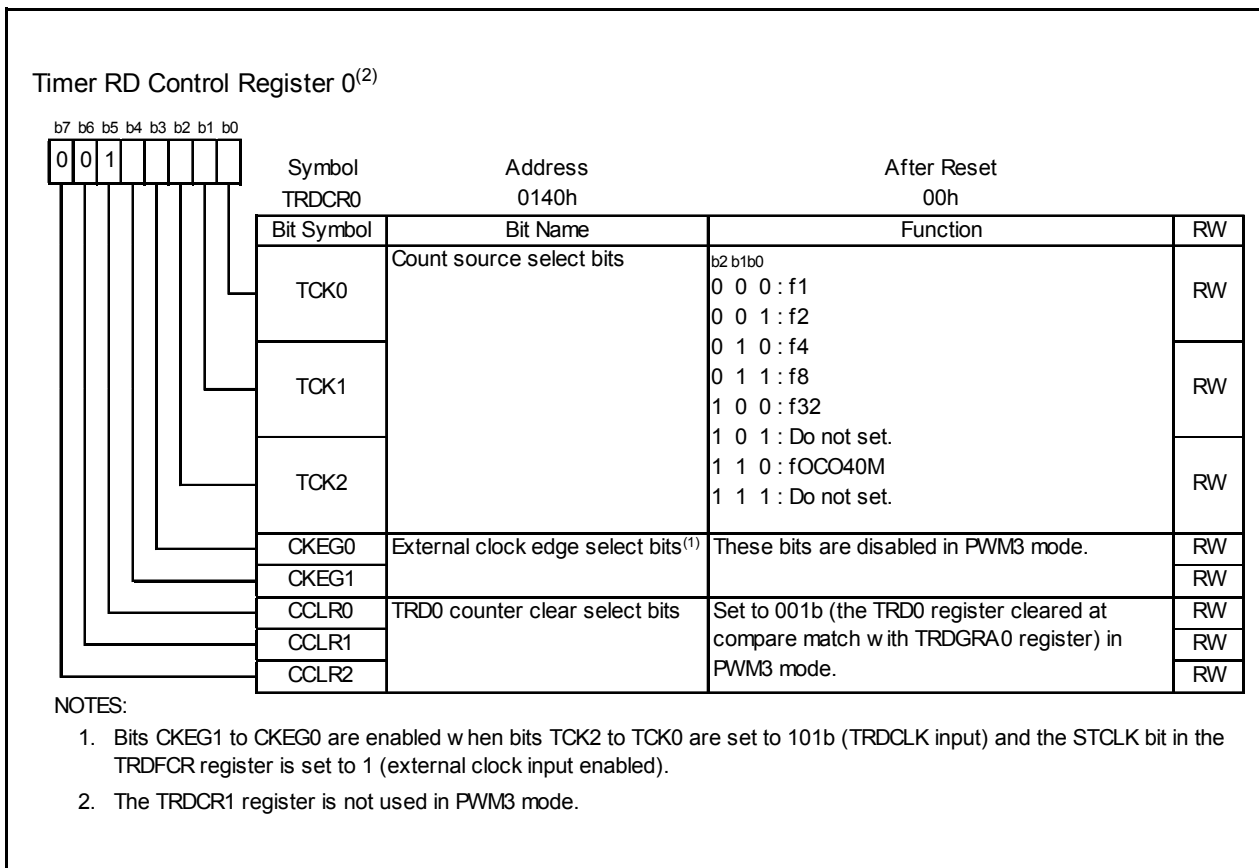


Figure 14.145 TRDCR0 Register in PWM3 Mode

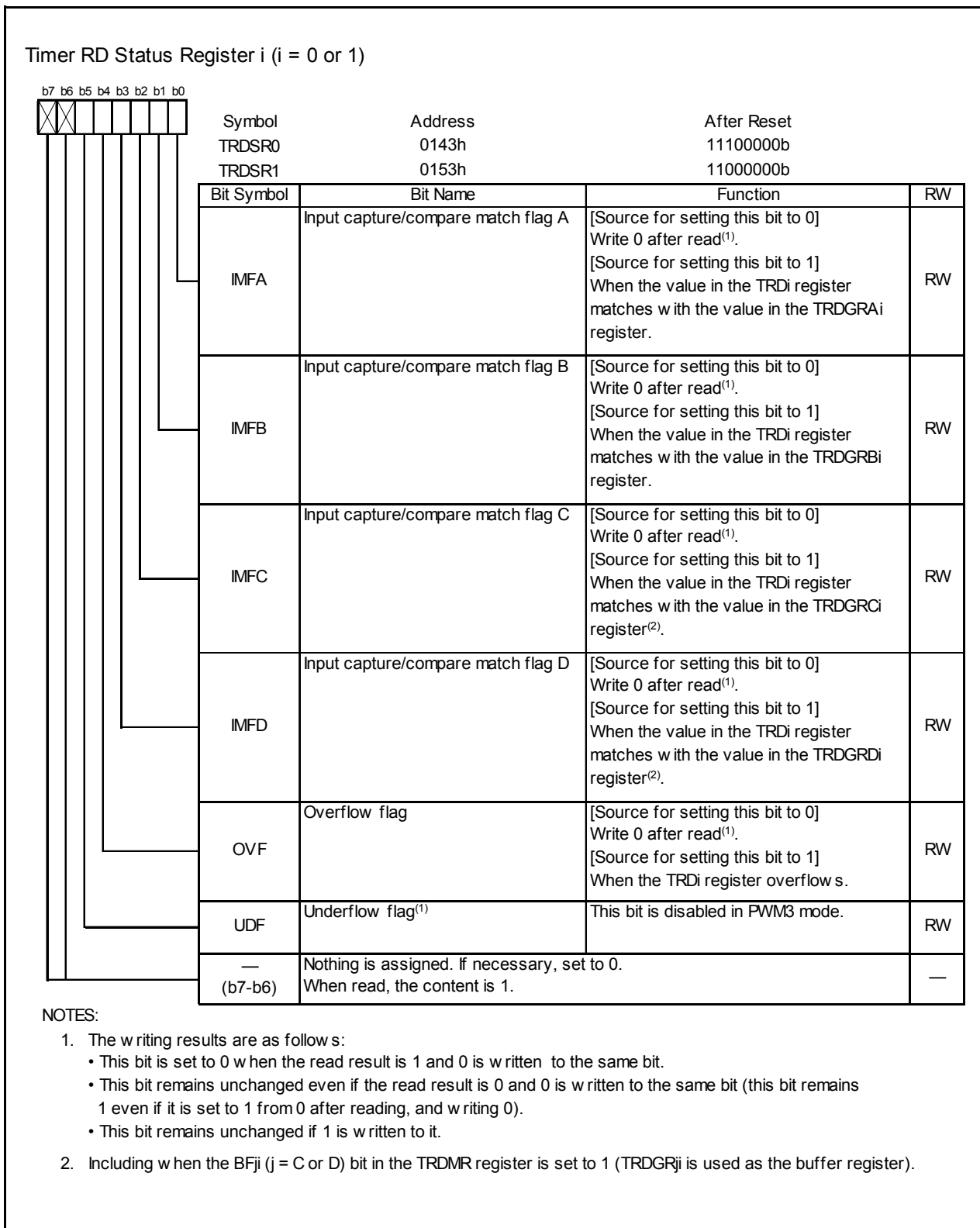


Figure 14.146 Registers TRDSR0 to TRDSR1 in PWM3 Mode

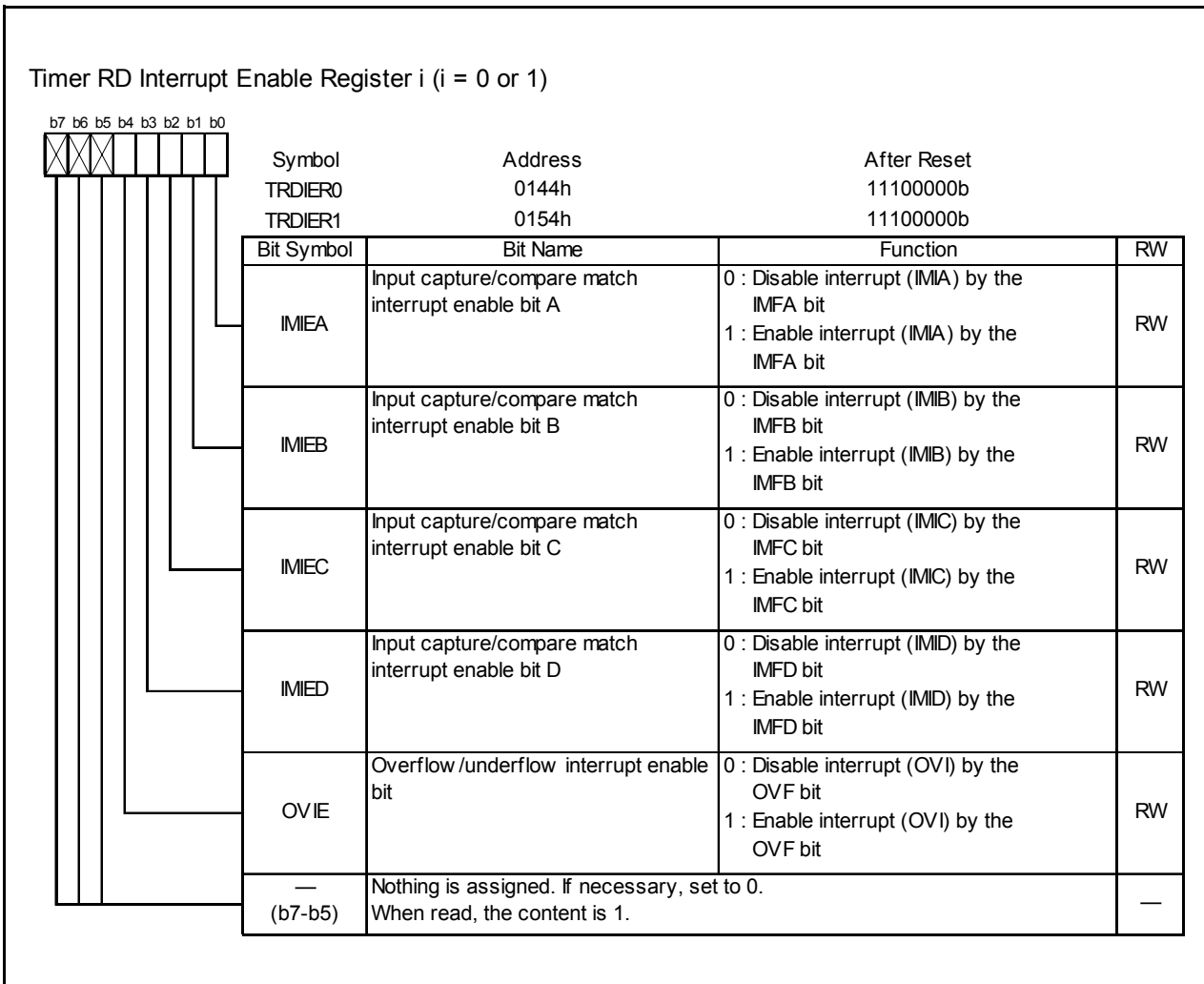


Figure 14.147 Registers TRDIER0 to TRDIER1 in PWM3 Mode

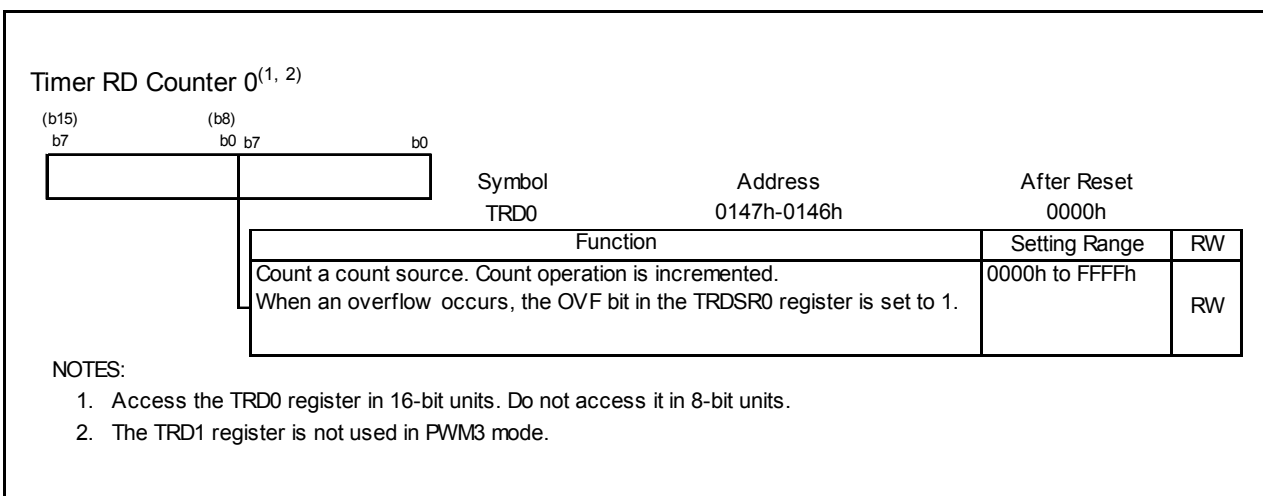


Figure 14.148 TRD0 Register in PWM3 Mode

Timer RD General Registers Ai, Bi, Ci, and Di (i = 0 or 1)⁽¹⁾

| (b15) b7 | | (b8) b0 b7 | | b0 | Symbol | Address | After Reset |
|--|--|---------------|--|----|---------|-------------|-------------|
| | | | | | TRDGRA0 | 0149h-0148h | FFFFh |
| | | | | | TRDGRB0 | 014Bh-014Ah | FFFFh |
| | | | | | TRDGRC0 | 014Dh-014Ch | FFFFh |
| | | | | | TRDGRD0 | 014Fh-014Eh | FFFFh |
| | | | | | TRDGRA1 | 0159h-0158h | FFFFh |
| | | | | | TRDGRB1 | 015Bh-015Ah | FFFFh |
| | | | | | TRDGRC1 | 015Dh-015Ch | FFFFh |
| | | | | | TRDGRD1 | 015Fh-015Eh | FFFFh |
| Function | | | | | | | RW |
| Refer to Table 14.49 TRDGRji Register Functions in PWM3 mode. | | | | | | | RW |

NOTE:
1. Access registers TRDGRAi to TRDGRDi in 16-bit units. Do not access them in 8-bit units.

Figure 14.149 Registers TRDGRAi, TRDGRBi, TRDGRCi, and TRDGRDi in PWM3 Mode

The following registers are disabled in the PWM3 mode function: TRDPMR, TRDDF0, TRDDF1, TRDIORA0, TRDIORC0, TRDPOCR0, TRDIORA1, TRDIORC1, and TRDPOCR1.

Table 14.49 TRDGR_{ji} Register Functions in PWM3 Mode

| Register | Setting | Register Function | PWM Output Pin |
|----------|----------|---|----------------|
| TRDGRA0 | – | General register. Set the PWM period. Setting range: Value set in TRDGRA1 register or above | TRDIOA0 |
| TRDGRA1 | | General register. Set the changing point (the active level timing) of PWM output. Setting range: Value set in TRDGRA0 register or below | |
| TRDGRB0 | | General register. Set the changing point (the timing that returns to initial output level) of PWM output. Setting range: Value set in TRDGRB1 register or above Value set in TRDGRA0 register or below | TRDIOB0 |
| TRDGRB1 | | General register. Set the changing point (active level timing) of PWM output. Setting range: Value set in TRDGRB0 register or below | |
| TRDGRC0 | BFC0 = 0 | (These registers is not used in PWM3 mode.) | – |
| TRDGRC1 | BFC1 = 0 | | |
| TRDGRD0 | BFD0 = 0 | | |
| TRDGRD1 | BFD1 = 0 | | |
| TRDGRC0 | BFC0 = 1 | Buffer register. Set the next PWM period. (Refer to 14.4.2 Buffer Operation .) Setting range: Value set in TRDGRC1 register or above | TRDIOA0 |
| TRDGRC1 | BFC1 = 1 | Buffer register. Set the changing point of next PWM output. (Refer to 14.4.2 Buffer Operation .) Setting range: Value set in TRDGRC0 register or below | |
| TRDGRD0 | BFD0 = 1 | Buffer register. Set the changing point of next PWM output. (Refer to 14.4.2 Buffer Operation .) Setting range: Value set in TRDGRD1 register or above, setting value or below in TRDGRC0 register. | TRDIOB0 |
| TRDGRD1 | BFD1 = 1 | Buffer register. Set the changing point of next PWM output. (Refer to 14.4.2 Buffer Operation .) Setting range: Value set in TRDGRD0 register or below | |

BFC0, BFD0, BFC1, BFD1: Bits in TRDMR register

Registers TRDGRC0, TRDGRC1, TRDGRD0, and TRDGRD1 are not used in PWM3 mode. To use them as buffer registers, set bits BFC0, BFC1, BFD0, and BFD1 to 0 (general register) and write a value to the TRDGRC0, TRDGRC1, TRDGRD0, or TRDGRD1 register. After this, bits BFC0, BFC1, BFD0, and BFD1 may be set to 1 (buffer register).

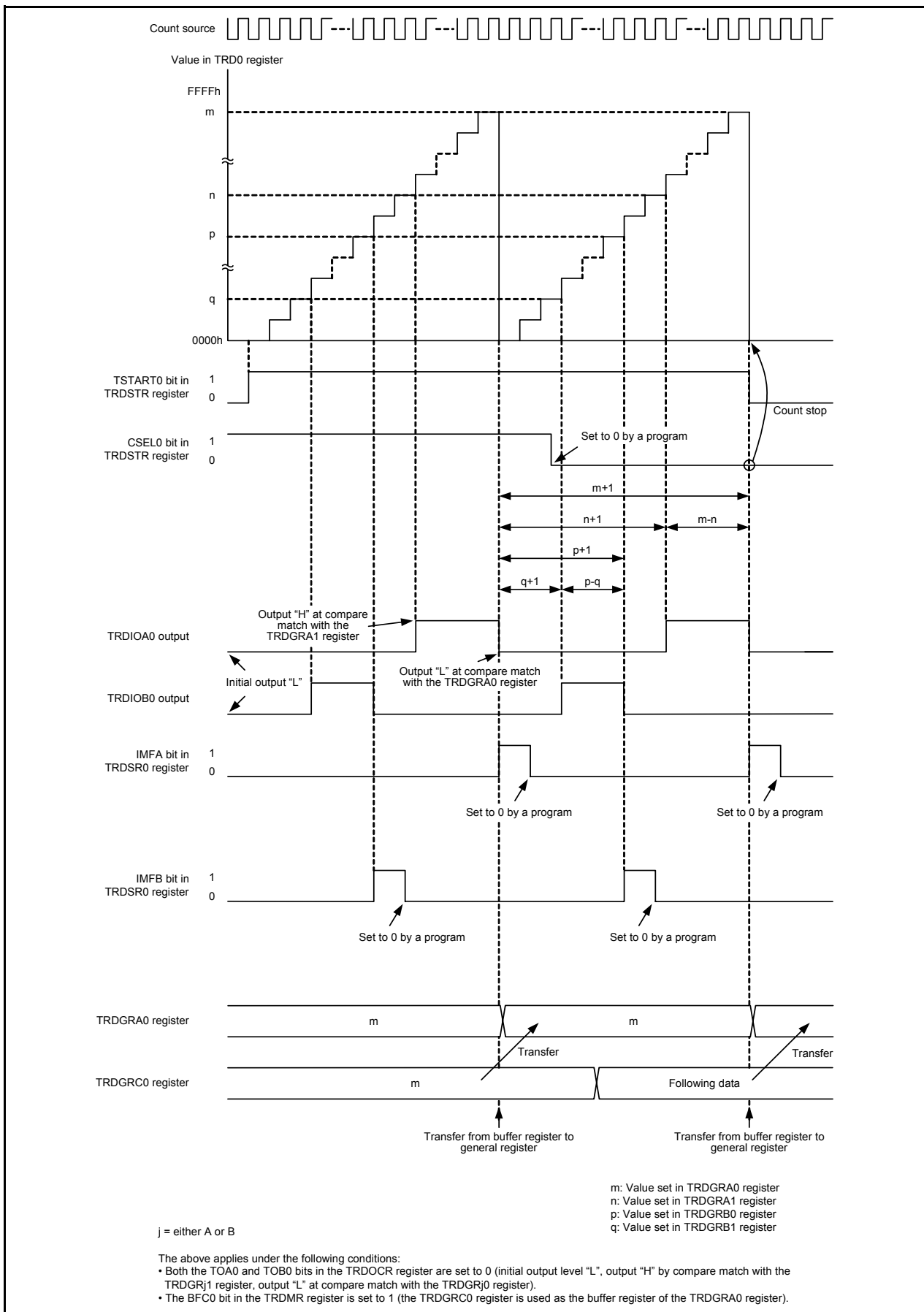


Figure 14.150 Operating Example of PWM3 Mode

14.4.11 Timer RD Interrupt

Timer RD generates the timer RD interrupt request based on 6 sources for each channel. The timer RD interrupt has 1 TRDiIC register (bits IR, and ILVL0 to ILVL2), and 1 vector for each channel. Table 14.50 lists the Registers Associated with Timer RD Interrupt, and Figure 14.151 shows a Block Diagram of Timer RD Interrupt.

Table 14.50 Registers Associated with Timer RD Interrupt

| | Timer RD Status Register | Timer RD Interrupt Enable Register | Timer RD Interrupt Control Register |
|-----------|--------------------------|------------------------------------|-------------------------------------|
| Channel 0 | TRDSR0 | TRDIER0 | TRD0IC |
| Channel 1 | TRDSR1 | TRDIER1 | TRD1IC |

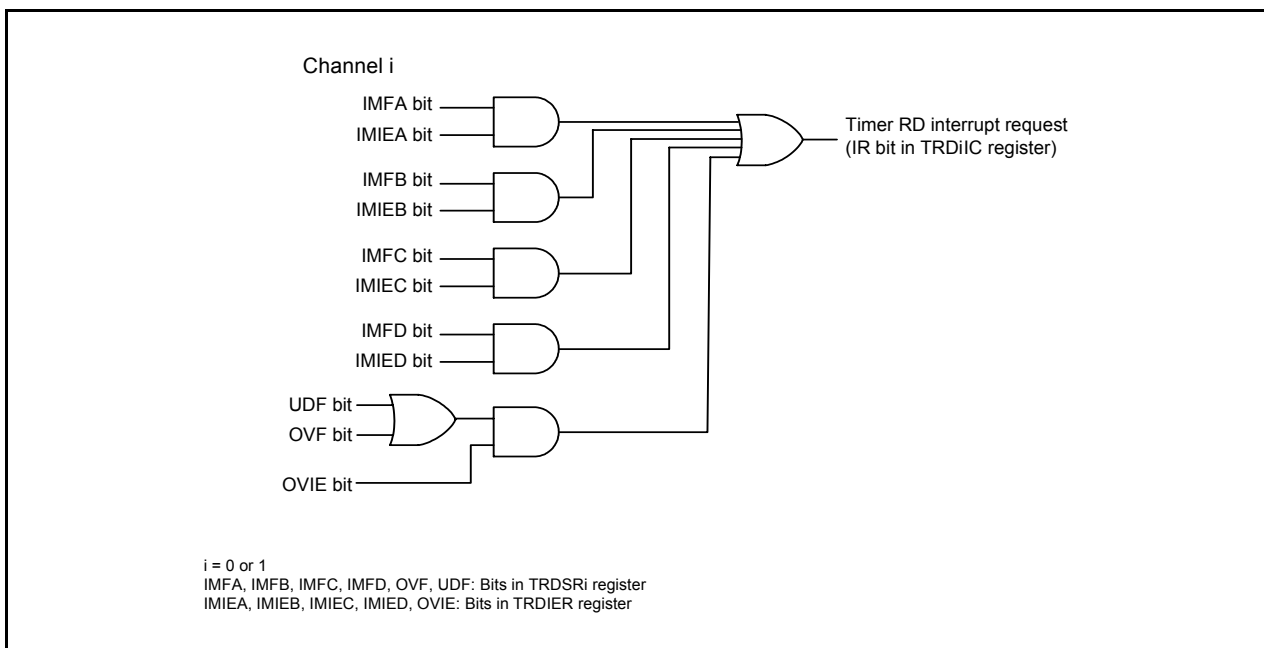


Figure 14.151 Block Diagram of Timer RD Interrupt

As with other maskable interrupts, the timer RD interrupt is controlled by the combination of the I flag, IR bit, bits ILVL0 to ILVL2, and IPL. However, since the interrupt source (timer RD interrupt) is generated by a combination of multiple interrupt request sources, the following differences from other maskable interrupts apply:

- When bits in the TRDSRi register corresponding to bits set to 1 in the TRDIERi register are set to 1 (enable interrupt), the IR bit in the TRDiIC register is set to 1 (interrupt requested).
- When either bits in the TRDSRi register or bits in the TRDIERi register corresponding to bits in the TRDSRi register, or both of them, are set to 0, the IR bit is set to 0 (interrupt not requested). Therefore, even though the interrupt is not acknowledged after the IR bit is set to 1, the interrupt request will not be maintained.
- When the conditions of other request sources are met, the IR bit remains 1.
- When multiple bits in the TRDIERi register are set to 1, which request source causes an interrupt is determined by the TRDSRi register.
- Since each bit in the TRDSRi register is not automatically set to 0 even if the interrupt is acknowledged, set each bit to 0 in the interrupt routine. For information on how to set these bits to 0, refer to the descriptions of the registers used in the different modes (Figures 14.77, 14.93, 14.107, 14.120, 14.132, and 14.146).

Refer to **Registers TRDSR0 to TRDSR1 in each mode (Figures 14.77, 14.93, 14.107, 14.120, 14.132, and 14.146)** for the TRDSRi register. Refer to **Registers TRDIER0 to TRDIER1 in each mode (Figures 14.78, 14.94, 14.108, 14.121, 14.133, and 14.147)** for the TRDIERi register.

Refer to **12.1.6 Interrupt Control** for information on the TRDiC register and **12.1.5.2 Relocatable Vector Tables** for the interrupt vectors.

14.4.12 Notes on Timer RD

14.4.12.1 TRDSTR Register

- Set the TRDSTR register using the MOV instruction.
- When the CSELi (i = 0 to 1) is set to 0 (the count stops at compare match of registers TRDi and TRDGRAi), the count does not stop and the TSTARTi bit remains unchanged even if 0 (count stops) is written to the TSTARTi bit.
- Therefore, set the TSTARTi bit to 0 to change other bits without changing the TSTARTi bit when the CSELi bit is set to 0.
- To stop counting by a program, set the TSTARTi bit after setting the CSELi bit to 1. Although the CSELi bit is set to 1 and the TSTARTi bit is set to 0 at the same time (with 1 instruction), the count cannot be stopped.
- Table 14.51 lists the TRDIOji (j = A, B, C, or D) Pin Output Level when Count Stops to use the TRDIOji (j = A, B, C, or D) pin with the timer RD output.

Table 14.51 TRDIOji (j = A, B, C, or D) Pin Output Level when Count Stops

| Count Stop | TRDIOji Pin Output when Count Stops |
|---|--|
| When the CSELi bit is set to 1, set the TSTARTi bit to 0 and the count stops. | Hold the output level immediately before the count stops. |
| When the CSELi bit is set to 0, the count stops at compare match of registers TRDi and TRDGRAi. | Hold the output level after output changes by compare match. |

14.4.12.2 TRDi Register (i = 0 or 1)

- When writing the value to the TRDi register by a program while the TSTARTi bit in the TRDSTR register is set to 1 (count starts), avoid overlapping with the timing for setting the TRDi register to 0000h, and then write. If the timing for setting the TRDi register to 0000h overlaps with the timing for writing the value to the TRDi register, the value is not written and the TRDi register is set to 0000h.
These precautions are applicable when selecting the following by bits CCLR2 to CCLR0 in the TRDCRi register.
 - 001b (Clear by the TRDi register at compare match with the TRDGRAi register.)
 - 010b (Clear by the TRDi register at compare match with the TRDGRBi register.)
 - 011b (Synchronous clear)
 - 101b (Clear by the TRDi register at compare match with the TRDGRCi register.)
 - 110b (Clear by the TRDi register at compare match with the TRDGRDi register.)

- When writing the value to the TRDi register and continuously reading the same register, the value before writing may be read. In this case, execute the JMP.B instruction between the writing and reading.

```

Program example      MOV.W      #XXXXh, TRD0      ;Writing
                    JMP.B      L1              ;JMP.B
                    L1:        MOV.W      TRD0,DATA    ;Reading
  
```

14.4.12.3 TRDSRi Register (i = 0 or 1)

When writing the value to the TRDSRi register and continuously reading the same register, the value before writing may be read. In this case, execute the JMP.B instruction between the writing and reading.

```

Program example      MOV.B      #XXh, TRDSR0    ;Writing
                    JMP.B      L1              ;JMP.B
                    L1:        MOV.B      TRDSR0,DATA ;Reading
  
```

14.4.12.4 Count Source Switch

- Switch the count source after the count stops.

Change procedure

- (1) Set the TSTART_i (i = 0 or 1) bit in the TRDSTR register to 0 (count stops).
- (2) Change bits TCK2 to TCK0 in the TRDCR_i register.

- When changing the count source from fOCO40M to another source and stopping fOCO40M, wait 2 cycles of f₁ or more after setting the clock switch, and then stop fOCO40M.

Change procedure

- (1) Set the TSTART_i (i = 0 or 1) bit in the TRDSTR register to 0 (count stops).
- (2) Change bits TCK2 to TCK0 in the TRDCR_i register.
- (3) Wait 2 or more cycles of f₁.
- (4) Set the FRA00 bit in the FRA0 register to 0 (high-speed on-chip oscillator stops).

14.4.12.5 Input Capture Function

- Set the pulse width of the input capture signal to 3 or more cycles of the timer RD operation clock (refer to **Table 14.26 Timer RD Operation Clocks**).
- The value in the TRD_i register is transferred to the TRDGR_{ji} register 2 to 3 cycles of the timer RD operation clock after the input capture signal is applied to the TRDIO_{ji} pin (i = 0 or 1, j = either A, B, C, or D) (no digital filter).

14.4.12.6 Reset Synchronous PWM Mode

- When reset synchronous PWM mode is used for motor control, make sure OLS0 = OLS1.
- Set to reset synchronous PWM mode by the following procedure:

Change procedure

- (1) Set the TSTART0 bit in the TRDSTR register to 0 (count stops).
- (2) Set bits CMD1 to CMD0 in the TRDFCR register to 00b (timer mode, PWM mode, and PWM3 mode).
- (3) Set bits CMD1 to CMD0 to 01b (reset synchronous PWM mode).
- (4) Set the other registers associated with timer RD again.

14.4.12.7 Complementary PWM Mode

- When complementary PWM mode is used for motor control, make sure OLS0 = OLS1.
- Change bits CMD1 to CMD0 in the TRDFCR register in the following procedure.

Change procedure: When setting to complementary PWM mode (including re-set), or changing the transfer timing from the buffer register to the general register in complementary PWM mode.

- (1) Set both the TSTART0 and TSTART1 bits in the TRDSTR register to 0 (count stops).
- (2) Set bits CMD1 to CMD0 in the TRDFCR register to 00b (timer mode, PWM mode, and PWM3 mode).
- (3) Set bits CMD1 to CMD0 to 10b or 11b (complementary PWM mode).
- (4) Set the registers associated with other timer RD again.

Change procedure: When stopping complementary PWM mode

- (1) Set both the TSTART0 and TSTART1 bits in the TRDSTR register to 0 (count stops).
- (2) Set bits CMD1 to CMD0 to 00b (timer mode, PWM mode, and PWM3 mode).

- Do not write to TRDGRA0, TRDGRB0, TRDGRA1, or TRDGRB1 register during operation. When changing the PWM waveform, transfer the values written to registers TRDGRD0, TRDGRC1, and TRDGRD1 to registers TRDGRB0, TRDGRA1, and TRDGRB1 using the buffer operation. However, to write data to the TRDGRD0, TRDGRC1, or TRDGRD1 register, set bits BFD0, BFC1, and BFD1 to 0 (general register). After this, bits BFD0, BFC1, and BFD1 may be set to 1 (buffer register). The PWM period cannot be changed.

- If the value in the TRDGRA0 register is assumed to be m , the TRD0 register counts $m-1$, m , $m+1$, m , $m-1$, in that order, when changing from increment to decrement operation.

When changing from m to $m+1$, the IMFA bit is set to 1. Also, bits CMD1 to CMD0 in the TRDFCR register are set to 11b (complementary PWM mode, buffer data transferred at compare match between registers TRD0 and TRDGRA0), the content in the buffer registers (TRDGRD0, TRDGRC1, and TRDGRD1) is transferred to the general registers (TRDGRB0, TRDGRA1, and TRDGRB1).

During $m+1$, m , and $m-1$ operation, the IMFA bit remains unchanged and data are not transferred to registers such as the TRDGRA0 register.

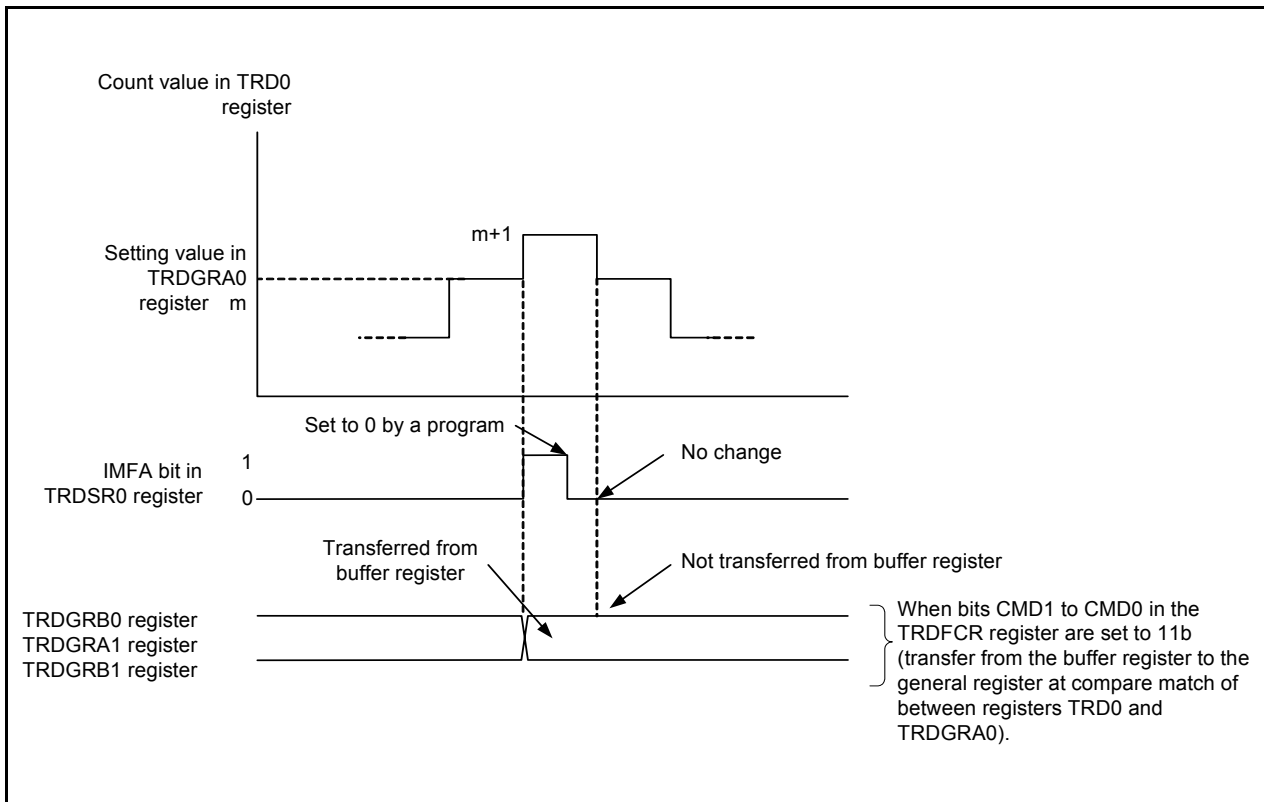


Figure 14.152 Operation at Compare Match between Registers TRD0 and TRDGRA0 in Complementary PWM Mode

- The TRD1 register counts 1, 0, FFFFh, 0, 1, in that order, when changing from decrement to increment operation.

The UDF bit is set to 1 when changing between 1, 0, and FFFFh operation. Also, when bits CMD1 to CMD0 in the TRDFCR register are set to 10b (complementary PWM mode, buffer data transferred at underflow in the TRD1 register), the content in the buffer registers (TRDGRD0, TRDGRC1, and TRDGRD1) is transferred to the general registers (TRDGRB0, TRDGRA1, and TRDGRB1). During FFFFh, 0, 1 operation, data are not transferred to registers such as the TRDGRB0 register. Also, at this time, the OVF bit remains unchanged.

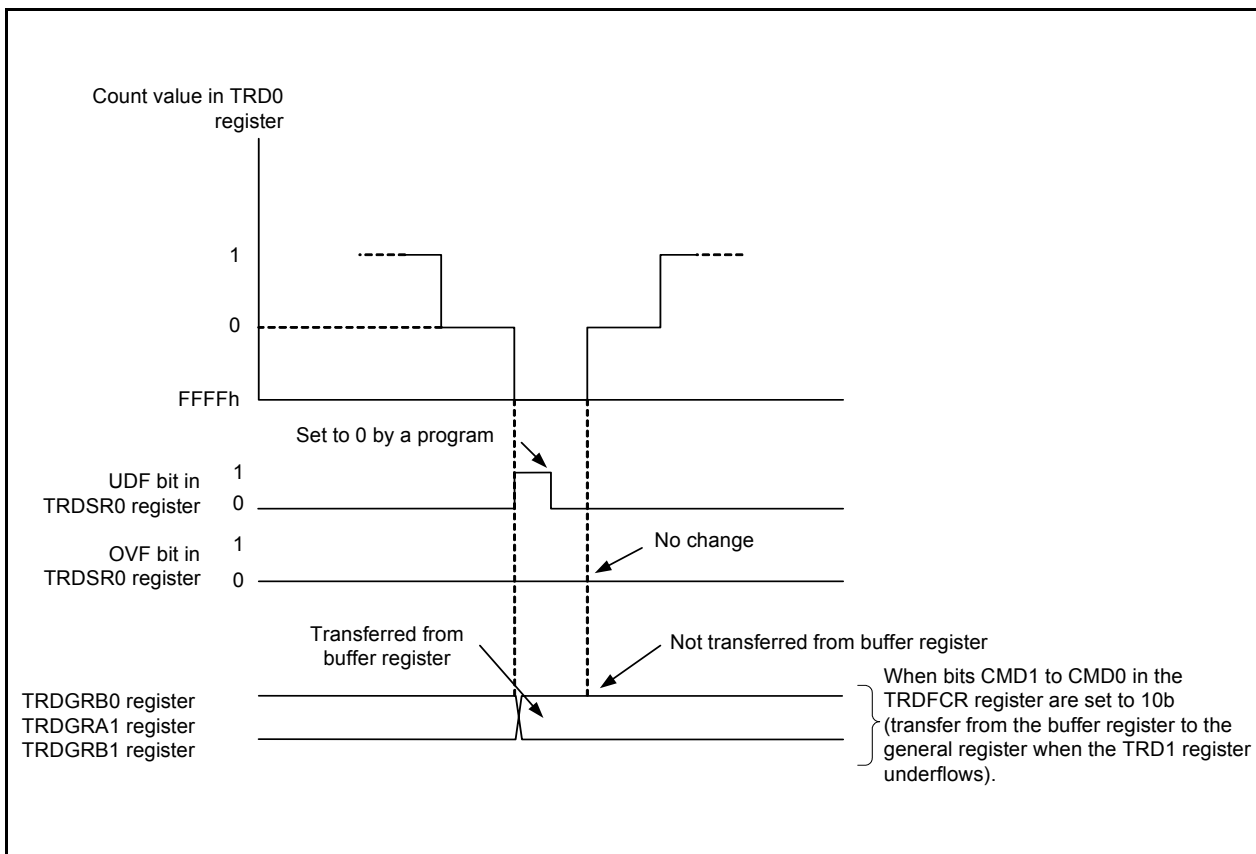


Figure 14.153 Operation when TRD1 Register Underflows in Complementary PWM Mode

- Select with bits CMD1 to CMD0 the timing of data transfer from the buffer register to the general register. However, transfer takes place with the following timing in spite of the value of bits CMD1 to CMD0 in the following cases:

Value in buffer register \geq value in TRDGRA0 register:

Transfer take place at underflow of the TRD1 register.

After this, when the buffer register is set to 0001h or above and a smaller value than the value of the TRDGRA0 register, and the TRD1 register underflows for the first time after setting, the value is transferred to the general register. After that, the value is transferred with the timing selected by bits CMD1 to CMD0.

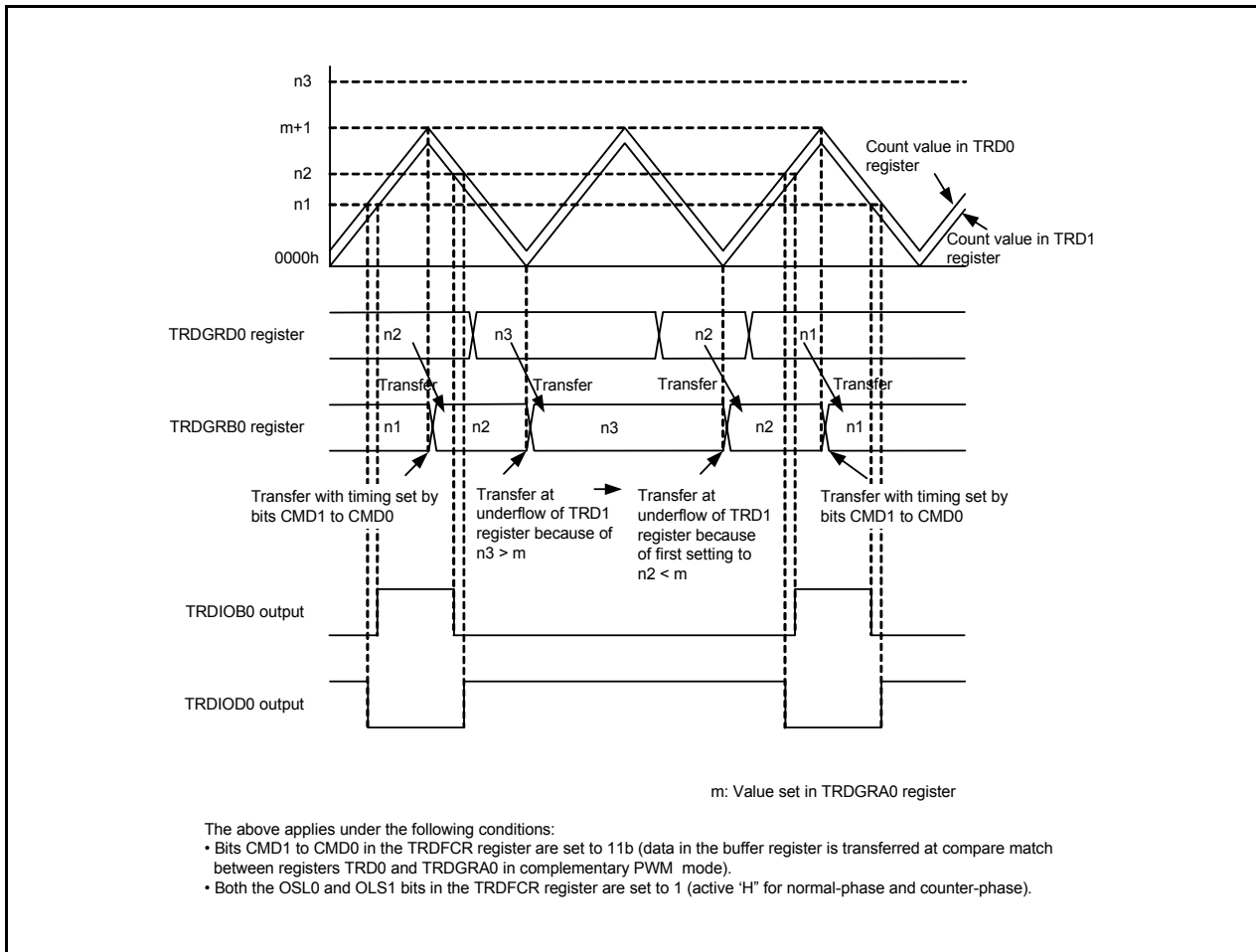


Figure 14.154 Operation when Value in Buffer Register \geq Value in TRDGRA0 Register in Complementary PWM Mode

When the value in the buffer register is set to 0000h:

Transfer takes place at compare match between registers TRD0 and TRDGRA0.

After this, when the buffer register is set to 0001h or above and a smaller value than the value of the TRDGRA0 register, and a compare match occurs between registers TRD0 and TRDGRA0 for the first time after setting, the value is transferred to the general register. After that, the value is transferred with the timing selected by bits CMD1 to CMD0.

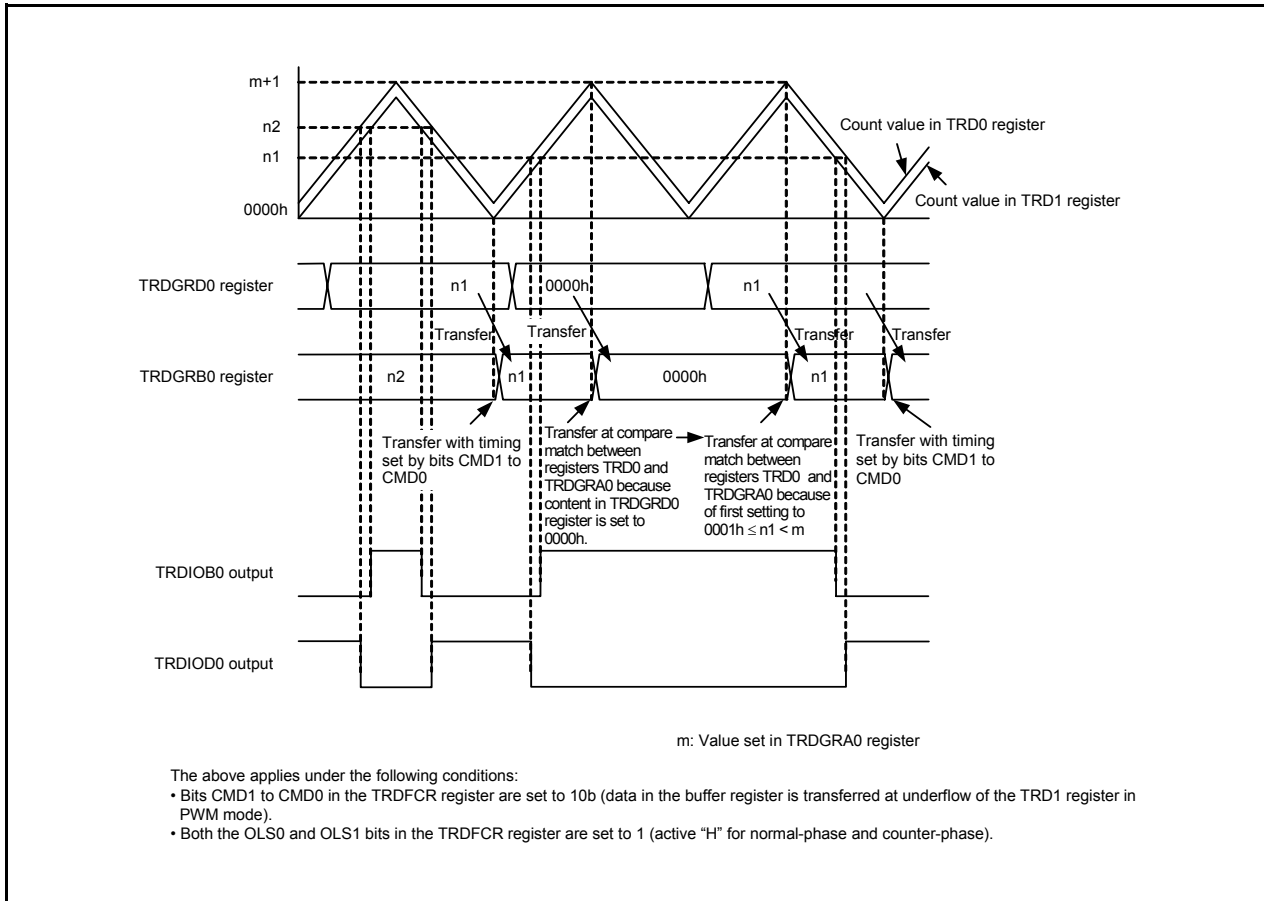


Figure 14.155 Operation when Value in Buffer Register Is Set to 0000h in Complementary PWM Mode

14.4.12.8 Count Source fOCO40M

- The count source fOCO40M can be used with supply voltage $V_{CC} = 3.0$ to 5.5 V. For supply voltage other than that, do not set bits TCK2 to TCK0 in registers TRDCR0 and TRDCR to 110b (select fOCO40M as the count source).

14.5 Timer RE

Timer RE has the 4-bit counter and 8-bit counter. Timer RE has the following 2 modes:

- Real-time clock mode Generate 1-second signal from fC4 and count seconds, minutes, hours, and days of the week.
- Output compare mode Count a count source and detect compare matches.

The count source for timer RE is the operating clock that regulates the timing of timer operations.

14.5.1 Real-Time Clock Mode

In real-time clock mode, a 1-second signal is generated from fC4 using a divide-by-2 frequency divider, 4-bit counter, and 8-bit counter and used to count seconds, minutes, hours, and days of the week. Figure 14.156 shows a Block Diagram of Real-Time Clock Mode and Table 14.52 lists the Real-Time Clock Mode Specifications. Figures 14.157 to 14.161 and 14.163 to 14.164 show the Registers Associated with Real-Time Clock Mode. Table 14.53 lists the Interrupt Sources, Figure 14.162 shows the Definition of Time Representation and Figure 14.165 shows the Operating Example in Real-Time Clock Mode.

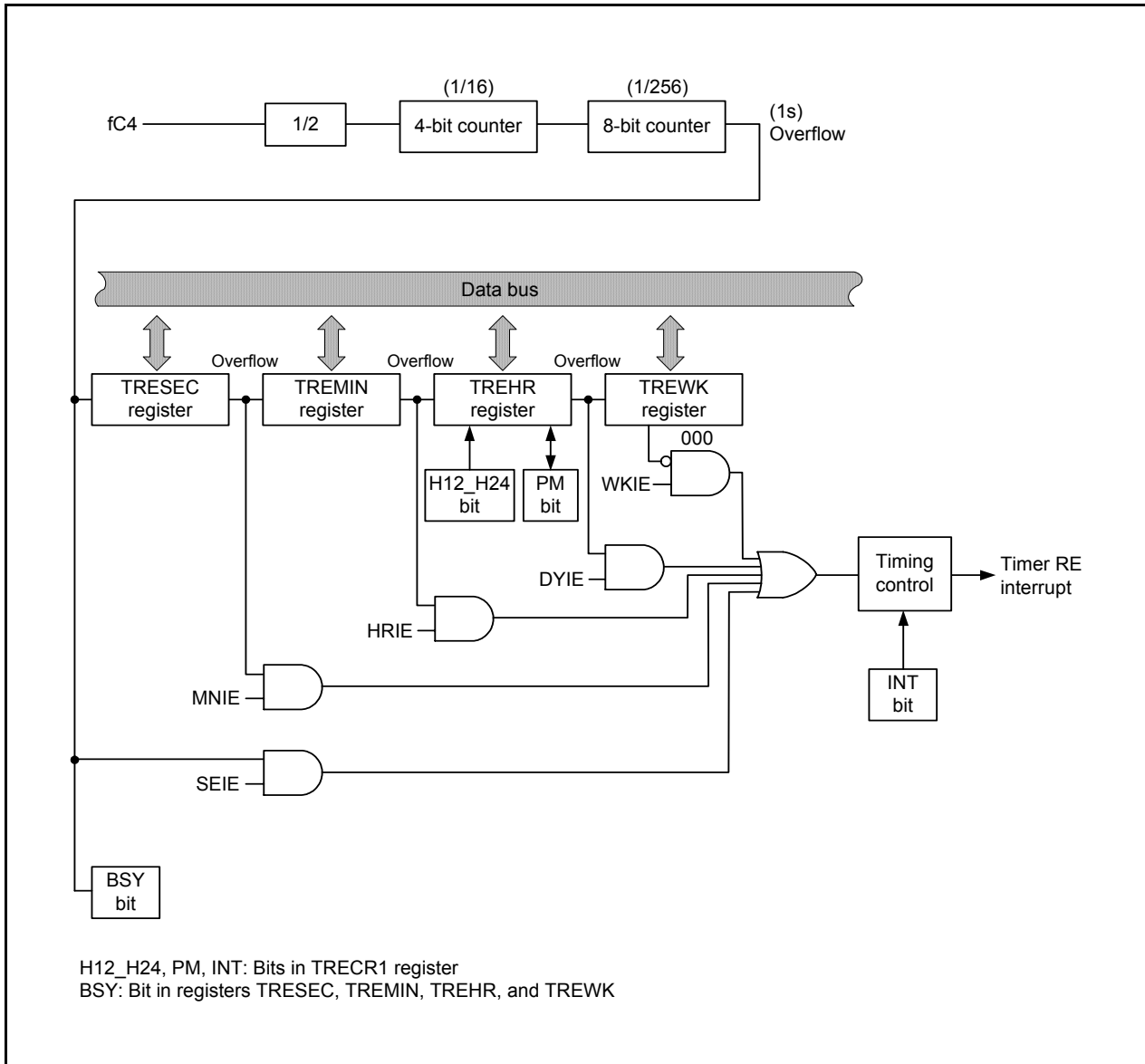


Figure 14.156 Block Diagram of Real-Time Clock Mode

Table 14.52 Real-Time Clock Mode Specifications

| Item | Specification |
|-------------------------------------|---|
| Count source | fC4 |
| Count operation | Increment |
| Count start condition | 1 (count starts) is written to TSTART bit in TRECR1 register |
| Count stop condition | 0 (count stops) is written to TSTART bit in TRECR1 register |
| Interrupt request generation timing | Select any one of the following: <ul style="list-style-type: none"> • Update second data • Update minute data • Update hour data • Update day of week data • When day of week data is set to 000b (Sunday) |
| TREO pin function | Programmable I/O ports or output of f2, f4, or f8 |
| Read from timer | When reading TRESEC, TREMIN, TREHR, or TREWK register, the count value can be read. The values read from registers TRESEC, TREMIN, and TREHR are represented by the BCD code. |
| Write to timer | When bits TSTART and TCSTF in the TRECR1 register are set to 0 (timer stops), the value can be written to registers TRESEC, TREMIN, TREHR, and TREWK. The values written to registers TRESEC, TREMIN, and TREHR are represented by the BCD codes. |
| Select function | • 12-hour mode/24-hour mode switch function |

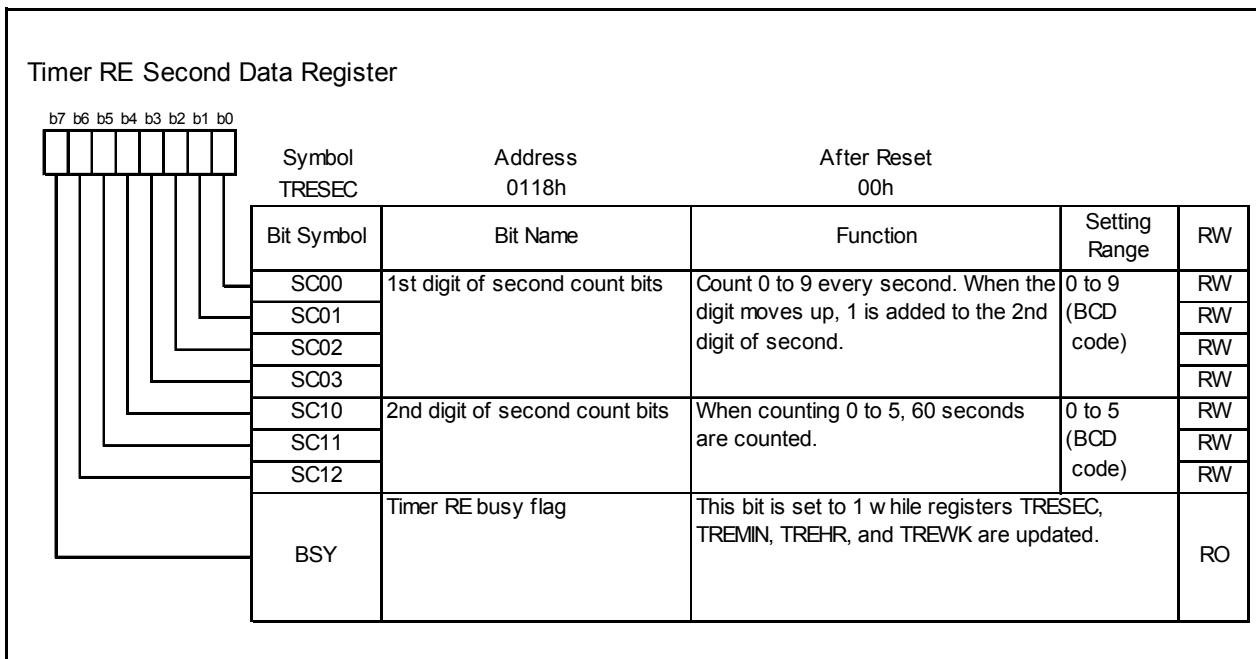


Figure 14.157 TRESEC Register in Real-Time Clock Mode

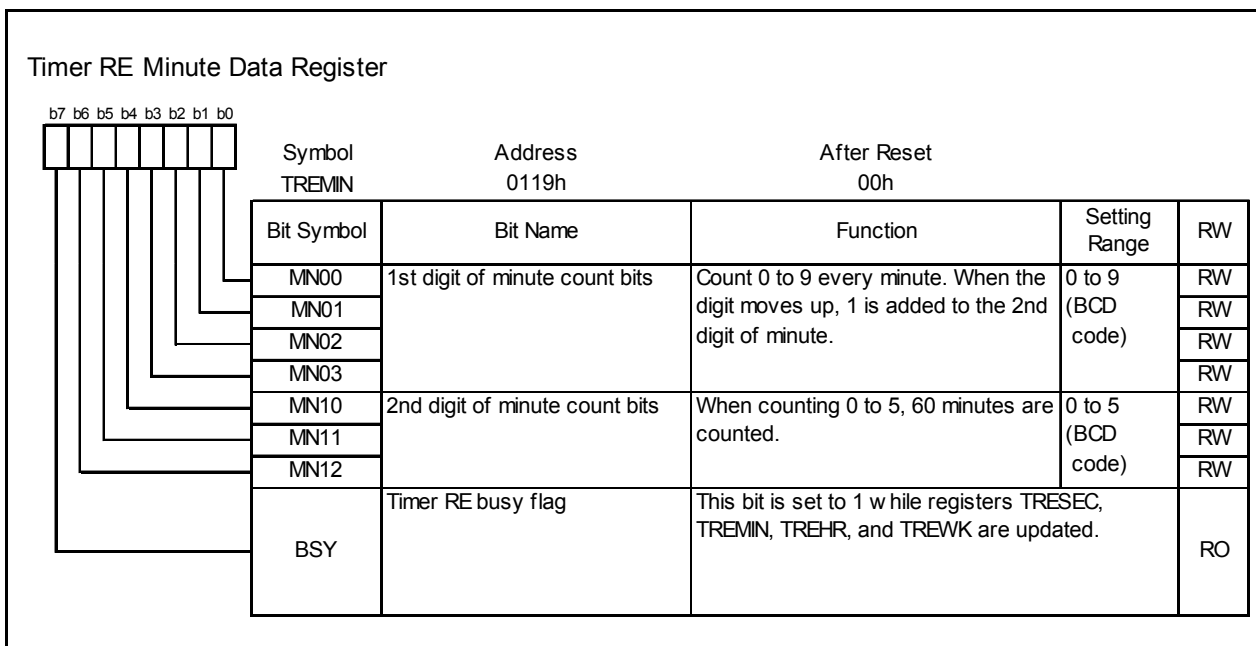


Figure 14.158 TREMIN Register in Real-Time Clock Mode

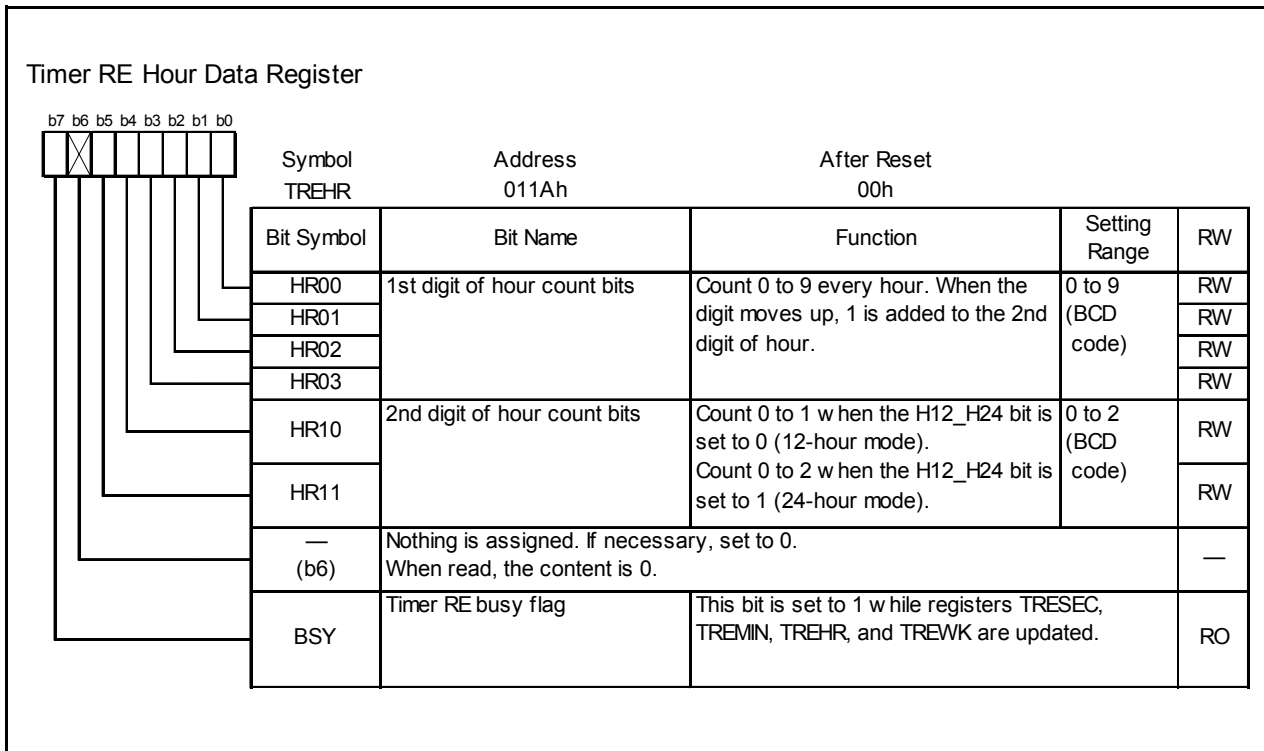


Figure 14.159 TREHR Register in Real-Time Clock Mode

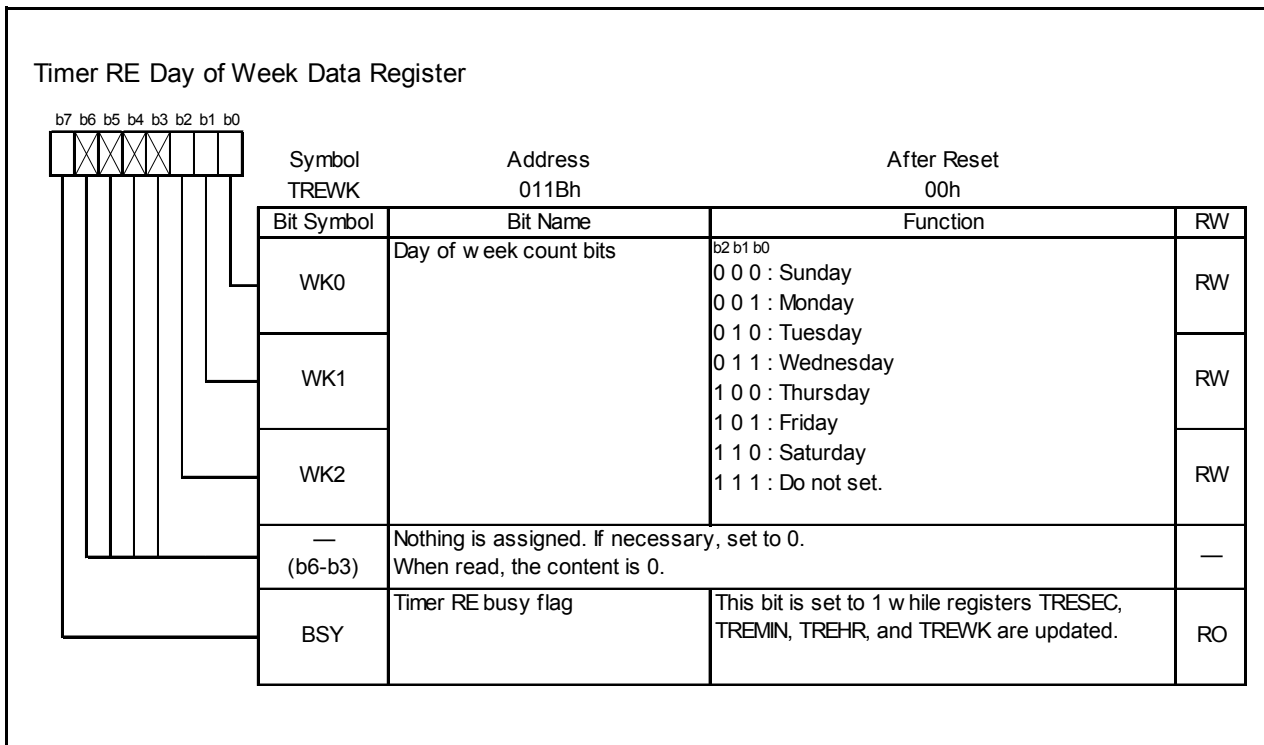


Figure 14.160 TREWK Register in Real-Time Clock Mode

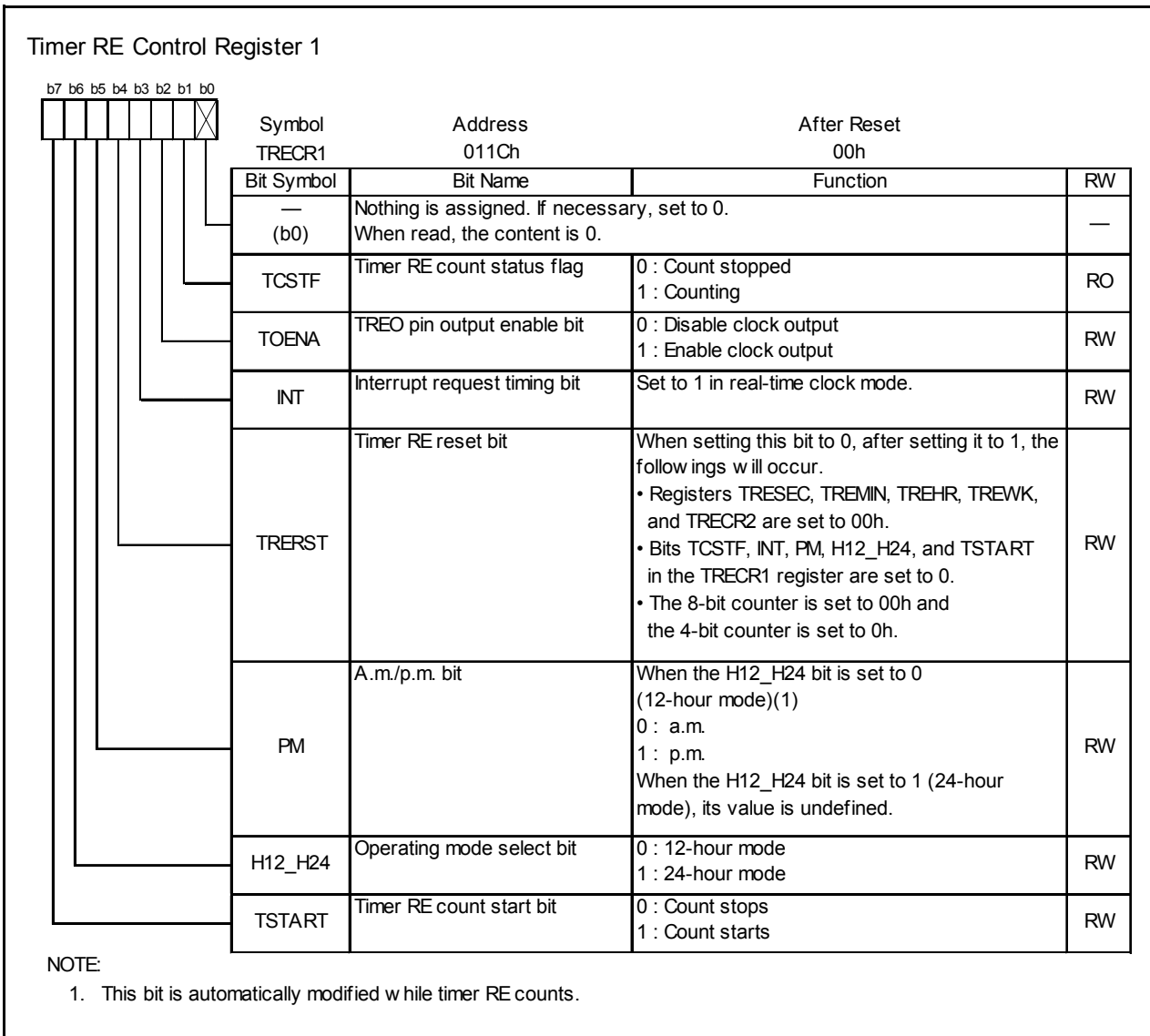


Figure 14.161 TRECR1 Register in Real-Time Clock Mode

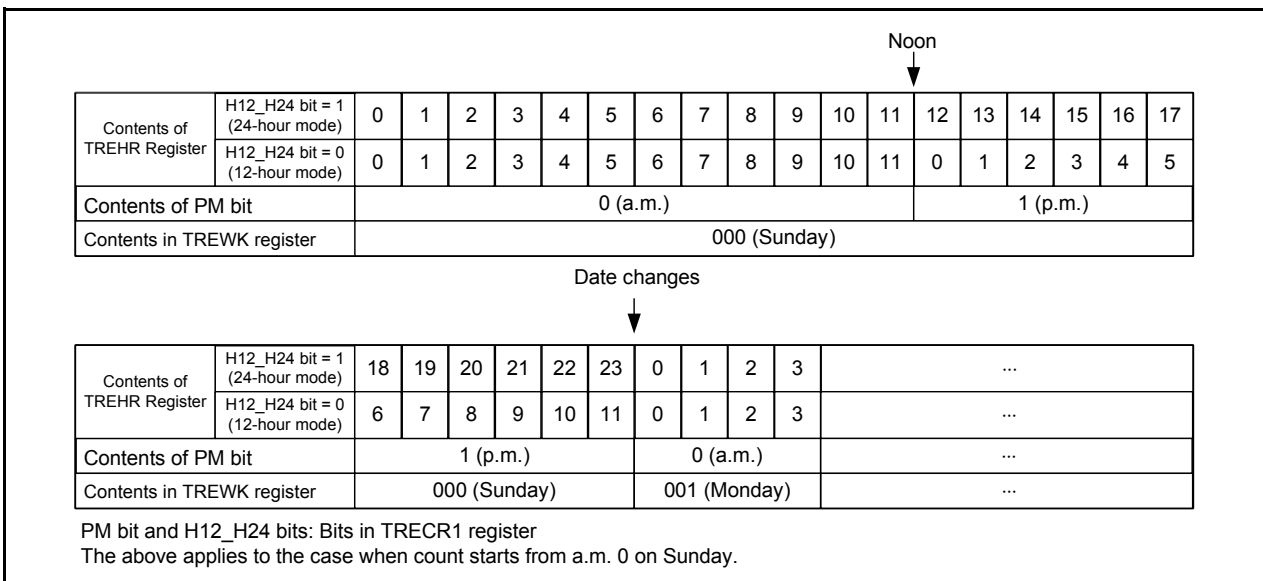


Figure 14.162 Definition of Time Representation

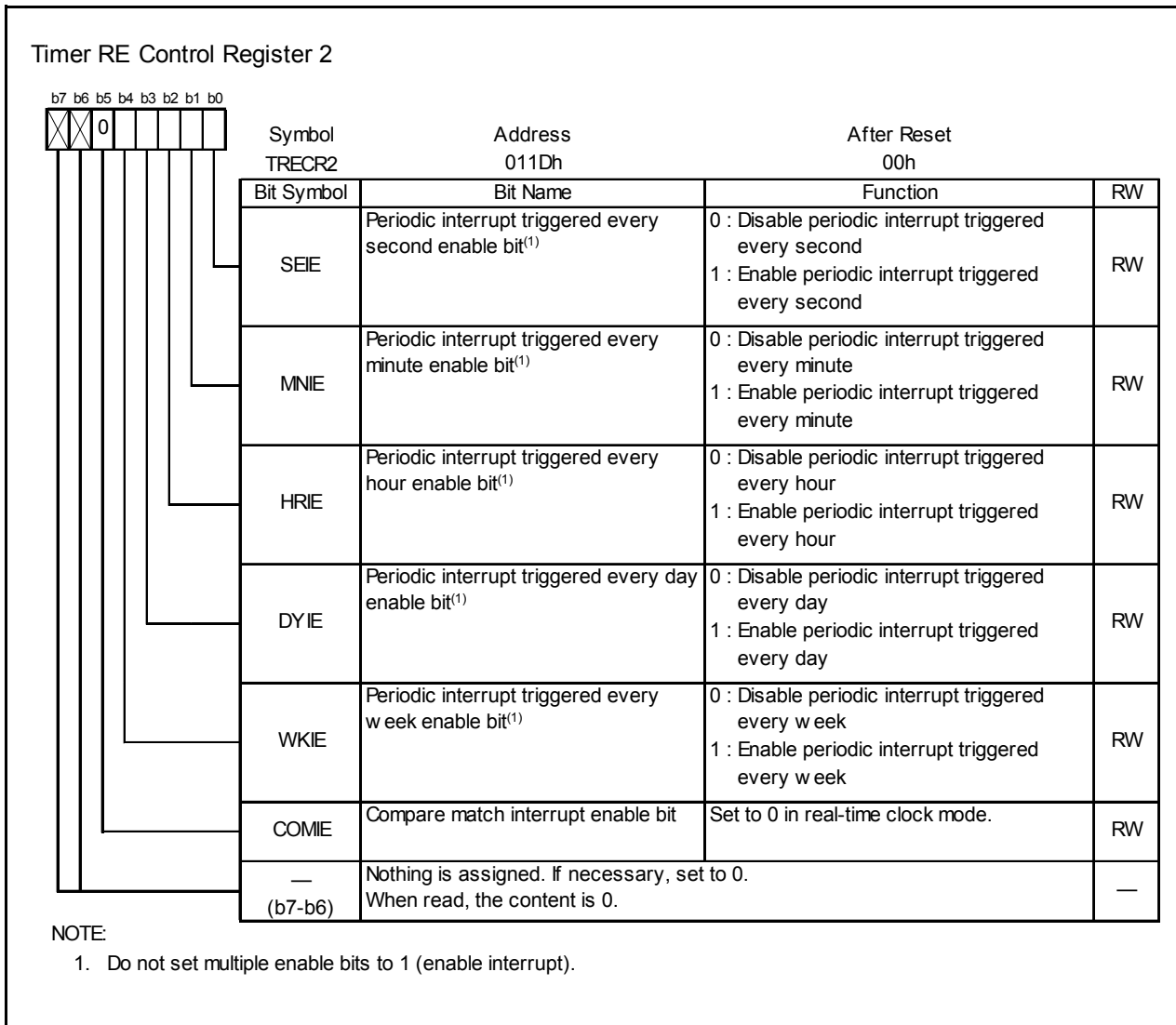


Figure 14.163 TREC2 Register in Real-Time Clock Mode

Table 14.53 Interrupt Sources

| Factor | Interrupt Source | Interrupt Enable Bit |
|---|---|----------------------|
| Periodic interrupt triggered every week | Value in TREWK register is set to 000b (Sunday) (1-week period) | WKIE |
| Periodic interrupt triggered every day | TREWK register is updated (1-day period) | DYIE |
| Periodic interrupt triggered every hour | TREHR register is updated (1-hour period) | HRIE |
| Periodic interrupt triggered every minute | TREMIN register is updated (1-minute period) | MNIE |
| Periodic interrupt triggered every second | TRESEC register is updated (1-second period) | SEIE |

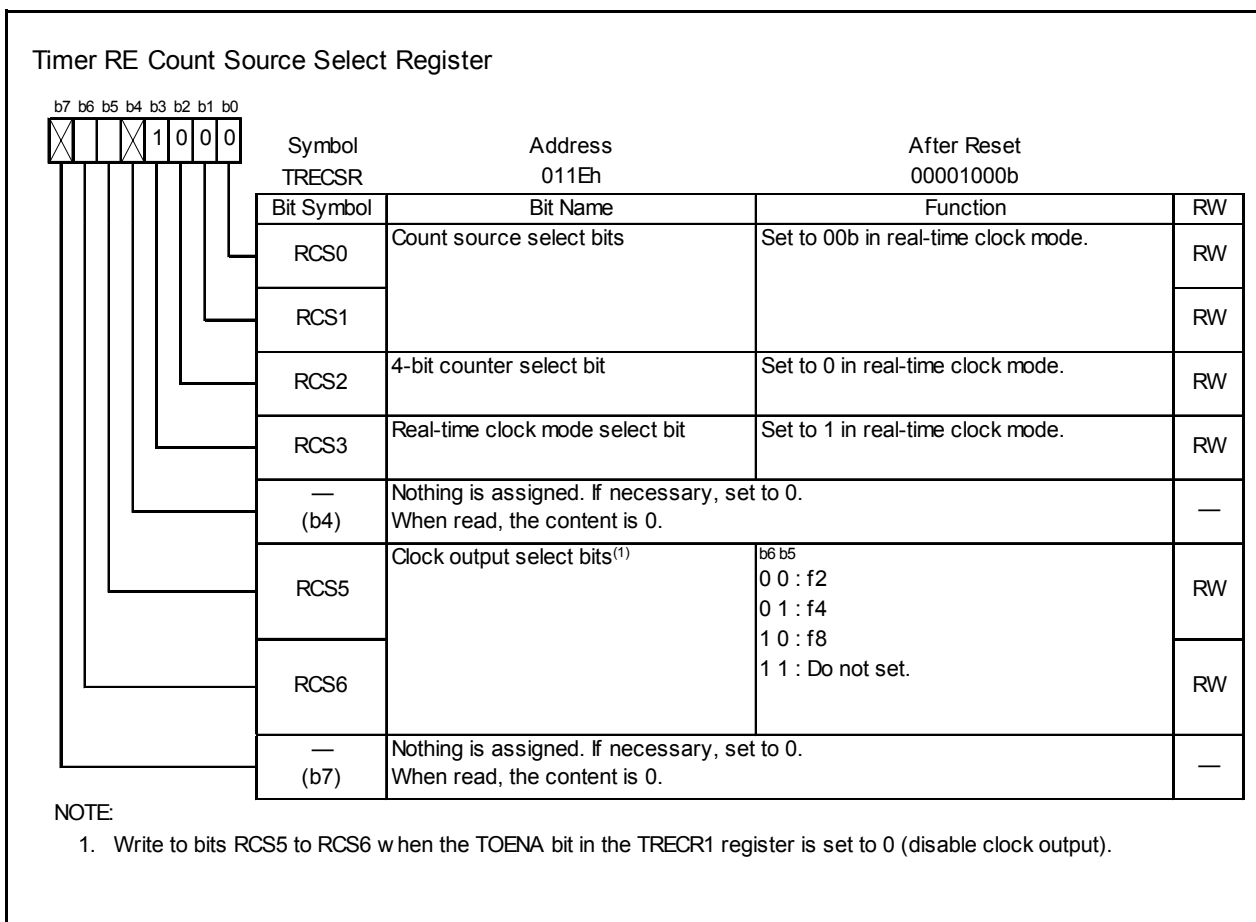


Figure 14.164 TRECSR Register in Real-Time Clock Mode

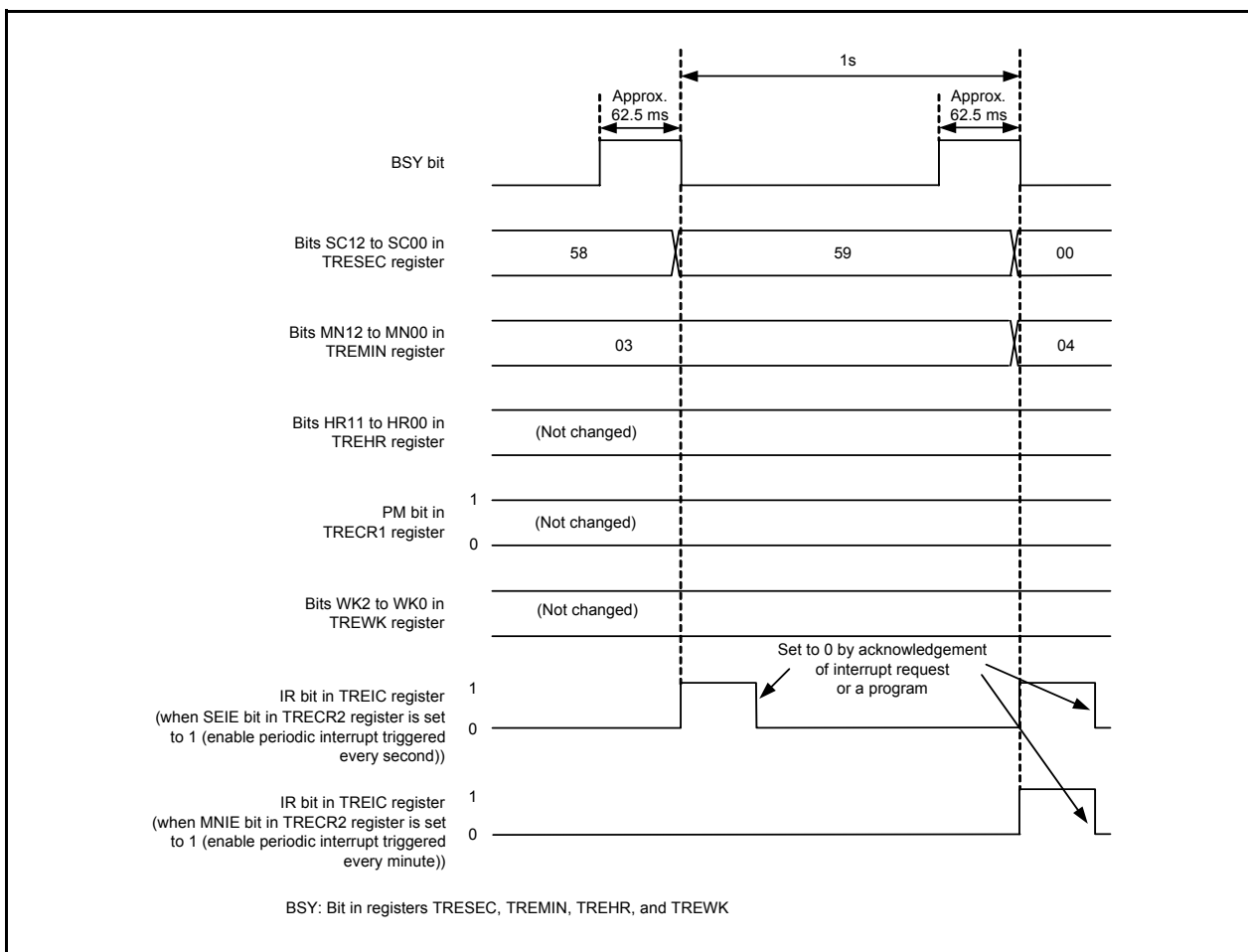


Figure 14.165 Operating Example in Real-Time Clock Mode

14.5.2 Output Compare Mode

In output compare mode, the internal count source divided by 2 is counted using the 4-bit or 8-bit counter and compare value match is detected with the 8-bit counter. Figure 14.166 shows a Block Diagram of Output Compare Mode and Table 14.54 lists the Output Compare Mode Specifications. Figures 14.167 to 14.171 show the registers associated with output compare mode, and Figure 14.172 shows the Operating Example in Output Compare Mode.

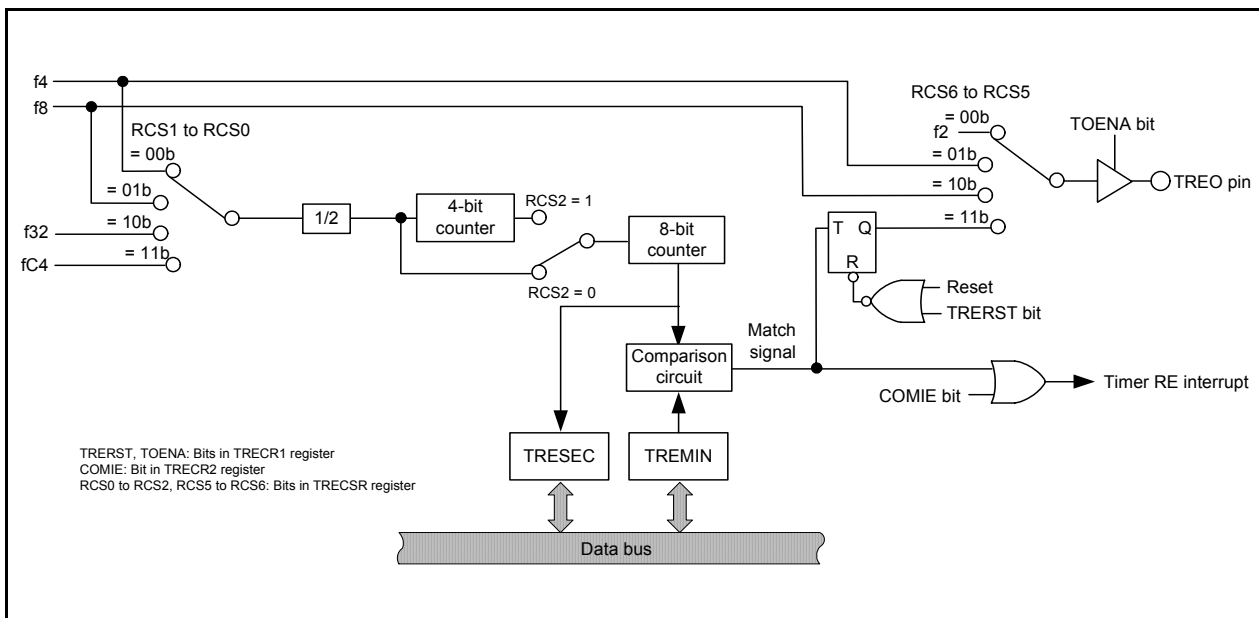


Figure 14.166 Block Diagram of Output Compare Mode

Table 14.54 Output Compare Mode Specifications

| Item | Specification |
|-------------------------------------|--|
| Count sources | f4, f8, f32, fC4 |
| Count operations | <ul style="list-style-type: none"> • Increment • When the 8-bit counter content matches with the TREMIN register content, the value returns to 00h and count continues. The count value is held while count stops. |
| Count period | <ul style="list-style-type: none"> • When RCS2 = 0 (4-bit counter is not used) $1/f_i \times 2 \times (n+1)$ • When RCS2 = 1 (4-bit counter is used) $1/f_i \times 32 \times (n+1)$ f _i : Frequency of count source n: Setting value of TREMIN register |
| Count start condition | 1 (count starts) is written to the TSTART bit in the TRECR1 register |
| Count stop condition | 0 (count stops) is written to the TSTART bit in the TRECR1 register |
| Interrupt request generation timing | When the 8-bit counter content matches with the TREMIN register content |
| TREO pin function | Select any one of the following: <ul style="list-style-type: none"> • Programmable I/O ports • Output f2, f4, or f8 • Compare output |
| Read from timer | When reading the TRESEC register, the 8-bit counter value can be read. When reading the TREMIN register, the compare value can be read. |
| Write to timer | Writing to the TRESEC register is disabled. When bits TSTART and TCSTF in the TRECR1 register are set to 0 (timer stops), writing to the TREMIN register is enabled. |
| Select functions | <ul style="list-style-type: none"> • Select use of 4-bit counter • Compare output function Every time the 8-bit counter value matches the TREMIN register value, TREO output polarity is reversed. The TREO pin outputs "L" after reset is deasserted and the timer RE is reset by the TRERST bit in the TRECR1 register. Output level is held by setting the TSTART bit to 0 (count stops). |

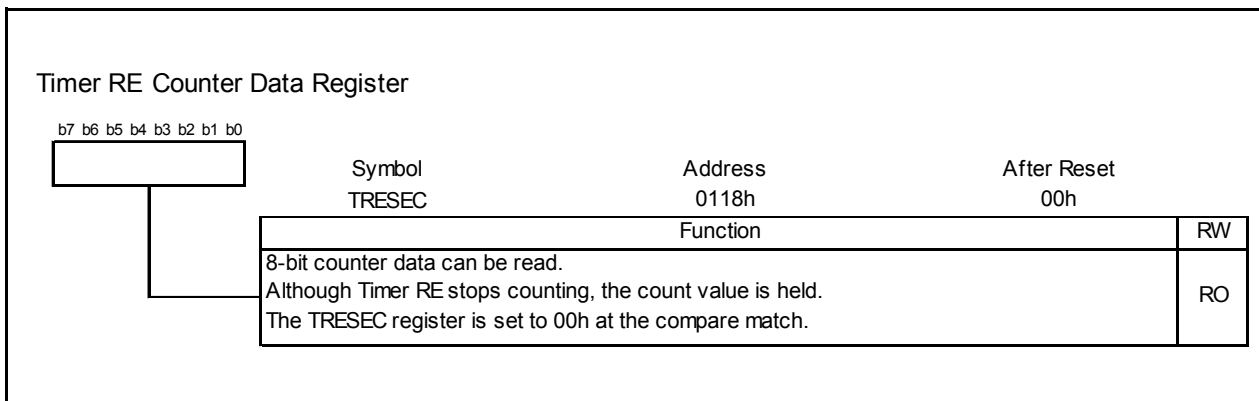


Figure 14.167 TRESEC Register in Output Compare Mode

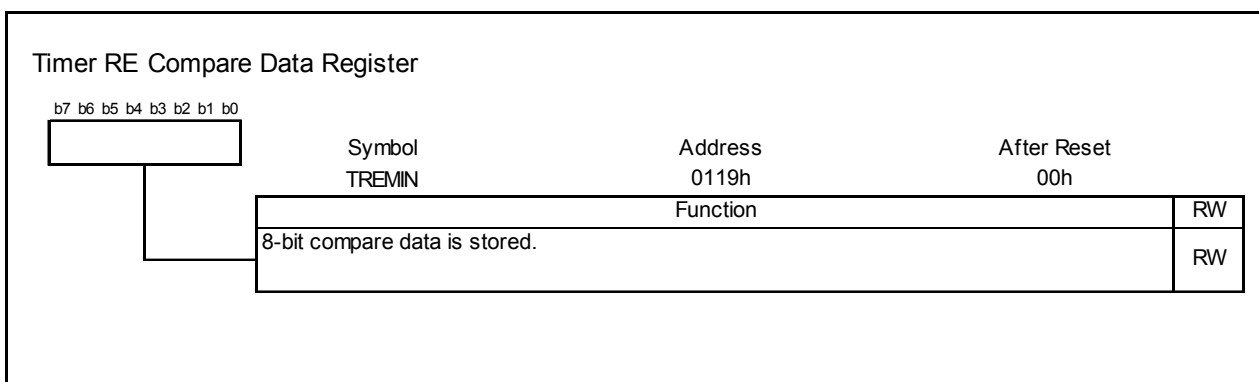


Figure 14.168 TREMIN Register in Output Compare Mode

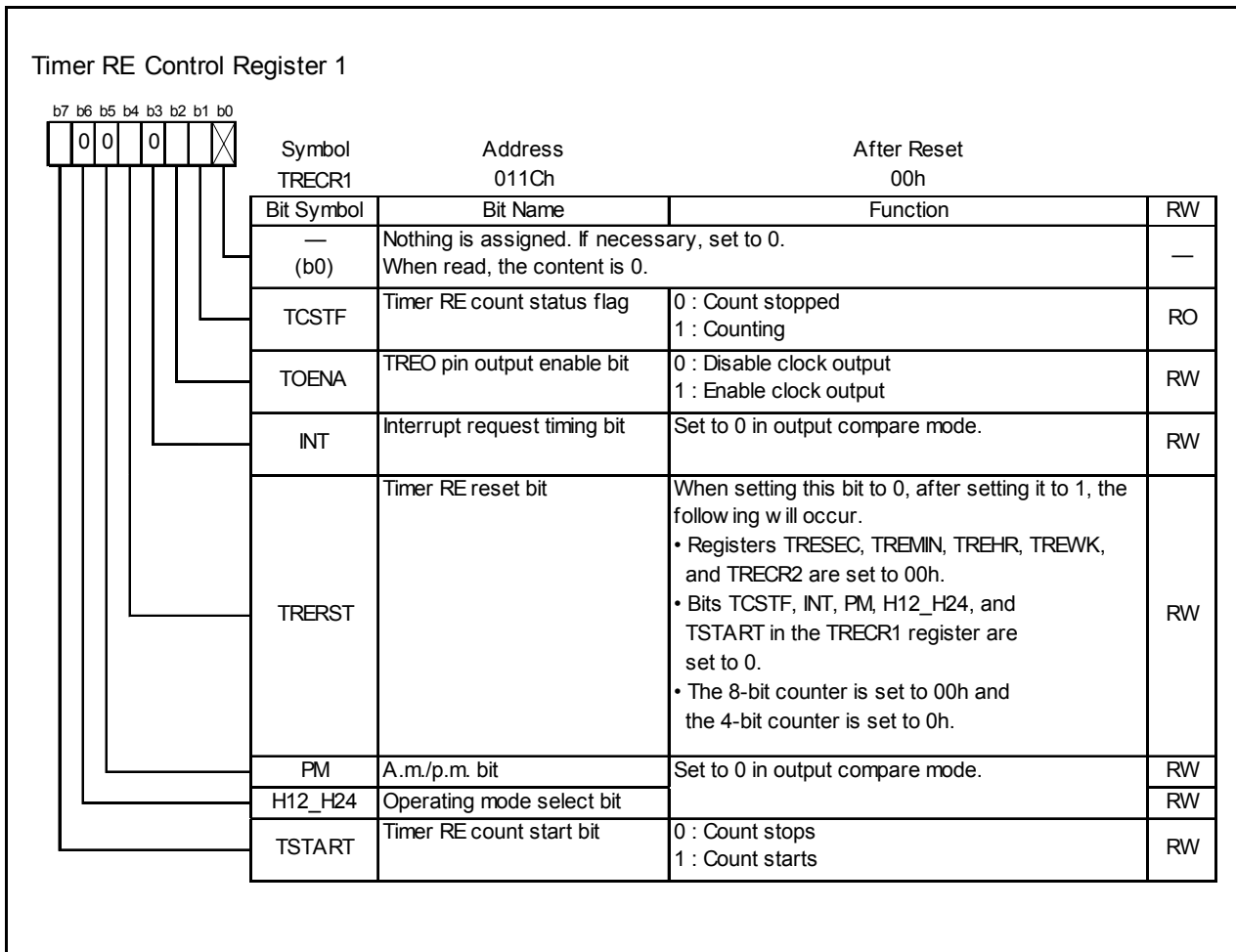


Figure 14.169 TREC1 Register in Output Compare Mode

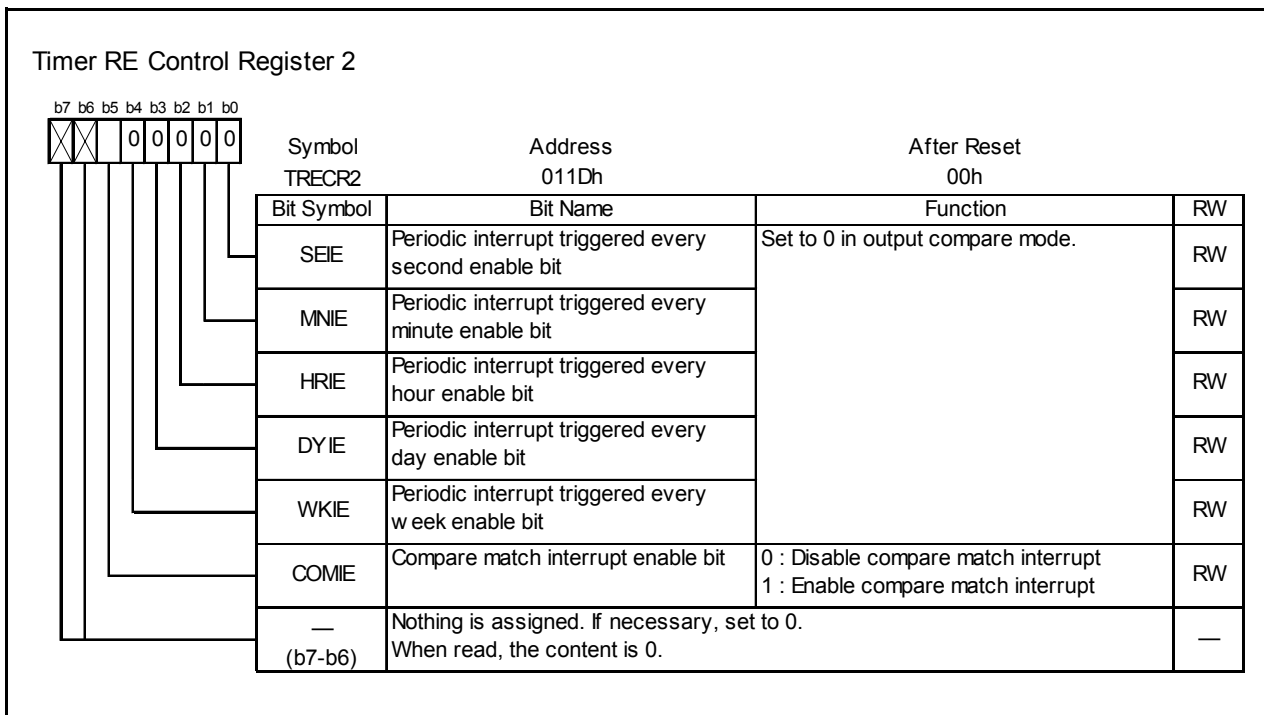


Figure 14.170 TREC2 Register in Output Compare Mode

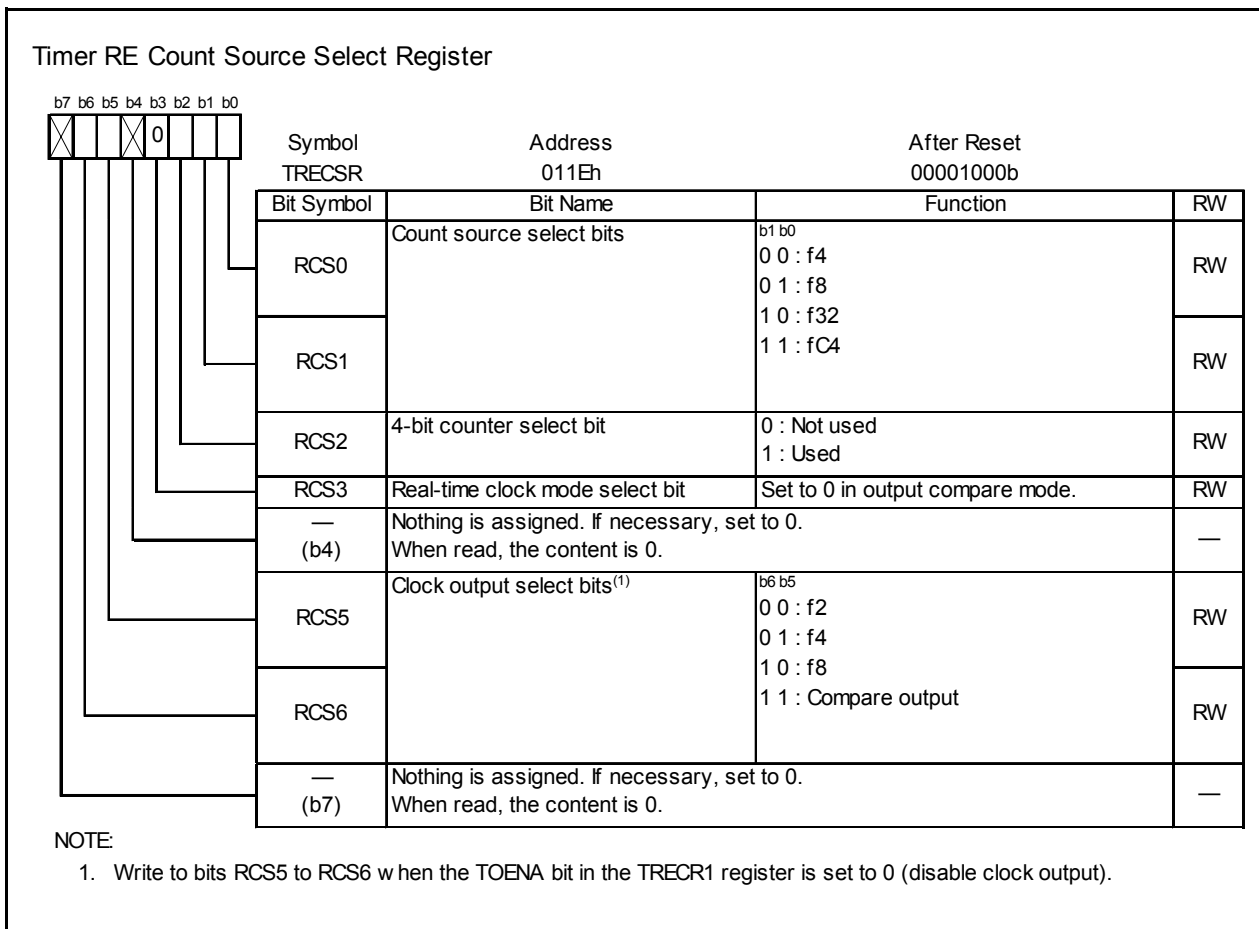


Figure 14.171 TRECSR Register in Output Compare Mode

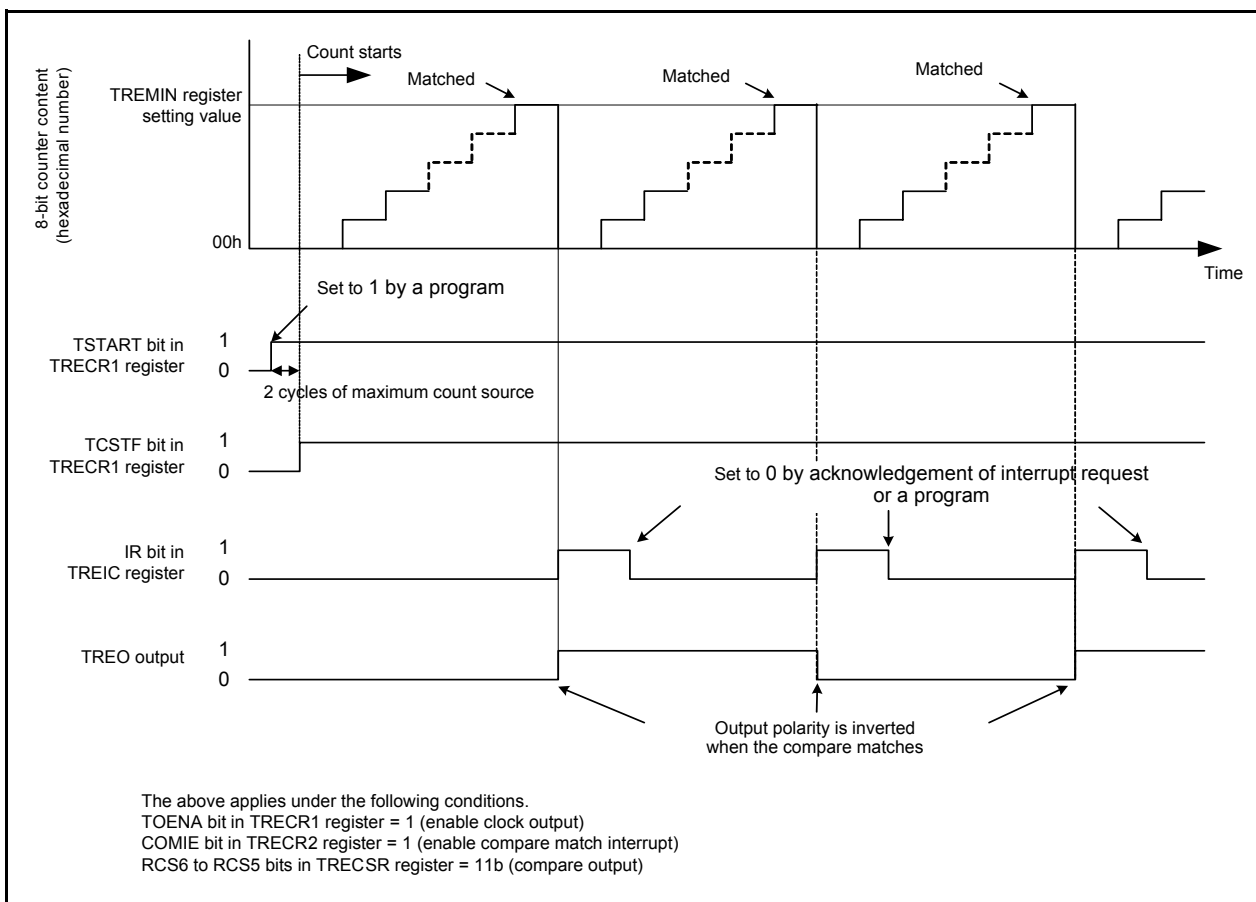


Figure 14.172 Operating Example in Output Compare Mode

14.5.3 Notes on Timer RE

14.5.3.1 Starting and Stopping Count

Timer RE has the TSTART bit for instructing the count to start or stop, and the TCSTF bit, which indicates count start or stop. Bits TSTART and TCSTF are in the TRECRC1 register.

Timer RE starts counting and the TCSTF bit is set to 1 (count starts) when the TSTART bit is set to 1 (count starts). It takes up to 2 cycles of the count source until the TCSTF bit is set to 1 after setting the TSTART bit to 1. During this time, do not access registers associated with timer RE⁽¹⁾ other than the TCSTF bit.

Also, timer RE stops counting when setting the TSTART bit to 0 (count stops) and the TCSTF bit is set to 0 (count stops). It takes the time for up to 2 cycles of the count source until the TCSTF bit is set to 0 after setting the TSTART bit to 0. During this time, do not access registers associated with timer RE other than the TCSTF bit.

NOTE:

1. Registers associated with timer RE: TRESEC, TREMIN, TREHR, TREWK, TRECRC1, TRECRC2, and TRECSR.

14.5.3.2 Register Setting

Write to the following registers or bits when timer RE is stopped.

- Registers TRESEC, TREMIN, TREHR, TREWK, and TRECRC2
- Bits H12_H24, PM, and INT in TRECRC1 register
- Bits RCS0 to RCS3 in TRECSR register

Timer RE is stopped when bits TSTART and TCSTF in the TRECRC1 register are set to 0 (timer RE stopped).

Also, set all above-mentioned registers and bits (immediately before timer RE count starts) before setting the TRECRC2 register.

Figure 14.173 shows a Setting Example in Real-Time Clock Mode.

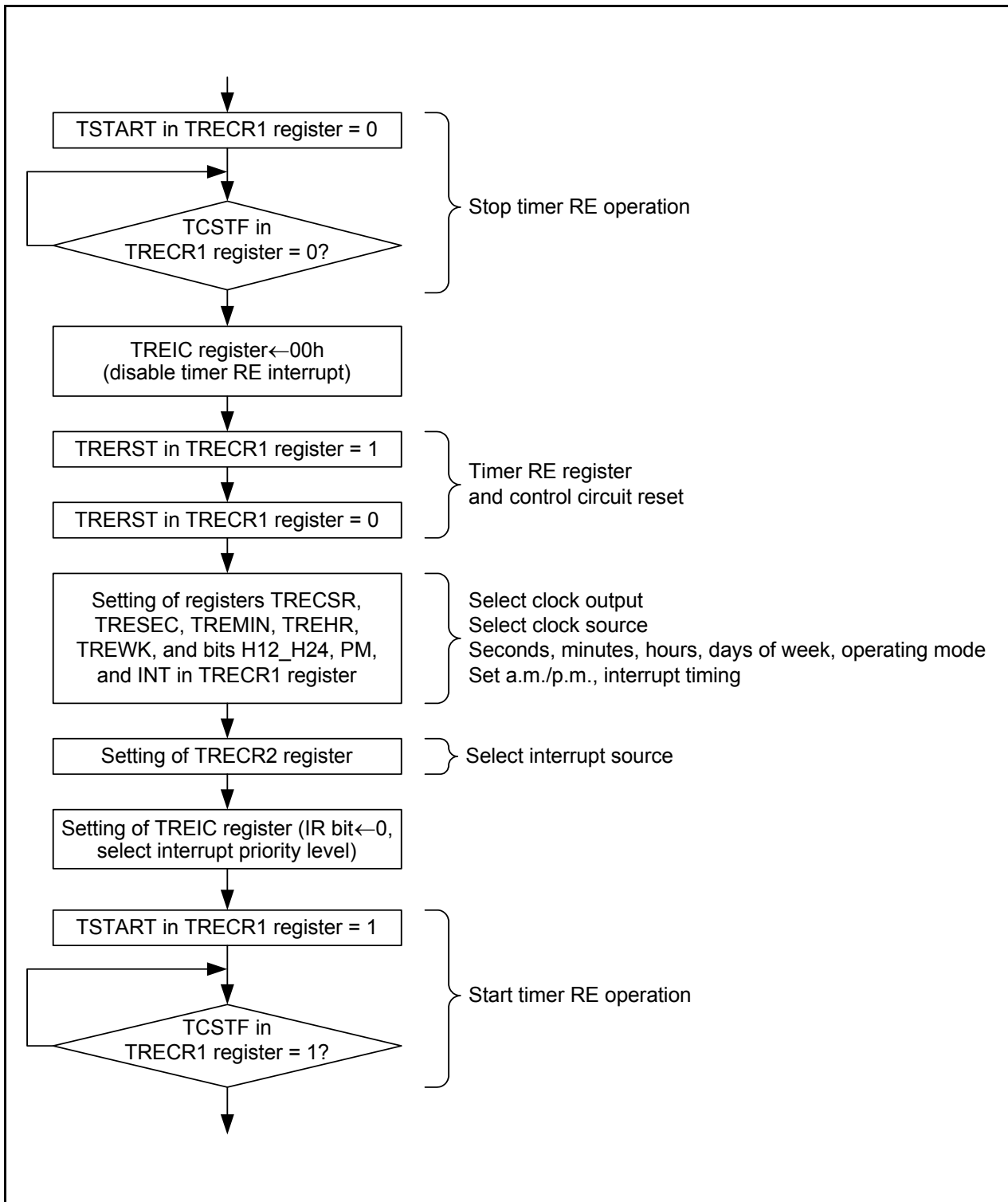


Figure 14.173 Setting Example in Real-Time Clock Mode

14.5.3.3 Time Reading Procedure of Real-Time Clock Mode

In real-time clock mode, read registers TRESEC, TREMIN, TREHR, and TREWK when time data is updated and read the PM bit in the TRECR1 register when the BSY bit is set to 0 (not while data is updated).

Also, when reading several registers, an incorrect time will be read if data is updated before another register is read after reading any register.

In order to prevent this, use the reading procedure shown below.

- Using an interrupt
Read necessary contents of registers TRESEC, TREMIN, TREHR, and TREWK and the PM bit in the TRECR1 register in the timer RE interrupt routine.
- Monitoring with a program 1
Monitor the IR bit in the TREIC register with a program and read necessary contents of registers TRESEC, TREMIN, TREHR, and TREWK and the PM bit in the TRECR1 register after the IR bit in the TREIC register is set to 1 (timer RE interrupt request generated).
- Monitoring with a program 2
 - (1) Monitor the BSY bit.
 - (2) Monitor until the BSY bit is set to 0 after the BSY bit is set to 1 (approximately 62.5 ms while the BSY bit is set to 1).
 - (3) Read necessary contents of registers TRESEC, TREMIN, TREHR, and TREWK and the PM bit in the TRECR1 register after the BSY bit is set to 0.
- Using read results if they are the same value twice
 - (1) Read necessary contents of registers TRESEC, TREMIN, TREHR, and TREWK and the PM bit in the TRECR1 register.
 - (2) Read the same register as (1) and compare the contents.
 - (3) Recognize as the correct value if the contents match. If the contents do not match, repeat until the read contents match with the previous contents.

Also, when reading several registers, read them as continuously as possible.

14.6 Timer RF

Timer RF is a 16-bit timer. The count source for timer RF is the operating clock that regulates the timing of timer operations. Figure 14.174 shows a Block Diagram of Timer RF. Figure 14.175 shows a Block Diagram of CMP Waveform Generation Unit. Figure 14.176 shows a Block Diagram of CMP Waveform Output Unit.

Timer RF has two modes: input capture mode and output compare mode. Figures 14.177 to 14.180 show the Timer C-associated registers.

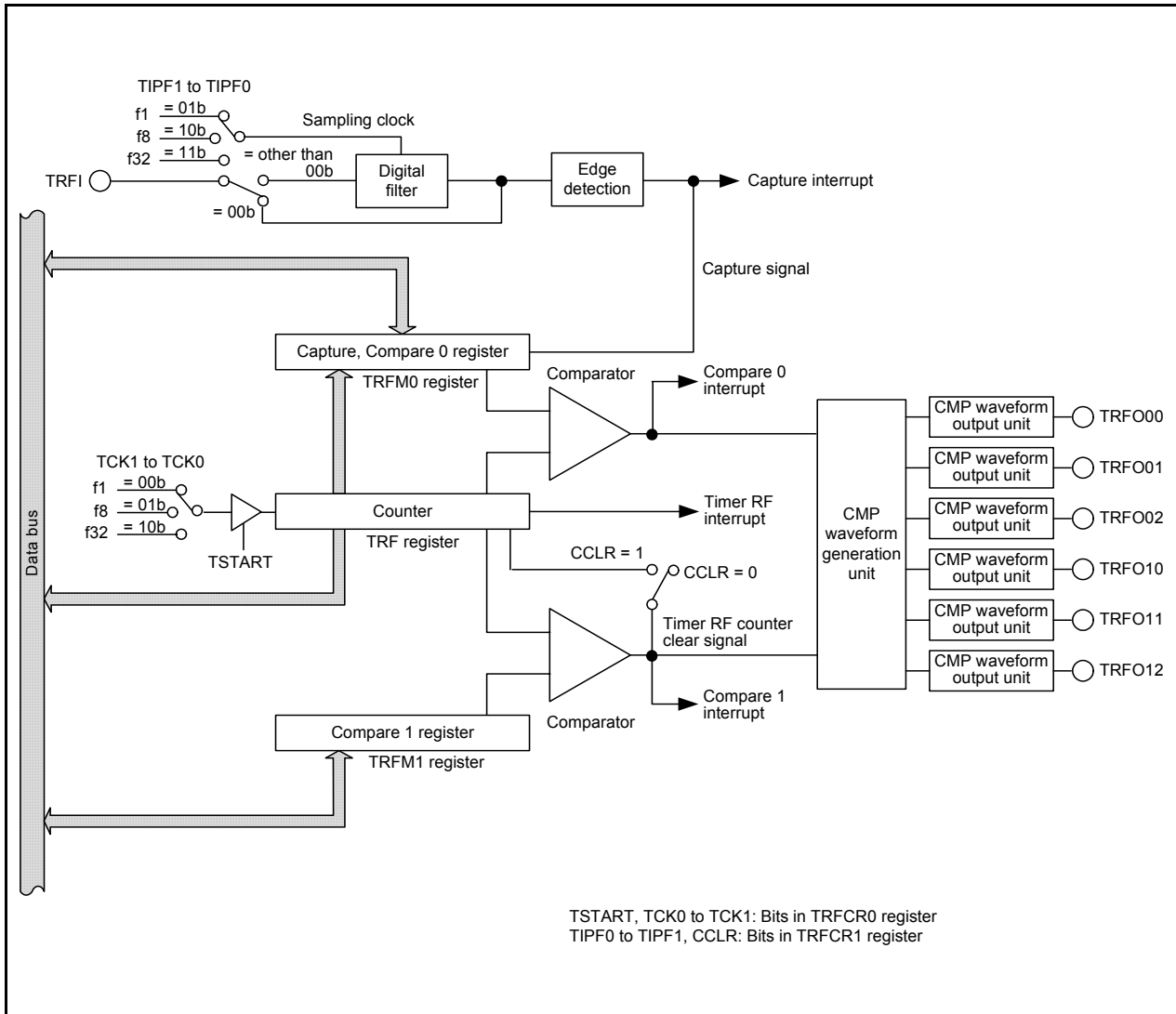


Figure 14.174 Block Diagram of Timer RF

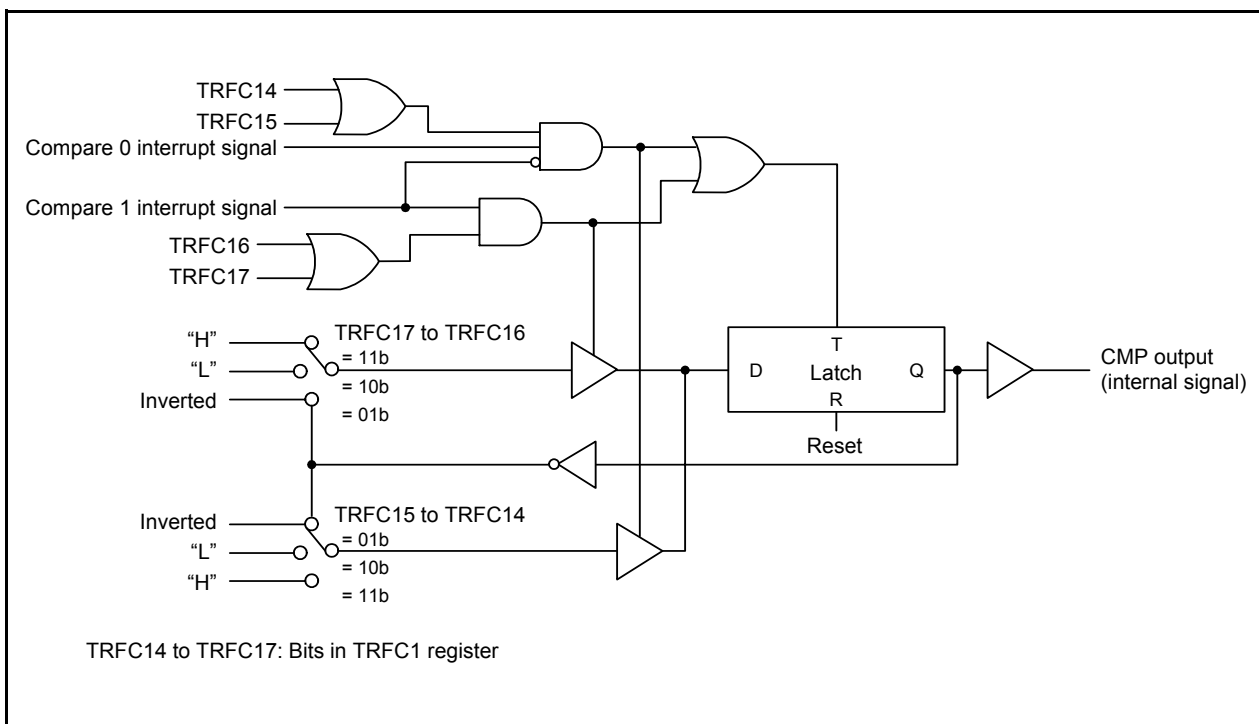


Figure 14.175 Block Diagram of CMP Waveform Generation Unit

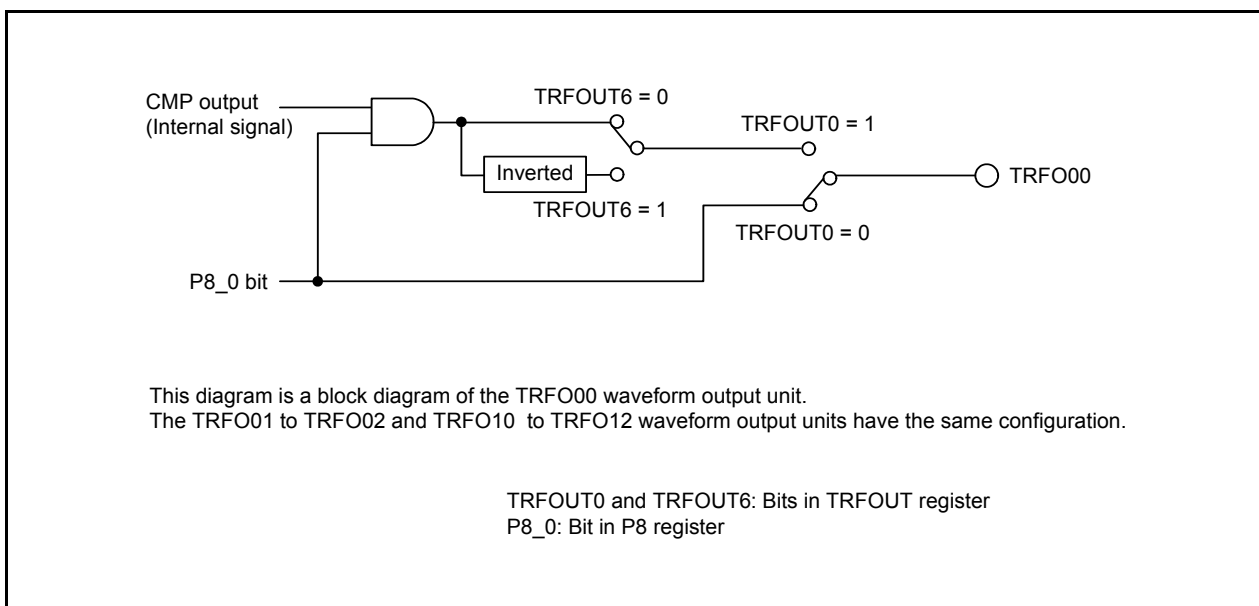


Figure 14.176 Block Diagram of CMP Waveform Output Unit

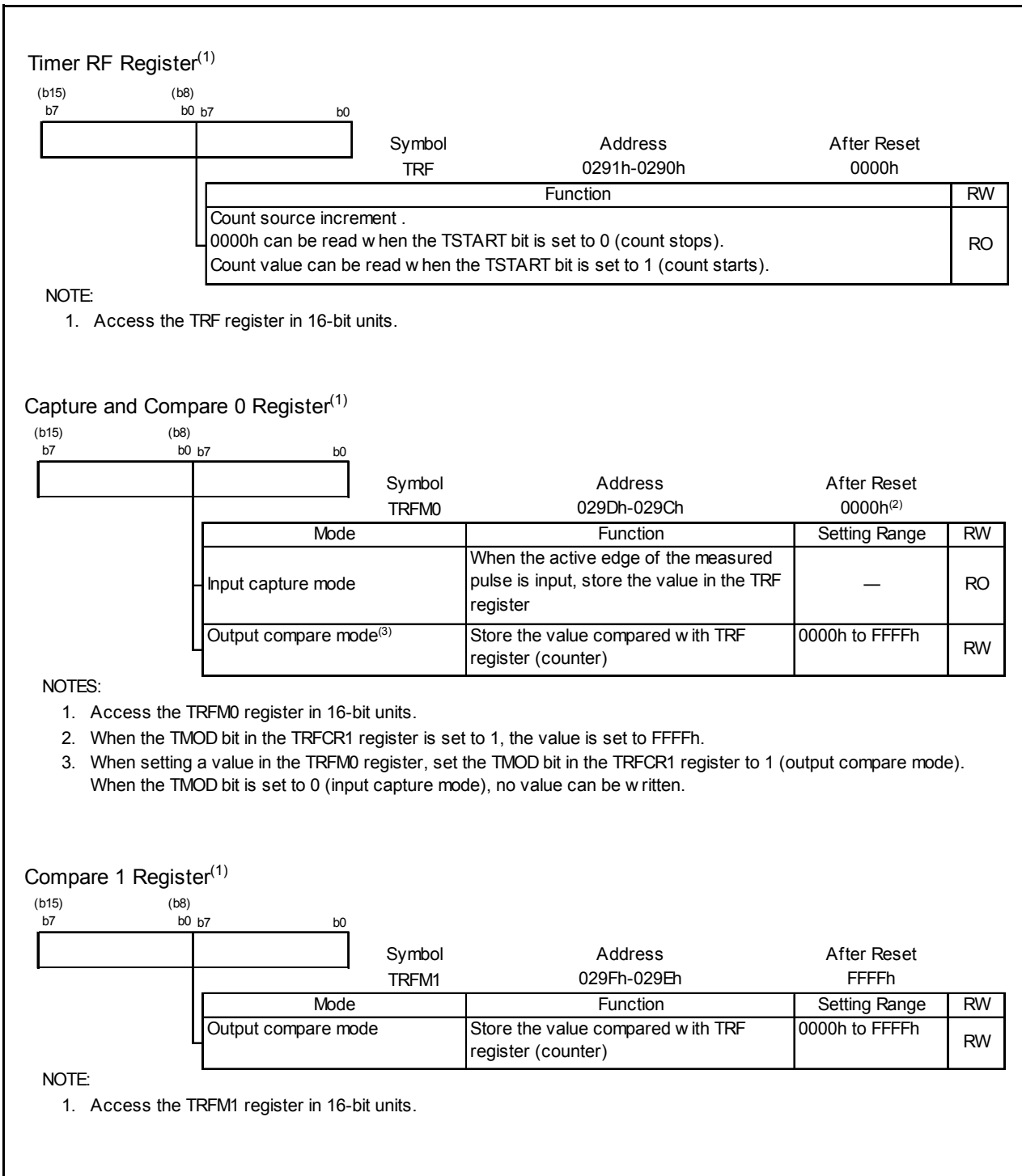


Figure 14.177 Registers TRF, TRFM0, and TRFM1

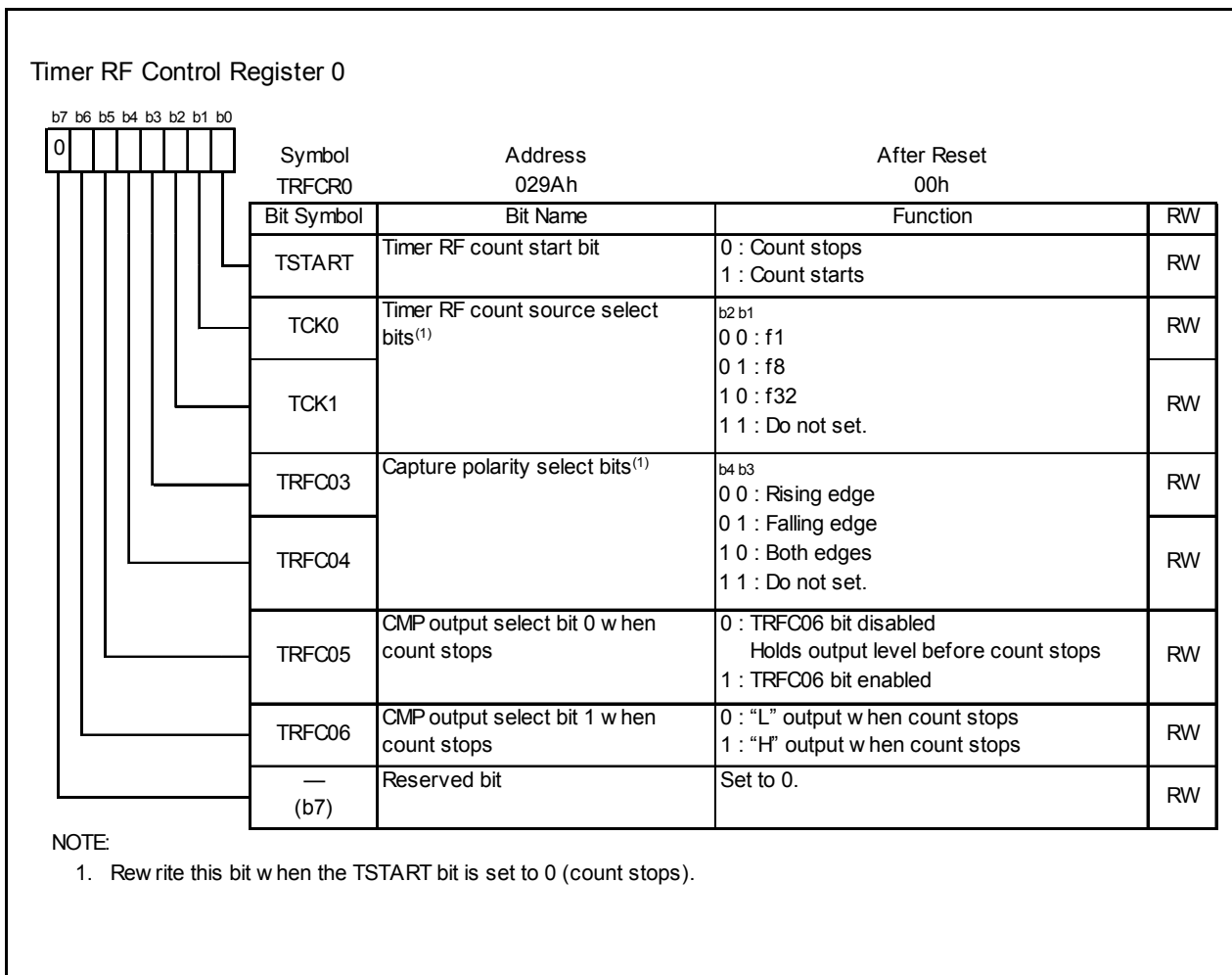


Figure 14.178 TRFCR0 Register

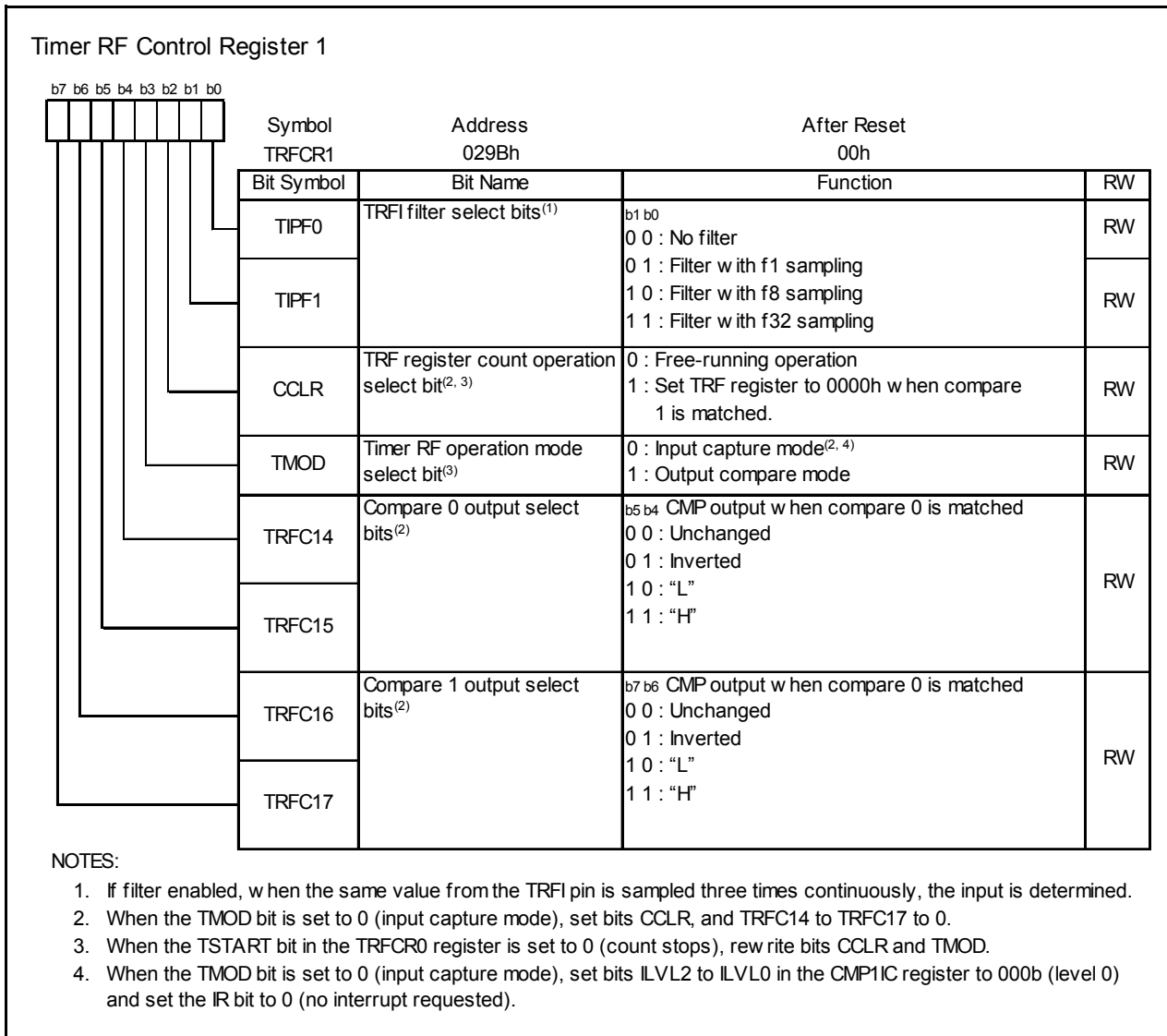


Figure 14.179 TRFCR1 Register

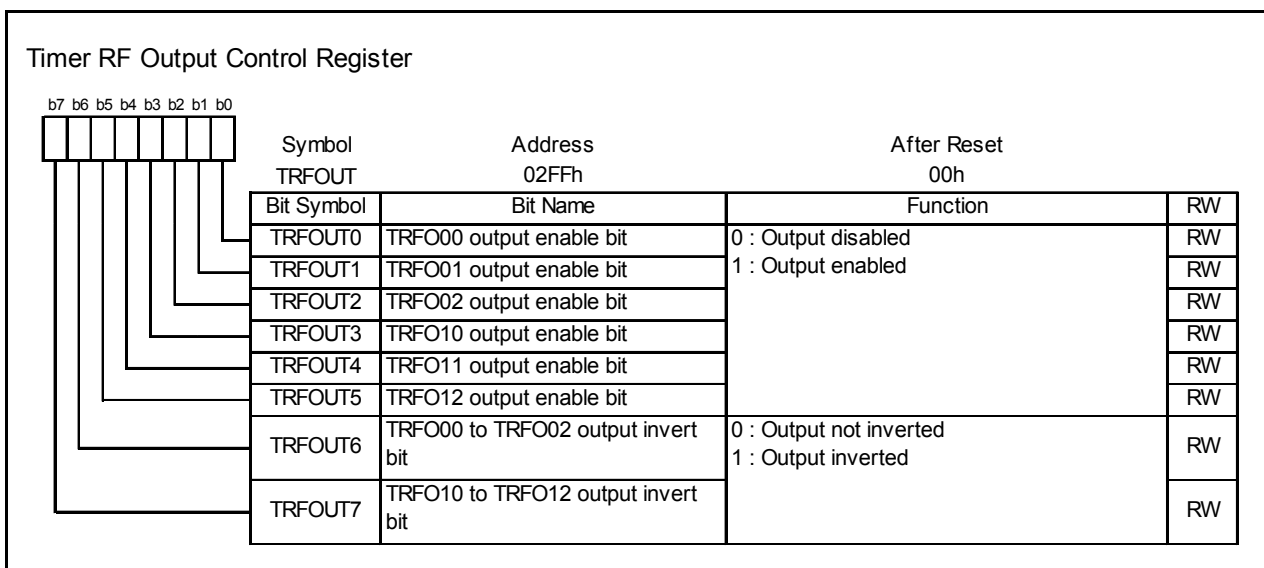


Figure 14.180 TRFOUT Register

14.6.1 Input Capture Mode

In input capture mode, the edge of the TRFI pin input signal is used as a trigger to latch the timer value and the width or the period of external signal is measured. The TRFI input is equipped with a digital filter, and this prevents errors caused by noise or the like from occurring. Table 14.55 shows the Input Capture Mode Specifications. Figure 14.181 shows an Operating Example in Input Capture Mode.

Table 14.55 Input Capture Mode Specifications

| Item | Specification |
|--|--|
| Count sources | f1, f8, f32 |
| Count operations | <ul style="list-style-type: none"> • Increment • Transfer the value in the TRF register to the TRFM0 register at the valid edge of the measured pulse. |
| Count period | $1/fk \times 65536$ fk: Frequency of count source |
| Count start condition | The TSTART bit in the TRFCR0 register is set to 1 (count starts). |
| Count stop condition | The TSTART bit in the TRFCR0 register is set to 0 (count stops). |
| Interrupt request generation timing | <ul style="list-style-type: none"> • The valid edge of TRFI input [capture interrupt] • When timer RF overflows [timer RF interrupt] |
| TRFI pin function | Measured pulse input |
| TRFO00 to TRFO02, TRFO11 to TRFO12 pin functions | Programmable I/O port |
| Counter value reset timing | In the following cases, the value in the TRF register is set to 0000h. <ul style="list-style-type: none"> • When the TSTART bit in the TRFCR0 register is set to 0 (count stops). |
| Read from timer | <ul style="list-style-type: none"> • The count value can be read out by reading the TRF register. • The count value at the measured pulse valid edge input can be read out by reading the TRFM0 register. |
| Write to timer | Write to the TRF and TRFM0 registers is disabled. |
| Select functions | <ul style="list-style-type: none"> • TRFI polarity selected Selects the valid edge of the measured pulse. (Bits TRFC03 to TRFC04 in the TRFCR0 register.) • Digital filter function The TRFI input is sampled, and when the sampled input level matches as three times, the level is determined. Selects the sampling clock of the digital filter. (Bits TIPF0 to TIPF1 in the TRFCR1 register.) |

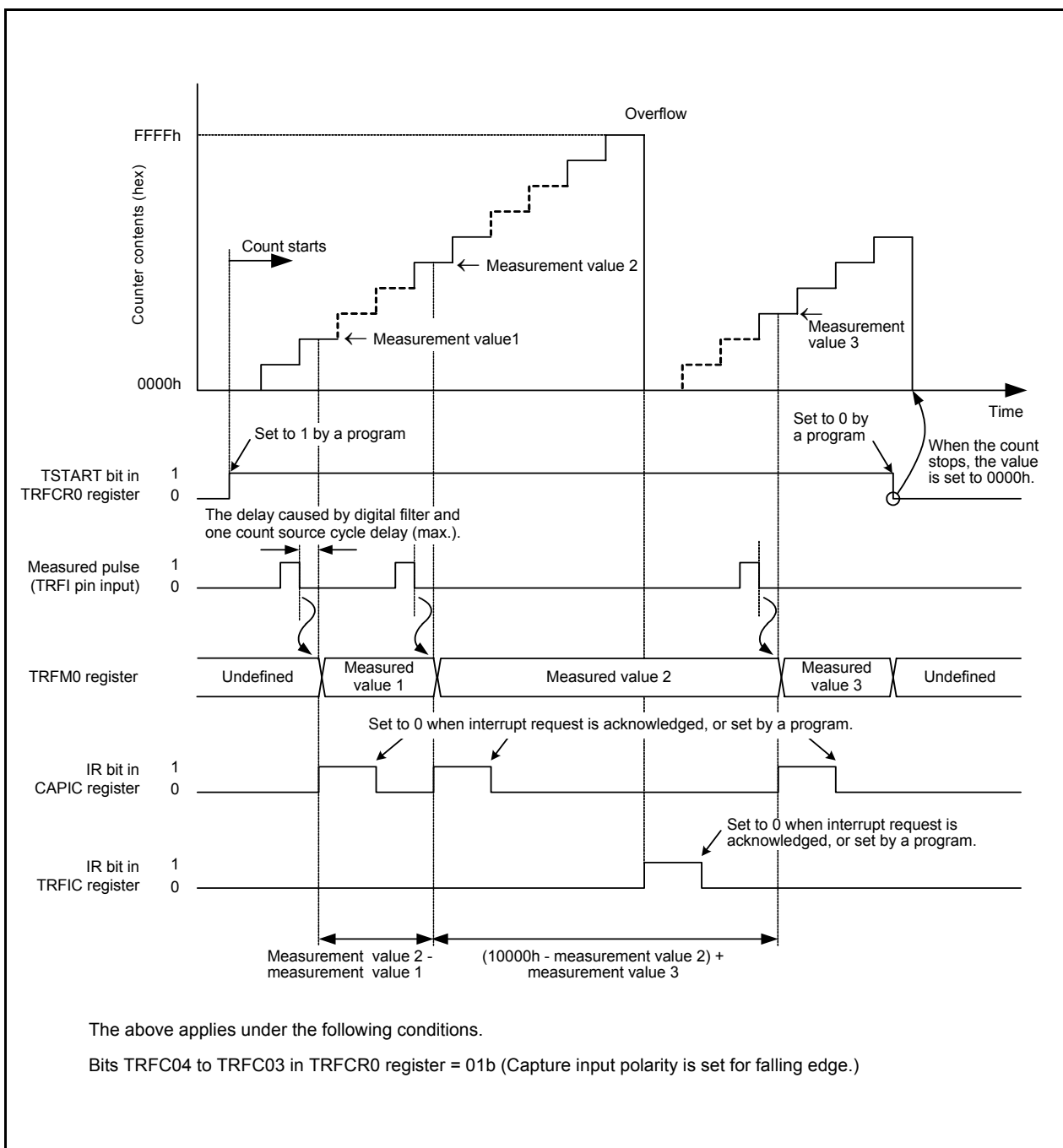


Figure 14.181 Operating Example in Input Capture Mode

14.6.1.1 Digital Filter

The TRFI input is sampled, and when the sampled input level matches three times, its level is determined. Select the digital filter function and sampling clock by the TRFCR1 register.

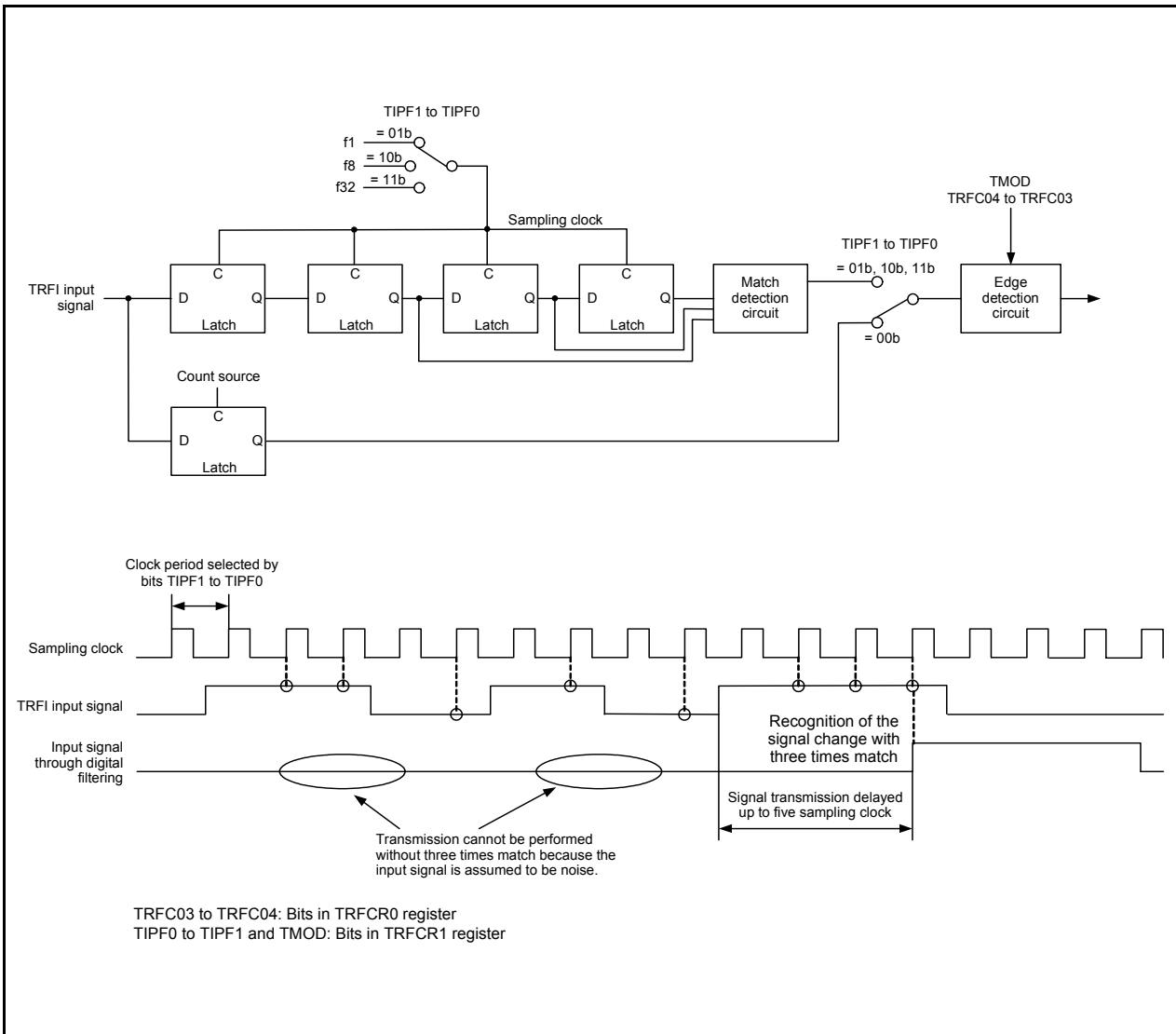


Figure 14.182 Block Diagram of Digital Filter

14.6.2 Output Compare Mode

In output compare mode, when the value of the TRF register matches the value of the TRFM0 (compare 0 match) or TRFM1 (compare 1 match) register, a user-set level is output mode from the output-compare output pin. Table 14.56 shows the Output Compare Mode Specifications. Table 14.57 shows the Output in Output Compare Mode (Example of TRFO00 Pin). Figure 14.183 shows an Operating Example in Output Compare Mode. Figure 14.184 shows the Operating Example in Output Compare Mode (“L” and “H” Held Output in Count Stops).

Table 14.56 Output Compare Mode Specifications

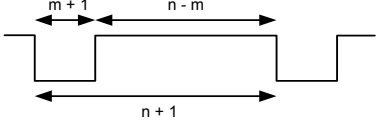
| Item | Specification |
|-------------------------------------|---|
| Count sources | f1, f8, f32 |
| Count operations | Increment |
| PWM waveform | <p>PWM period: $1/f_k \times (n + 1)$ “L” level width: $1/f_k \times (m + 1)$ “H” level width: $1/f_k \times (n - m)$ f_k: Frequency of count source m: Value set in the TRFM0 register n: Value set in the TRFM1 register</p>  <p>It applies under the following conditions.</p> <ul style="list-style-type: none"> • CMP output “H” when compare 0 is matched • CMP output “L” when compare 1 is matched • CMP output not inverted |
| Count start condition | The TSTART bit in the TRFCR0 register is set to 1 (count starts). |
| Count stop condition | The TSTART bit in the TRFCR0 register is set to 0 (count stops). |
| Interrupt request generation timing | <ul style="list-style-type: none"> • When compare 0 match is generated [compare 0 interrupt] • When compare 1 match is generated [compare 1 interrupt] • When time RF overflows [timer RF interrupt]. |
| TRFO00 to TRFO12 pins function | Programmable I/O port or output-compare output |
| Counter value reset timing | <p>In the following cases, the value in the TRF register is set to 0000h.</p> <ul style="list-style-type: none"> • When the TSTART bit in the TRFCR0 register is set to 0 (count stops). • The CCLR bit in the TRFCR1 register is set to 1 (the TRF register is set to 0000h at compare 1 match) in the compare 1 matches. |
| Read from timer | <ul style="list-style-type: none"> • The count value can be read out by reading the TRF register. • The value in the compare register can be read out by reading registers TRFM0 and TRFM1. |
| Write to timer | Write to the TRF register is disabled |
| Select functions | <ul style="list-style-type: none"> • Output-compare output pin selected Either 1 pin or multiple pins among TRFO00 to TRFO02, or TRFO10 to TRFO12 (bits TRFOUT0 to TRFOUT5 in the TRFOUT register). • Output level at the compare match Selects “H”, “L”, inverted, or unchanged (bits TRFC14 to TRFC17 in the TRFCR1 register). • Output level inverted Selects output level inverted or not inverted (bits TRFOUT6 to TRFOUT7 in the TRFOUT register). • Output level at the count stops Selects “H”, “L”, or unchanged (bits TRFC05 to TRFC06 in the TRFCR0 register). • Timing to set the TRF register to 0000h Overflow or compare 1 match in the TRFM1 register (the CCLR bit in the TRFCR1 register). |

Table 14.57 Output in Output Compare Mode (Example of TRFO00 Pin)

| TRFO00 Output | | Bit Setting Value | | | | | |
|----------------|--|-------------------|--------|--------|-----------------|---------|-------------|
| | | TRFCR0 Register | | | TRFOUT Register | | P8 Register |
| | | TRFC06 | TRFC05 | TSTART | TRFOUT6 | TRFOUT0 | P8_0 |
| Counting | CMP output | X | X | 1 | 0 | 1 | 1 |
| | Inverted output of CMP output | X | X | 1 | 1 | 1 | 1 |
| | "L" output | X | X | 1 | 0 | 1 | 0 |
| | "H" output | X | X | 1 | 1 | 1 | 0 |
| Count stops | Holds output level before count stops | X | 0 | 0 | X | 1 | 1 |
| | "L" output | 0 | 1 | 0 | X | 1 | 1 |
| | "H" output | 1 | 1 | 0 | X | 1 | 1 |

X: 0 or 1

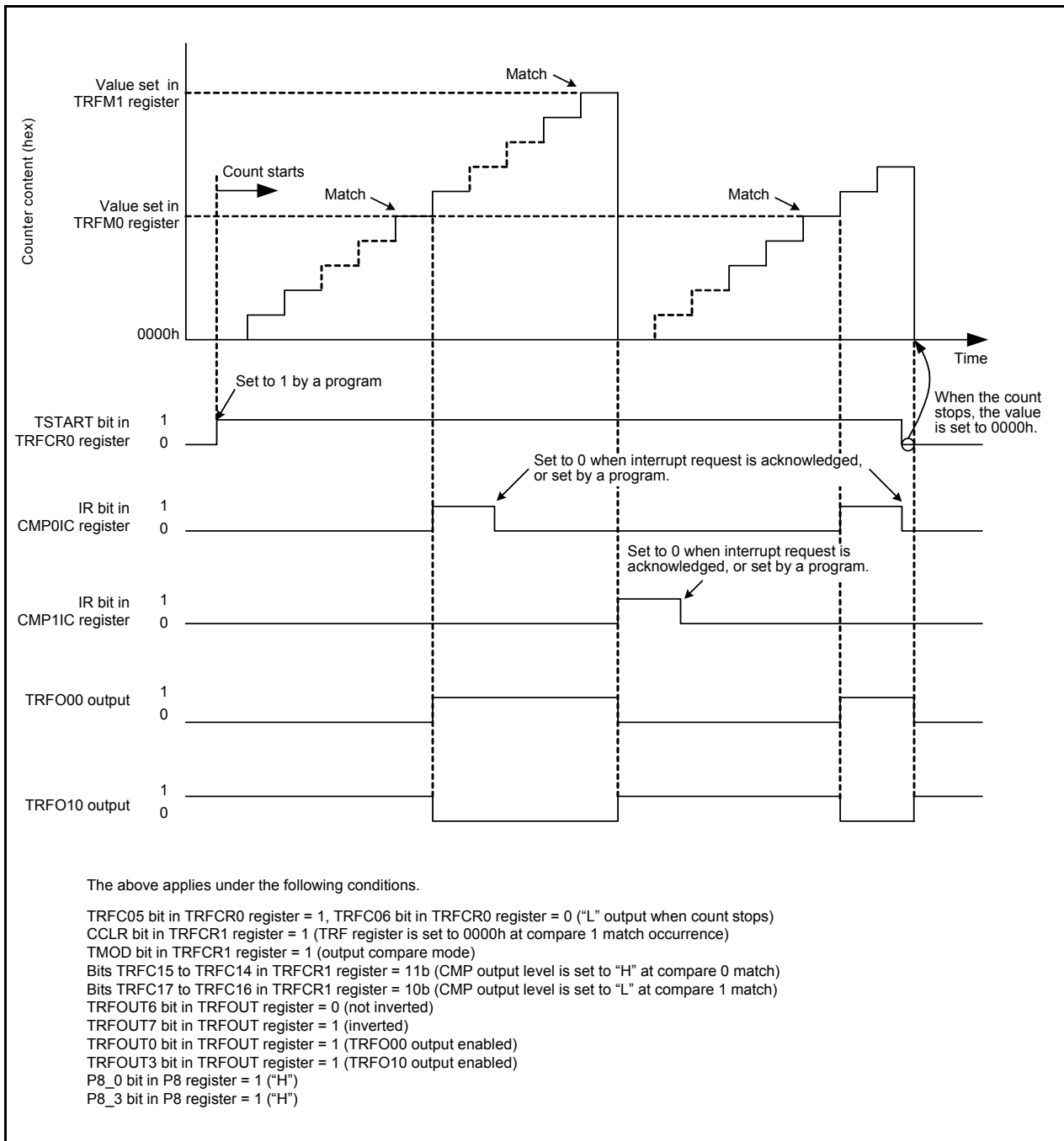


Figure 14.183 Operating Example in Output Compare Mode

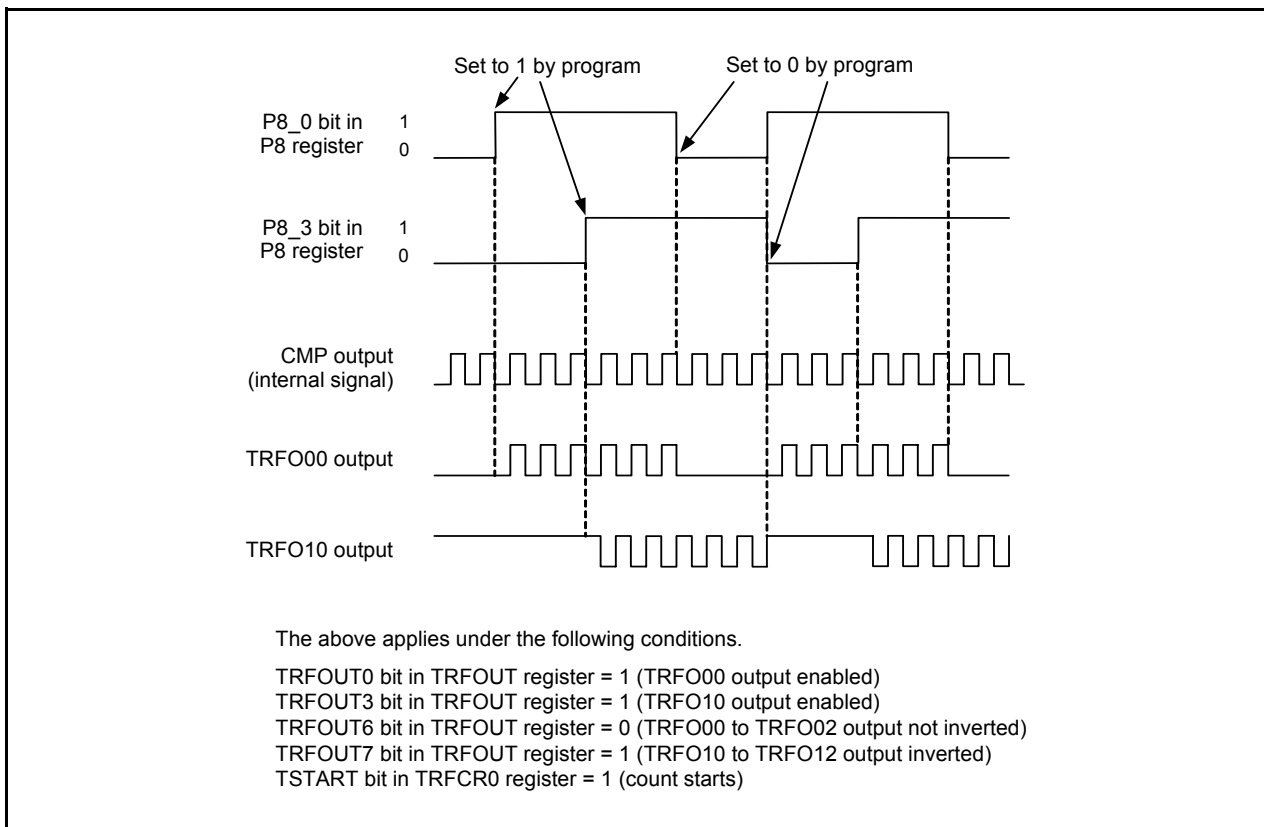


Figure 14.184 Operating Example in Output Compare Mode (“L” and “H” Held Output in Count Stops)

In output compare mode, the same PWM waveform is output from all of pins TRFO00 to TRFO02 and TRFO10 to TRFO12 during count operation. Note that the output waveform can be inverted for pins TRFO00 to TRFO02 or for pins TRFO10 to TRFO12. The output can also be fixed at “L” or “H” for individual pins for a given period.

The behavior when count operation stops can be selected from the following two options: the output level before the count stops is maintained, or output is fixed at “L” or “H”.

The values in the compare *i* register can be read by reading the TRFM_{*i*} (*i* = 0 or 1) register. Writing to the TRFM_{*i*} register causes the values to be stored in the compare *i* register in the following timing:

- If the TSTART bit is set to 0 (count stops)
Values are stored simultaneously with the write to the TRFM_{*i*} register.
- If the TSTART bit is set to 1 (count starts) and the CCLR bit in the TRFCR1 register is set to 0 (free running)
Values are stored when the TRF register (counter) overflows.
- If the TSTART bit is set to 1 and the CCLR bit is set to 1 (TRF register set to 0000h at compare 1 match)
Values are stored when the compare 1 and TRF register (counter) values match.

14.6.3 Notes on Timer RF

- Access registers TRF, TRFM0, and TRFM1 in 16-bit units.

Example of reading timer RF:

```
MOV.W    0290H,R0    ; Read out timer RF
```

- In input capture mode, a capture interrupt request is generated by inputting an edge selected by bits TRFC03 and TRFC04 in the TRFCR0 register even when the TSTART bit in the TRFCR0 register is set to 0 (count stops).

15. Serial Interface

The serial interface consists of three channels (UART0 to UART2). Each UARTi (i = 0 to 2) has an exclusive timer to generate the transfer clock and operates independently.

Figure 15.1 shows a UARTi (i = 0 to 2) Block Diagram. Figure 15.2 shows a UARTi Transmit/Receive Unit. Figure 15.3 shows a Block Diagram of CLK1 and CLK2 Pin Switching Unit.

UARTi has two modes: clock synchronous serial I/O mode and clock asynchronous serial I/O mode (UART mode). Figures 15.4 to 15.8 show the registers associated with UARTi.

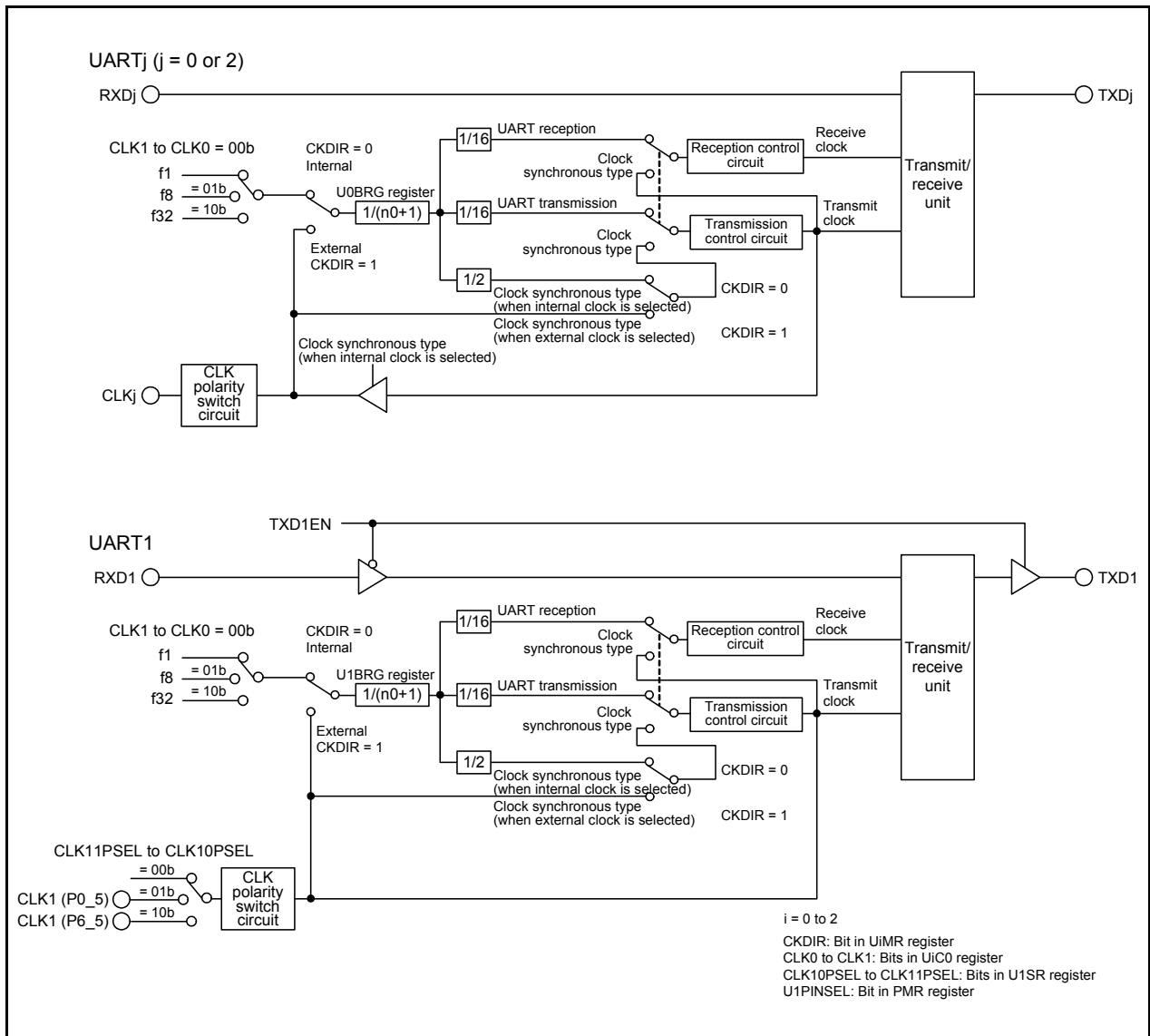


Figure 15.1 UARTi (i = 0 to 2) Block Diagram

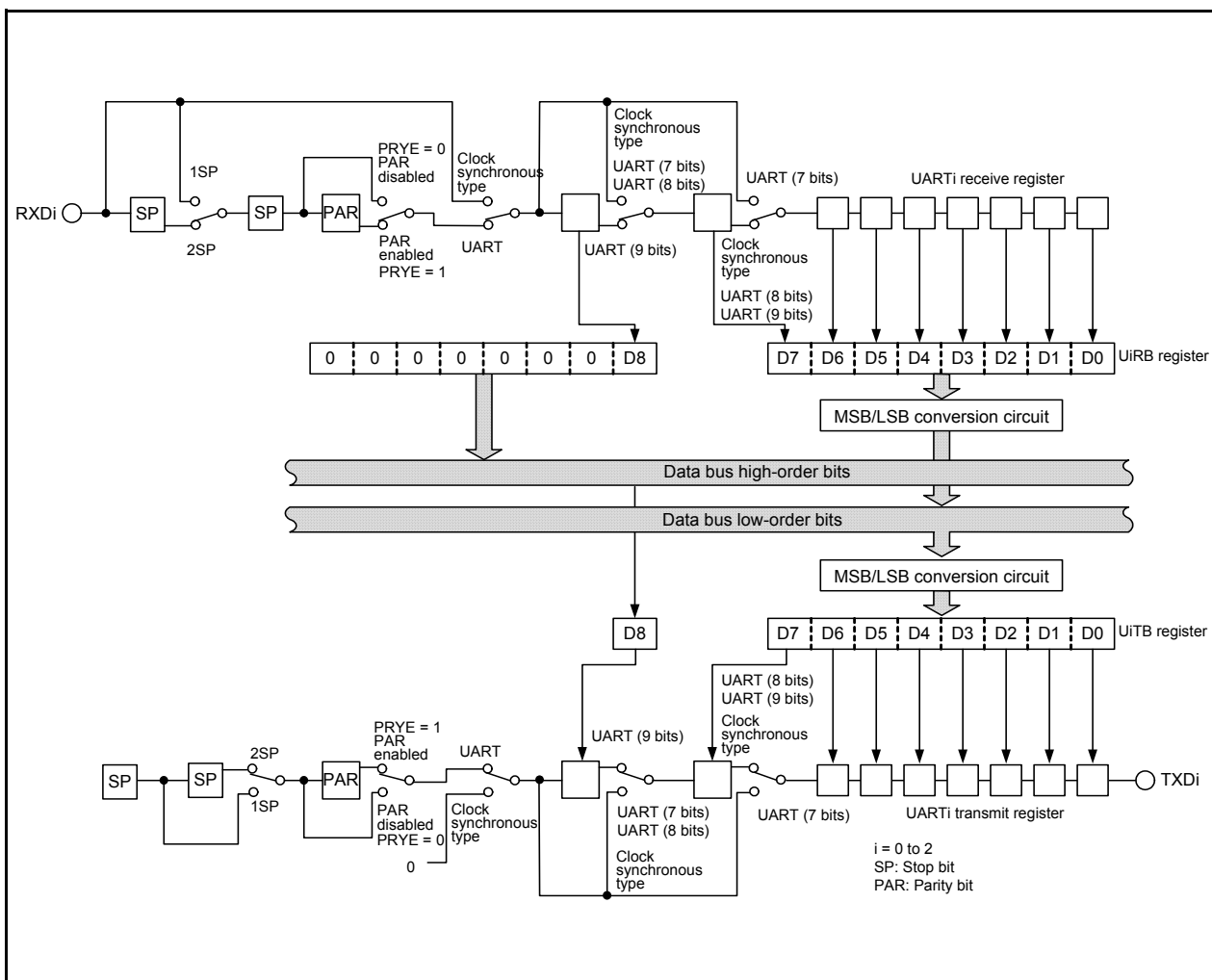


Figure 15.2 UARTi Transmit/Receive Unit

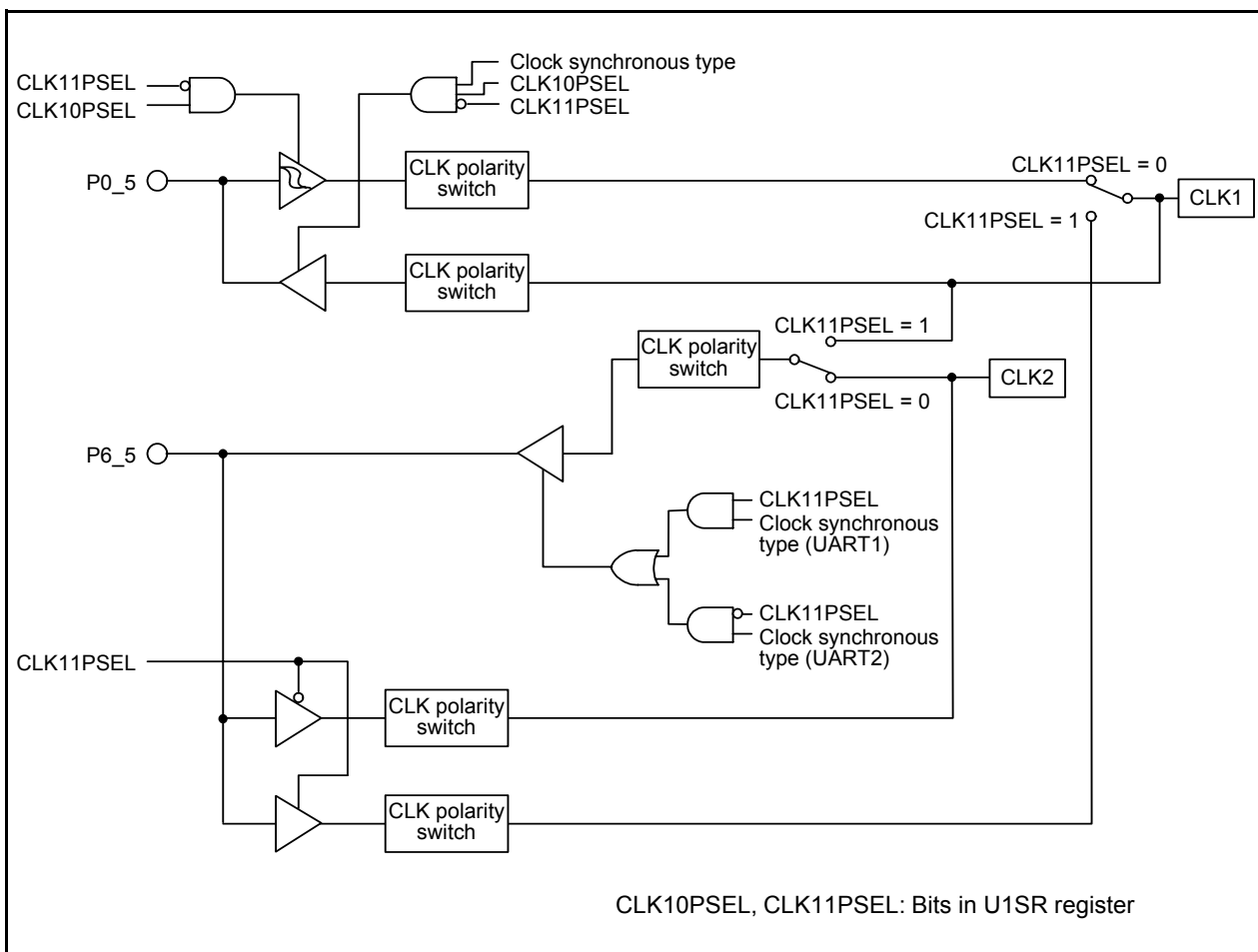


Figure 15.3 Block Diagram of CLK1 and CLK2 Pin Switching Unit

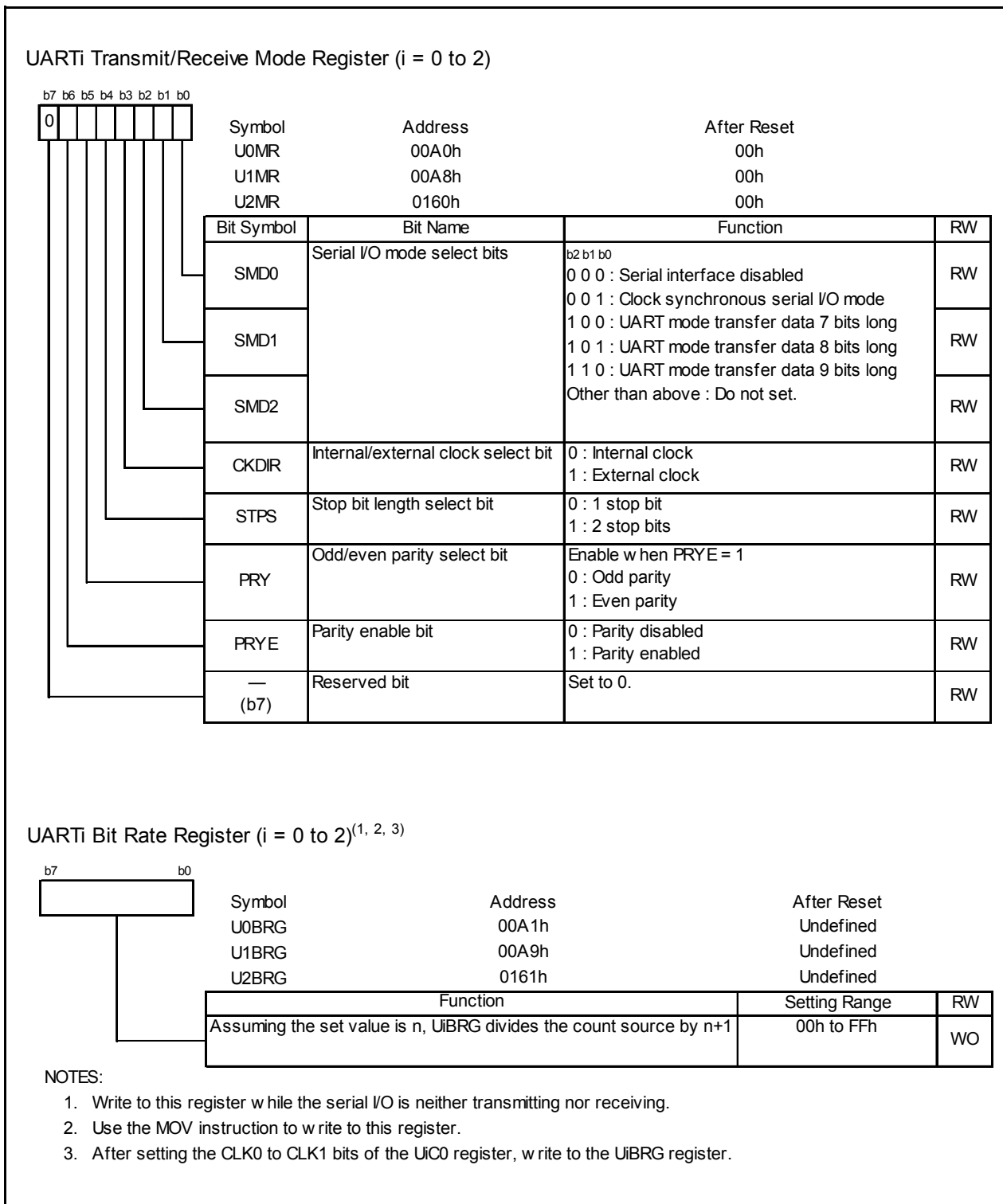


Figure 15.4 Registers U0MR to U2MR and U0BRG to U2BRG

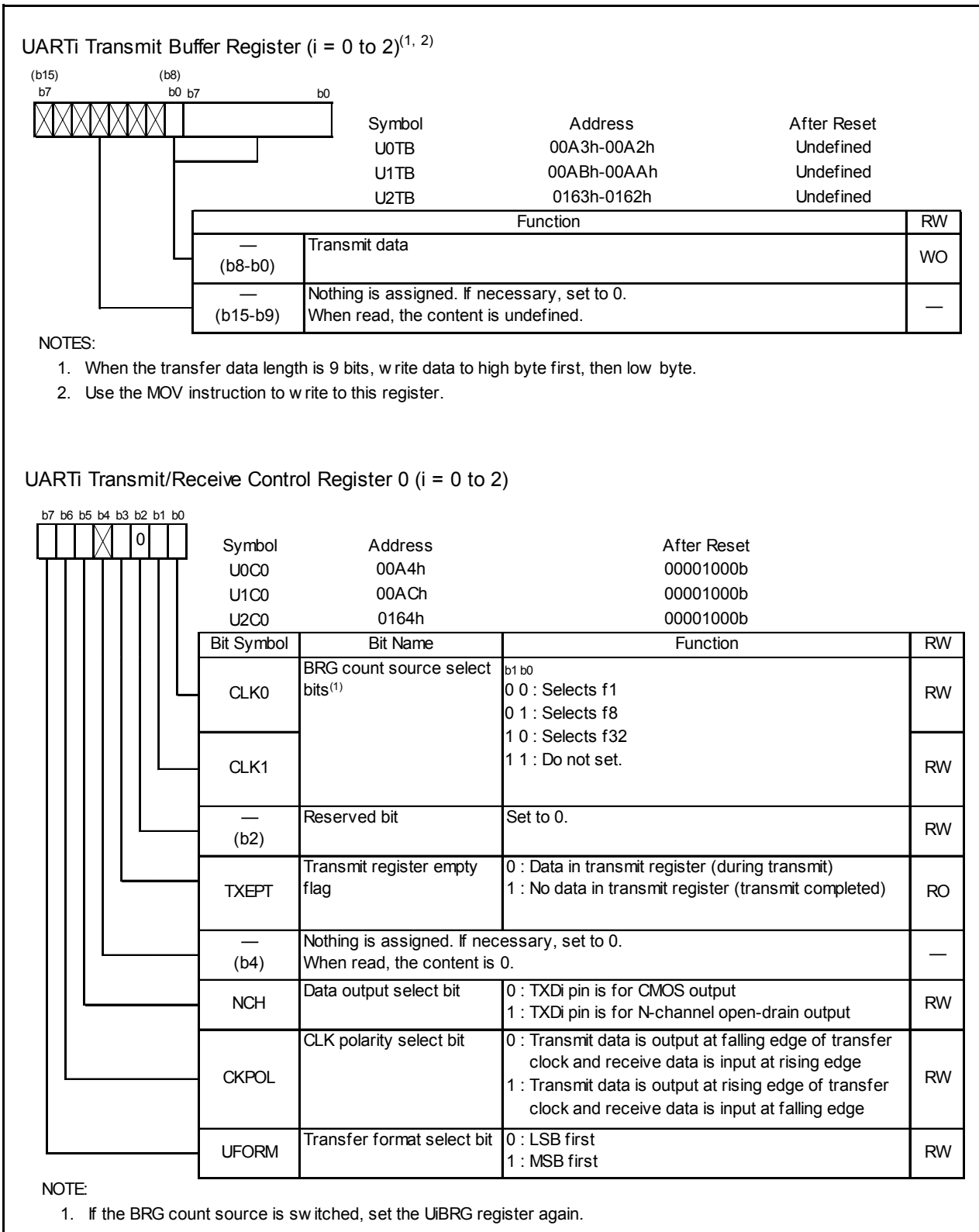


Figure 15.5 Registers U0TB to U2TB and U0C0 to U2C0

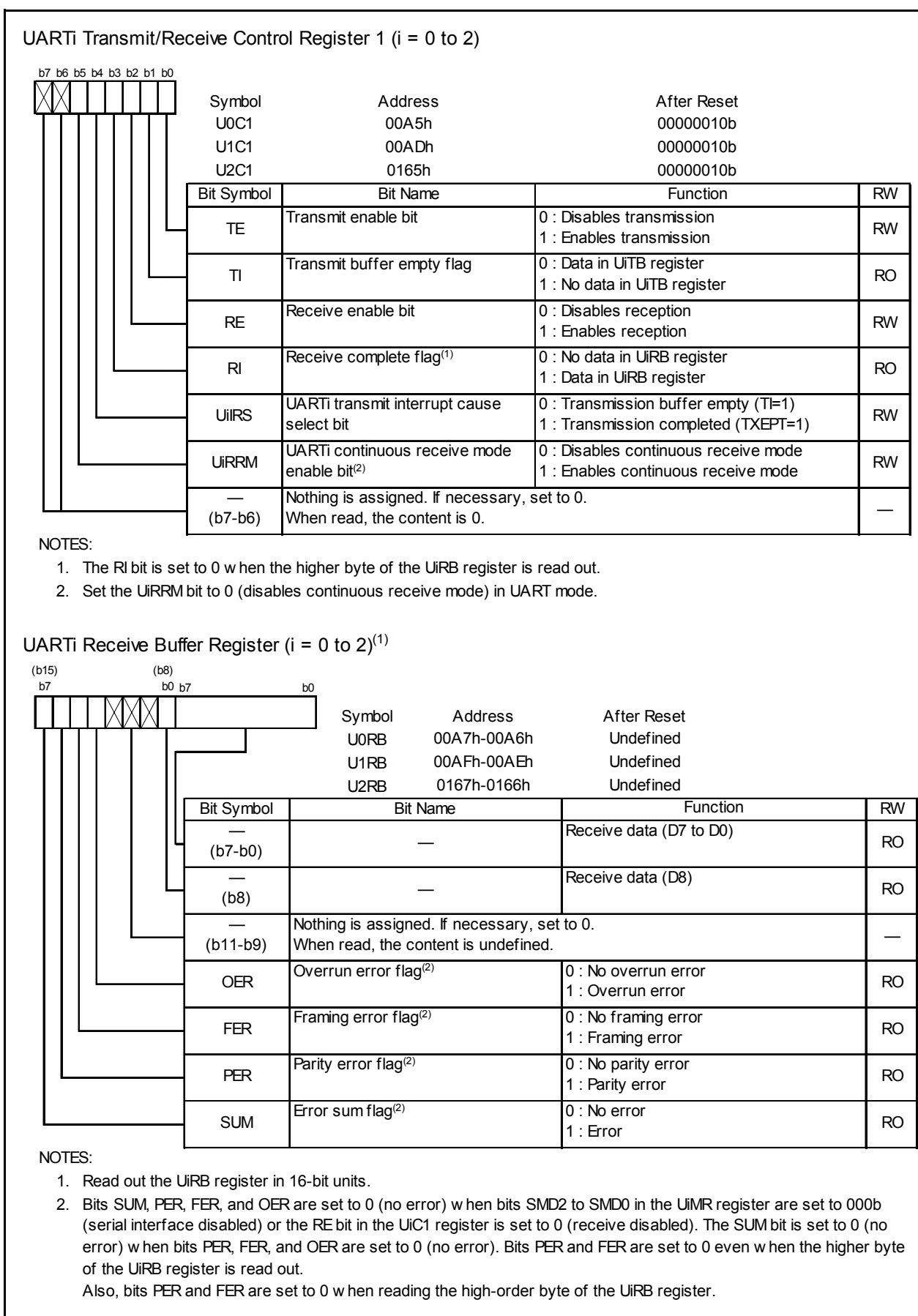


Figure 15.6 Registers U0C1 to U2C1 and U0RB to U2RB

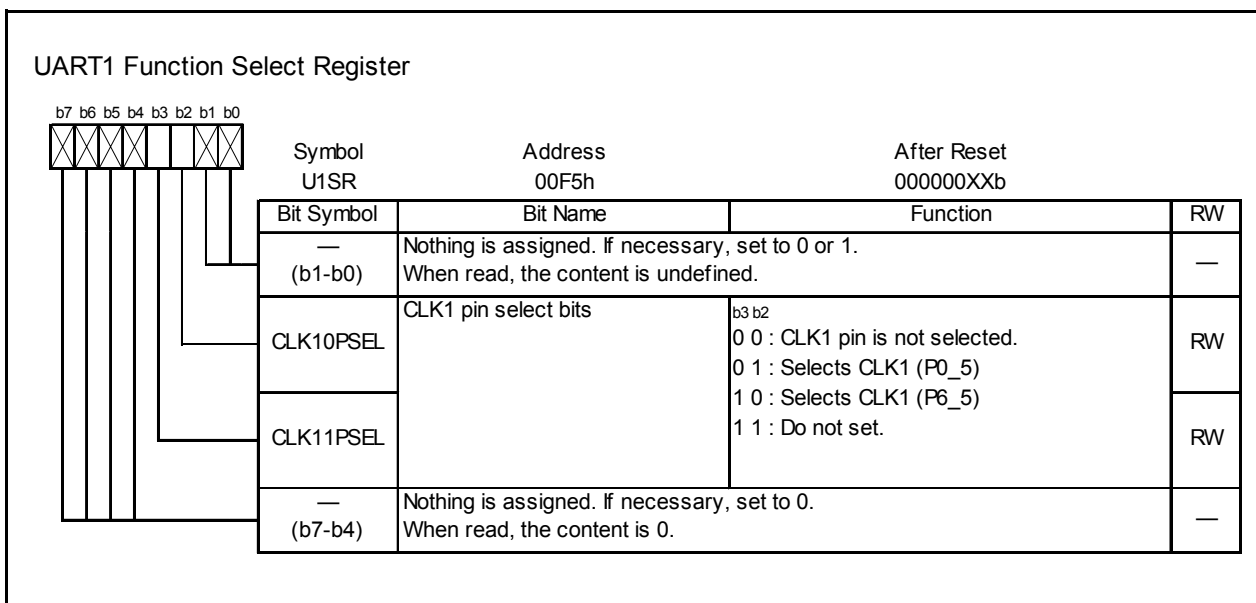


Figure 15.7 U1SR Register

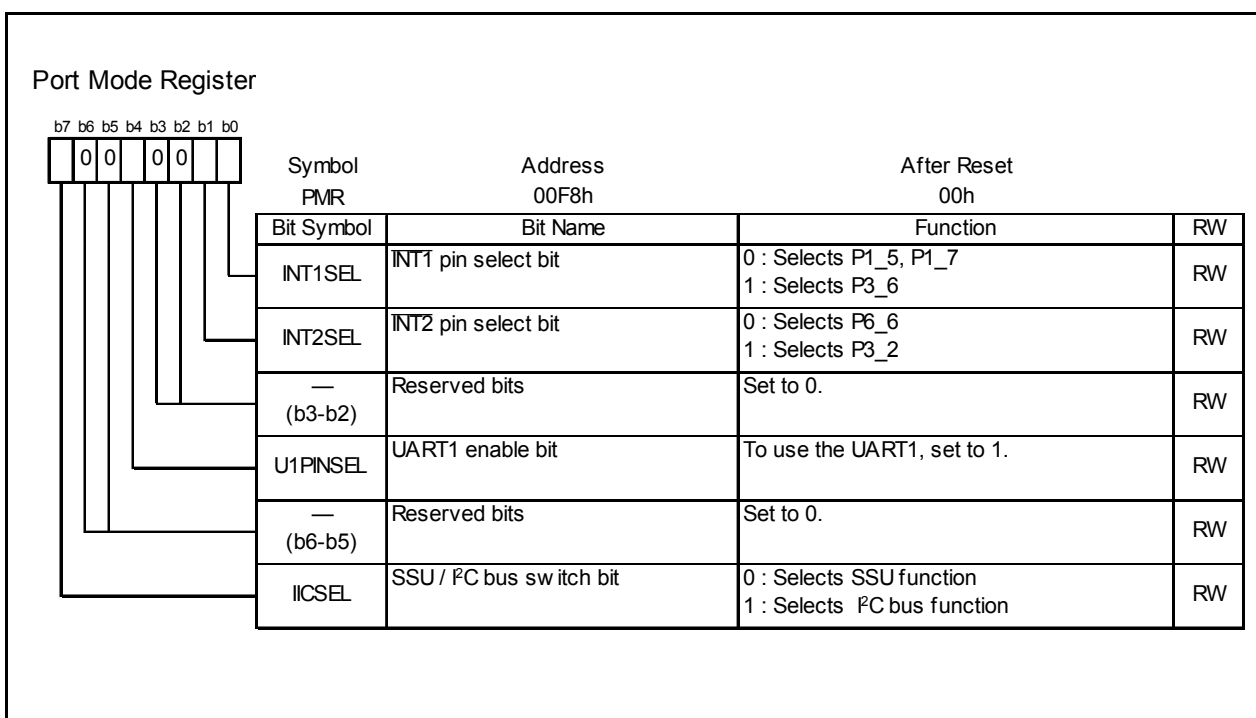


Figure 15.8 PMR Register

15.1 Clock Synchronous Serial I/O Mode

In clock synchronous serial I/O mode, data is transmitted and received using a transfer clock.

Table 15.1 lists the Clock Synchronous Serial I/O Mode Specifications. Table 15.2 lists the Registers Used and Settings in Clock Synchronous Serial I/O Mode⁽¹⁾.

Table 15.1 Clock Synchronous Serial I/O Mode Specifications

| Item | Specification |
|-------------------------------------|--|
| Transfer data format | <ul style="list-style-type: none"> Transfer data length: 8 bits |
| Transfer clocks | <ul style="list-style-type: none"> CKDIR bit in UiMR register is set to 0 (internal clock): $f_i/(2(n+1))$ $f_i = f_1, f_8, f_{32}$ $n =$ value set in UiBRG register: 00h to FFh The CKDIR bit is set to 1 (external clock): input from CLKi pin |
| Transmit start conditions | <ul style="list-style-type: none"> Before transmission starts, the following requirements must be met⁽¹⁾ <ul style="list-style-type: none"> The TE bit in the UiC1 register is set to 1 (transmission enabled) The TI bit in the UiC1 register is set to 0 (data in the UiTB register) |
| Receive start conditions | <ul style="list-style-type: none"> Before reception starts, the following requirements must be met⁽¹⁾ <ul style="list-style-type: none"> The RE bit in the UiC1 register is set to 1 (reception enabled) The TE bit in the UiC1 register is set to 1 (transmission enabled) The TI bit in the UiC1 register is set to 0 (data in the UiTB register) |
| Interrupt request generation timing | <ul style="list-style-type: none"> When transmitting, one of the following conditions can be selected <ul style="list-style-type: none"> The UiIRS bit is set to 0 (transmit buffer empty): When transferring data from the UiTB register to UARTi transmit register (when transmission starts). The UiIRS bit is set to 1 (transmission completes): When completing data transmission from UARTi transmit register. When receiving When data transfer from the UARTi receive register to the UiRB register (when reception completes). |
| Error detection | <ul style="list-style-type: none"> Overrun error⁽²⁾ This error occurs if the serial interface starts receiving the next data item before reading the UiRB register and receives the 7th bit of the next data. |
| Select functions | <ul style="list-style-type: none"> CLK polarity selection Transfer data input/output can be selected to occur synchronously with the rising or the falling edge of the transfer clock. LSB first, MSB first selection Whether transmitting or receiving data begins with bit 0 or begins with bit 7 can be selected. Continuous receive mode selection Receive is enabled immediately by reading the UiRB register. |

i = 0 to 2

NOTES:

- If an external clock is selected, ensure that the external clock is "H" when the CKPOL bit in the UiC0 register is set to 0 (transmit data output at falling edge and receive data input at rising edge of transfer clock), and that the external clock is "L" when the CKPOL bit is set to 1 (transmit data output at rising edge and receive data input at falling edge of transfer clock).
- If an overrun error occurs, the receive data (b0 to b8) of the UiRB register will be undefined. The IR bit in the SiRIC register remains unchanged.

Table 15.2 Registers Used and Settings in Clock Synchronous Serial I/O Mode⁽¹⁾

| Register | Bit | Function |
|----------|--------------|--|
| UiTB | 0 to 7 | Set data transmission |
| UiRB | 0 to 7 | Data reception can be read |
| | OER | Overrun error flag |
| UiBRG | 0 to 7 | Set bit rate |
| UiMR | SMD2 to SMD0 | Set to 001b |
| | CKDIR | Select the internal clock or external clock |
| UiC0 | CLK1 to CLK0 | Select the count source in the UiBRG register |
| | TXEPT | Transmit register empty flag |
| | NCH | Select TXDi pin output mode |
| | CKPOL | Select the transfer clock polarity |
| | UFORM | Select the LSB first or MSB first |
| UiC1 | TE | Set this bit to 1 to enable transmission/reception |
| | TI | Transmit buffer empty flag |
| | RE | Set this bit to 1 to enable reception |
| | RI | Reception complete flag |
| | UiIRS | Select the UARTi transmit interrupt source |
| | UiRRM | Set this bit to 1 to use continuous receive mode |

i = 0 to 2

NOTE:

1. Set bits which are not in this table to 0 when writing to the above registers in clock synchronous serial I/O mode.

Table 15.3 lists the I/O Pin Functions in Clock Synchronous Serial I/O Mode. The TXDi pin outputs “H” level between the operating mode selection of UARTi (i = 0 to 2) and transfer start. (If the NCH bit is set to 1 (N-channel open-drain output), this pin is in a high-impedance state.)

Table 15.3 I/O Pin Functions in Clock Synchronous Serial I/O Mode

| Pin Name | Function | Selection Method |
|------------------------|-----------------------|--|
| TXD0 (P1_4) | Output serial data | (Outputs dummy data when performing reception only) |
| RXD0 (P1_5) | Input serial data | PD1_5 bit in PD1 register = 0 (P1_5 can be used as an input port when performing transmission only) |
| CLK0 (P1_6) | Output transfer clock | CKDIR bit in U0MR register = 0 |
| | Input transfer clock | CKDIR bit in U0MR register = 1 PD1_6 bit in PD1 register = 0 |
| TXD1 (P6_6) | Output serial data | U1PINSEL bit in PMR register = 1 (Outputs dummy data when performing reception only) |
| RXD1 (P6_7) | Input serial data | U1PINSEL bit in PMR register = 1 PD6_7 bit in PD6 register = 0 (P6_7 can be used as an input port when performing transmission only) |
| CLK1 (P0_5 or P6_5) | Output transfer clock | <ul style="list-style-type: none"> • When CLK1 (P0_5) Bits CLK11PSEL to CLK10PSEL in U1SR register = 01b (P0_5) CKDIR bit in U1MR register = 0 • When CLK1 (P6_5) Bits CLK11PSEL to CLK10PSEL in U1SR register = 10b (P6_5) CKDIR bit in U1MR register = 0 |
| | Input transfer clock | <ul style="list-style-type: none"> • When CLK1 (P0_5) Bits CLK11PSEL to CLK10PSEL in U1SR register = 01b (P0_5) PD0_5 bit in PD0 register = 0 CKDIR bit in U1MR register = 1 • When CLK1 (P6_5) Bits CLK11PSEL to CLK10PSEL in U1SR register = 10b (P6_5) PD6_5 bit in PD6 register = 0 CKDIR bit in U1MR register = 1 |
| TXD2 (P6_3) | Output serial data | (Outputs dummy data when performing reception only) |
| RXD2 (P6_4) | Input serial data | PD6_4 bit in PD6 register = 0 (P6_4 can be used as an input port when performing transmission only) |
| CLK2 (P6_5) | Output transfer clock | CKDIR bit in U2MR register = 0 |
| | Input transfer clock | CKDIR bit in U2MR register = 1 PD6_6 bit in PD6 register = 0 |

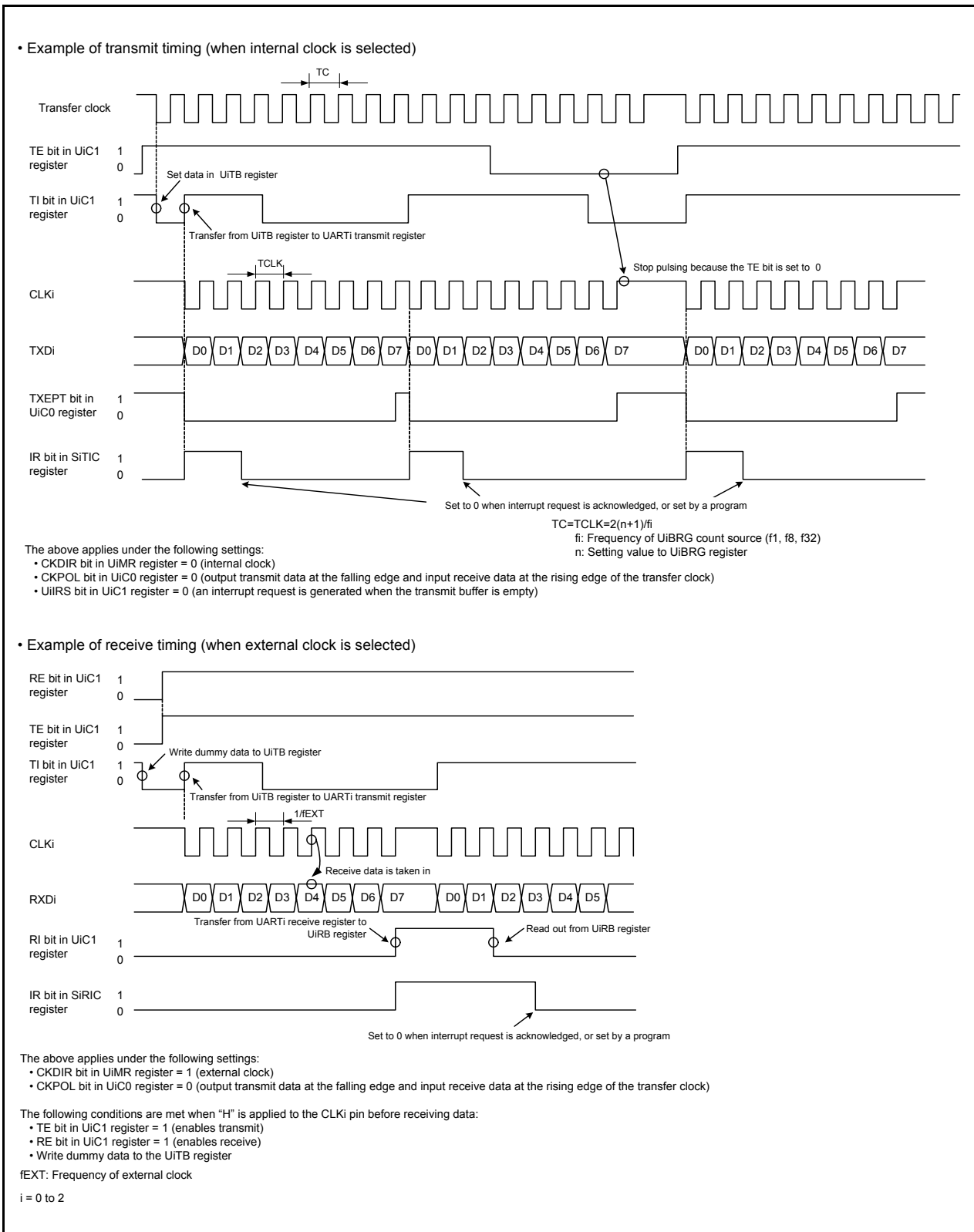


Figure 15.9 Transmit and Receive Timing Example in Clock Synchronous Serial I/O Mode

15.1.1 Polarity Select Function

Figure 15.10 shows the Transfer Clock Polarity. Use the CKPOL bit in the UiC0 (i = 0 to 2) register to select the transfer clock polarity.

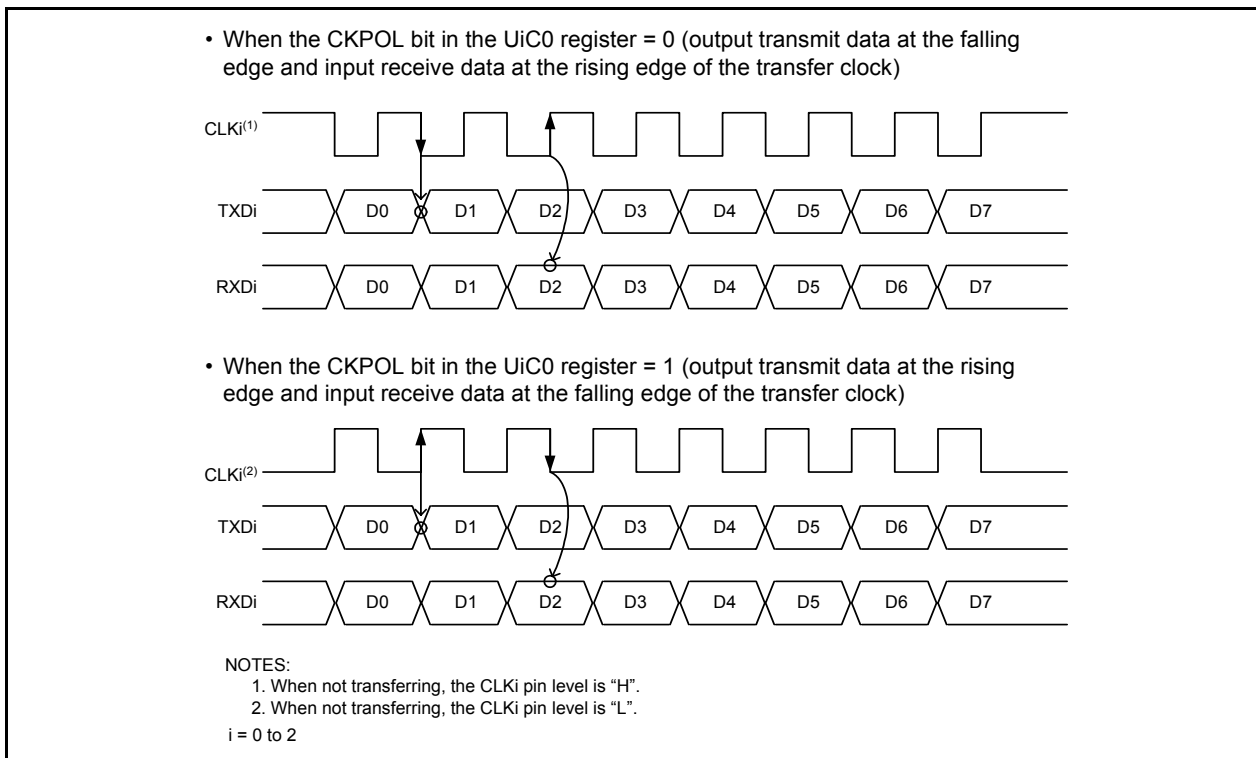


Figure 15.10 Transfer Clock Polarity

15.1.2 LSB First/MSB First Select Function

Figure 15.11 shows the Transfer Format. Use the UFORM bit in the UiC0 (i = 0 to 2) register to select the transfer format.

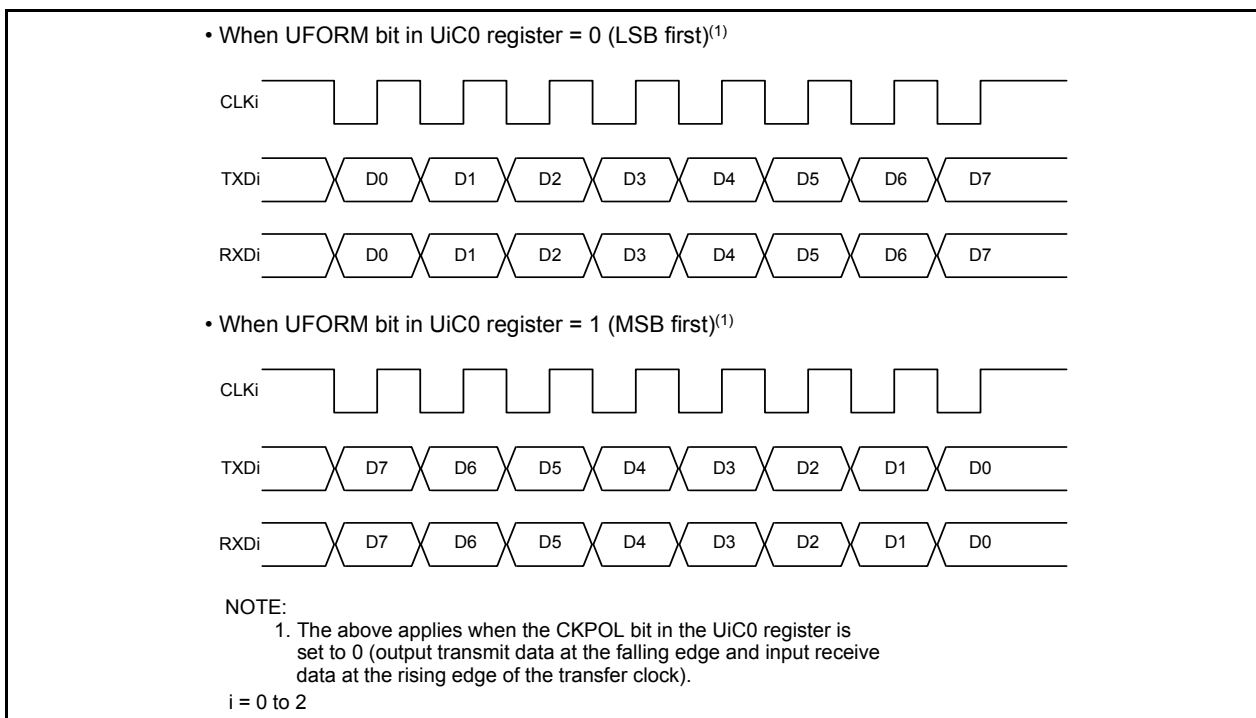


Figure 15.11 Transfer Format

15.1.3 Continuous Receive Mode

Continuous receive mode is selected by setting the UiRRM ($i = 0$ to 2) bit in the UiC1 register to 1 (enables continuous receive mode). In this mode, reading the UiRB register sets the TI bit in the UiC1 register to 0 (data in the UiTB register). When the UiRRM bit is set to 1, do not write dummy data to the UiTB register by a program.

15.2 Clock Asynchronous Serial I/O (UART) Mode

The UART mode allows data transmission and reception after setting the desired bit rate and transfer data format. Table 15.4 lists the UART Mode Specifications. Table 15.5 lists the Registers Used and Settings for UART Mode.

Table 15.4 UART Mode Specifications

| Item | Specification |
|-------------------------------------|---|
| Transfer data formats | <ul style="list-style-type: none"> • Character bit (transfer data): Selectable among 7, 8 or 9 bits • Start bit: 1 bit • Parity bit: Selectable among odd, even, or none • Stop bit: Selectable among 1 or 2 bits |
| Transfer clocks | <ul style="list-style-type: none"> • CKDIR bit in UiMR register is set to 0 (internal clock): $f_j/(16(n+1))$ $f_j = f_1, f_8, f_{32}$ $n =$ value set in UiBRG register: 00h to FFh • CKDIR bit is set to 1 (external clock): $f_{EXT}/(16(n+1))$ f_{EXT}: Input from CLKi pin, $n =$ value set in UiBRG register: 00h to FFh |
| Transmit start conditions | <ul style="list-style-type: none"> • Before transmission starts, the following are required <ul style="list-style-type: none"> - TE bit in UiC1 register is set to 1 (transmission enabled) - TI bit in UiC1 register is set to 0 (data in UiTB register) |
| Receive start conditions | <ul style="list-style-type: none"> • Before reception starts, the following are required <ul style="list-style-type: none"> - RE bit in UiC1 register is set to 1 (reception enabled) - Start bit detected |
| Interrupt request generation timing | <ul style="list-style-type: none"> • When transmitting, one of the following conditions can be selected <ul style="list-style-type: none"> - UiIRS bit is set to 0 (transmit buffer empty): When transferring data from the UiTB register to UARTi transmit register (when transmission starts). - UiIRS bit is set to 1 (transfer ends): When serial interface completes transmitting data from the UARTi transmit register • When receiving When transferring data from the UARTi receive register to UiRB register (when reception ends). |
| Error detection | <ul style="list-style-type: none"> • Overrun error⁽¹⁾ This error occurs if the serial interface starts receiving the next data item before reading the UiRB register and receive the bit preceding the final stop bit of the next data item. • Framing error This error occurs when the set number of stop bits is not detected. • Parity error This error occurs when parity is enabled, and the number of 1's in parity and character bits do not match the number of 1's set. • Error sum flag This flag is set is set to 1 when an overrun, framing, or parity error is generated. |

i = 0 to 2

NOTE:

1. If an overrun error occurs, the receive data (b0 to b8) of the UiRB register will be undefined. The IR bit in the SiRIC register remains unchanged.

Table 15.5 Registers Used and Settings for UART Mode

| Register | Bit | Function |
|----------|--------------------|---|
| UiTB | 0 to 8 | Set transmit data ⁽¹⁾ |
| UiRB | 0 to 8 | Receive data can be read ^(1, 2) |
| | OER, FER, PER, SUM | Error flag |
| UiBRG | 0 to 7 | Set a bit rate |
| UiMR | SMD2 to SMD0 | Set to 100b when transfer data is 7 bits long Set to 101b when transfer data is 8 bits long Set to 110b when transfer data is 9 bits long |
| | CKDIR | Select the internal clock or external clock |
| | STPS | Select the stop bit |
| | PRY, PRYE | Select whether parity is included and whether odd or even |
| UiC0 | CLK0, CLK1 | Select the count source for the UiBRG register |
| | TXEPT | Transmit register empty flag |
| | NCH | Select TXDi pin output mode |
| | CKPOL | Set to 0 |
| | UFORM | LSB first or MSB first can be selected when transfer data is 8 bits long. Set to 0 when transfer data is 7 or 9 bits long. |
| UiC1 | TE | Set to 1 to enable transmit |
| | TI | Transmit buffer empty flag |
| | RE | Set to 1 to enable receive |
| | RI | Receive complete flag |
| | UiIRS | Select the source of UARTi transmit interrupt |
| | UiRRM | Set to 0 |

i = 0 to 2

NOTES:

1. The bits used for transmit/receive data are as follows: Bits 0 to 6 when transfer data is 7 bits long; bits 0 to 7 when transfer data is 8 bits long; bits 0 to 8 when transfer data is 9 bits long.
2. The following bits are undefined: Bits 7 and 8 when transfer data is 7 bits long; bit 8 when transfer data is 8 bits long.

Table 15.6 lists the I/O Pin Functions in UART Mode. After the UART_i (i = 0 to 2) operating mode is selected, the TXD_i pin outputs “H” level. (If the NCH bit is set to 1 (N-channel open-drain output), this pin is in a high-impedance state) until transfer starts.)

Table 15.6 I/O Pin Functions in UART Mode

| Pin name | Function | Selection Method |
|---------------------|-----------------------|--|
| TXD0 (P1_4) | Output serial data | (Cannot be used as a port when performing reception only) |
| RXD0 (P1_5) | Input serial data | PD1_5 bit in PD1 register = 0 (P1_5 can be used as an input port when performing transmission only) |
| CLK0 (P1_6) | Programmable I/O Port | CKDIR bit in U0MR register = 0 |
| | Input transfer clock | CKDIR bit in U0MR register = 1 PD1_6 bit in PD1 register = 0 |
| TXD1 (P6_6) | Output serial data | U1PINSEL bit in PMR register = 1 (Cannot be used as a port when performing reception only) |
| RXD1 (P6_7) | Input serial data | U1PINSEL bit in PMR register = 1 PD6_7 bit in PD6 register = 0 (P6_7 can be used as an input port when performing transmission only) |
| CLK1 (P0_5 or P6_5) | Programmable I/O Port | Bits CLK11PSEL to CLK10PSEL in U1SR register = 00b (CLK1 pin is not selected) |
| | Input transfer clock | <ul style="list-style-type: none"> • When CLK1 (P0_5) Bits CLK11PSEL to CLK10PSEL in U1SR register = 01b (P0_5) PD0_5 bit in PD0 register = 0 CKDIR bit in U1MR register = 1 • When CLK1 (P6_5) Bits CLK11PSEL to CLK10PSEL in U1SR register = 10b (P6_5) PD6_5 bit in PD6 register = 0 CKDIR bit in U1MR register = 1 |
| TXD2 (P6_3) | Output serial data | (Cannot be used as a port when performing reception only) |
| RXD2 (P6_4) | Input serial data | PD6_4 bit in PD6 register = 0 (P6_4 can be used as an input port when performing transmission only) |
| CLK2 (P6_5) | Programmable I/O Port | CKDIR bit in U2MR register = 0 |
| | Input transfer clock | CKDIR bit in U2MR register = 1 PD6_6 bit in PD6 register = 0 |

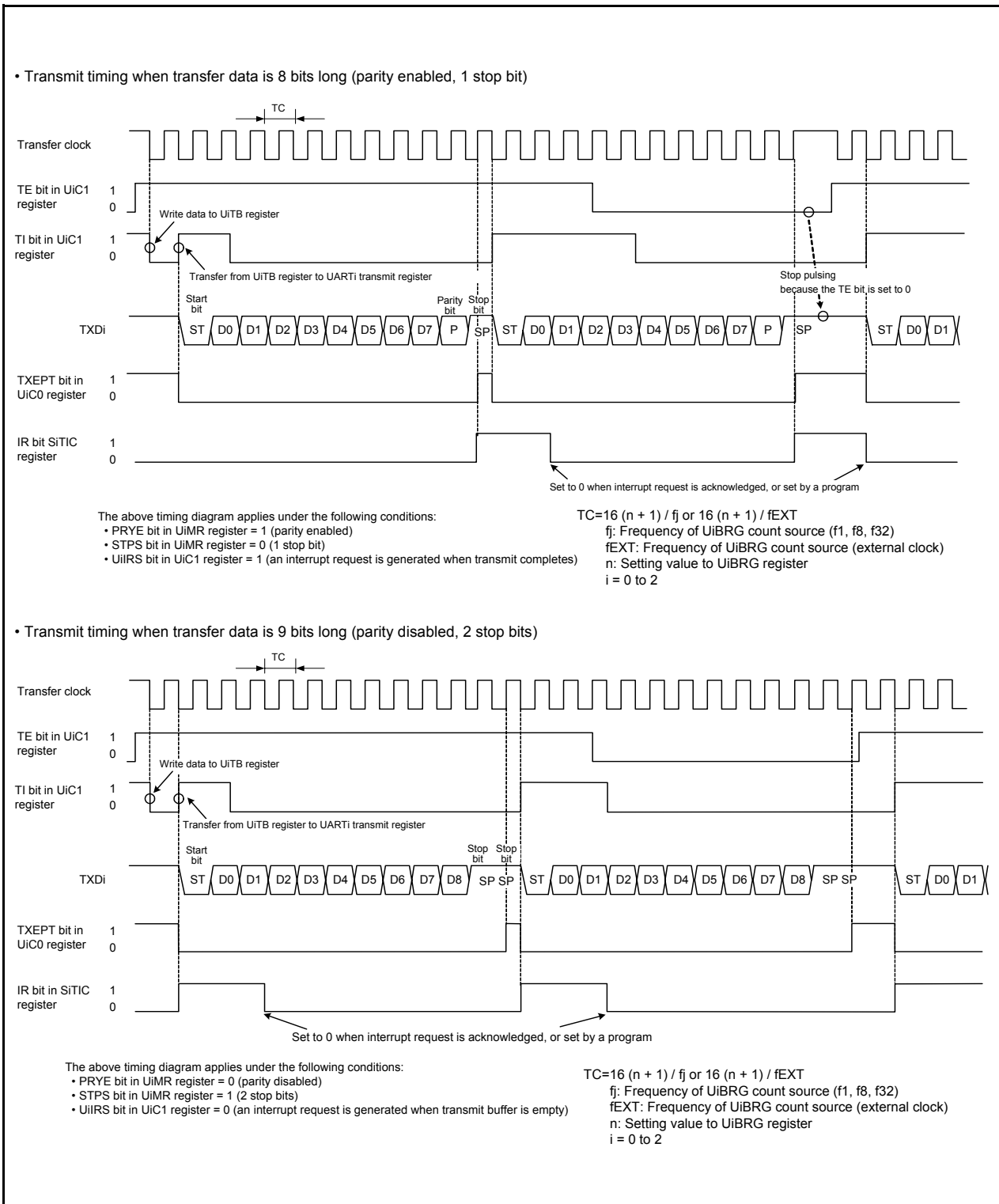


Figure 15.12 Transmit Timing in UART Mode

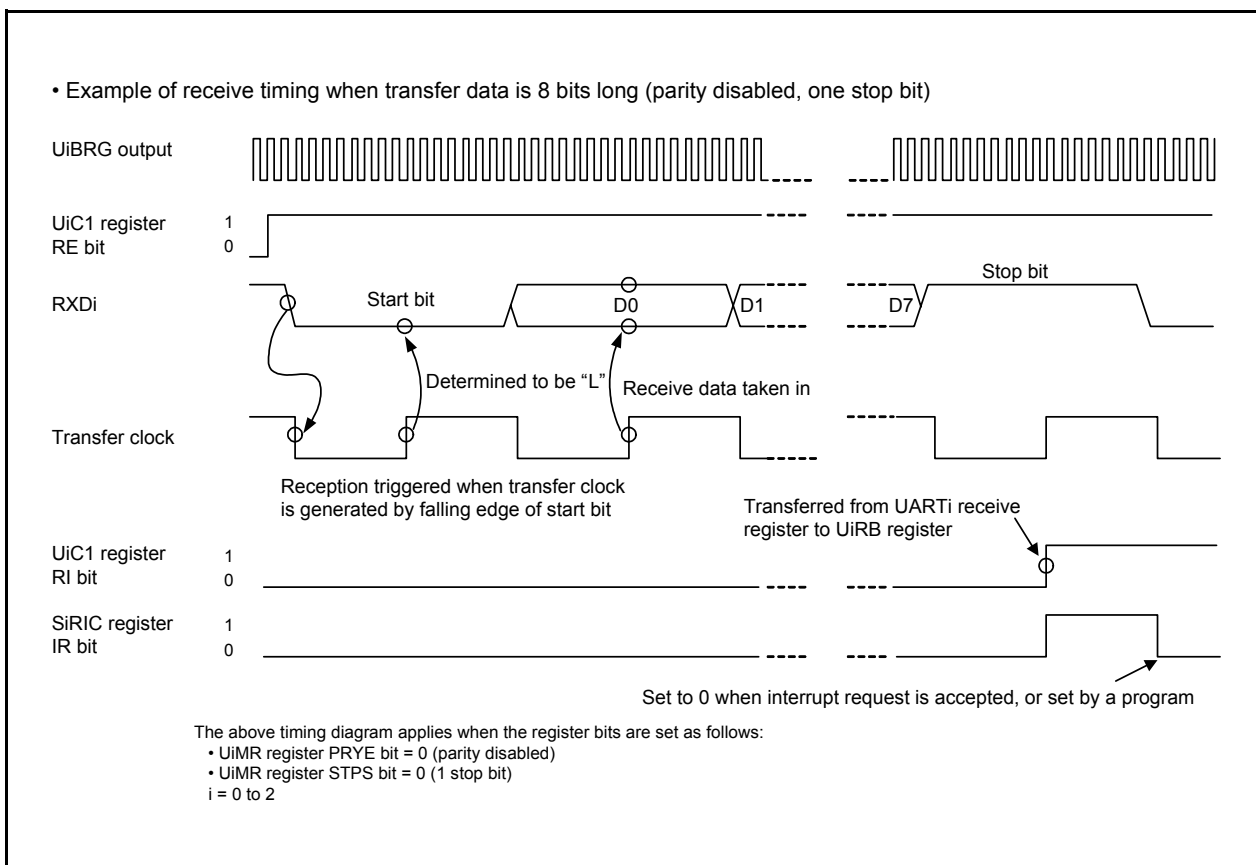


Figure 15.13 Receive Timing Example in UART Mode

15.2.1 Bit Rate

In UART mode, the bit rate is the frequency divided by the UiBRG (i = 0 to 2) register.

| | |
|--|--|
| UART mode | |
| • Internal clock selected | |
| $\text{UiBRG register setting value} = \frac{f_j}{\text{Bit Rate} \times 16} - 1$ | |
| Fj: Count source frequency of the UiBRG register (f1, f8, or f32) | |
| • External clock selected | |
| $\text{UiBRG register setting value} = \frac{f_{\text{EXT}}}{\text{Bit Rate} \times 16} - 1$ | |
| fEXT: Count source frequency of the UiBRG register (external clock) | |
| i = 0 to 2 | |

Figure 15.14 Calculation Formula of UiBRG (i = 0 to 2) Register Setting Value

Table 15.7 Bit Rate Setting Example in UART Mode (Internal Clock Selected)

| Bit Rate (bps) | BRG Count Source | System Clock = 20 MHz | | | System Clock = 8 MHz | | |
|----------------|------------------|-----------------------|-------------------|-----------|----------------------|-------------------|-----------|
| | | UiBRG Setting Value | Actual Time (bps) | Error (%) | UiBRG Setting Value | Actual Time (bps) | Error (%) |
| 1200 | f8 | 129 (81h) | 1201.92 | 0.16 | 51 (33h) | 1201.92 | 0.16 |
| 2400 | f8 | 64 (40h) | 2403.85 | 0.16 | 25 (19h) | 2403.85 | 0.16 |
| 4800 | f8 | 32 (20h) | 4734.85 | -1.36 | 12 (0Ch) | 4807.69 | 0.16 |
| 9600 | f1 | 129 (81h) | 9615.38 | 0.16 | 51 (33h) | 9615.38 | 0.16 |
| 14400 | f1 | 86 (56h) | 14367.82 | -0.22 | 34 (22h) | 14285.71 | -0.79 |
| 19200 | f1 | 64 (40h) | 19230.77 | 0.16 | 25 (19h) | 19230.77 | 0.16 |
| 28800 | f1 | 42 (2Ah) | 29069.77 | 0.94 | 16 (10h) | 29411.76 | 2.12 |
| 31250 | f1 | 39 (27h) | 31250.00 | 0.00 | 15 (0Fh) | 31250.00 | 0.00 |
| 38400 | f1 | 32 (20h) | 37878.79 | -1.36 | 12 (0Ch) | 38461.54 | 0.16 |
| 51200 | f1 | 23 (17h) | 52083.33 | 1.73 | 9 (09h) | 50000.00 | -2.34 |

i = 0 to 2

15.3 Notes on Serial Interface

- When reading data from the UiRB (i = 0 to 2) register either in the clock synchronous serial I/O mode or in the clock asynchronous serial I/O mode. Ensure the data is read in 16-bit units. When the high-order byte of the UiRB register is read, bits PER and FER in the UiRB register and the RI bit in the UiC1 register are set to 0. To check receive errors, read the UiRB register and then use the read data.

Example (when reading receive buffer register):

```
MOV.W    00A6H,R0    ; Read the UORB register
```

- When writing data to the UiTB register in the clock asynchronous serial I/O mode with 9-bit transfer data length, write data to the high-order byte first then the low-order byte, in 8-bit units.

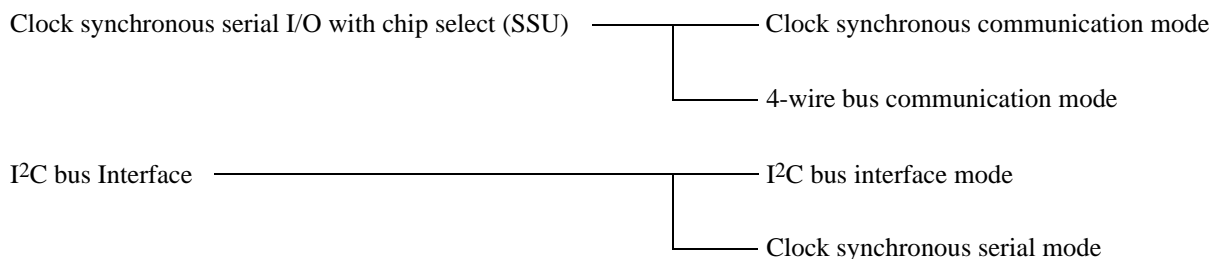
Example (when reading transmit buffer register):

```
MOV.B    #XXH,00A3H  ; Write the high-order byte of UOTB register  
MOV.B    #XXH,00A2H  ; Write the low-order byte of UOTB register
```

16. Clock Synchronous Serial Interface

The clock synchronous serial interface is configured as follows.

Clock synchronous serial interface



The clock synchronous serial interface uses the registers at addresses 00B8h to 00BFh. Registers, bits, symbols, and functions vary even for the same addresses depending on the mode. Refer to the register diagrams of each function for details.

Also, the differences between clock synchronous communication mode and clock synchronous serial mode are the options of the transfer clock, clock output format, and data output format.

16.1 Mode Selection

The clock synchronous serial interface has four modes.

Table 16.1 lists the Mode Selections. Refer to **16.2 Clock Synchronous Serial I/O with Chip Select (SSU)** and the sections that follow for details of each mode.

Table 16.1 Mode Selections

| IICSEL Bit in PMR Register | Bit 7 in 00B8h (ICE Bit in ICCR1 Register) | Bit 0 in 00BDh (SSUMS Bit in SSMR2 Register, FS Bit in SAR Register) | Function | Mode |
|----------------------------|--|--|---|--------------------------------------|
| 0 | 0 | 0 | Clock synchronous serial I/O with chip select | Clock synchronous communication mode |
| 0 | 0 | 1 | | 4-wire bus communication mode |
| 1 | 1 | 0 | I ² C bus interface | I ² C bus interface mode |
| 1 | 1 | 1 | | Clock synchronous serial mode |

16.2 Clock Synchronous Serial I/O with Chip Select (SSU)

Clock synchronous serial I/O with chip select supports clock synchronous serial data communication.

Table 16.2 shows a Clock Synchronous Serial I/O with Chip Select Specifications and Figure 16.1 shows a Block Diagram of Clock Synchronous Serial I/O with Chip Select. Figures 16.2 to 16.10 show clock synchronous serial I/O with chip select associated registers.

Table 16.2 Clock Synchronous Serial I/O with Chip Select Specifications

| Item | Specification |
|-----------------------------|---|
| Transfer data format | <ul style="list-style-type: none"> Transfer data length: 8 bits Continuous transmission and reception of serial data are supported since both transmitter and receiver have buffer structures. |
| Operating modes | <ul style="list-style-type: none"> Clock synchronous communication mode 4-wire bus communication mode (including bidirectional communication) |
| Master/slave device | Selectable |
| I/O pins | SSCK (I/O): Clock I/O pin SSI (I/O): Data I/O pin SSO (I/O): Data I/O pin $\overline{\text{SCS}}$ (I/O): Chip-select I/O pin |
| Transfer clocks | <ul style="list-style-type: none"> When the MSS bit in the SSCRH register is set to 0 (operates as slave device), external clock is selected (input from SSCK pin). When the MSS bit in the SSCRH register is set to 1 (operates as master device), internal clock (selectable among f1/256, f1/128, f1/64, f1/32, f1/16, f1/8 and f1/4, output from SSCK pin) is selected. Clock polarity and phase of SSCK can be selected. |
| Receive error detection | <ul style="list-style-type: none"> Overrun error Overrun error occurs during reception and completes in error. While the RDRF bit in the SSSR register is set to 1 (data in the SSRDR register) and when next serial data receive is completed, the ORER bit is set to 1. |
| Multimaster error detection | <ul style="list-style-type: none"> Conflict error When the SSUMS bit in the SSMR2 register is set to 1 (4-wire bus communication mode) and the MSS bit in the SSCRH register is set to 1 (operates as master device) and when starting a serial communication, the CE bit in the SSSR register is set to 1 if "L" applies to the $\overline{\text{SCS}}$ pin input. When the SSUMS bit in the SSMR2 register is set to 1 (4-wire bus communication mode), the MSS bit in the SSCRH register is set to 0 (operates as slave device) and the $\overline{\text{SCS}}$ pin input changes state from "L" to "H", the CE bit in the SSSR register is set to 1. |
| Interrupt requests | 5 interrupt requests (transmit-end, transmit-data-empty, receive-data-full, overrun error, and conflict error) ⁽¹⁾ . |
| Select functions | <ul style="list-style-type: none"> Data transfer direction Selects MSB-first or LSB-first SSCK clock polarity Selects "L" or "H" level when clock stops SSCK clock phase Selects edge of data change and data download |

NOTE:

1. Clock synchronous serial I/O with chip select has only one interrupt vector table.

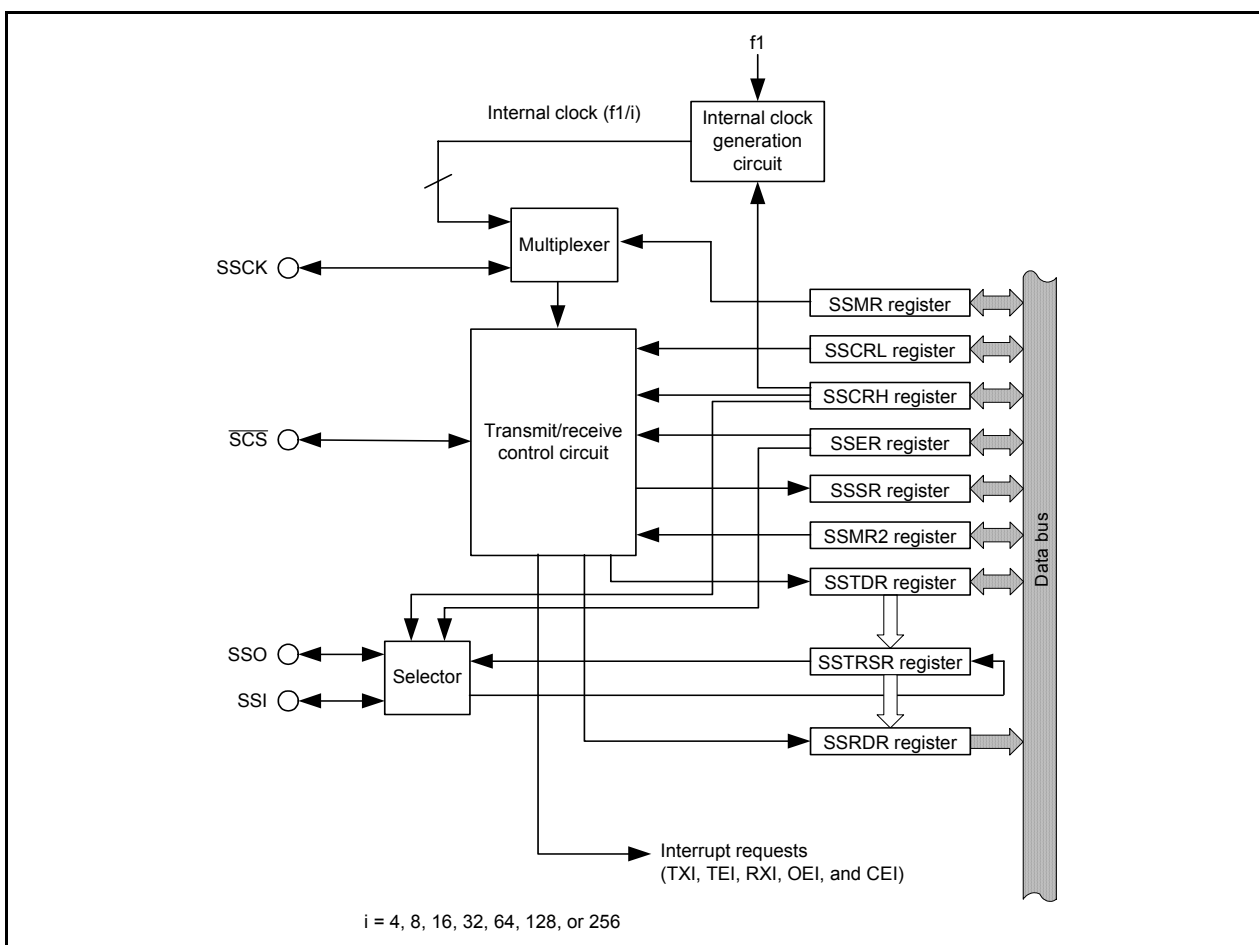


Figure 16.1 Block Diagram of Clock Synchronous Serial I/O with Chip Select

Module Operation Enable Register

| Bit | Symbol | Address | After Reset |
|---------|--------|--|--|
| b7 | — | 0008h | 00h |
| b6 | — | | |
| b5 | — | | |
| b4 | — | | |
| b3 | — | | |
| b2 | MSTIIC | SSU, \bar{P} C bus operation enable bit | 0: Disable ⁽¹⁾ 1: Enable |
| b1 | MSTTRD | Timer RD operation enable bit | 0: Disable ⁽²⁾ 1: Enable |
| b0 | MSTTRC | Timer RC operation enable bit | 0: Disable ⁽³⁾ 1: Enable |
| (b7-b6) | — | Nothing is assigned. If necessary, set to 0. When read, the content is 0. | — |

NOTES:

- When the MSTIIC bit is set to 0 (disable), any access to the SSU or the \bar{P} C bus associated registers (addresses 00B8h to 00BFh) is disabled.
- When the MSTTRD bit is set to 0 (disable), any access to the timer RD associated registers (addresses 0137h to 015Fh) is disabled.
- When the MSTTRC bit is set to 0 (disable), any access to the timer RC associated registers (addresses 0120h to 0132h) is disabled.

Figure 16.2 MSTCR Register

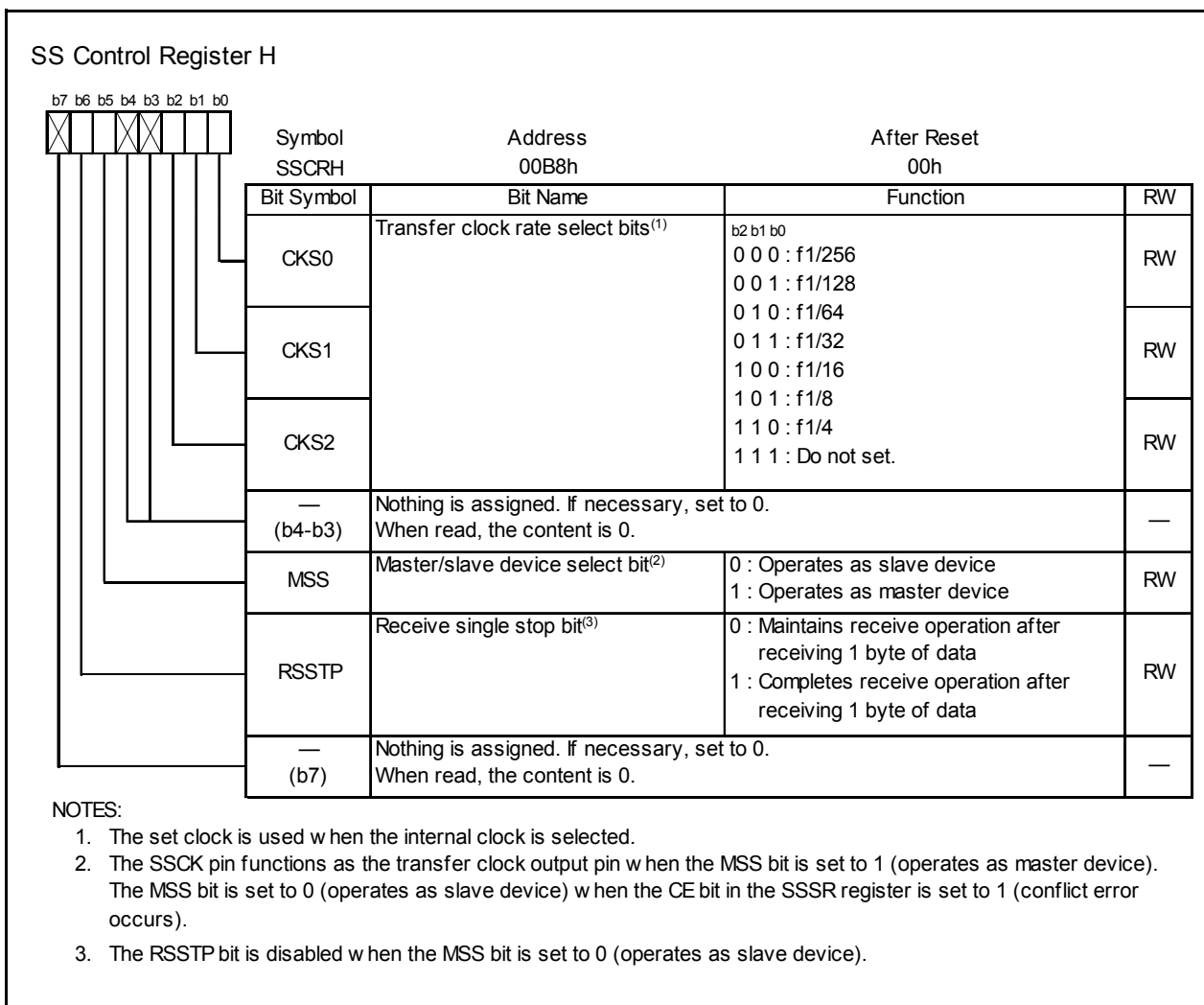


Figure 16.3 SSCRH Register

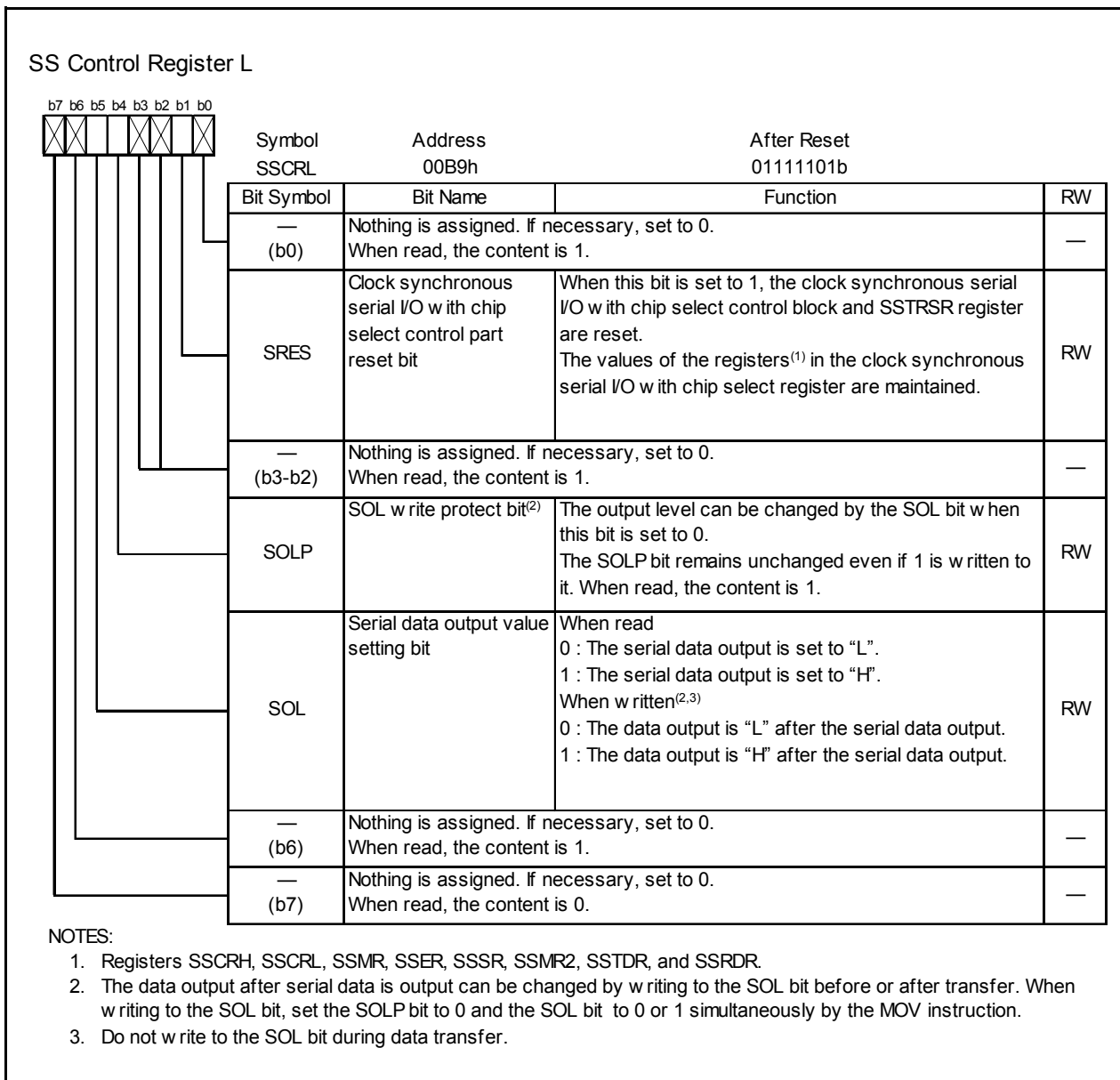


Figure 16.4 SSCRL Register

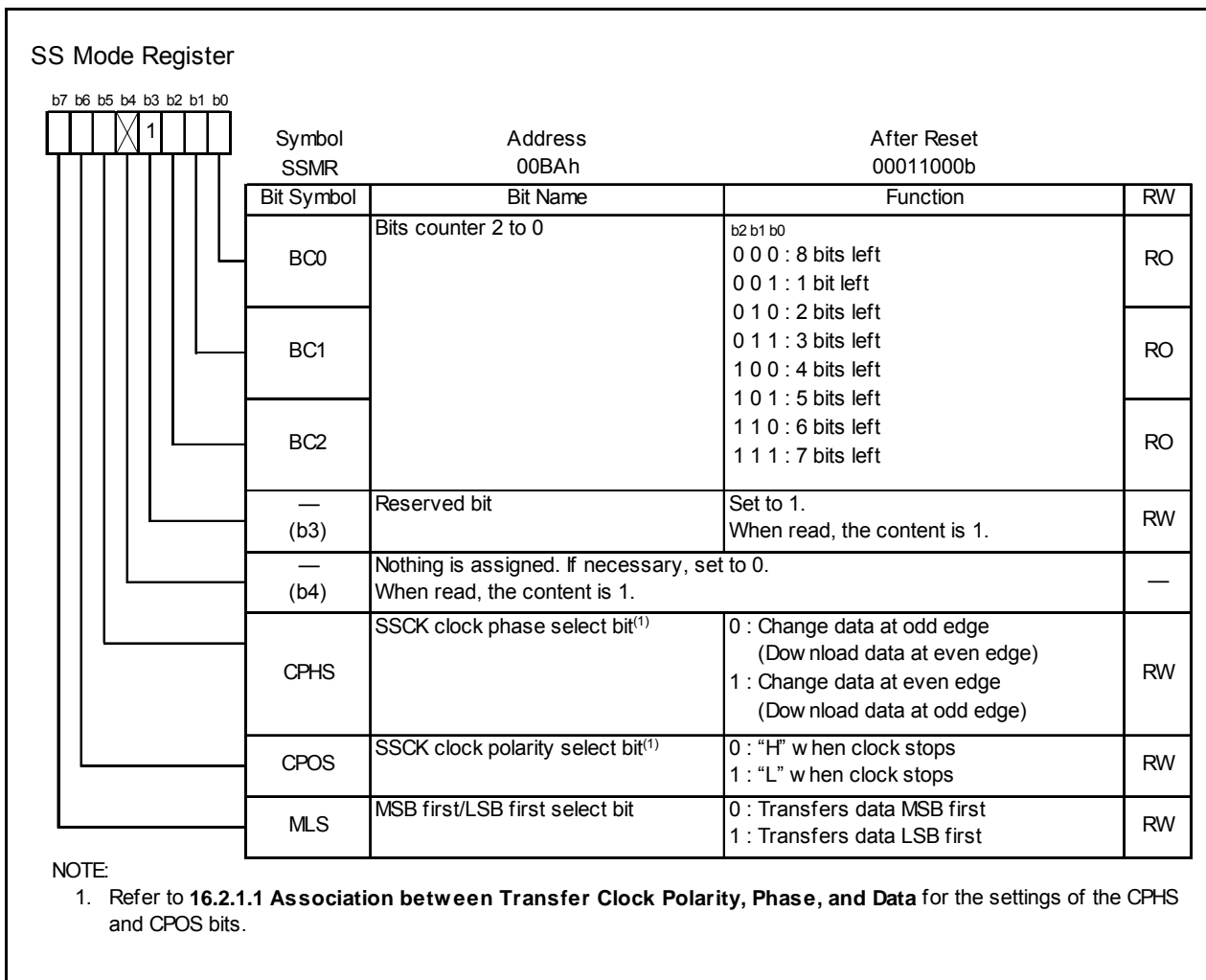


Figure 16.5 SSMR Register

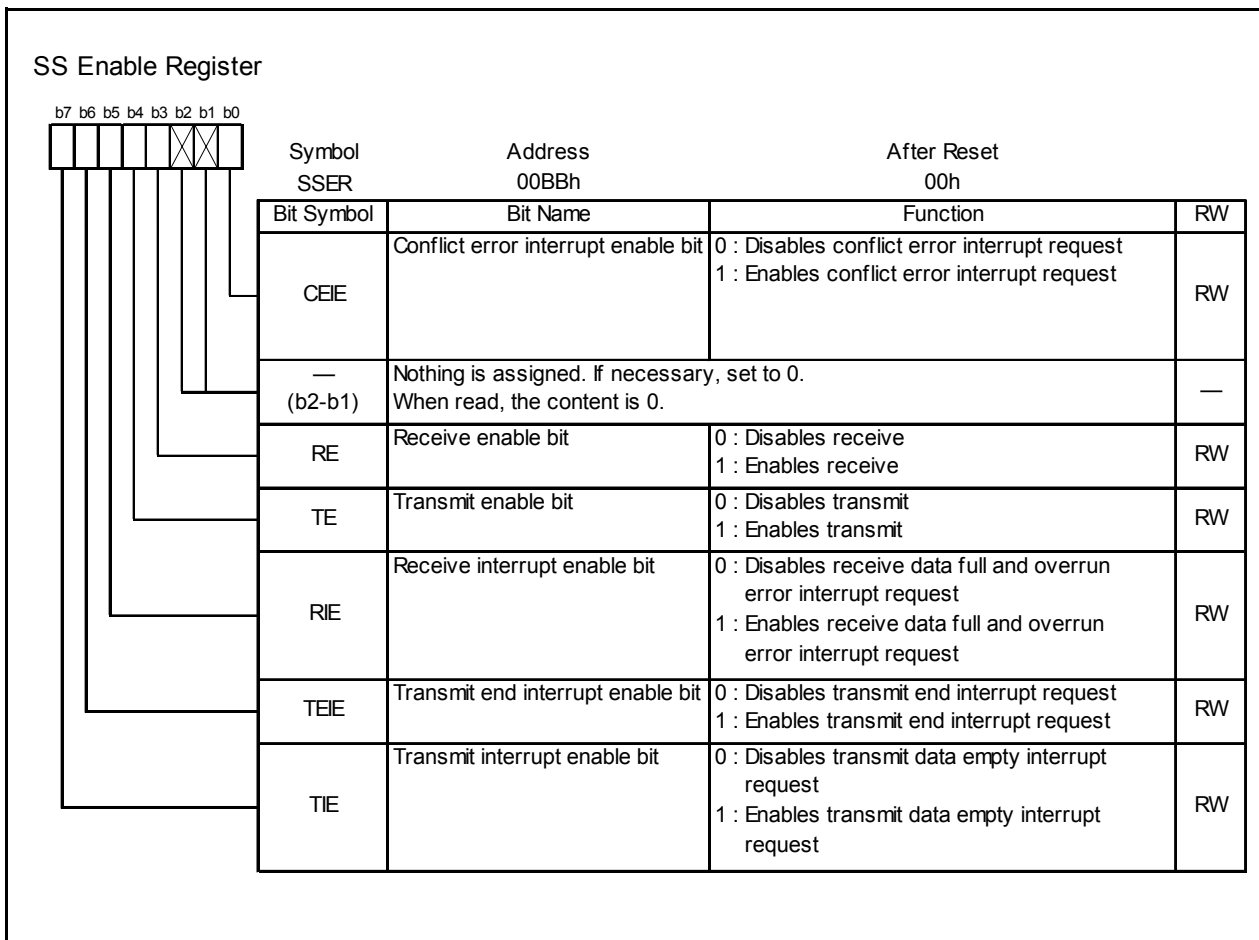


Figure 16.6 SSER Register

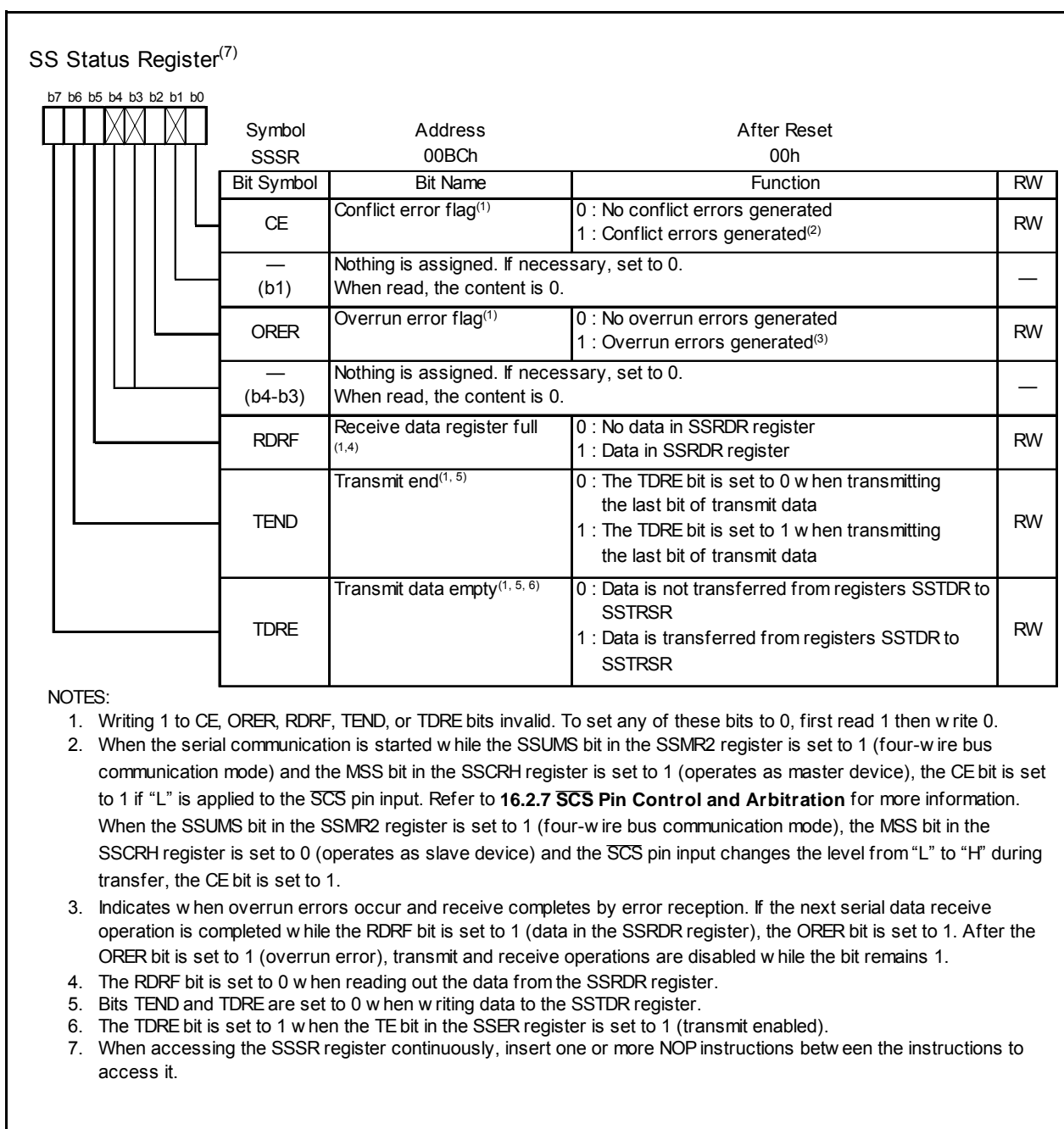


Figure 16.7 SSSR Register

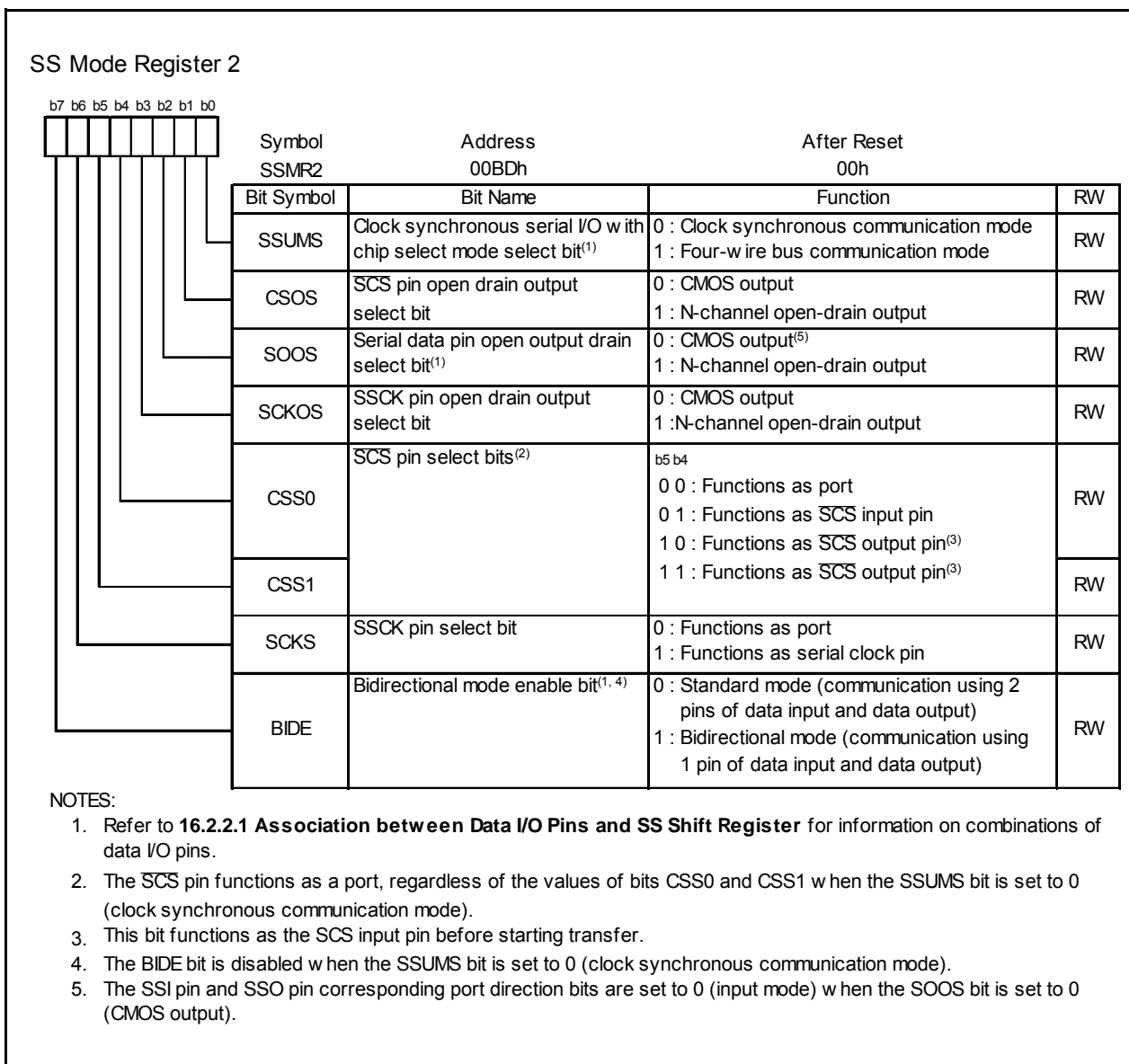


Figure 16.8 SSMR2 Register

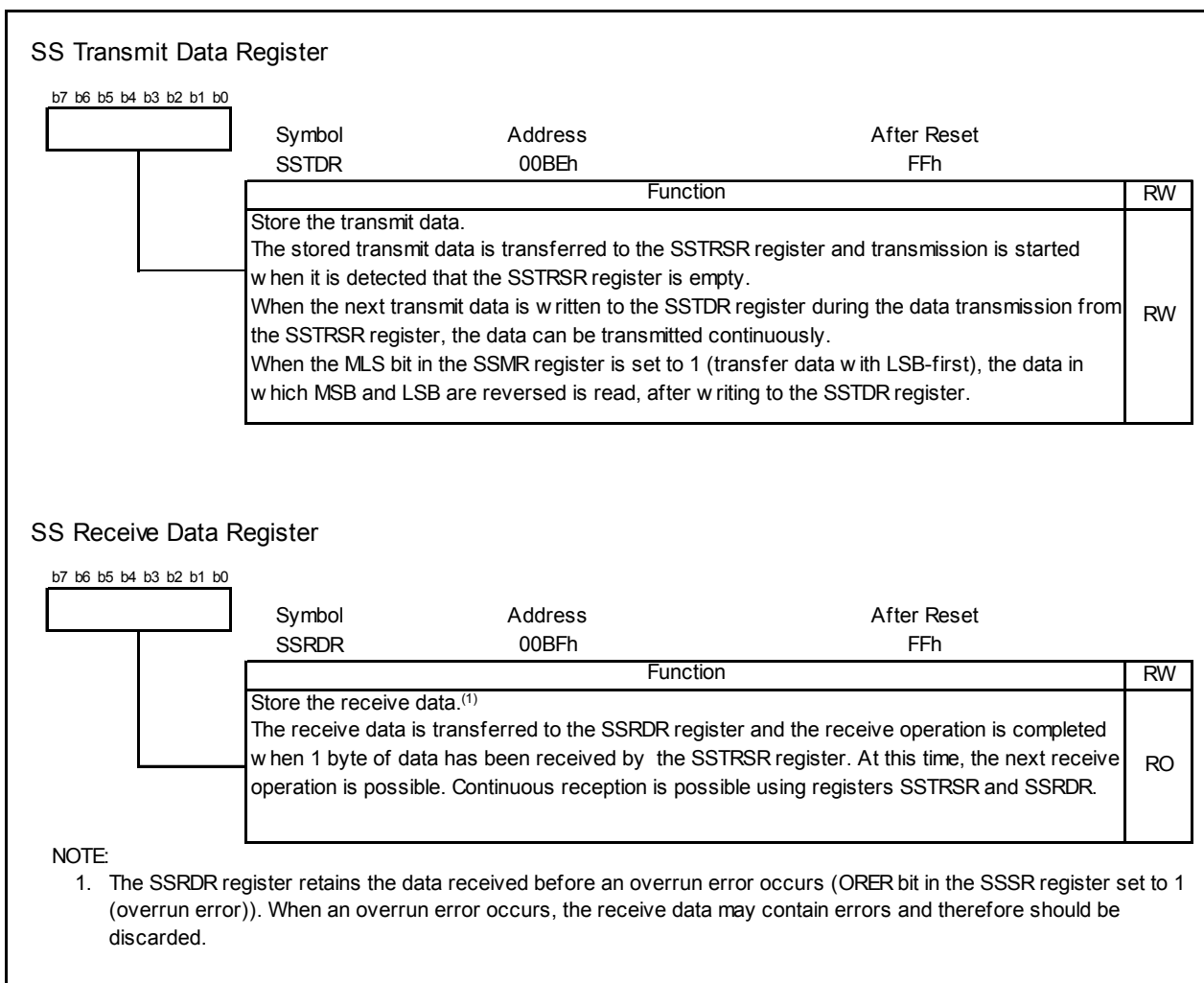


Figure 16.9 Registers SSTDR and SSRDR

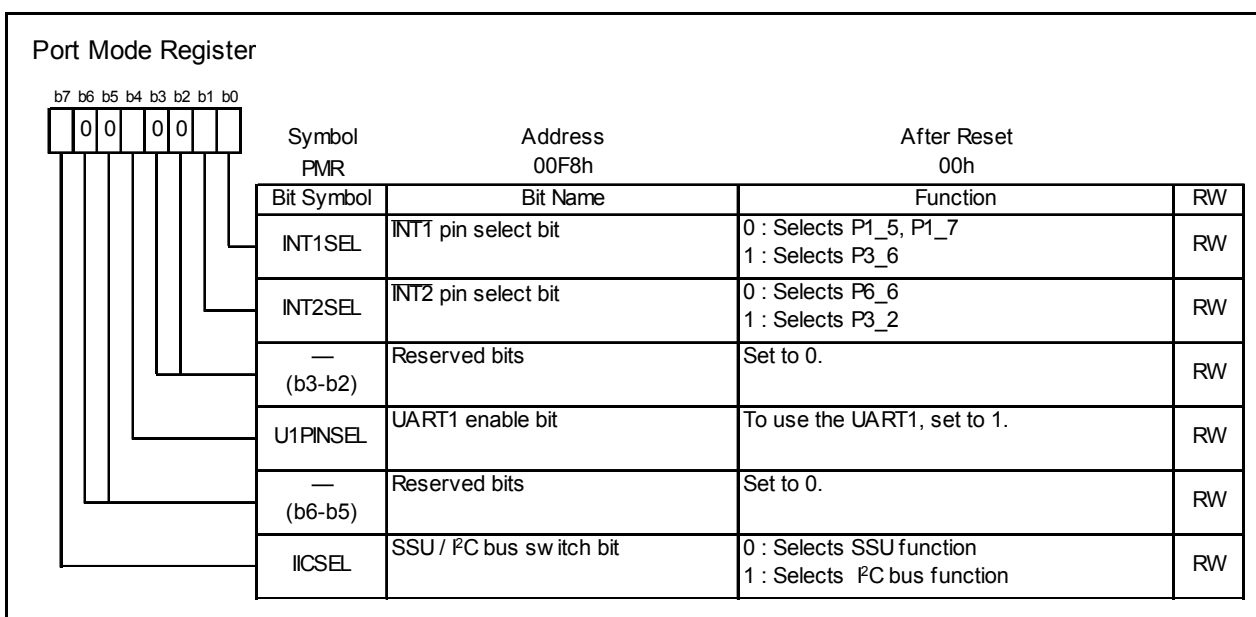


Figure 16.10 PMR Register

16.2.1 Transfer Clock

The transfer clock can be selected from among seven internal clocks (f1/256, f1/128, f1/64, f1/32, f1/16, f1/8, and f1/4) and an external clock.

When using clock synchronous serial I/O with chip select, set the SCKS bit in the SSMR2 register to 1 and select the SSCK pin as the serial clock pin.

When the MSS bit in the SSCRH register is set to 1 (operates as master device), an internal clock can be selected and the SSCK pin functions as output. When transfer is started, the SSCK pin outputs clocks of the transfer rate selected by bits CKS0 to CKS2 in the SSCRH register.

When the MSS bit in the SSCRH register is set to 0 (operates as slave device), an external clock can be selected and the SSCK pin functions as input.

16.2.1.1 Association between Transfer Clock Polarity, Phase, and Data

The association between the transfer clock polarity, phase and data changes according to the combination of the SSUMS bit in the SSMR2 register and bits CPHS and CPOS in the SSMR register.

Figure 16.11 shows the Association between Transfer Clock Polarity, Phase, and Transfer Data.

Also, the MSB-first transfer or LSB-first transfer can be selected by setting the MLS bit in the SSMR register. When the MLS bit is set to 1, transfer is started from the LSB and proceeds to the MSB. When the MLS bit is set to 0, transfer is started from the MSB and proceeds to the LSB.

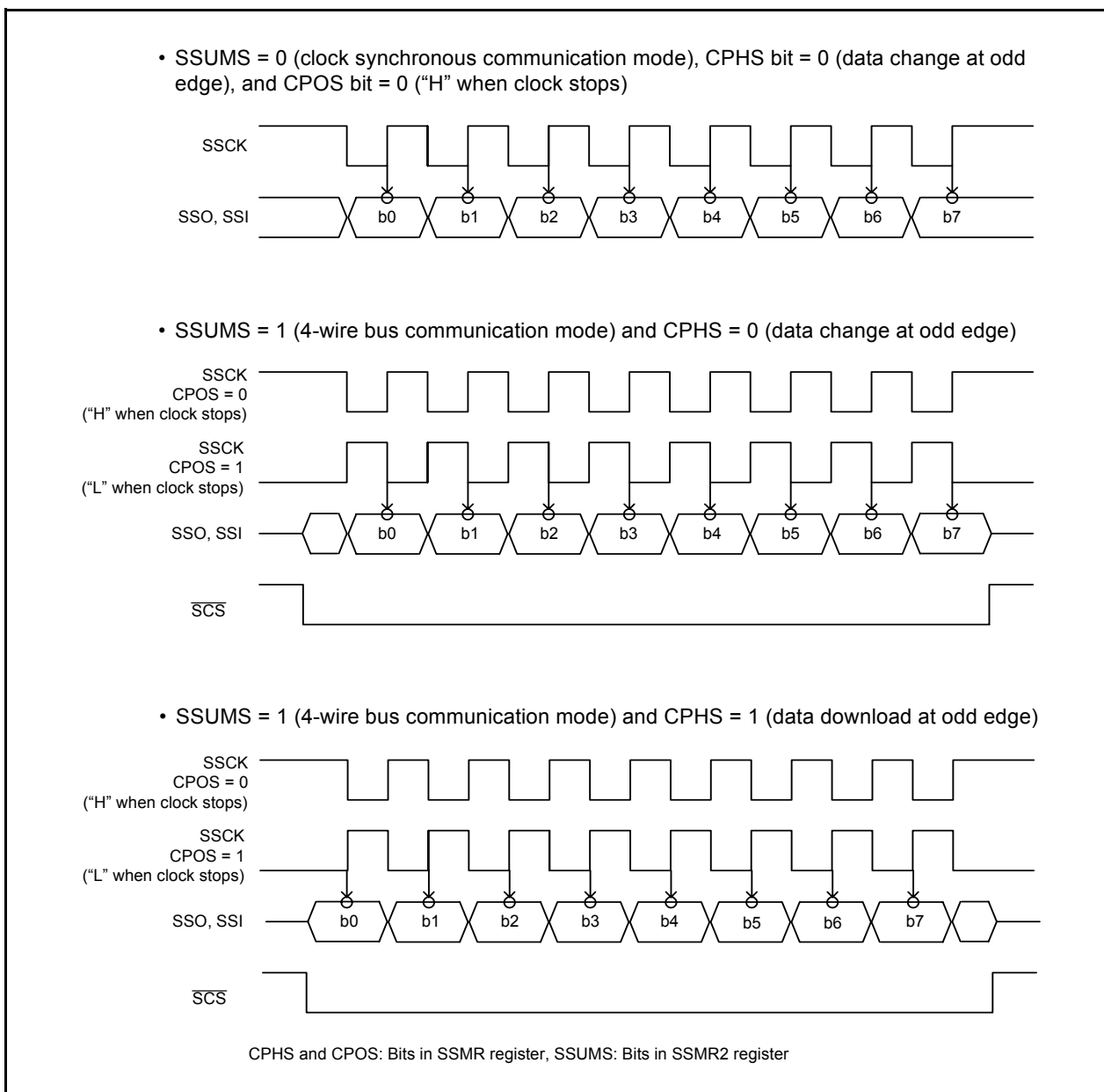


Figure 16.11 Association between Transfer Clock Polarity, Phase, and Transfer Data

16.2.2 SS Shift Register (SSTRSR)

The SSTRSR register is a shift register for transmitting and receiving serial data.

When transmit data is transferred from the SSTDR register to the SSTRSR register and the MLS bit in the SSMR register is set to 0 (MSB-first), the bit 0 in the SSTDR register is transferred to bit 0 in the SSTRSR register. When the MLS bit is set to 1 (LSB-first), bit 7 in the SSTDR register is transferred to bit 0 in the SSTRSR register.

16.2.2.1 Association between Data I/O Pins and SS Shift Register

The connection between the data I/O pins and SSTRSR register (SS shift register) changes according to a combination of the MSS bit in the SSCRH register and the SSUMS bit in the SSMR2 register. The connection also changes according to the BIDE bit in the SSMR2 register.

Figure 16.12 shows the Association between Data I/O Pins and SSTRSR Register.

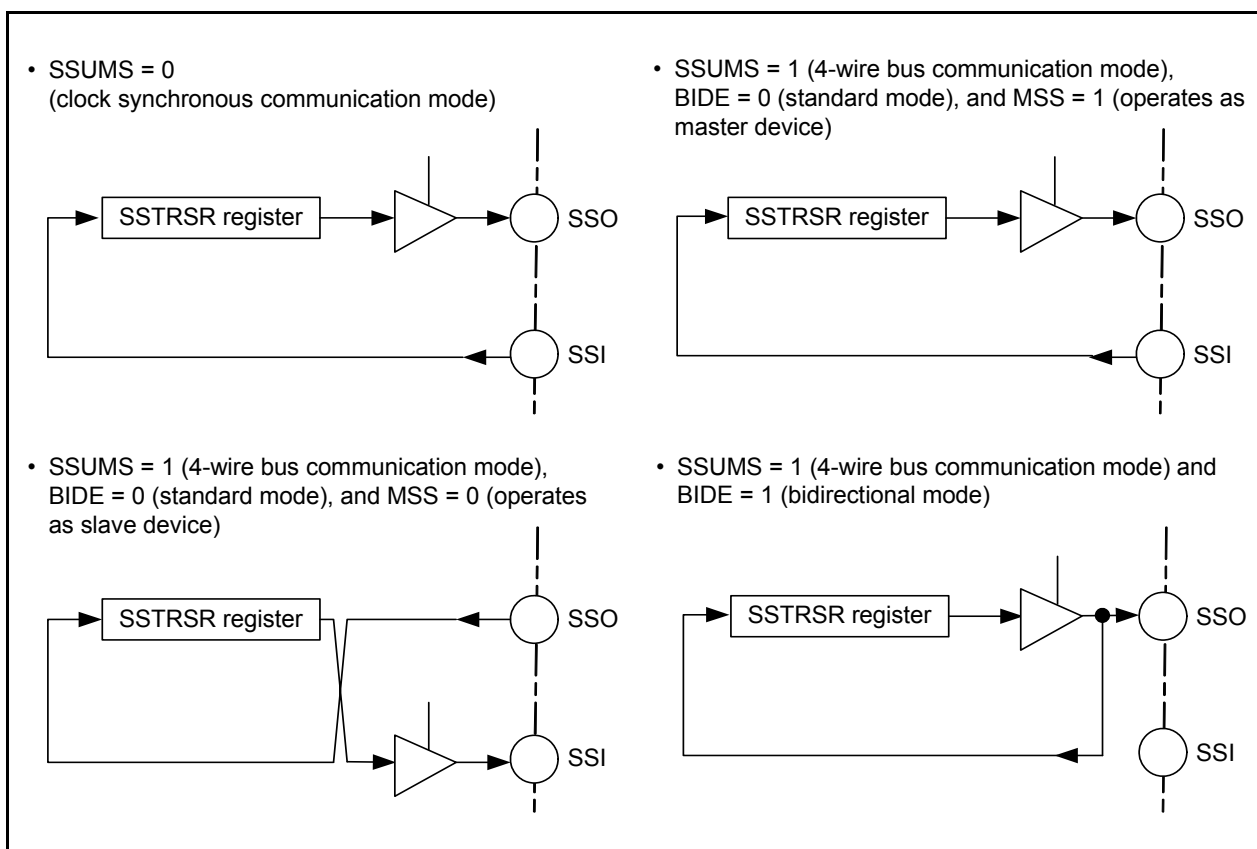


Figure 16.12 Association between Data I/O Pins and SSTRSR Register

16.2.3 Interrupt Requests

Clock synchronous serial I/O with chip select 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 clock synchronous serial I/O with chip select interrupt vector table, determining interrupt sources by flags is required. Table 16.3 shows the Clock Synchronous Serial I/O with Chip Select Interrupt Requests.

Table 16.3 Clock Synchronous Serial I/O with Chip Select Interrupt Requests

| Interrupt Request | Abbreviation | Generation Condition |
|---------------------|--------------|----------------------|
| Transmit data empty | TXI | TIE = 1, TDRE = 1 |
| Transmit end | TEI | TEIE = 1, TEND = 1 |
| Receive data full | RXI | RIE = 1, RDRF = 1 |
| Overrun error | OEI | RIE = 1, ORER = 1 |
| Conflict error | CEI | CEIE = 1, CE = 1 |

CEIE, RIE, TEIE and TIE: Bits in SSER register

ORER, RDRF, TEND and TDRE: Bits in SSSR register

If the generation conditions in Table 16.3 are met, a clock synchronous serial I/O with chip select interrupt request is generated. Set each interrupt source to 0 by a clock synchronous serial I/O with chip select interrupt routine.

However, the TDRE and TEND bits are automatically set to 0 by writing transmit data to the SSTDR register and the RDRF bit is automatically set to 0 by reading the SSRDR register. In particular, the TDRE bit is set to 1 (data transmitted from registers SSTDR to SSTRSR) at the same time transmit data is written to the SSTDR register. Setting the TDRE bit to 0 (data not transmitted from registers SSTDR to SSTRSR) can cause an additional byte of data to be transmitted.

16.2.4 Communication Modes and Pin Functions

Clock synchronous serial I/O with chip select switches the functions of the I/O pins in each communication mode according to the setting of the MSS bit in the SSCRH register and bits RE and TE in the SSER register. Table 16.4 shows the Association between Communication Modes and I/O Pins.

Table 16.4 Association between Communication Modes and I/O Pins

| Communication Mode | Bit Setting | | | | | Pin State | | |
|--|-------------|----------|-----|----|----|-----------|--------|--------|
| | SSUMS | BIDE | MSS | TE | RE | SSI | SSO | SSCK |
| Clock synchronous communication mode | 0 | Disabled | 0 | 0 | 1 | Input | –(1) | Input |
| | | | | 1 | 0 | –(1) | Output | Input |
| | | | | 1 | 1 | Input | Output | Input |
| | | | 1 | 0 | 1 | Input | –(1) | Output |
| | | | | 1 | 0 | –(1) | Output | Output |
| | | | | 1 | 1 | Input | Output | Output |
| 4-wire bus communication mode | 1 | 0 | 0 | 0 | 1 | –(1) | Input | Input |
| | | | | 1 | 0 | Output | –(1) | Input |
| | | | | 1 | 1 | Output | Input | Input |
| | | | 1 | 0 | 1 | Input | –(1) | Output |
| | | | | 1 | 0 | –(1) | Output | Output |
| | | | | 1 | 1 | Input | Output | Output |
| 4-wire bus (bidirectional) communication mode(2) | 1 | 1 | 0 | 0 | 1 | –(1) | Input | Input |
| | | | | 1 | 0 | –(1) | Output | Input |
| | | | 1 | 0 | 1 | –(1) | Input | Output |
| | | | | 1 | 0 | –(1) | Output | Output |

NOTES:

1. This pin can be used as a programmable I/O port.
2. Do not set both bits TE and RE to 1 in 4-wire bus (bidirectional) communication mode.

SSUMS and BIDE: Bits in SSMR2 register

MSS: Bit in SSCRH register

TE and RE: Bits in SSER register

16.2.5 Clock Synchronous Communication Mode

16.2.5.1 Initialization in Clock Synchronous Communication Mode

Figure 16.13 shows Initialization in Clock Synchronous Communication Mode. To initialize, set the TE bit in the SSER register to 0 (transmit disabled) and the RE bit to 0 (receive disabled) before data transmission or reception.

Set the TE bit to 0 and the RE bit to 0 before changing the communication mode or format.

Setting the RE bit to 0 does not change the contents of flags RDRF and ORER or the contents of the SSRDR register.

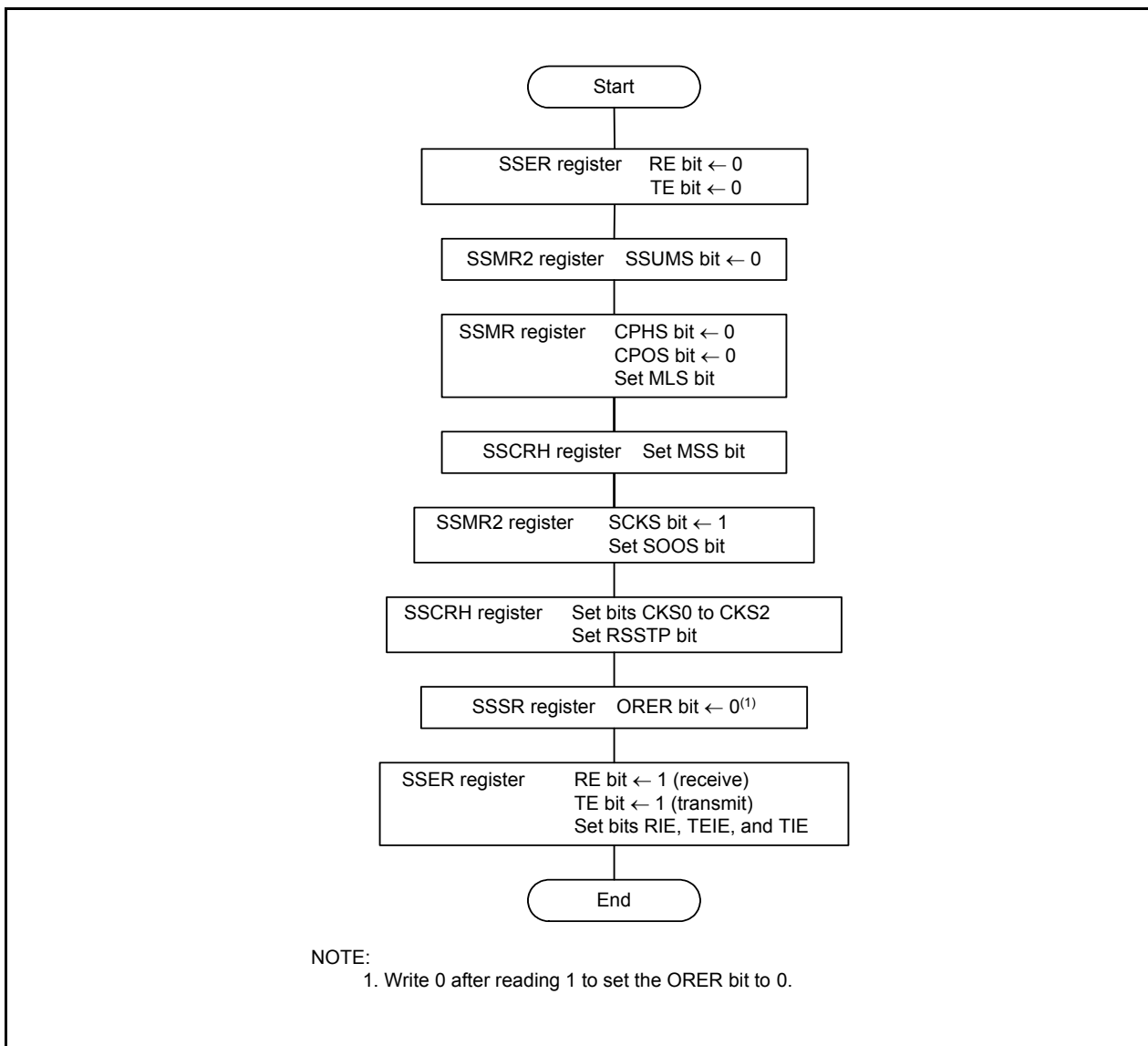


Figure 16.13 Initialization in Clock Synchronous Communication Mode

16.2.5.2 Data Transmission

Figure 16.14 shows an Example of Clock Synchronous Serial I/O with Chip Select Operation for Data Transmission (Clock Synchronous Communication Mode). During data transmission, the clock synchronous serial I/O with chip select operates as described below.

When clock synchronous serial I/O with chip select is set as a master device, it outputs a synchronous clock and data. When clock synchronous serial I/O with chip select is set as a slave device, it outputs data synchronized with the input clock.

When the TE bit is set to 1 (transmit enabled) before writing the transmit data to the SSTDR register, the TDRE bit is automatically set to 0 (data not transferred from registers SSTDR to SSTRSR) and the data is transferred from registers SSTDR to SSTRSR.

After the TDRE bit is set to 1 (data transferred from registers SSTDR to SSTRSR), transmission starts. When the TIE bit in the SSER register is set to 1, the TXI interrupt request is generated. When one frame of data is transferred while the TDRE bit is set to 0, data is transferred from registers SSTDR to SSTRSR and transmission of the next frame is started. If the 8th bit is transmitted while the TDRE bit is set to 1, the TEND bit in the SSSR register is set to 1 (the TDRE bit is set to 1 when the last bit of the transmit data is transmitted) and the state is retained. The TEI interrupt request is generated when the TEIE bit in the SSER register is set to 1 (transmit-end interrupt request enabled). The SSCK pin is fixed "H" after transmit-end.

Transmission cannot be performed while the ORER bit in the SSSR register is set to 1 (overrun error). Confirm that the ORER bit is set to 0 before transmission.

Figure 16.15 shows a Sample Flowchart of Data Transmission (Clock Synchronous Communication Mode).

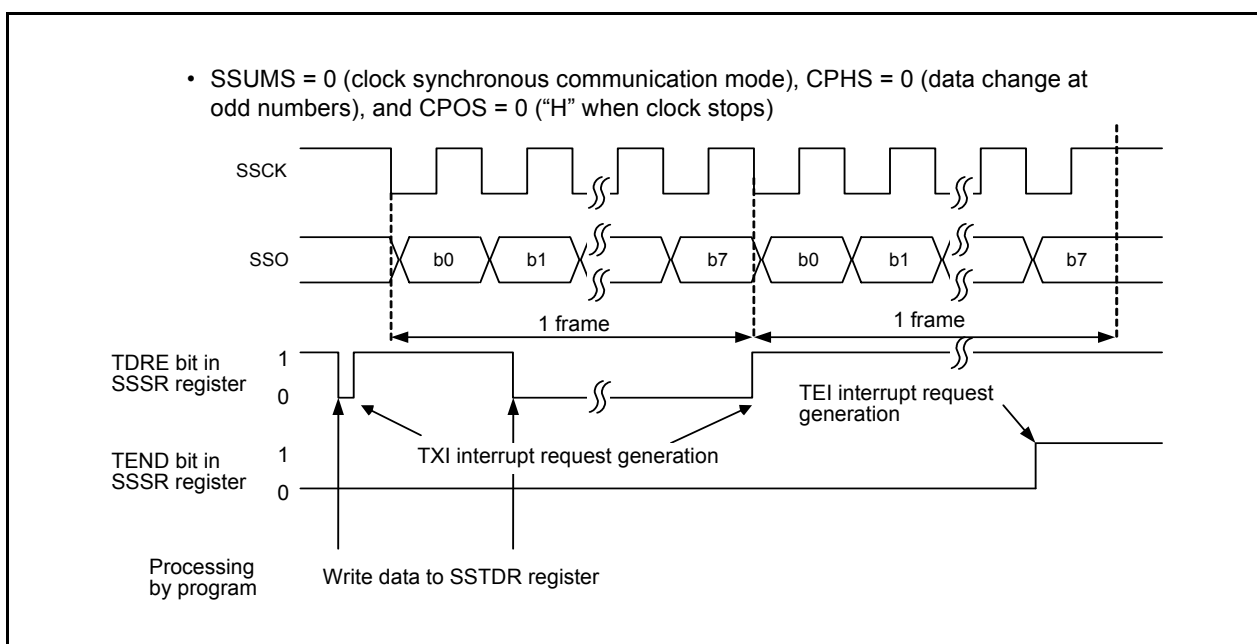


Figure 16.14 Example of Clock Synchronous Serial I/O with Chip Select Operation for Data Transmission (Clock Synchronous Communication Mode)

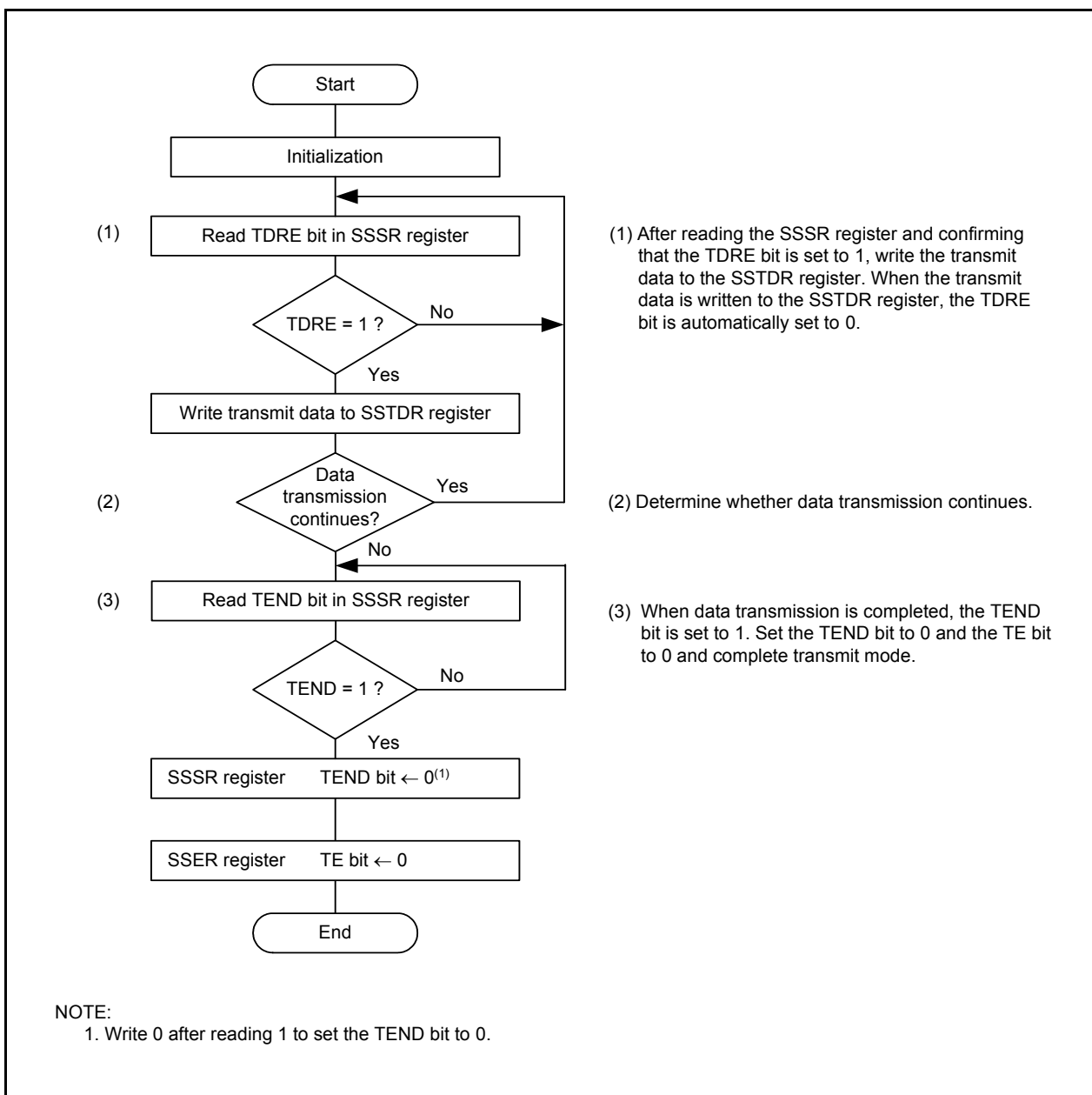


Figure 16.15 Sample Flowchart of Data Transmission (Clock Synchronous Communication Mode)

16.2.5.3 Data Reception

Figure 16.16 shows an Example of Clock Synchronous Serial I/O with Chip Select Operation for Data Reception (Clock Synchronous Communication Mode).

During data reception, clock synchronous serial I/O with chip select operates as described below. When the clock synchronous serial I/O with chip select is set as the master device, it outputs a synchronous clock and inputs data. When clock synchronous serial I/O with chip select is set as a slave device, it inputs data synchronized with the input clock.

When clock synchronous serial I/O with chip select is set as a master device, it outputs a receive clock and starts receiving by performing dummy read of the SSRDR register.

After 8 bits of data are received, the RDRF bit in the SSSR register is set to 1 (data in the SSRDR register) and receive data is stored in the SSRDR register. When the RIE bit in the SSER register is set to 1 (RXI and OEI interrupt requests enabled), the RXI interrupt request is generated. If the SSRDR register is read, the RDRF bit is automatically set to 0 (no data in the SSRDR register).

Read the receive data after setting the RSSTP bit in the SSCRH register to 1 (after receiving 1 byte of data, the receive operation is completed). Clock synchronous serial I/O with chip select outputs a clock for receiving 8 bits of data and stops. After that, set the RE bit in the SSER register to 0 (receive disabled) and the RSSTP bit to 0 (receive operation is continued after receiving the 1 byte of data) and read the receive data. If the SSRDR register is read while the RE bit is set to 1 (receive enabled), a receive clock is output again.

When the 8th clock rises while the RDRF bit is set to 1, the ORER bit in the SSSR register is set to 1 (overrun error: OEI) and the operation is stopped. When the ORER bit is set to 1, receive cannot be performed. Confirm that the ORER bit is set to 0 before restarting receive.

Figure 16.17 shows a Sample Flowchart of Data Reception (MSS = 1) (Clock Synchronous Communication Mode).

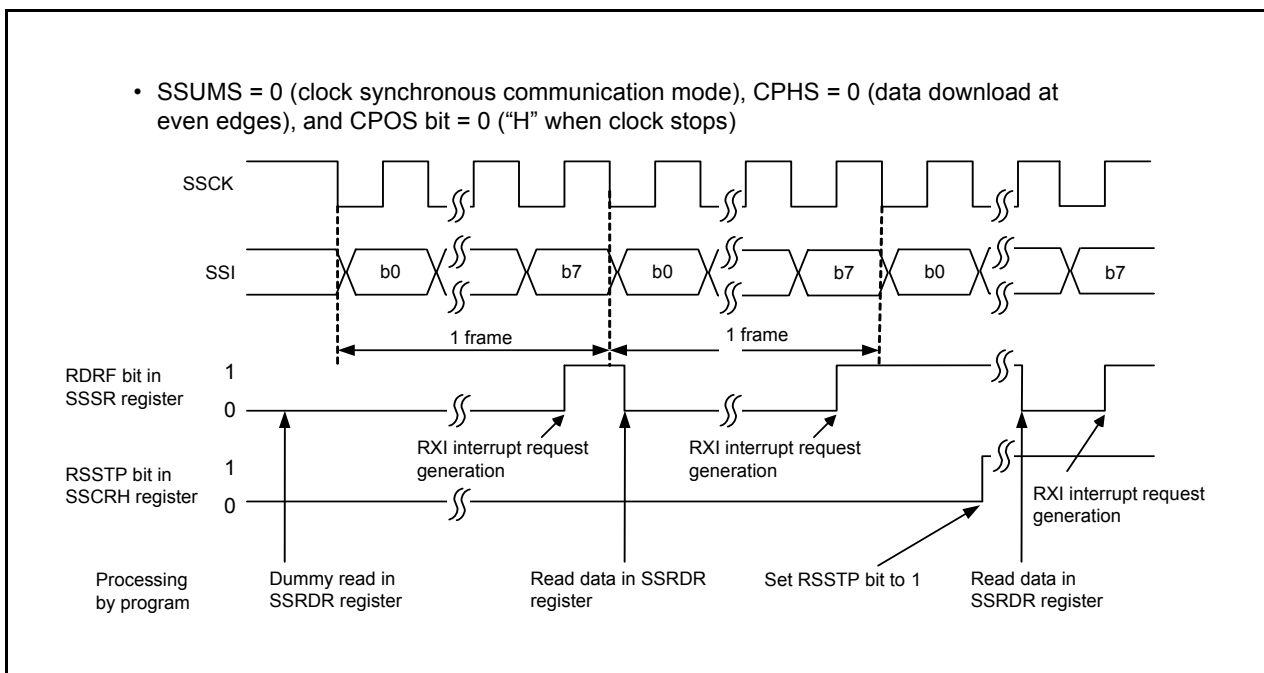


Figure 16.16 Example of Clock Synchronous Serial I/O with Chip Select Operation for Data Reception (Clock Synchronous Communication Mode)

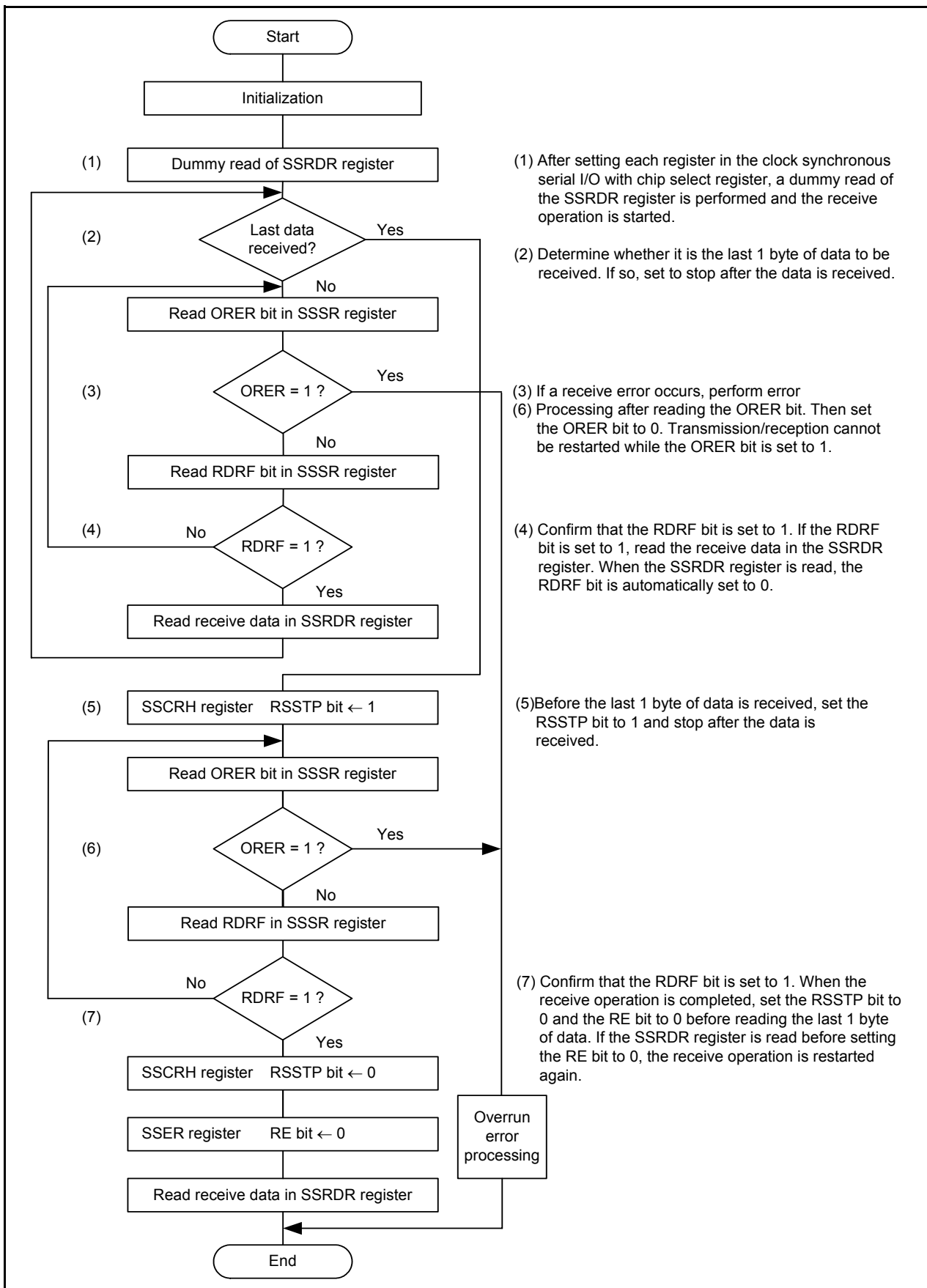


Figure 16.17 Sample Flowchart of Data Reception (MSS = 1) (Clock Synchronous Communication Mode)

16.2.5.4 Data Transmission/Reception

Data transmission/reception is an operation combining data transmission and reception which were described earlier. Transmission/reception is started by writing data to the SSTDR register.

When the 8th clock rises or the ORER bit is set to 1 (overrun error) while the TDRE bit is set to 1 (data is transferred from registers SSTDR to SSTRSR), the transmit/receive operation is stopped.

When switching from transmit mode (TE = 1) or receive mode (RE = 1) to transmit/receive mode (Te = RE = 1), set the TE bit to 0 and RE bit to 0 before switching. After confirming that the TEND bit is set to 0 (the TDRE bit is set to 0 when the last bit of the transmit data is transmitted), the RDRF bit is set to 0 (no data in the SSRDR register), and the ORER bit is set to 0 (no overrun error), set bits TE and RE to 1.

Figure 16.18 shows a Sample Flowchart of Data Transmission/Reception (Clock Synchronous Communication Mode).

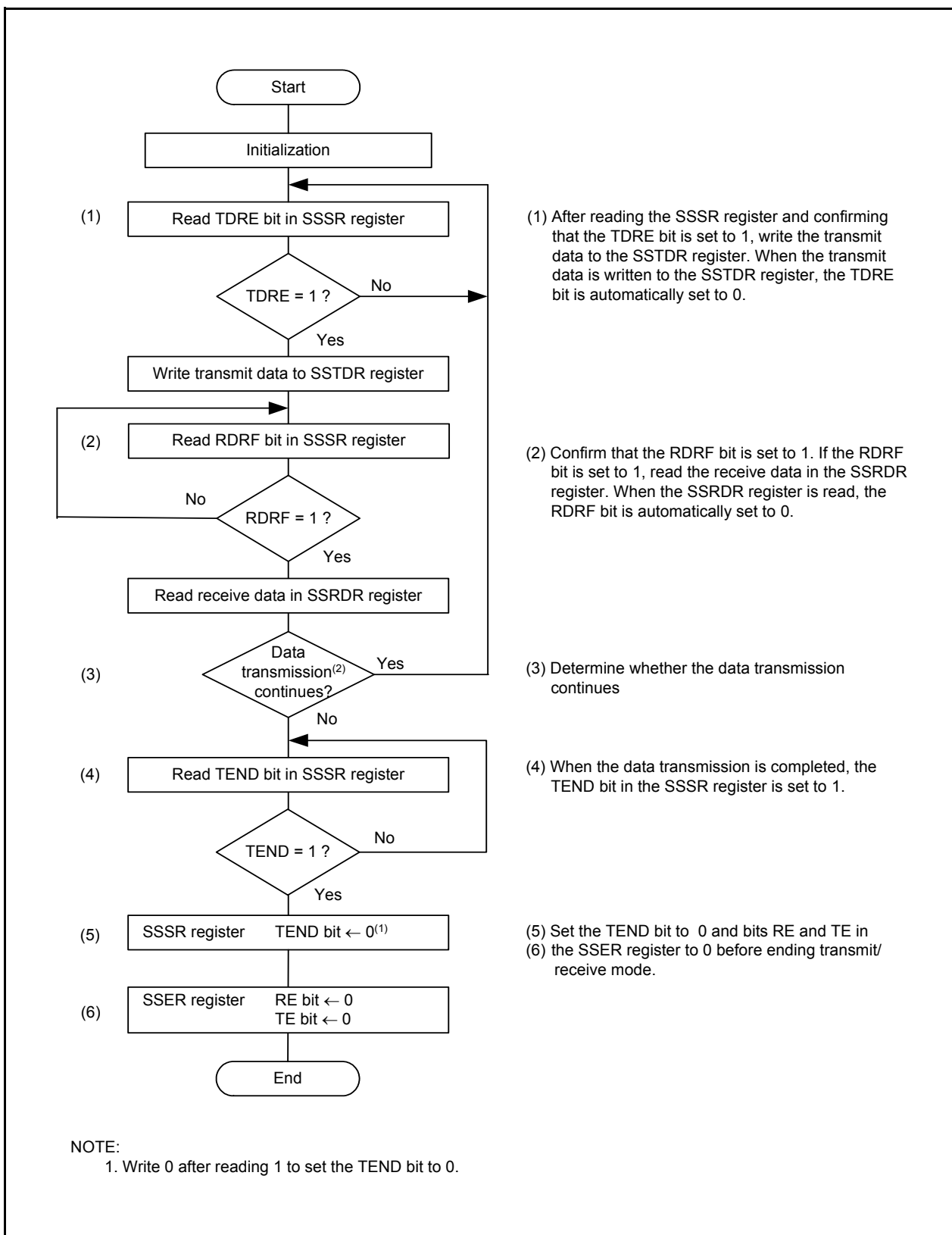


Figure 16.18 Sample Flowchart of Data Transmission/Reception (Clock Synchronous Communication Mode)

16.2.6 Operation in 4-Wire Bus Communication Mode

In 4-wire bus communication mode, a 4-wire bus consisting of a clock line, a data input line, a data output line, and a chip select line is used for communication. This mode includes bidirectional mode in which the data input line and data output line function as a single pin.

The data input line and output line change according to the settings of the MSS bit in the SSCRH register and the BIDE bit in the SSMR2 register. For details, refer to **16.2.2.1 Association between Data I/O Pins and SS Shift Register**. In this mode, clock polarity, phase, and data settings are performed by bits CPOS and CPHS in the SSMR register. For details, refer to **16.2.1.1 Association between Transfer Clock Polarity, Phase, and Data**.

When this MCU is set as the master device, the chip select line controls output. When clock synchronous serial I/O with chip select is set as a slave device, the $\overline{\text{chip}}$ select line controls input. When it is set as the master device, the chip select line controls output of the $\overline{\text{SCS}}$ pin or controls output of a general port according to the setting of the CSS1 bit in the SSMR2 register. When the MCU is set as a slave device, the chip select line sets the $\overline{\text{SCS}}$ pin as an input pin by setting bits CSS1 and CSS0 in the SSMR2 register to 01b.

In 4-wire bus communication mode, the MLS bit in the SSMR register is set to 0 and communication is performed MSB-first.

16.2.6.1 Initialization in 4-Wire Bus Communication Mode

Figure 16.19 shows Initialization in 4-Wire Bus Communication Mode. Before the data transit/receive operation, set the TE bit in the SSER register to 0 (transmit disabled), the RE bit in the SSER register to 0 (receive disabled), and initialize the clock synchronous serial I/O with chip select.

To change the communication mode or format, set the TE bit to 0 and the RE bit to 0 before making the change. Setting the RE bit to 0 does not change the settings of flags RDRF and ORER or the contents of the SSRDR register.

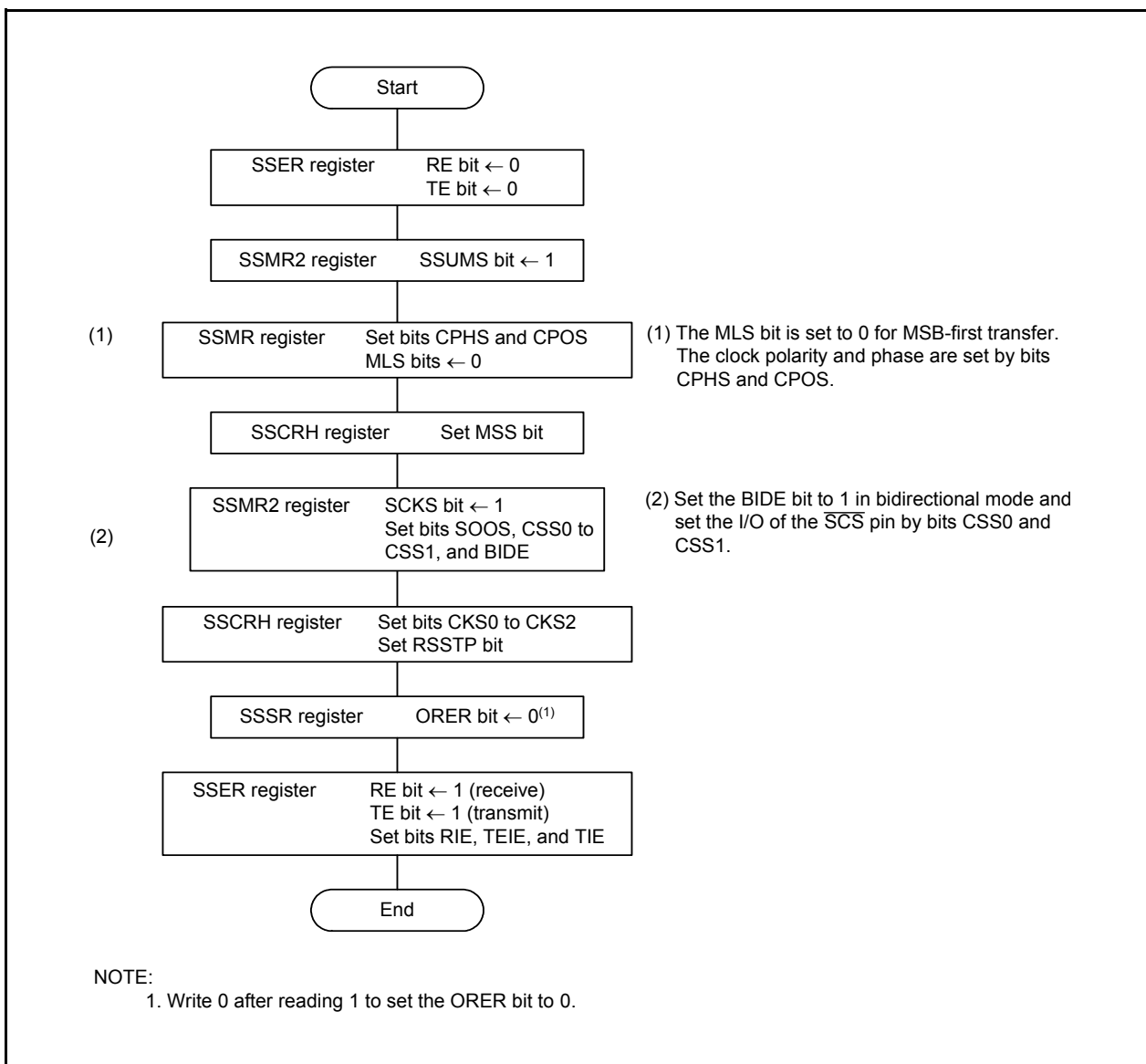


Figure 16.19 Initialization in 4-Wire Bus Communication Mode

16.2.6.2 Data Transmission

Figure 16.20 shows an Example of Clock Synchronous Serial I/O with Chip Select Operation during Data Transmission (4-Wire Bus Communication Mode). During the data transmit operation, clock synchronous serial I/O with chip select operates as described below.

When the MCU is set as the master device, it outputs a synchronous clock and data. When the MCU is set as a slave device, it outputs data in synchronization with the input clock while the \overline{SCS} pin is "L".

When the transmit data is written to the SSTDR register after setting the TE bit to 1 (transmit enabled), the TDRE bit is automatically set to 0 (data has not been transferred from registers SSTDR to SSTRSR) and the data is transferred from registers SSTDR to SSTRSR. After the TDRE bit is set to 1 (data is transferred from registers SSTDR to SSTRSR), transmission starts. When the TIE bit in the SSER register is set to 1, a TXI interrupt request is generated.

After 1 frame of data is transferred while the TDRE bit is set to 0, the data is transferred from registers SSTDR to SSTRSR and transmission of the next frame is started. If the 8th bit is transmitted while TDRE is set to 1, TEND in the SSSR register is set to 1 (when the last bit of the transmit data is transmitted, the TDRE bit is set to 1) and the state is retained. If the TEIE bit in the SSER register is set to 1 (transmit-end interrupt requests enabled), a TEI interrupt request is generated. The SSCK pin remains "H" after transmit-end and the \overline{SCS} pin is held "H". When transmitting continuously while the \overline{SCS} pin is held "L", write the next transmit data to the SSTDR register before transmitting the 8th bit.

Transmission cannot be performed while the ORER bit in the SSSR register is set to 1 (overrun error). Confirm that the ORER bit is set to 0 before transmission.

In contrast to the clock synchronous communication mode, the SSO pin is placed in high-impedance state while the \overline{SCS} pin is placed in high-impedance state when operating as a master device and the SSI pin is placed in high-impedance state while the \overline{SCS} pin is placed in "H" input state when operating as a slave device.

The sample flowchart is the same as that for the clock synchronous communication mode (refer to **Figure 16.15 Sample Flowchart of Data Transmission (Clock Synchronous Communication Mode)**).

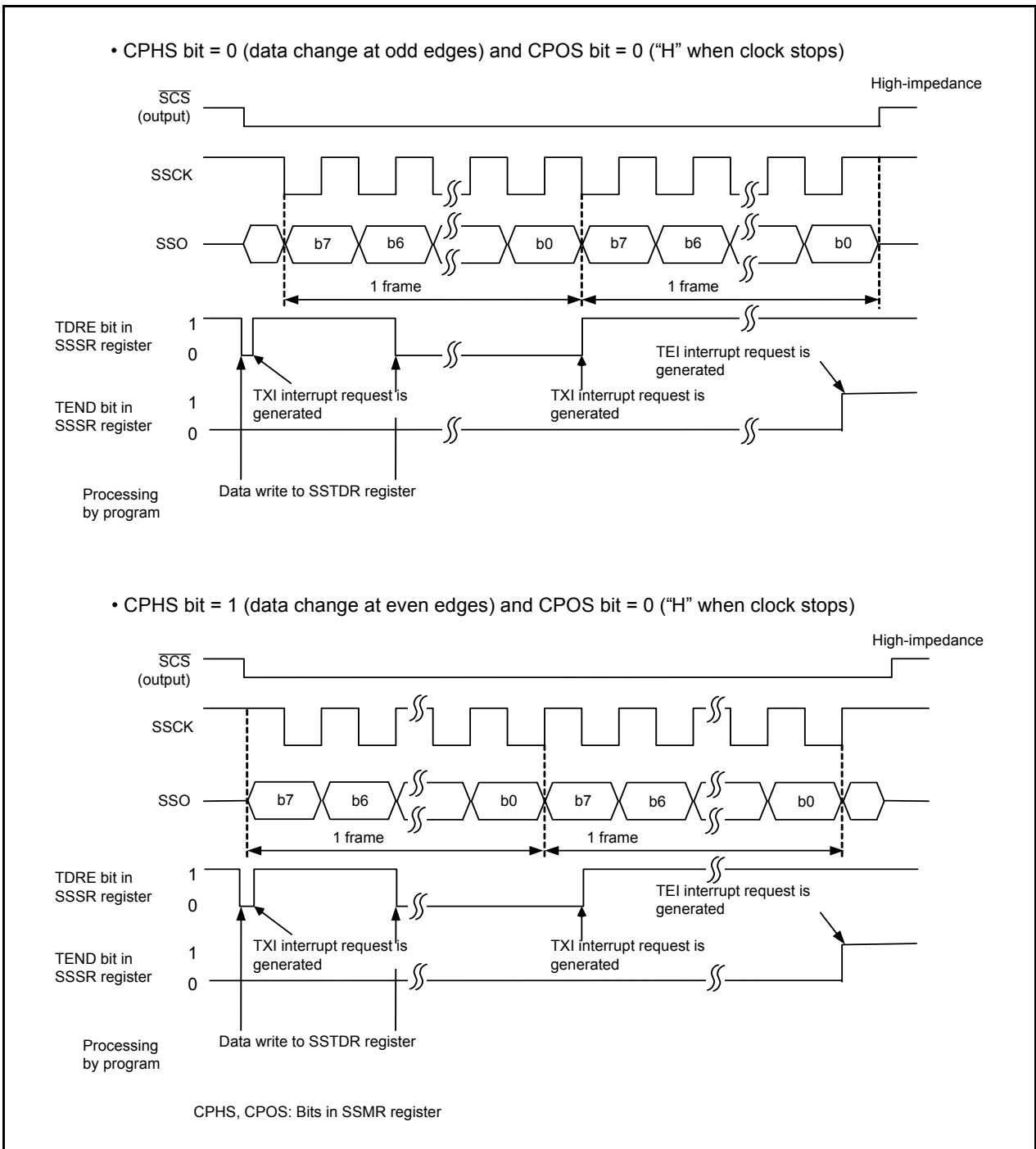


Figure 16.20 Example of Clock Synchronous Serial I/O with Chip Select Operation during Data Transmission (4-Wire Bus Communication Mode)

16.2.6.3 Data Reception

Figure 16.21 shows an Example of Clock Synchronous Serial I/O with Chip Select Operation during Data Reception (4-Wire Bus Communication Mode). During data reception, clock synchronous serial I/O with chip select operates as described below.

When the MCU is set as the master device, it outputs a synchronous clock and inputs data. When the MCU is set as a slave device, it outputs data synchronized with the input clock while the \overline{SCS} pin receives “L” input. When the MCU is set as the master device, it outputs a receive clock and starts receiving by performing a dummy read of the SSRDR register.

After 8 bits of data are received, the RDRF bit in the SSSR register is set to 1 (data in the SSRDR register) and receive data is stored in the SSRDR register. When the RIE bit in the SSER register is set to 1 (RXI and OEI interrupt requests enabled), an RXI interrupt request is generated. When the SSRDR register is read, the RDRF bit is automatically set to 0 (no data in the SSRDR register).

Read the receive data after setting the RSSTP bit in the SSCRH register to 1 (after receiving 1-byte data, the receive operation is completed). Clock synchronous serial I/O with chip select outputs a clock for receiving 8 bits of data and stops. After that, set the RE bit in the SSER register to 0 (receive disabled) and the RSSTP bit to 0 (receive operation is continued after receiving 1-byte data) and read the receive data. When the SSRDR register is read while the RE bit is set to 1 (receive enabled), a receive clock is output again.

When the 8th clock rises while the RDRF bit is set to 1, the ORER bit in the SSSR register is set to 1 (overrun error: OEI) and the operation is stopped. When the ORER bit is set to 1, reception cannot be performed. Confirm that the ORER bit is set to 0 before restarting reception.

The timing with which bits RDRF and ORER are set to 1 varies depending on the setting of the CPHS bit in the SSMR register. Figure 16.21 shows when bits RDRF and ORER are set to 1.

When the CPHS bit is set to 1 (data download at the odd edges), bits RDRF and ORER are set to 1 at some point during the frame.

The sample flowchart is the same as that for the clock synchronous communication mode (refer to **Figure 16.17 Sample Flowchart of Data Reception (MSS = 1) (Clock Synchronous Communication Mode)**).

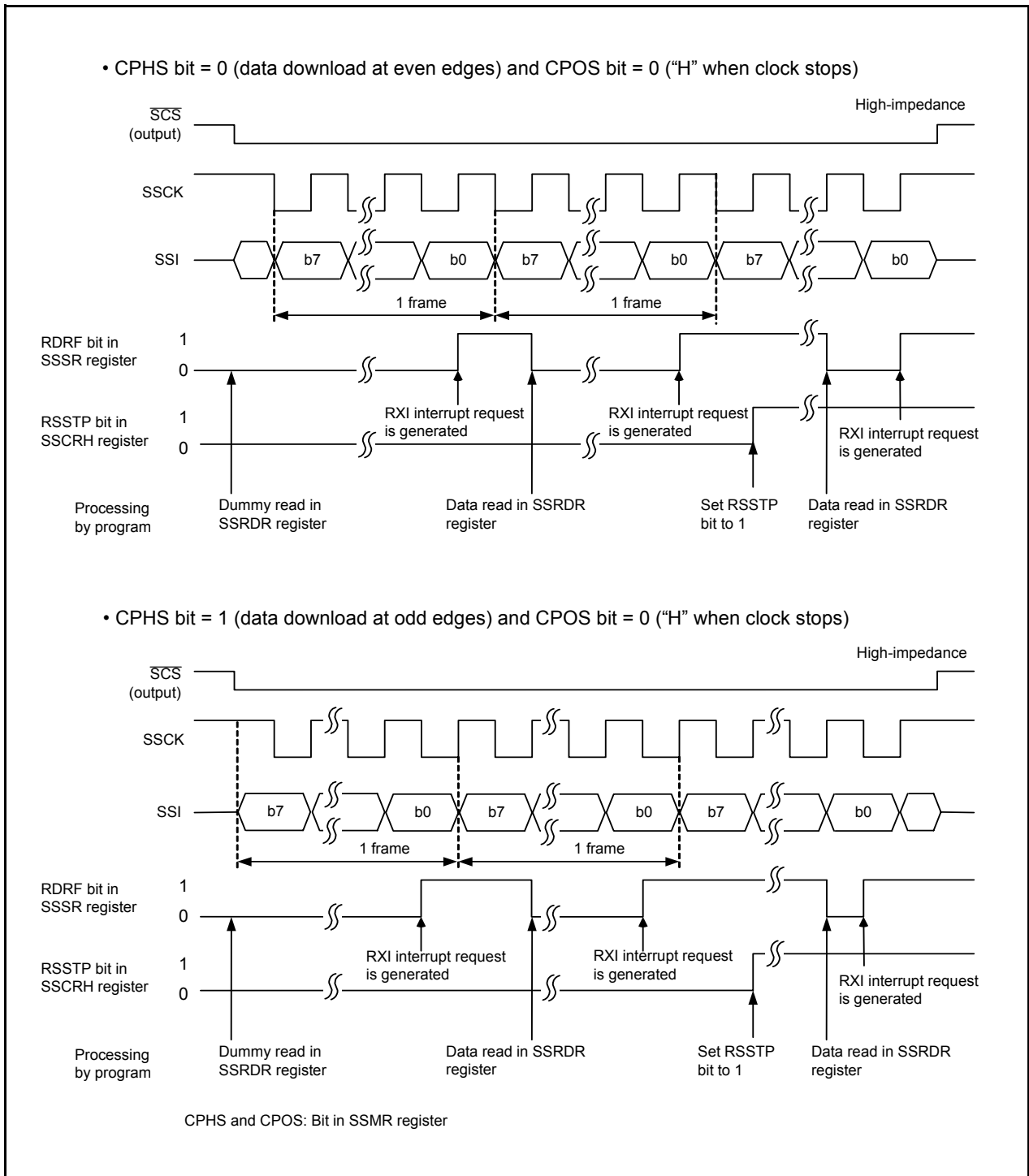


Figure 16.21 Example of Clock Synchronous Serial I/O with Chip Select Operation during Data Reception (4-Wire Bus Communication Mode)

16.2.7 $\overline{\text{SCS}}$ Pin Control and Arbitration

When setting the SSUMS bit in the SSMR2 register to 1 (4-wire bus communication mode) and the CSS1 bit in the SSMR2 register to 1 (functions as $\overline{\text{SCS}}$ output pin), set the MSS bit in the SSCRH register to 1 (operates as the master device) and check the arbitration of the $\overline{\text{SCS}}$ pin before starting serial transfer. If clock synchronous serial I/O with chip select detects that the synchronized internal $\overline{\text{SCS}}$ signal is held "L" in this period, the CE bit in the SSSR register is set to 1 (conflict error) and the MSS bit is automatically set to 0 (operates as a slave device).

Figure 16.22 shows the Arbitration Check Timing.

Future transmit operations are not performed while the CE bit is set to 1. Set the CE bit to 0 (no conflict error) before starting transmission.

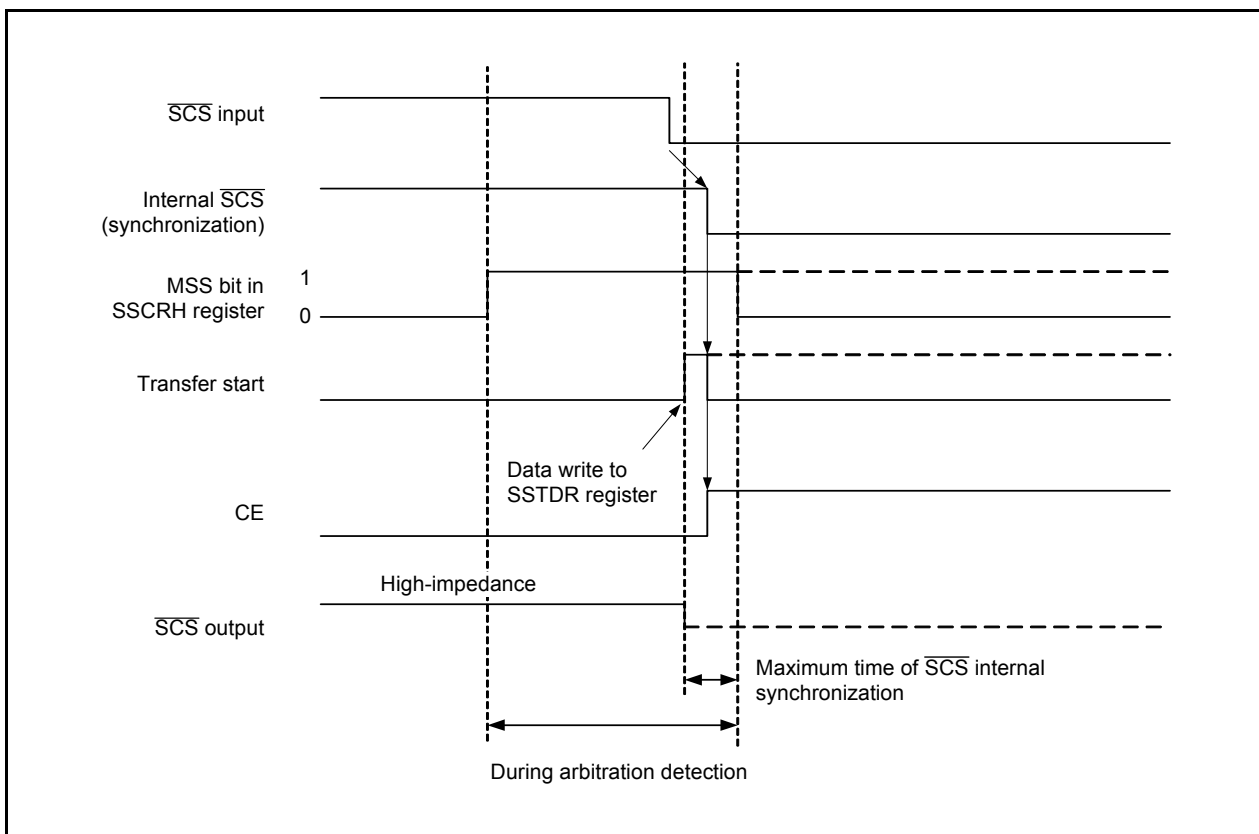


Figure 16.22 Arbitration Check Timing

16.2.8 Notes on Clock Synchronous Serial I/O with Chip Select

Set the IICSEL bit in the PMR register to 0 (select clock synchronous serial I/O with chip select function) to use the clock synchronous serial I/O with chip select function.

16.3 I²C bus Interface

The I²C bus interface is the circuit that performs serial communication based on the data transfer format of the Philips I²C bus.

Table 16.5 lists the I²C bus Interface Specifications, Figure 16.23 shows a Block Diagram of I²C bus interface, and Figure 16.24 shows the External Circuit Connection Example of Pins SCL and SDA. Figures 16.25 to 16.33 show the registers associated with the I²C bus interface.

* I²C bus is a trademark of Koninklijke Philips Electronics N. V.

Table 16.5 I²C bus Interface Specifications

| Item | Specification |
|-------------------------|---|
| Communication formats | <ul style="list-style-type: none"> • I²C bus format <ul style="list-style-type: none"> - Selectable as master/slave device - Continuous transmit/receive operation (because the shift register, transmit data register, and receive data register are independent) - Start/stop conditions are automatically generated in master mode - Automatic loading of acknowledge bit during transmission - Bit synchronization/wait function (In master mode, the state of the SCL signal is monitored per bit and the timing is synchronized automatically. If the transfer is not possible yet, the SCL signal goes "L" and the interface stands by.) - Support for direct drive of pins SCL and SDA (N-channel open-drain output) • Clock synchronous serial format <ul style="list-style-type: none"> - Continuous transmit/receive operation (because the shift register, transmit data register, and receive data register are independent) |
| I/O pins | SCL (I/O): Serial clock I/O pin SDA (I/O): Serial data I/O pin |
| Transfer clocks | <ul style="list-style-type: none"> • When the MST bit in the ICCR1 register is set to 0 The external clock (input from the SCL pin) • When the MST bit in the ICCR1 register is set to 1 The internal clock selected by bits CKS0 to CKS3 in the ICCR1 register (output from the SCL pin) |
| Receive error detection | <ul style="list-style-type: none"> • Overrun error detection (clock synchronous serial format) Indicates an overrun error during reception. When the last bit of the next data item is received while the RDRF bit in the ICSR register is set to 1 (data in the ICDRR register), the AL bit is set to 1. |
| Interrupt sources | <ul style="list-style-type: none"> • I²C bus format 6 sources⁽¹⁾ Transmit data empty (including when slave address matches), transmit ends, receive data full (including when slave address matches), arbitration lost, NACK detection, and stop condition detection. • Clock synchronous serial format 4 sources⁽¹⁾ Transmit data empty, transmit ends, receive data full and overrun error |
| Select functions | <ul style="list-style-type: none"> • I²C bus format <ul style="list-style-type: none"> - Selectable output level for acknowledge signal during reception • Clock synchronous serial format <ul style="list-style-type: none"> - MSB-first or LSB-first selectable as data transfer direction |

NOTE:

1. All sources use one interrupt vector for I²C bus interface.

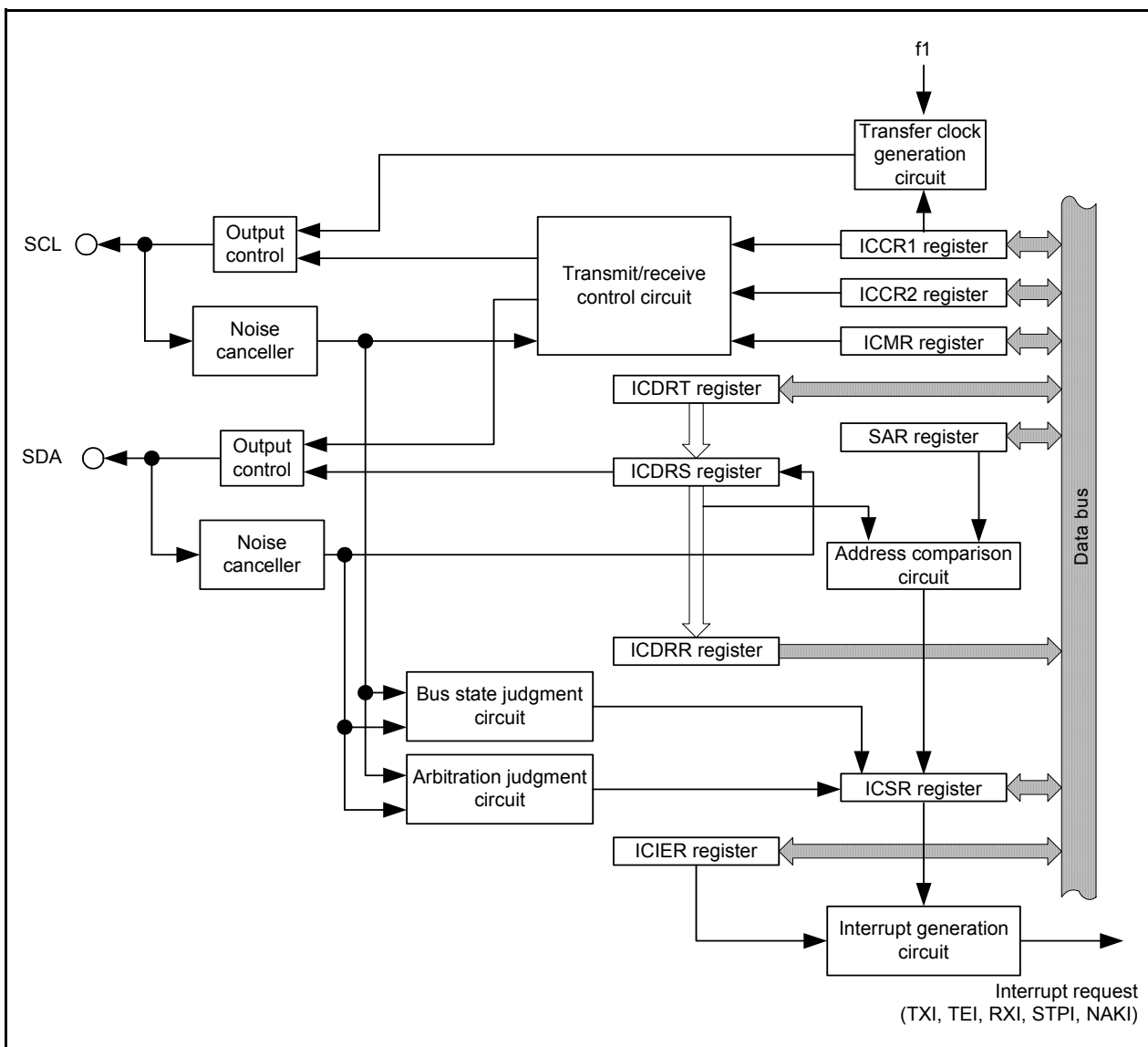


Figure 16.23 Block Diagram of I²C bus interface

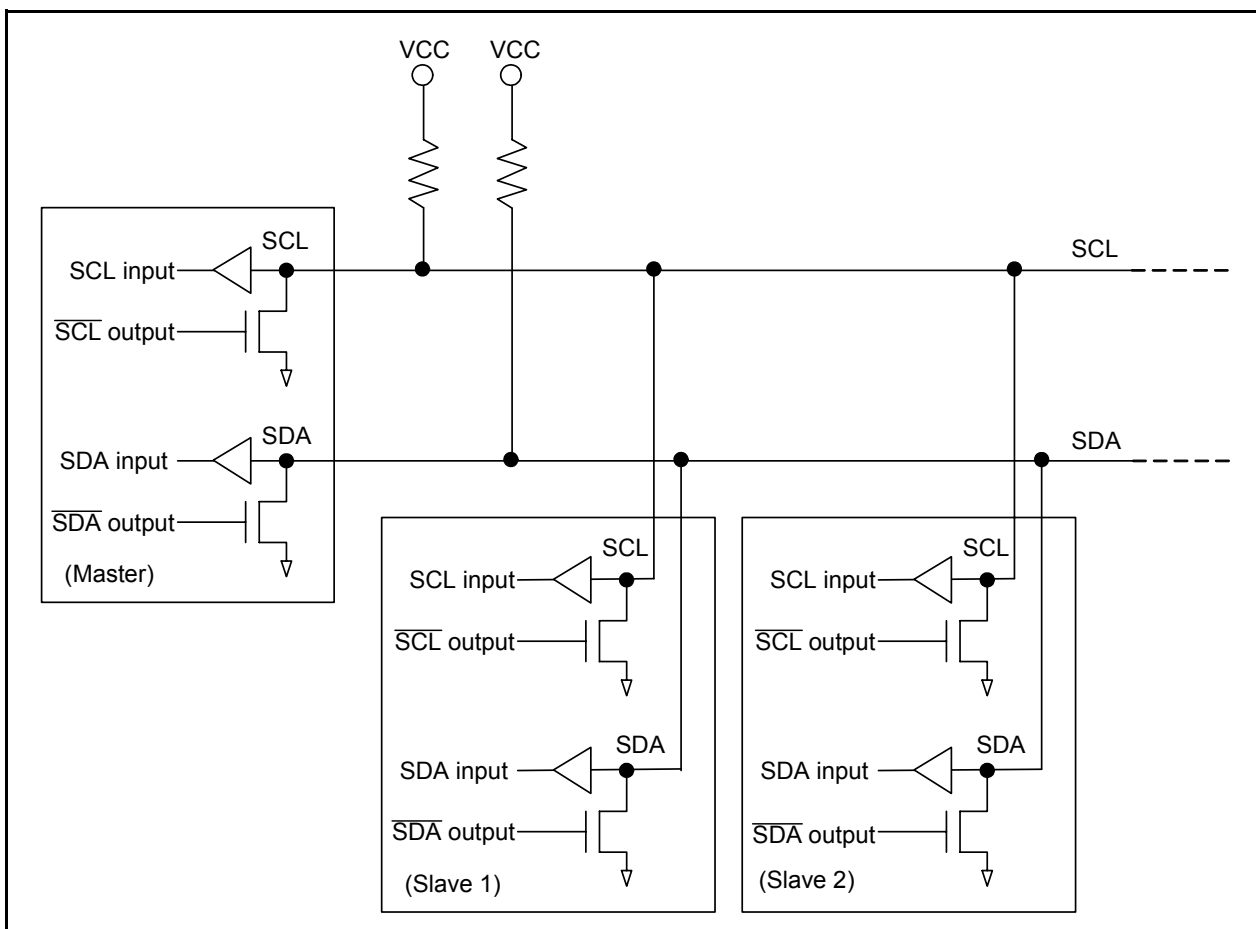


Figure 16.24 External Circuit Connection Example of Pins SCL and SDA

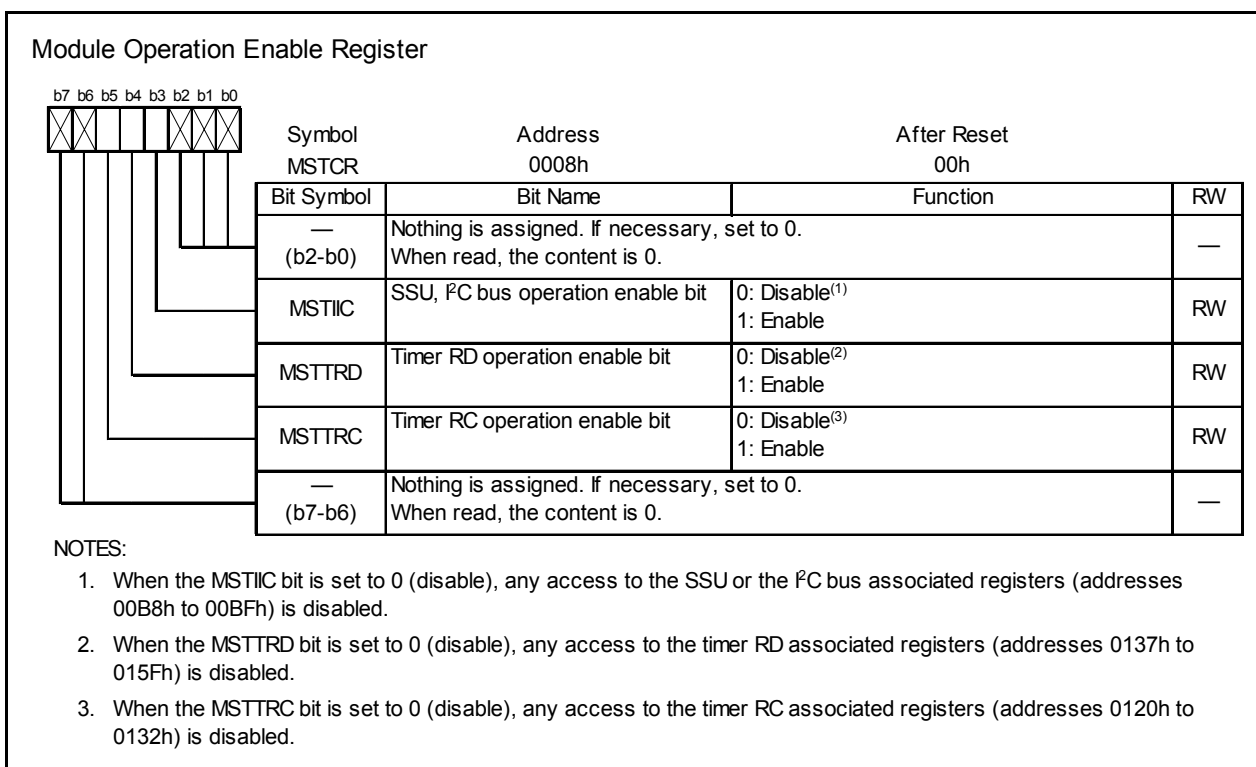


Figure 16.25 MSTCR Register

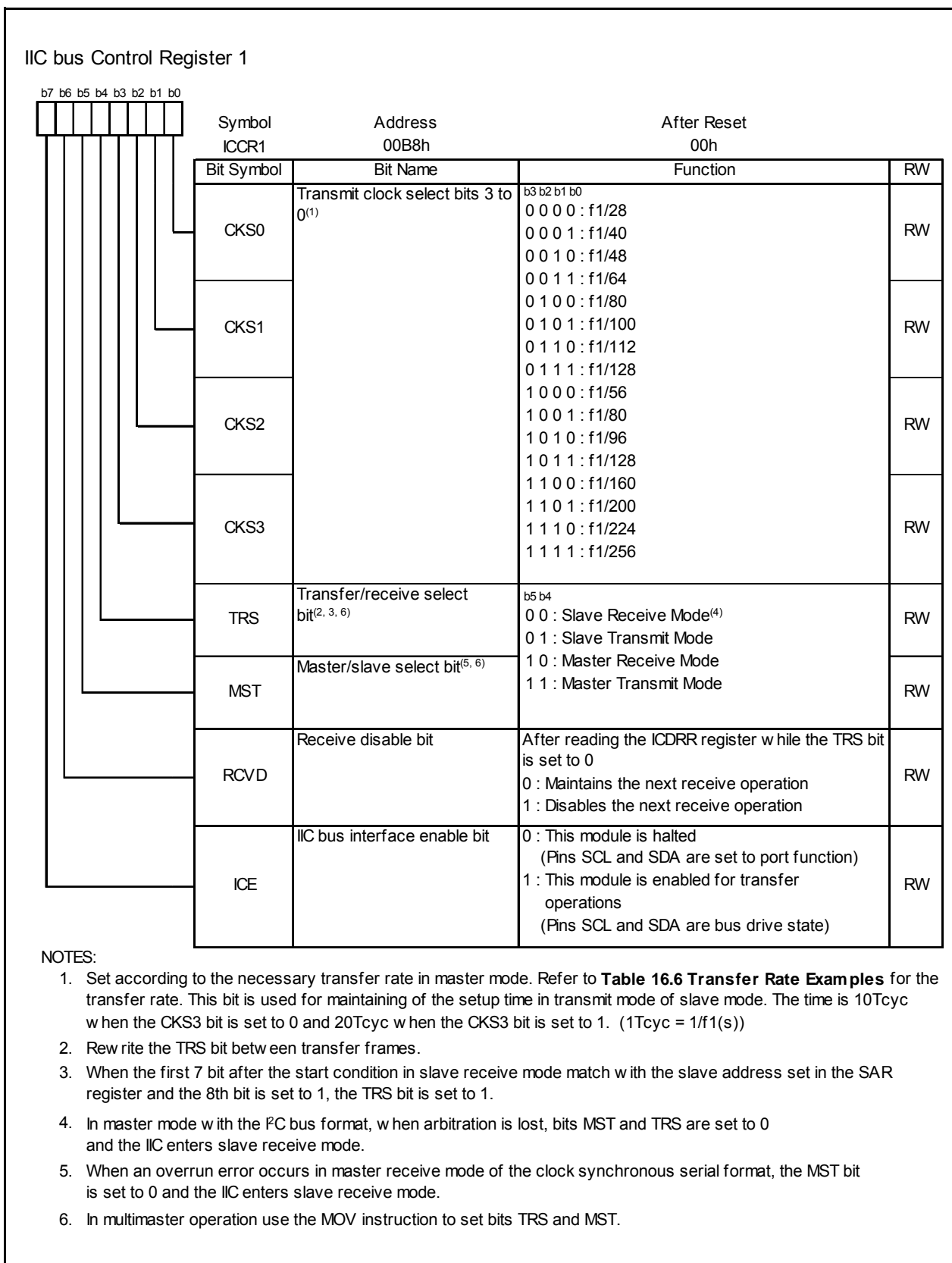


Figure 16.26 ICCR1 Register

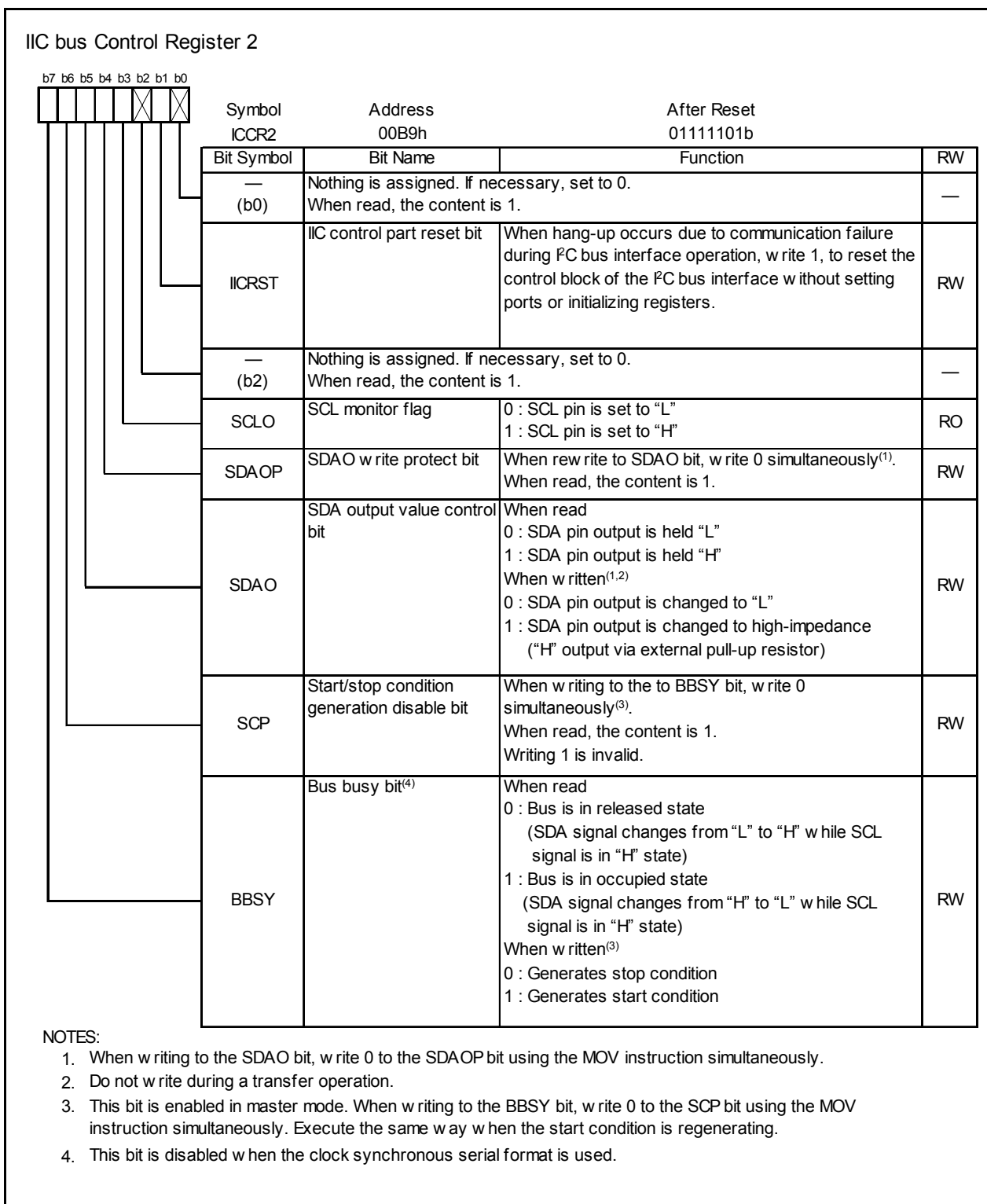


Figure 16.27 ICCR2 Register

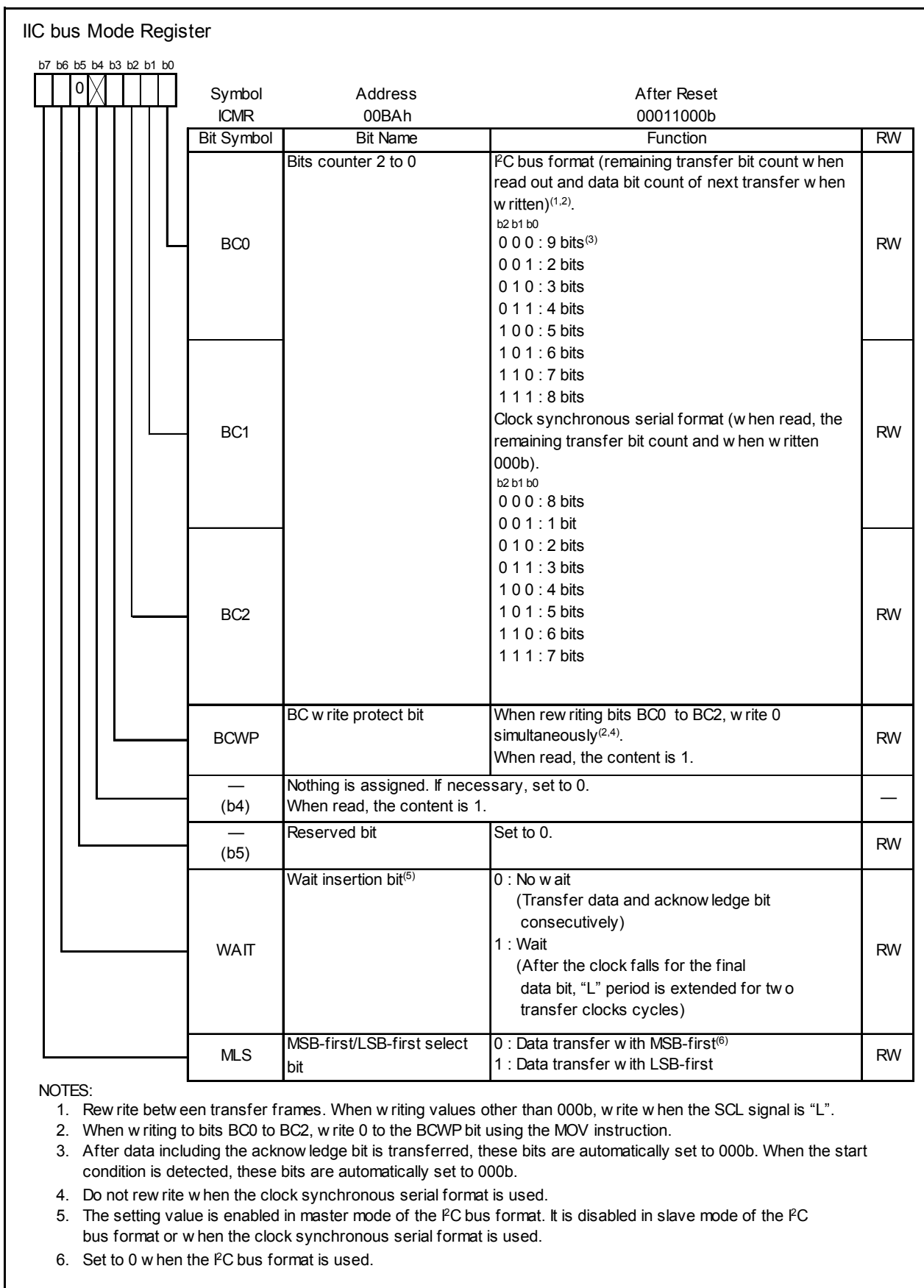


Figure 16.28 ICMR Register

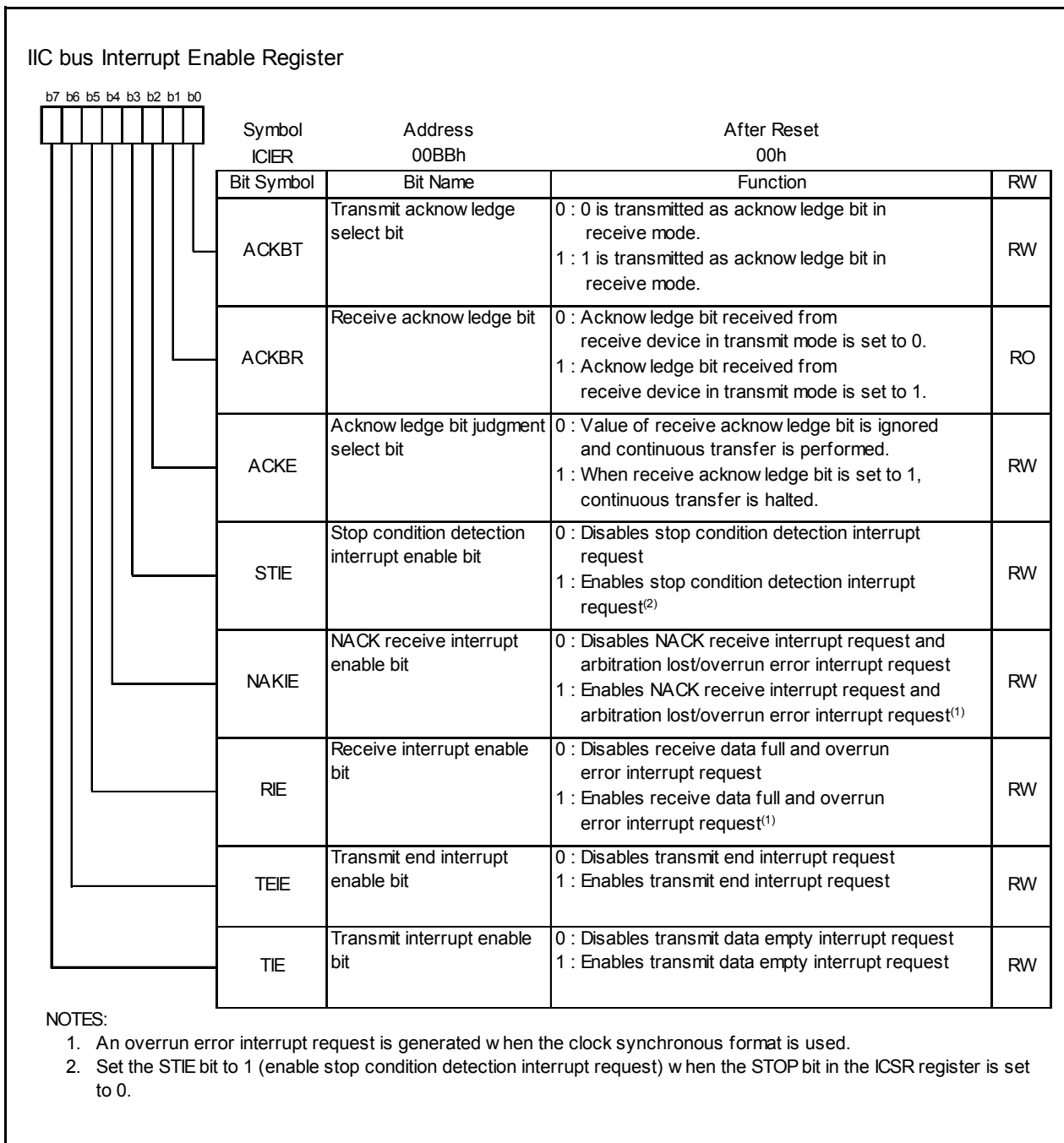


Figure 16.29 ICIER Register

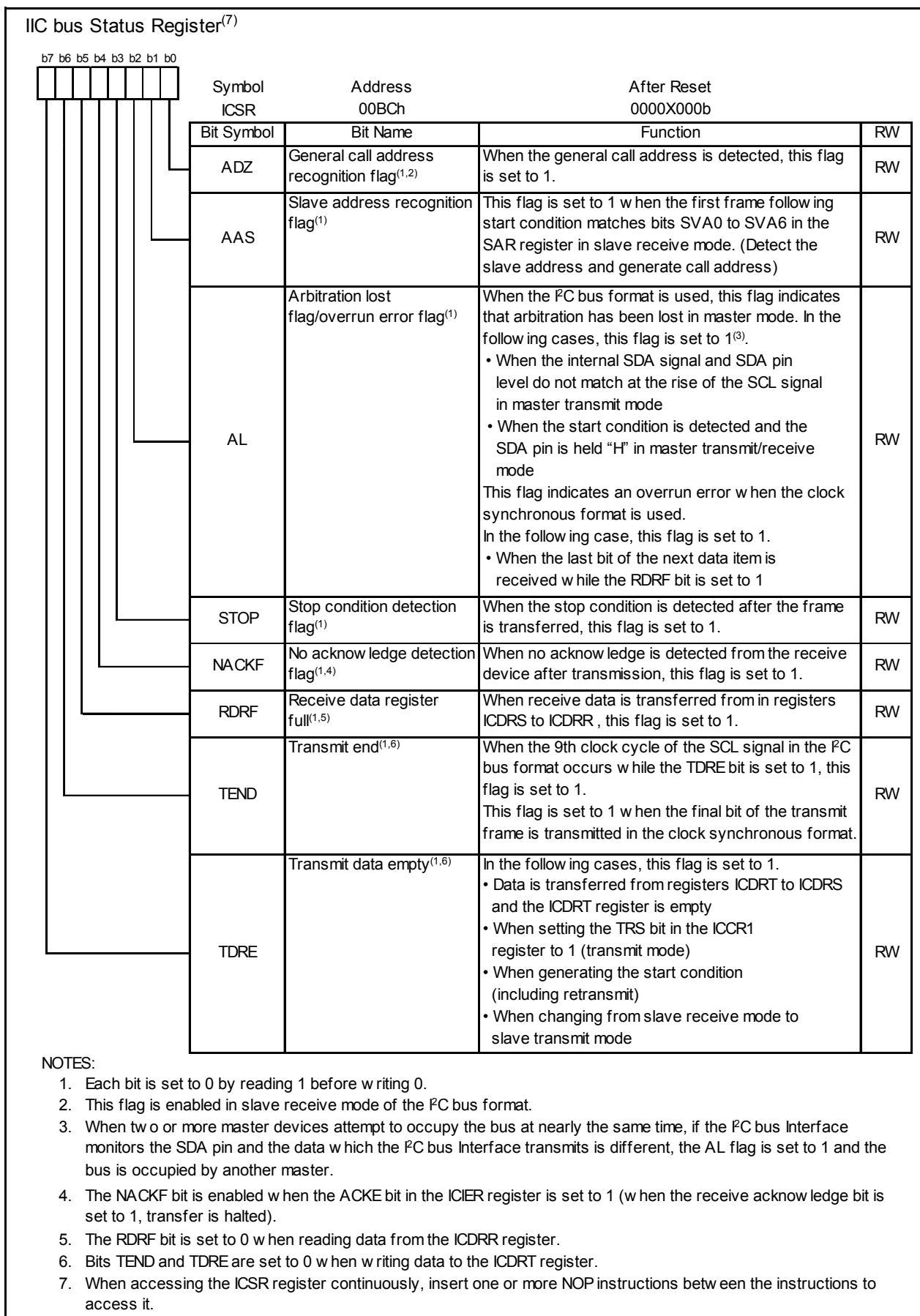


Figure 16.30 ICSR Register

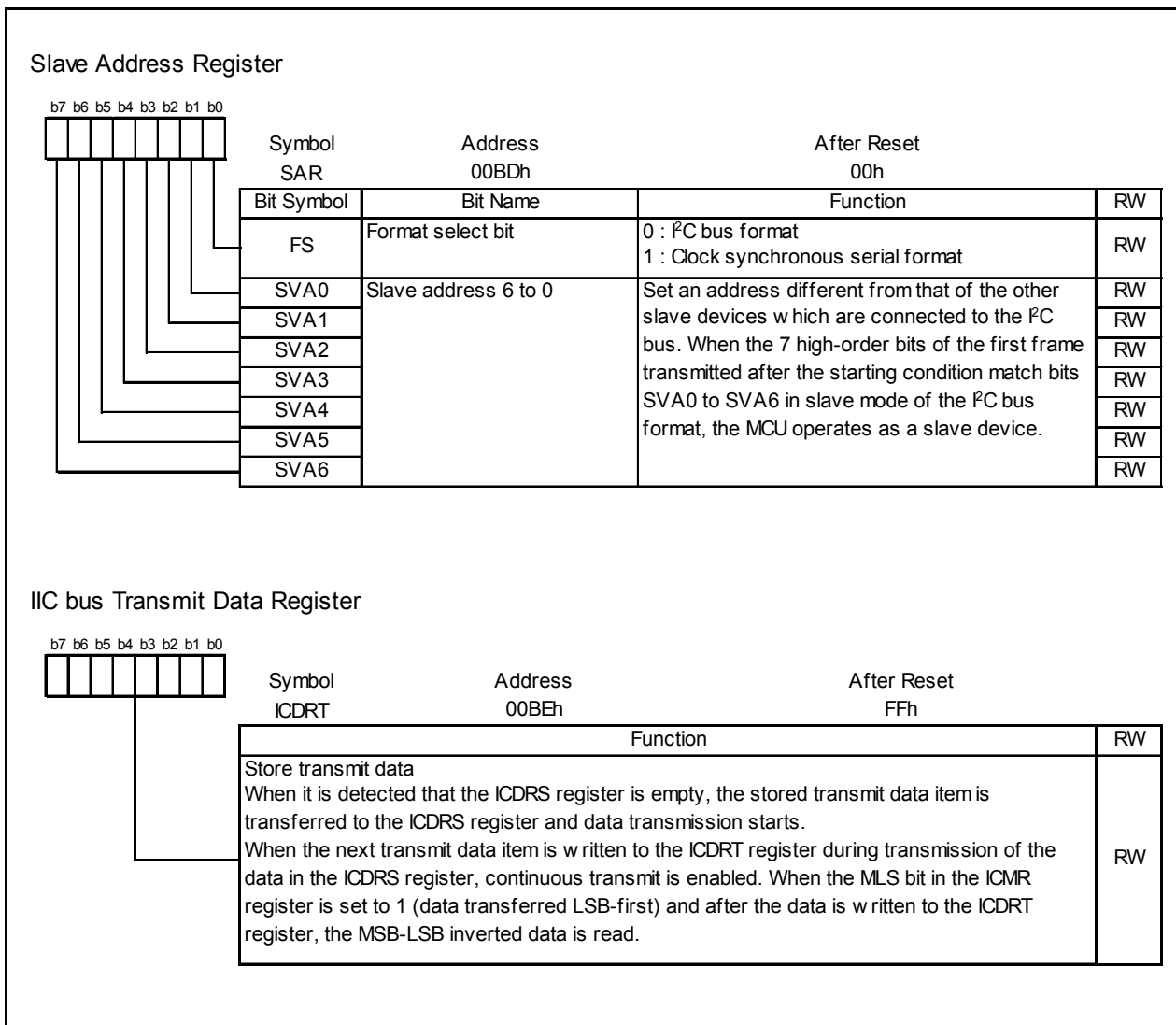


Figure 16.31 Registers SAR and ICDRT

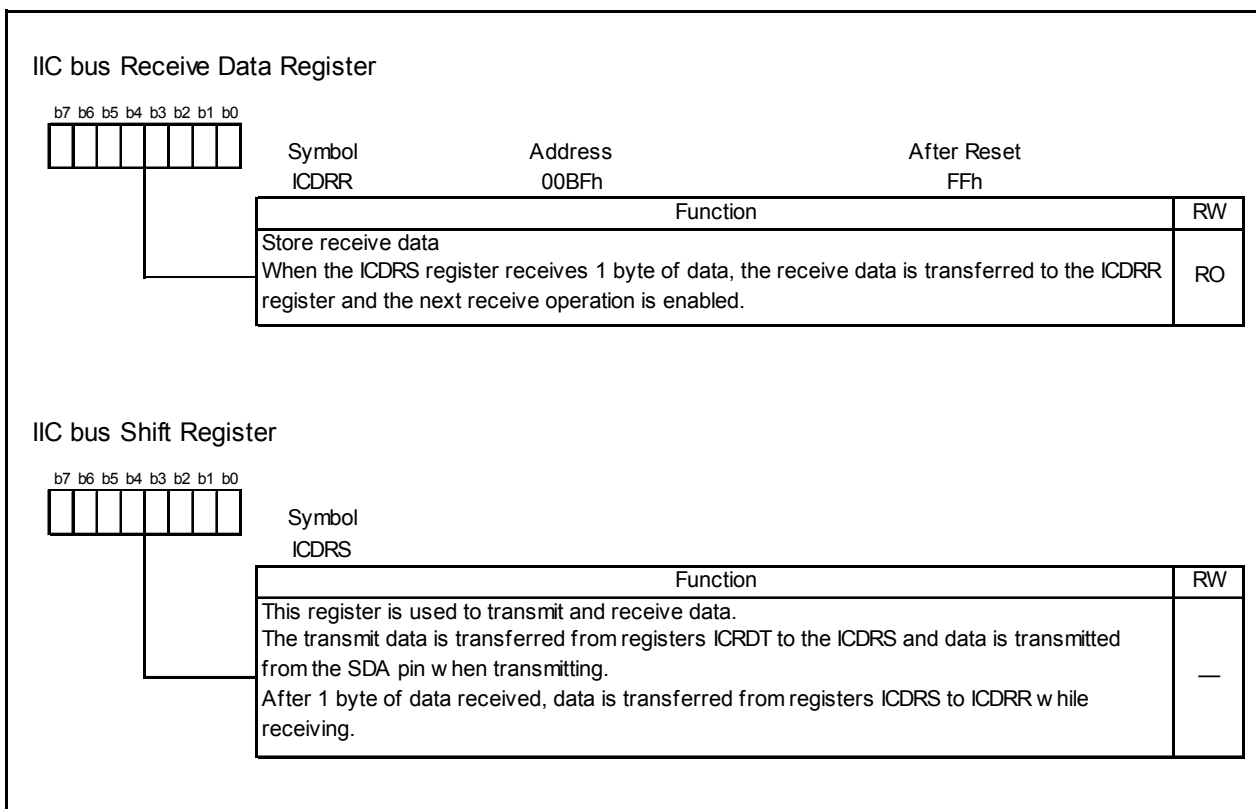


Figure 16.32 Registers ICDRR and ICDRS

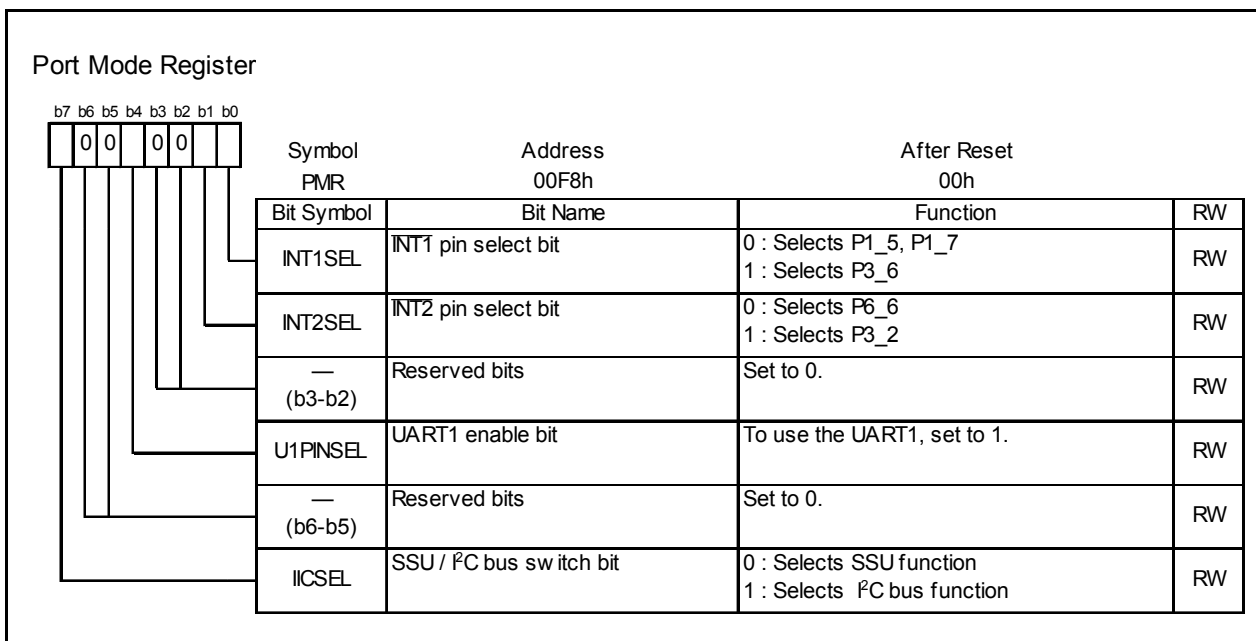


Figure 16.33 PMR Register

16.3.1 Transfer Clock

When the MST bit in the ICCR1 register is set to 0, the transfer clock is the external clock input from the SCL pin. When the MST bit in the ICCR1 register is set to 1, the transfer clock is the internal clock selected by bits CKS0 to CKS3 in the ICCR1 register and the transfer clock is output from the SCL pin.

Table 16.6 lists the Transfer Rate Examples.

Table 16.6 Transfer Rate Examples

| ICCR1 Register | | | | Transfer Clock | Transfer Rate | | | | |
|----------------|------|------|------|----------------|---------------|------------|-------------|-------------|-------------|
| CKS3 | CKS2 | CKS1 | CKS0 | | f1 = 5 MHz | f1 = 8 MHz | f1 = 10 MHz | f1 = 16 MHz | f1 = 20 MHz |
| 0 | 0 | 0 | 0 | f1/28 | 179 kHz | 286 kHz | 357 kHz | 571 kHz | 714 kHz |
| | | | 1 | f1/40 | 125 kHz | 200 kHz | 250 kHz | 400 kHz | 500 kHz |
| | | 1 | 0 | f1/48 | 104 kHz | 167 kHz | 208 kHz | 333 kHz | 417 kHz |
| | | | 1 | f1/64 | 78.1 kHz | 125 kHz | 156 kHz | 250 kHz | 313 kHz |
| | 1 | 0 | 0 | f1/80 | 62.5 kHz | 100 kHz | 125 kHz | 200 kHz | 250 kHz |
| | | | 1 | f1/100 | 50.0 kHz | 80.0 kHz | 100 kHz | 160 kHz | 200 kHz |
| | | 1 | 0 | f1/112 | 44.6 kHz | 71.4 kHz | 89.3 kHz | 143 kHz | 179 kHz |
| | | | 1 | f1/128 | 39.1 kHz | 62.5 kHz | 78.1 kHz | 125 kHz | 156 kHz |
| 1 | 0 | 0 | 0 | f1/56 | 89.3 kHz | 143 kHz | 179 kHz | 286 kHz | 357 kHz |
| | | | 1 | f1/80 | 62.5 kHz | 100 kHz | 125 kHz | 200 kHz | 250 kHz |
| | | 1 | 0 | f1/96 | 52.1 kHz | 83.3 kHz | 104 kHz | 167 kHz | 208 kHz |
| | | | 1 | f1/128 | 39.1 kHz | 62.5 kHz | 78.1 kHz | 125 kHz | 156 kHz |
| | 1 | 0 | 0 | f1/160 | 31.3 kHz | 50.0 kHz | 62.5 kHz | 100 kHz | 125 kHz |
| | | | 1 | f1/200 | 25.0 kHz | 40.0 kHz | 50.0 kHz | 80.0 kHz | 100 kHz |
| | | 1 | 0 | f1/224 | 22.3 kHz | 35.7 kHz | 44.6 kHz | 71.4 kHz | 89.3 kHz |
| | | | 1 | f1/256 | 19.5 kHz | 31.3 kHz | 39.1 kHz | 62.5 kHz | 78.1 kHz |

16.3.2 Interrupt Requests

The I²C bus interface has six interrupt requests when the I²C bus format is used and four interrupt requests when the clock synchronous serial format is used.

Table 16.7 lists the Interrupt Requests of I²C bus Interface.

Since these interrupt requests are allocated at the I²C bus interface interrupt vector table, determining the source bit by bit is necessary.

Table 16.7 Interrupt Requests of I²C bus Interface

| Interrupt Request | | Generation Condition | Format | |
|--------------------------------|------|--|----------------------|--------------------------|
| | | | I ² C bus | Clock Synchronous Serial |
| Transmit data empty | TXI | TIE = 1 and TDRE = 1 | Enabled | Enabled |
| Transmit ends | TEI | TEIE = 1 and TEND = 1 | Enabled | Enabled |
| Receive data full | RXI | RIE = 1 and RDRF = 1 | Enabled | Enabled |
| Stop condition detection | STPI | STIE = 1 and STOP = 1 | Enabled | Disabled |
| NACK detection | NAKI | NAKIE = 1 and AL = 1 (or NAKIE = 1 and NACKF = 1) | Enabled | Disabled |
| Arbitration lost/overrun error | | | Enabled | Enabled |

STIE, NAKIE, RIE, TEIE, TIE: Bits in ICIER register

AL, STOP, NACKF, RDRF, TEND, TDRE: Bits in ICSR register

When the generation conditions listed in Table 16.7 are met, an I²C bus interface interrupt request is generated. Set the interrupt generation conditions to 0 by the I²C bus interface interrupt routine. However, bits TDRE and TEND are automatically set to 0 by writing transmit data to the ICDRT register and the RDRF bit is automatically set to 0 by reading the ICDRR register. When writing transmit data to the ICDRT register, the TDRE bit is set to 0. When data is transferred from registers ICDRT to ICDRS, the TDRE bit is set to 1 and by further setting the TDRE bit to 0, 1 additional byte may be transmitted.

Set the STIE bit to 1 (enable stop condition detection interrupt request) when the STOP bit is set to 0.

16.3.3 I²C bus Interface Mode

16.3.3.1 I²C bus Format

Setting the FS bit in the SAR register to 0 enables communication in I²C bus format.

Figure 16.34 shows the I²C bus Format and Bus Timing. The 1st frame following the start condition consists of 8 bits.

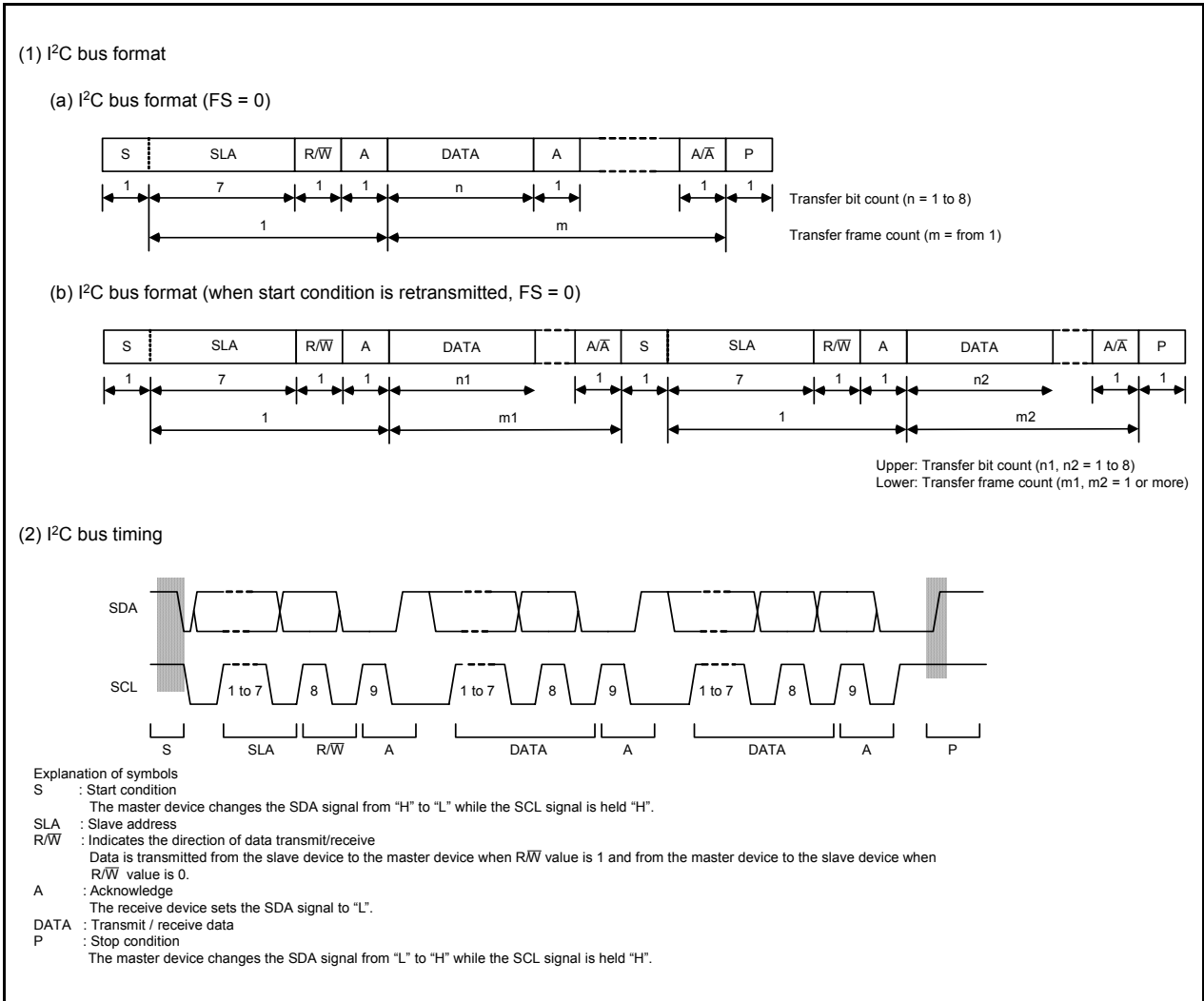


Figure 16.34 I²C bus Format and Bus Timing

16.3.3.2 Master Transmit Operation

In master transmit mode, the master device outputs the transmit clock and data, and the slave device returns an acknowledge signal.

Figures 16.35 and 16.36 show the Operating Timing in Master Transmit Mode (I²C bus Interface Mode).

The transmit procedure and operation in master transmit mode are as follows.

- (1) Set the STOP bit in the ICSR register to 0 to reset it. Then set the ICE bit in the ICCR1 register to 1 (transfer operation enabled). Then set bits WAIT and MLS in the ICMR register and set bits CKS0 to CKS3 in the ICCR1 register (initial setting).
- (2) Read the BBSY bit in the ICCR2 register to confirm that the bus is free. Set bits TRS and MST in the ICCR1 register to master transmit mode. The start condition is generated by writing 1 to the BBSY bit and 0 to the SCP bit by the MOV instruction.
- (3) After confirming that the TDRE bit in the ICSR register is set to 1 (data is transferred from registers ICDRT to ICDRS), write transmit data to the ICDRT register (data in which a slave address and R/W are indicated in the 1st byte). At this time, the TDRE bit is automatically set to 0, data is transferred from registers ICDRT to ICDRS, and the TDRE bit is set to 1 again.
- (4) When transmission of 1 byte of data is completed while the TDRE bit is set to 1, the TEND bit in the ICSR register is set to 1 at the rise of the 9th transmit clock pulse. Read the ACKBR bit in the ICIER register, and confirm that the slave is selected. Write the 2nd byte of data to the ICDRT register. Since the slave device is not acknowledged when the ACKBR bit is set to 1, generate the stop condition. The stop condition is generated by the writing 0 to the BBSY bit and 0 to the SCP bit by the MOV instruction. The SCL signal is held "L" until data is available and the stop condition is generated.
- (5) Write the transmit data after the 2nd byte to the ICDRT register every time the TDRE bit is set to 1.
- (6) When writing the number of bytes to be transmitted to the ICDRT register, wait until the TEND bit is set to 1 while the TDRE bit is set to 1. Or wait for NACK (the NACKF bit in the ICSR register is set to 1) from the receive device while the ACKE bit in the ICIER register is set to 1 (when the receive acknowledge bit is set to 1, transfer is halted). Then generate the stop condition before setting bits TEND and NACKF to 0.
- (7) When the STOP bit in the ICSR register is set to 1, return to slave receive mode.

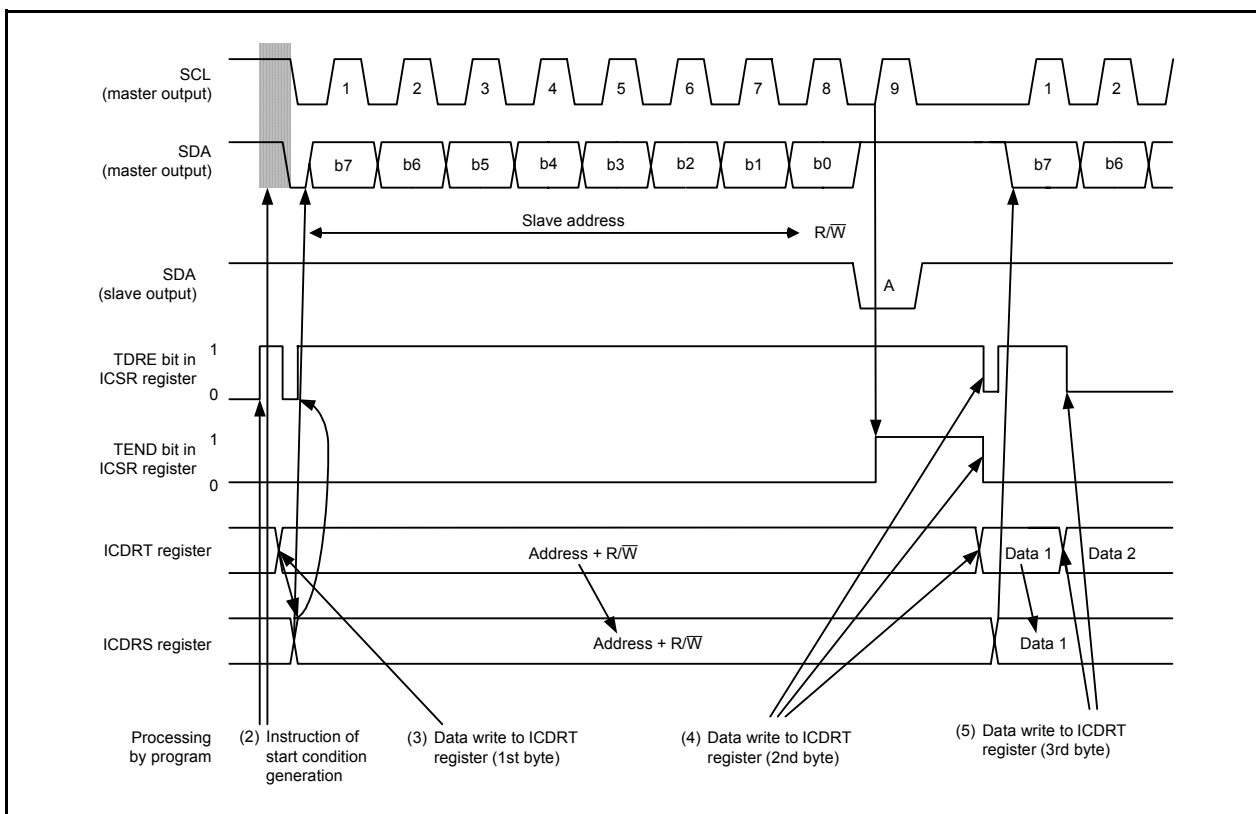


Figure 16.35 Operating Timing in Master Transmit Mode (I²C bus Interface Mode) (1)

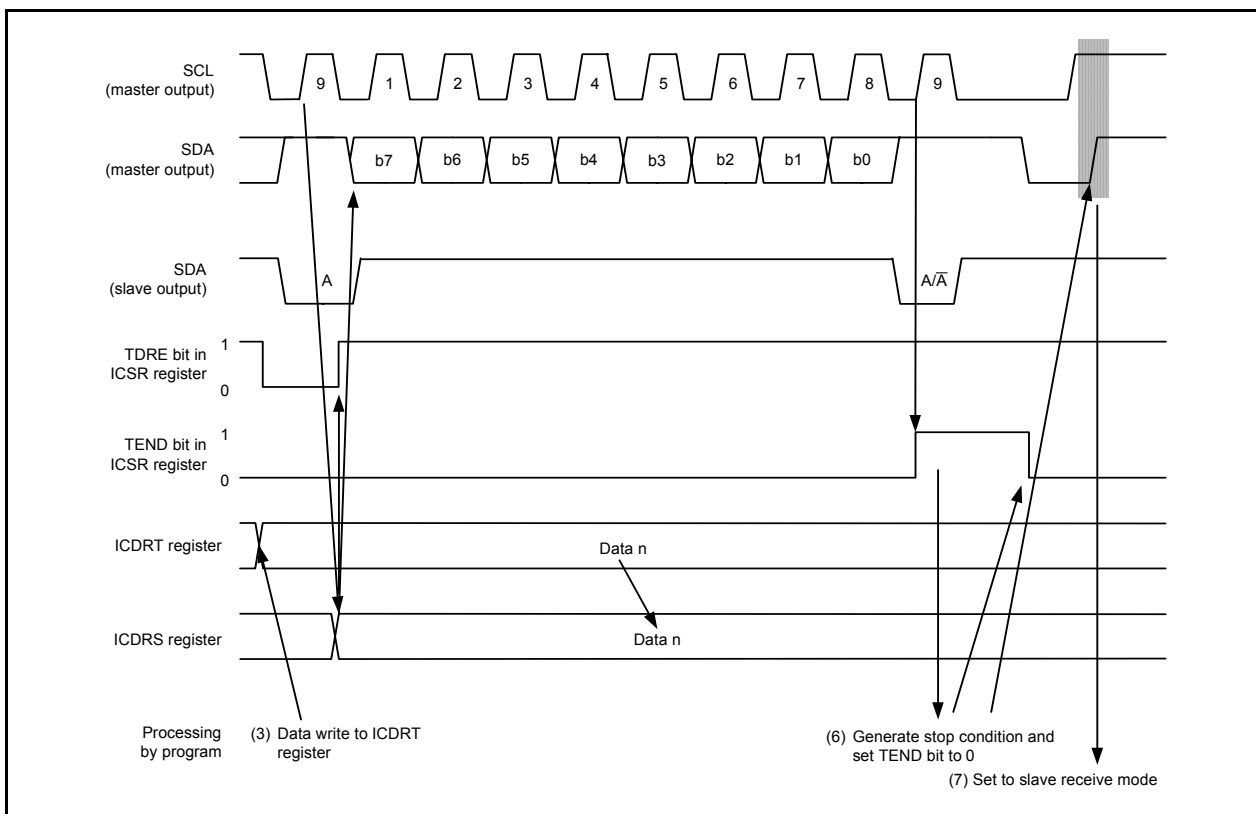


Figure 16.36 Operating Timing in Master Transmit Mode (I²C bus Interface Mode) (2)

16.3.3.3 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.

Figures 16.37 and 16.38 show the Operating Timing in Master Receive Mode (I²C bus Interface Mode).

The receive procedure and operation in master receive mode are shown below.

- (1) After setting the TEND bit in the ICSR register to 0, switch from master transmit mode to master receive mode by setting the TRS bit in the ICCR1 register to 0. Also, set the TDRE bit in the ICSR register to 0.
- (2) When performing the dummy read of the ICDRR register and starting the receive operation, the receive clock is output in synchronization with the internal clock and data is received. The master device outputs the level set by the ACKBT bit in the ICIER register to the SDA pin at the rising edge of the 9th clock cycle of the receive clock.
- (3) The 1-frame data receive is completed and the RDRF bit in the ICSR register is set to 1 at the rise of the 9th clock cycle. At this time, when reading the ICDRR register, the received data can be read and the RDRF bit is set to 0 simultaneously.
- (4) Continuous receive operation is enabled by reading the ICDRR register every time the RDRF bit is set to 1. If the 8th clock cycle falls after the ICDRR register is read by another process while the RDRF bit is set to 1, the SCL signal is fixed "L" until the ICDRR register is read.
- (5) If the next frame is the last receive frame and the RCVD bit in the ICCR1 register is set to 1 (disables the next receive operation) before reading the ICDRR register, stop condition generation is enabled after the next receive operation.
- (6) When the RDRF bit is set to 1 at the rise of the 9th clock cycle of the receive clock, generate the stop condition.
- (7) When the STOP bit in the ICSR register is set to 1, read the ICDRR register and set the RCVD bit to 0 (maintain the following receive operation).
- (8) Return to slave receive mode.

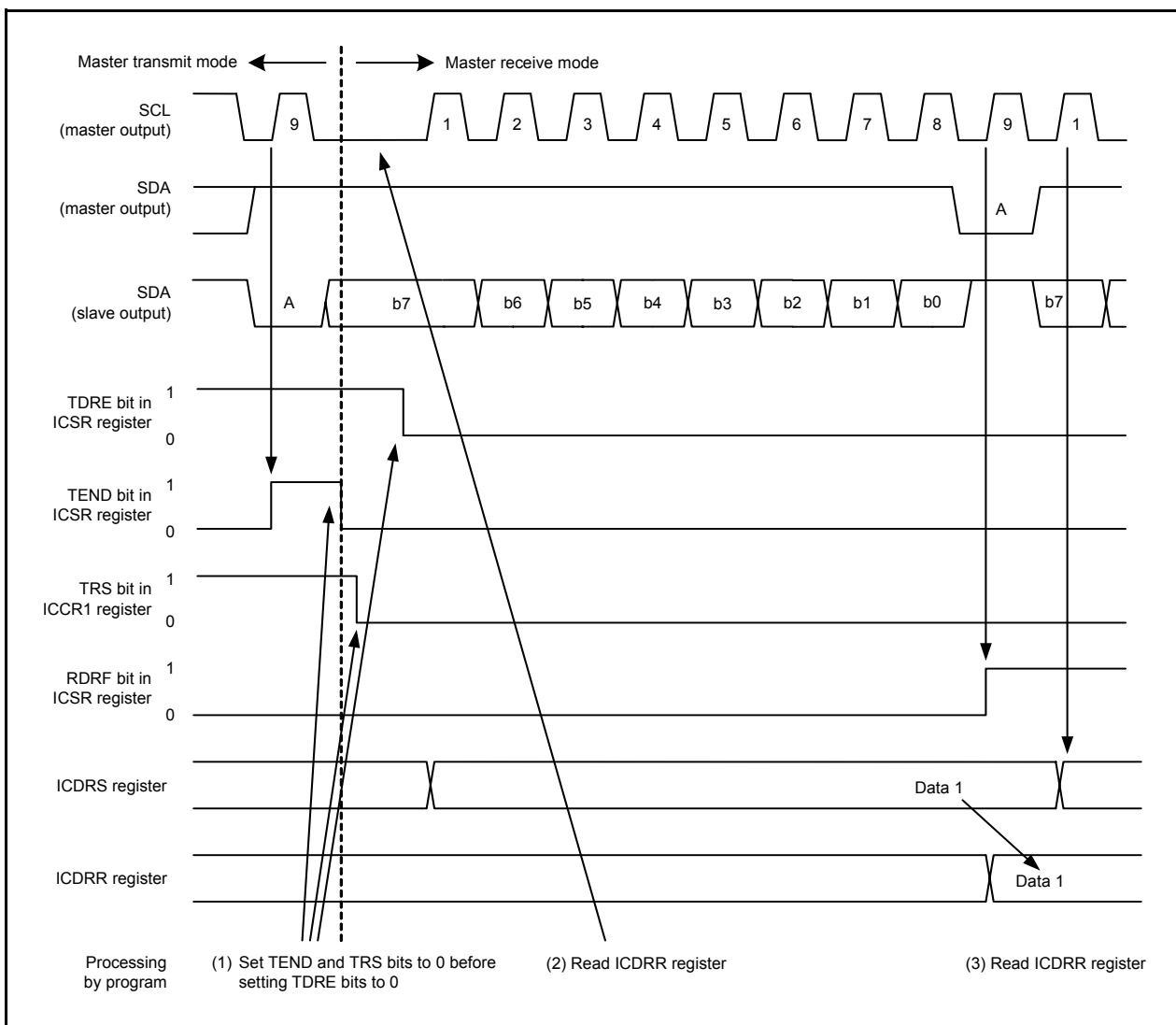


Figure 16.37 Operating Timing in Master Receive Mode (I²C bus Interface Mode) (1)

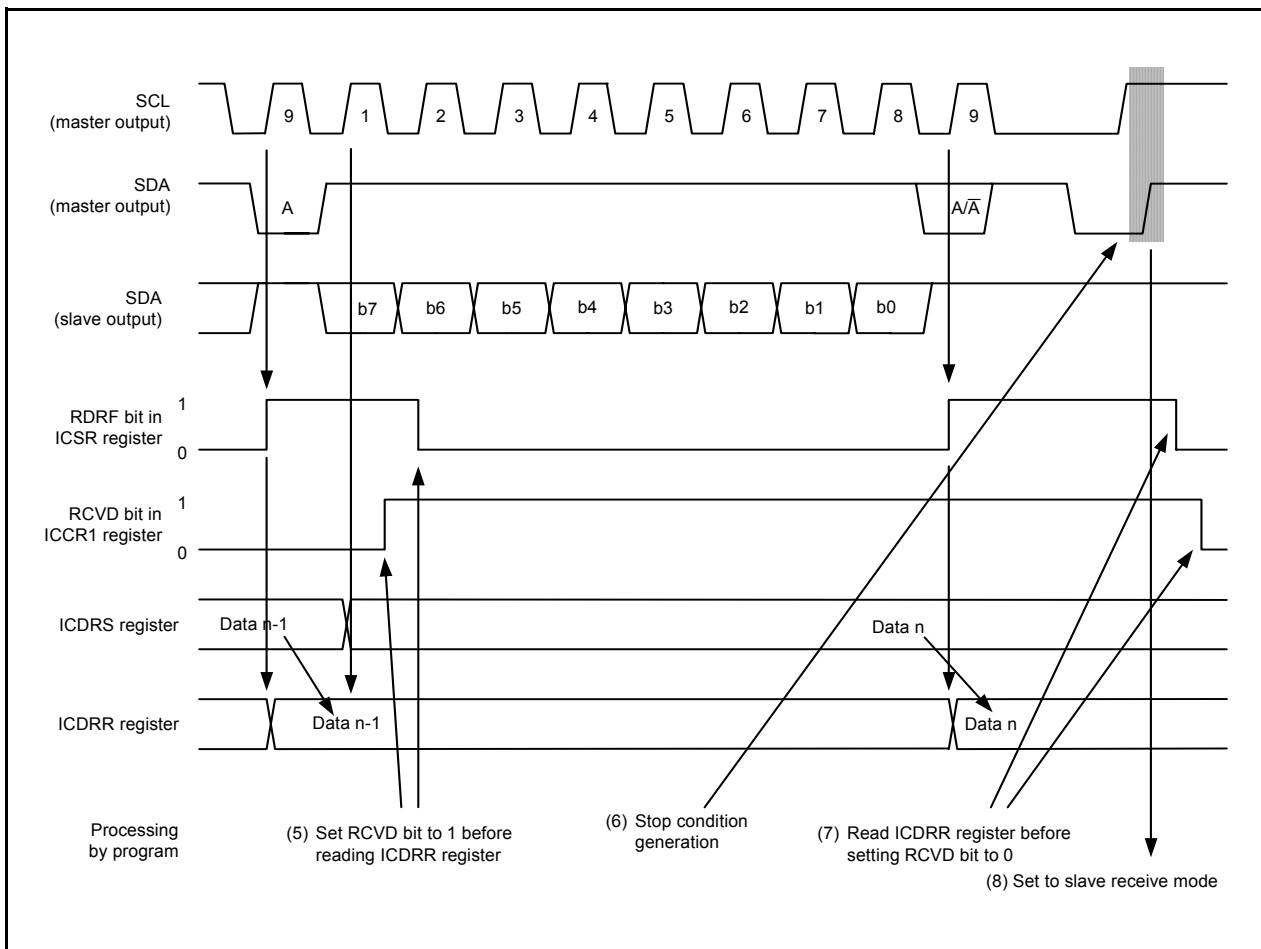


Figure 16.38 Operating Timing in Master Receive Mode (I²C bus Interface Mode) (2)

16.3.3.4 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.

Figures 16.39 and 16.40 show the Operating Timing in Slave Transmit Mode (I²C bus Interface Mode).

The transmit procedure and operation in slave transmit mode are as follows.

- (1) Set the ICE bit in the ICCR1 register to 1 (transfer operation enabled). Set bits WAIT and MLS in the ICMR register and bits CKS0 to CKS3 in the ICCR1 register (initial setting). Set bits TRS and MST in the ICCR1 register to 0 and wait until the slave address matches in slave receive mode.
- (2) When the slave address matches at the 1st frame after detecting the start condition, the slave device outputs the level set by the ACKBT bit in the ICIER register to the SDA pin at the rise of the 9th clock cycle. At this time, if the 8th bit of data (R/\overline{W}) is 1, bits TRS and TDRE in the ICSR register are set to 1, and the mode is switched to slave transmit mode automatically. Continuous transmission is enabled by writing transmit data to the ICDRT register every time the TDRE bit is set to 1.
- (3) When the TDRE bit in the ICDRT register is set to 1 after writing the last transmit data to the ICDRT register, wait until the TEND bit in the ICSR register is set to 1 while the TDRE bit is set to 1. When the TEND bit is set to 1, set the TEND bit to 0.
- (4) The SCL signal is released by setting the TRS bit to 0 and performing a dummy read of the ICDRR register to end the process.
- (5) Set the TDRE bit to 0.

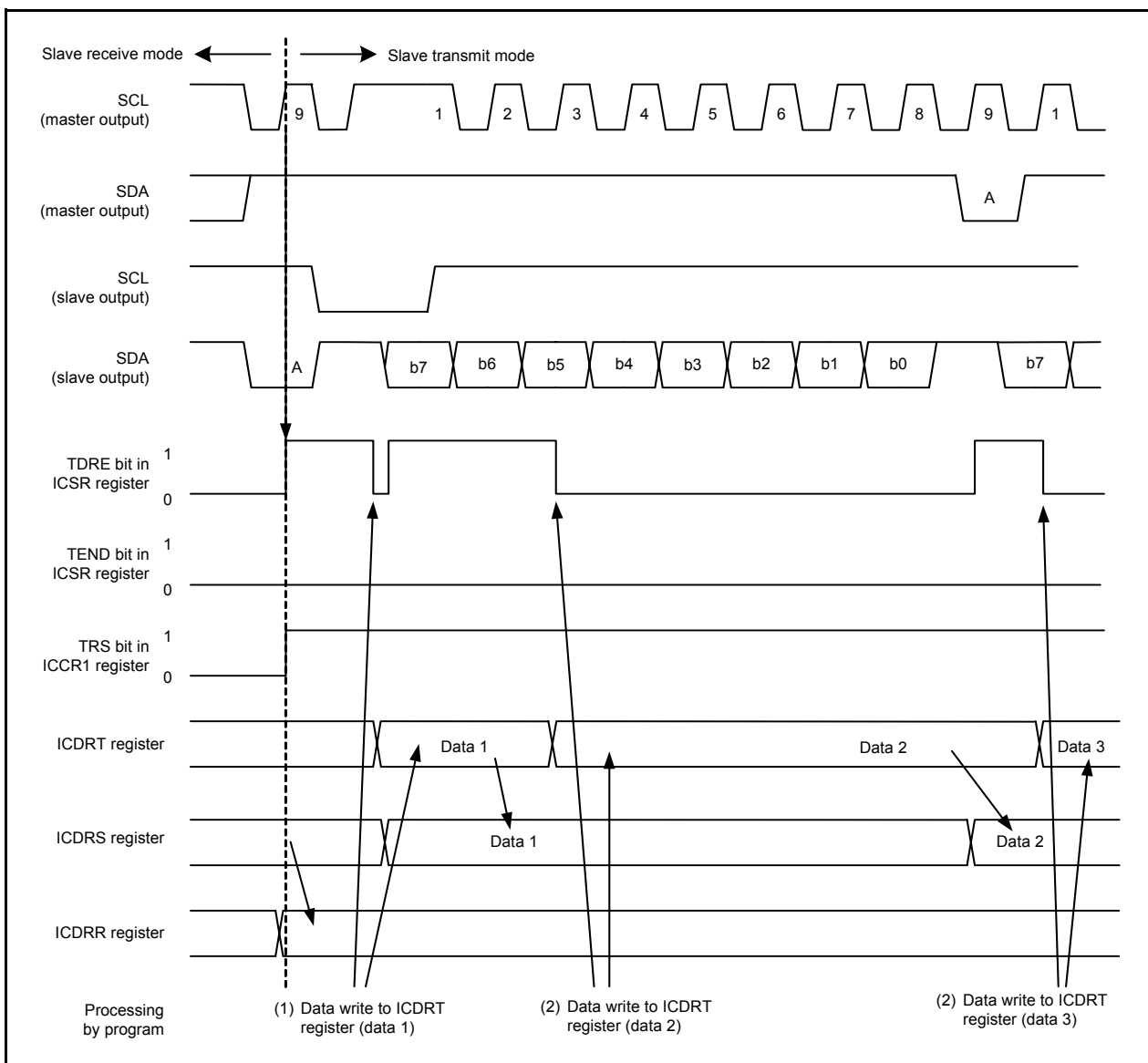


Figure 16.39 Operating Timing in Slave Transmit Mode (I2C bus Interface Mode) (1)

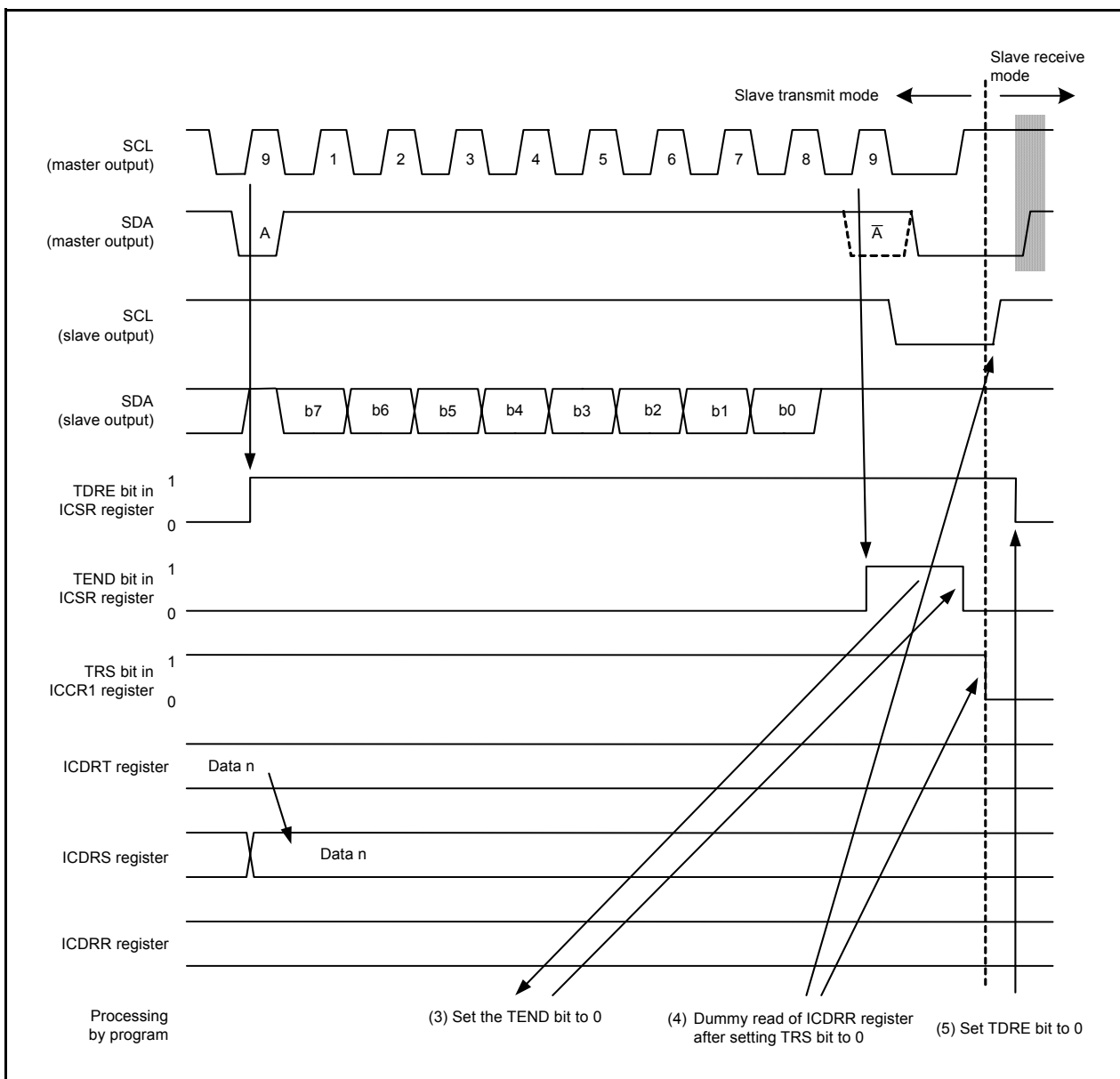


Figure 16.40 Operating Timing in Slave Transmit Mode (I2C bus interface Mode) (2)

16.3.3.5 Slave Receive Operation

In slave receive mode, the master device outputs the transmit clock and data, and the slave device returns an acknowledge signal.

Figures 16.41 and 16.42 show the Operating Timing in Slave Receive Mode (I²C bus Interface Mode).

The receive procedure and operation in slave receive mode are as follows.

- (1) Set the ICE bit in the ICCR1 register to 1 (transfer operation enabled). Set bits WAIT and MLS in the ICMR register and bits CKS0 to CKS3 in the ICCR1 register (initial setting). Set bits TRS and MST in the ICCR1 register to 0 and wait until the slave address matches in slave receive mode.
- (2) When the slave address matches at the 1st frame after detecting the start condition, the slave device outputs the level set in the ACKBT bit in the ICIER register to the SDA pin at the rise of the 9th clock cycle. Since the RDRF bit in the ICSR register is set to 1 simultaneously, perform the dummy read (the read data is unnecessary because it indicates the slave address and $\overline{R/W}$).
- (3) Read the ICDRR register every time the RDRF bit is set to 1. If the 8th clock cycle falls while the RDRF bit is set to 1, the SCL signal is fixed "L" until the ICDRR register is read. The setting change of the acknowledge signal returned to the master device before reading the ICDRR register takes affect from the following transfer frame.
- (4) Reading the last byte is performed by reading the ICDRR register in like manner.

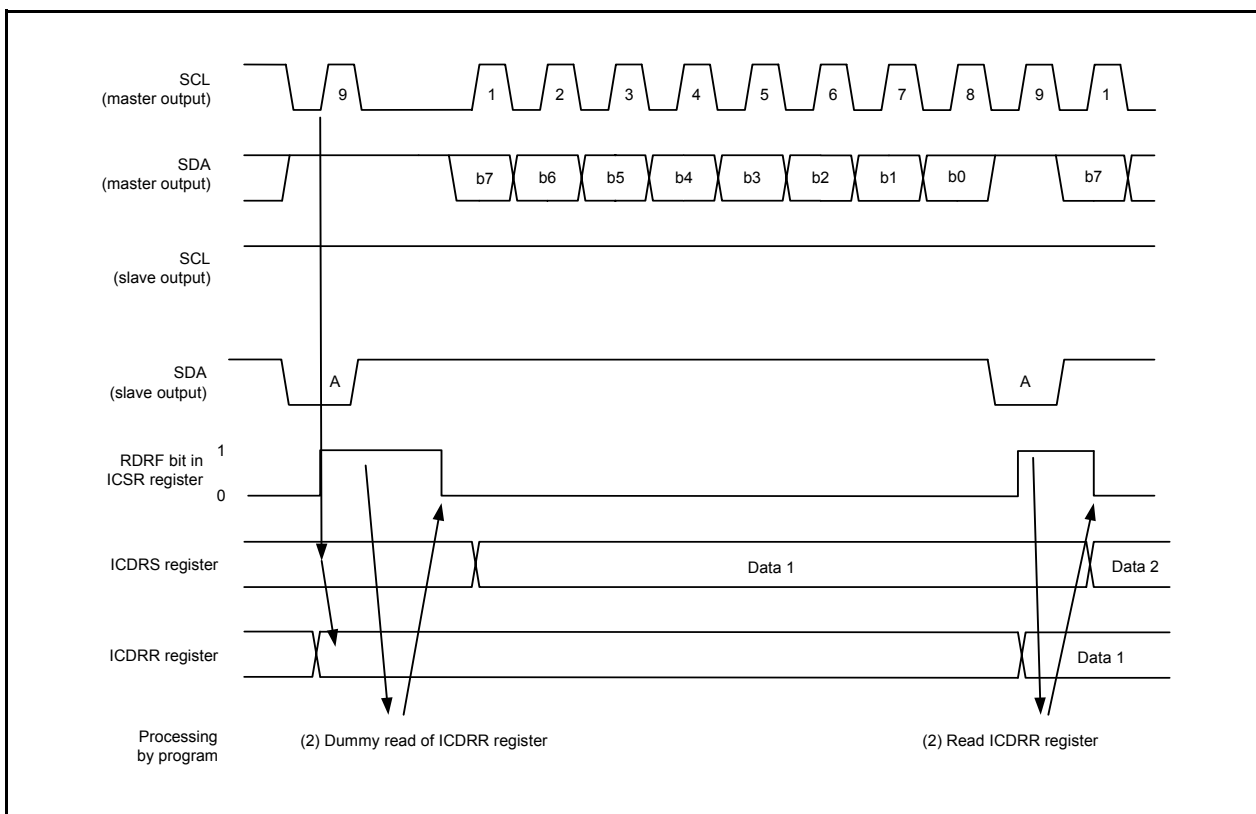


Figure 16.41 Operating Timing in Slave Receive Mode (I²C bus Interface Mode) (1)

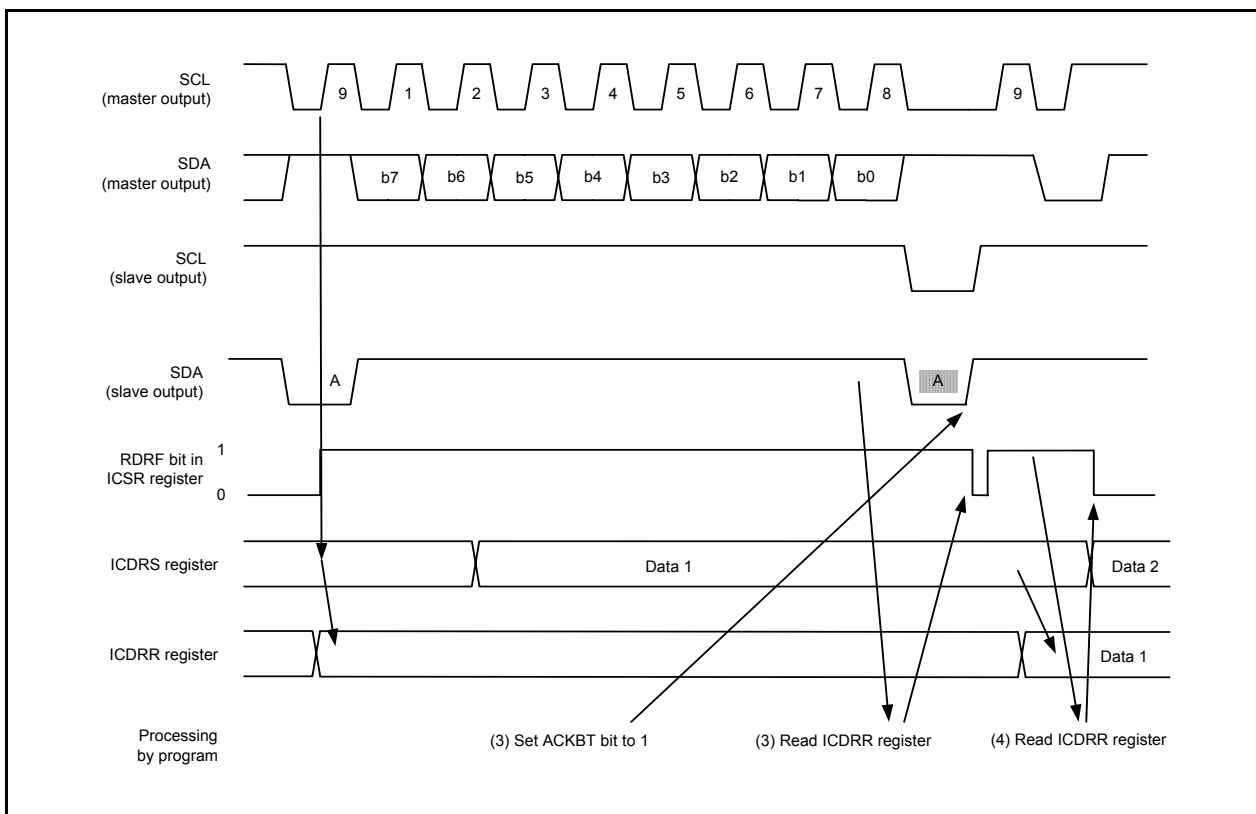


Figure 16.42 Operating Timing in Slave Receive Mode (I²C bus Interface Mode) (2)

16.3.4 Clock Synchronous Serial Mode

16.3.4.1 Clock Synchronous Serial Format

Set the FS bit in the SAR register to 1 to use the clock synchronous serial format for communication. Figure 16.43 shows the Transfer Format of Clock Synchronous Serial Format.

When the MST bit in the ICCR1 register is set to 1, the transfer clock is output from the SCL pin, and when the MST bit is set to 0, the external clock is input.

The transfer data is output between successive falling edges of the SCL clock, and data is determined at the rising edge of the SCL clock. MSB-first or LSB-first can be selected as the order of the data transfer by setting the MLS bit in the ICMR register. The SDA output level can be changed by the SDAO bit in the ICCR2 register during transfer standby.

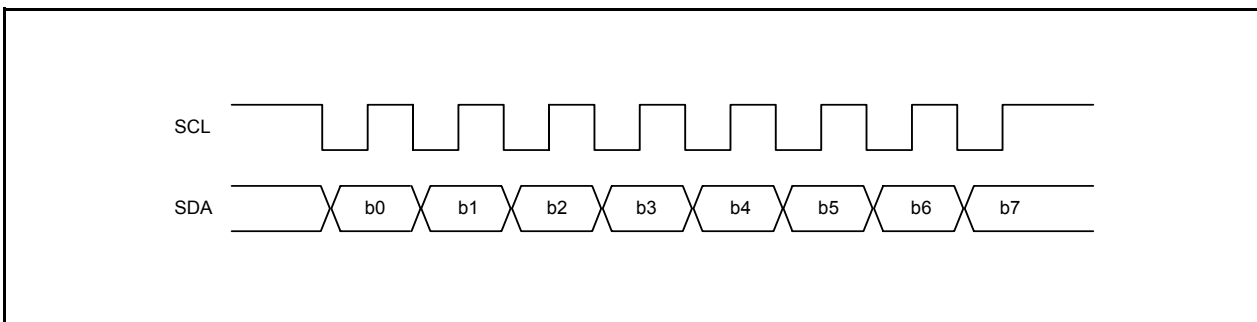


Figure 16.43 Transfer Format of Clock Synchronous Serial Format

16.3.4.2 Transmit Operation

In transmit mode, transmit data is output from the SDA pin in synchronization with the falling edge of the transfer clock. The transfer clock is output when the MST bit in the ICCR1 register is set to 1 and input when the MST bit is set to 0.

Figure 16.44 shows the Operating Timing in Transmit Mode (Clock Synchronous Serial Mode).

The transmit procedure and operation in transmit mode are as follows.

- (1) Set the ICE bit in the ICCR1 register to 1 (transfer operation enabled). Set bits CKS0 to CKS3 in the ICCR1 register and set the MST bit (initial setting).
- (2) The TDRE bit in the ICSR register is set to 1 by selecting transmit mode after setting the TRS bit in the ICCR1 register to 1.
- (3) Data is transferred from registers ICDRT to ICDRS and the TDRE bit is automatically set to 1 by writing transmit data to the ICDRT register after confirming that the TDRE bit is set to 1. Continuous transmission is enabled by writing data to the ICDRT register every time the TDRE bit is set to 1. When switching from transmit to receive mode, set the TRS bit to 0 while the TDRE bit is set to 1.

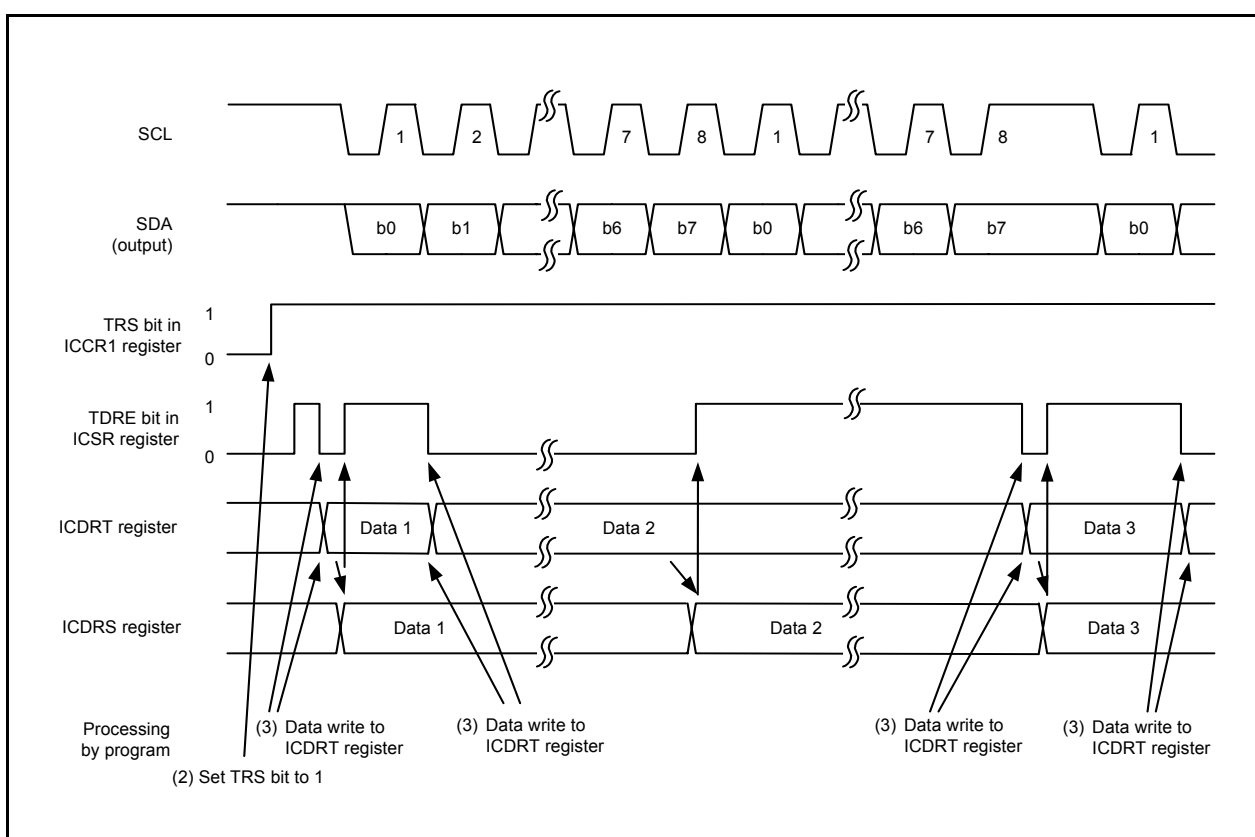


Figure 16.44 Operating Timing in Transmit Mode (Clock Synchronous Serial Mode)

16.3.4.3 Receive Operation

In receive mode, data is latched at the rising edge of the transfer clock. The transfer clock is output when the MST bit in the ICCR1 register is set to 1 and input when the MST bit is set to 0.

Figure 16.45 shows the Operating Timing in Receive Mode (Clock Synchronous Serial Mode).

The receive procedure and operation in receive mode are as follows.

- (1) Set the ICE bit in the ICCR1 register to 1 (transfer operation enabled). Set bits CKS0 to CKS3 in the ICCR1 register and set the MST bit (initial setting).
- (2) The output of the receive clock starts when the MST bit is set to 1 while the transfer clock is being output.
- (3) Data is transferred from registers ICDRS to ICDRR and the RDRF bit in the ICSR register is set to 1, when the receive operation is completed. Since the next byte of data is enabled when the MST bit is set to 1, the clock is output continuously. Continuous reception is enabled by reading the ICDRR register every time the RDRF bit is set to 1. An overrun is detected at the rise of the 8th clock cycle while the RDRF bit is set to 1, and the AL bit in the ICSR register is set to 1. At this time, the last receive data is retained in the ICDRR register.
- (4) When the MST bit is set to 1, set the RCVD bit in the ICCR1 register to 1 (disables the next receive operation) and read the ICDRR register. The SCL signal is fixed “H” after reception of the following byte of data is completed.

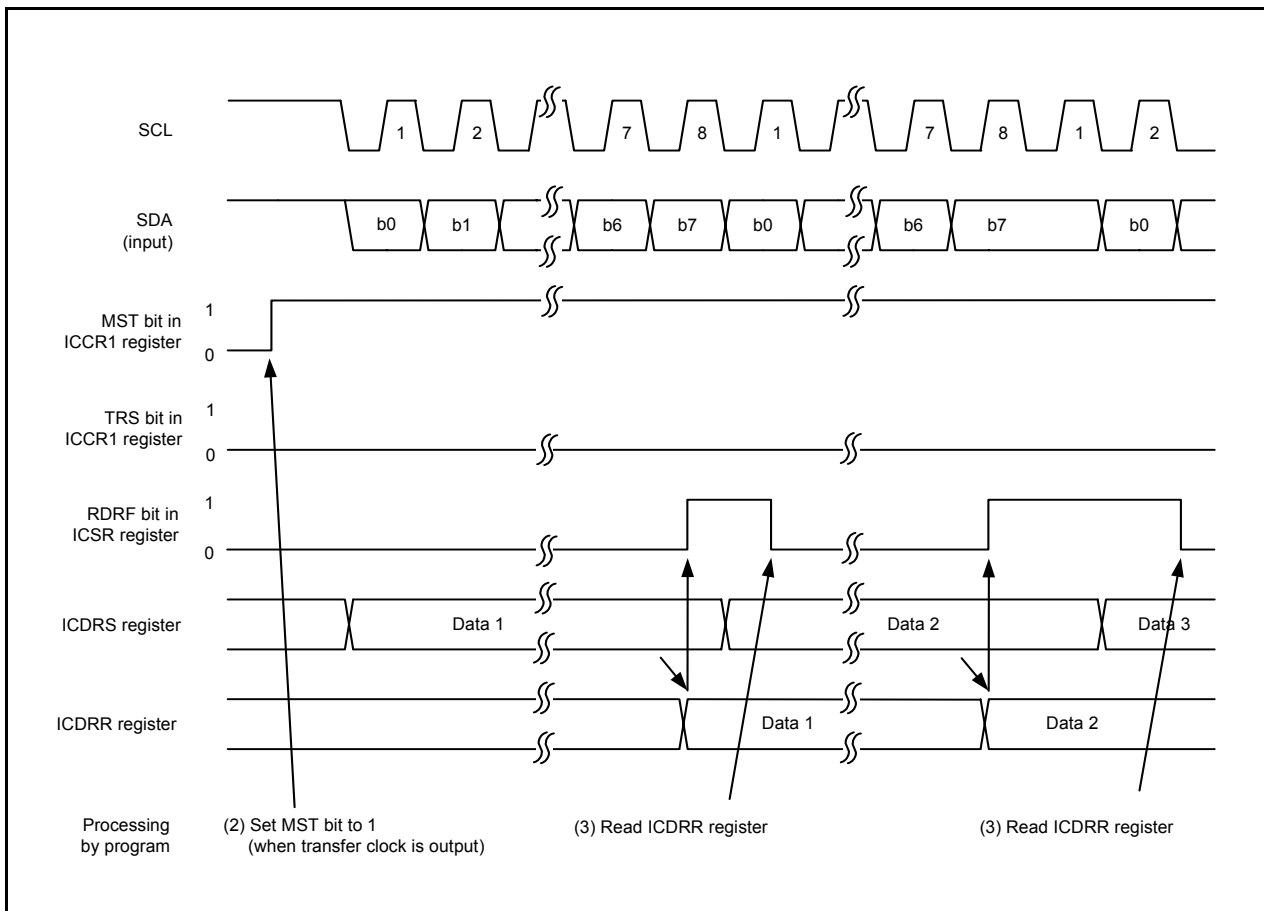


Figure 16.45 Operating Timing in Receive Mode (Clock Synchronous Serial Mode)

16.3.5 Examples of Register Setting

Figures 16.46 to 16.49 show Examples of Register Setting When Using I²C bus interface.

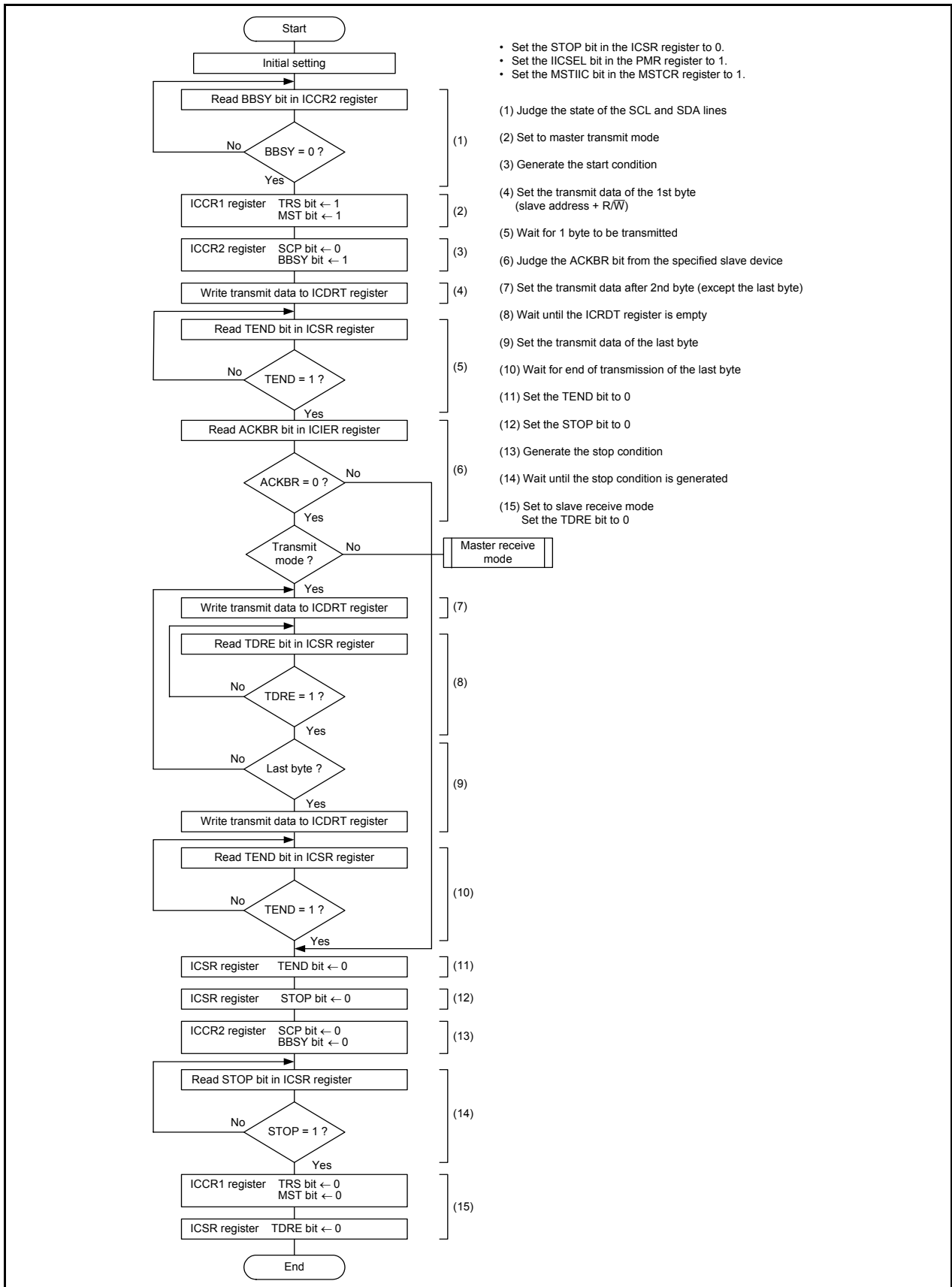


Figure 16.46 Example of Register Setting in Master Transmit Mode (I²C bus Interface Mode)

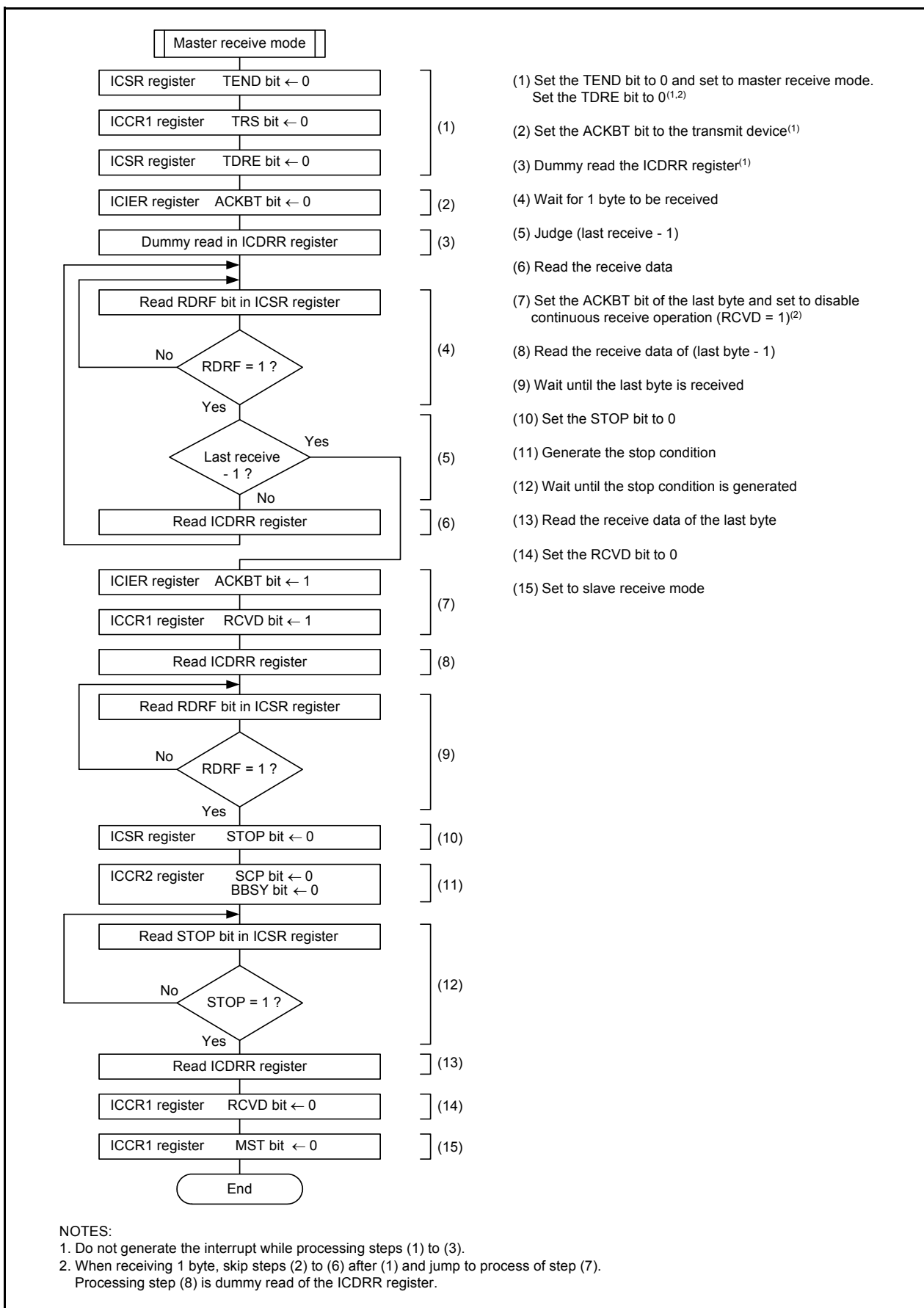


Figure 16.47 Example of Register Setting in Master Receive Mode (I²C bus Interface Mode)

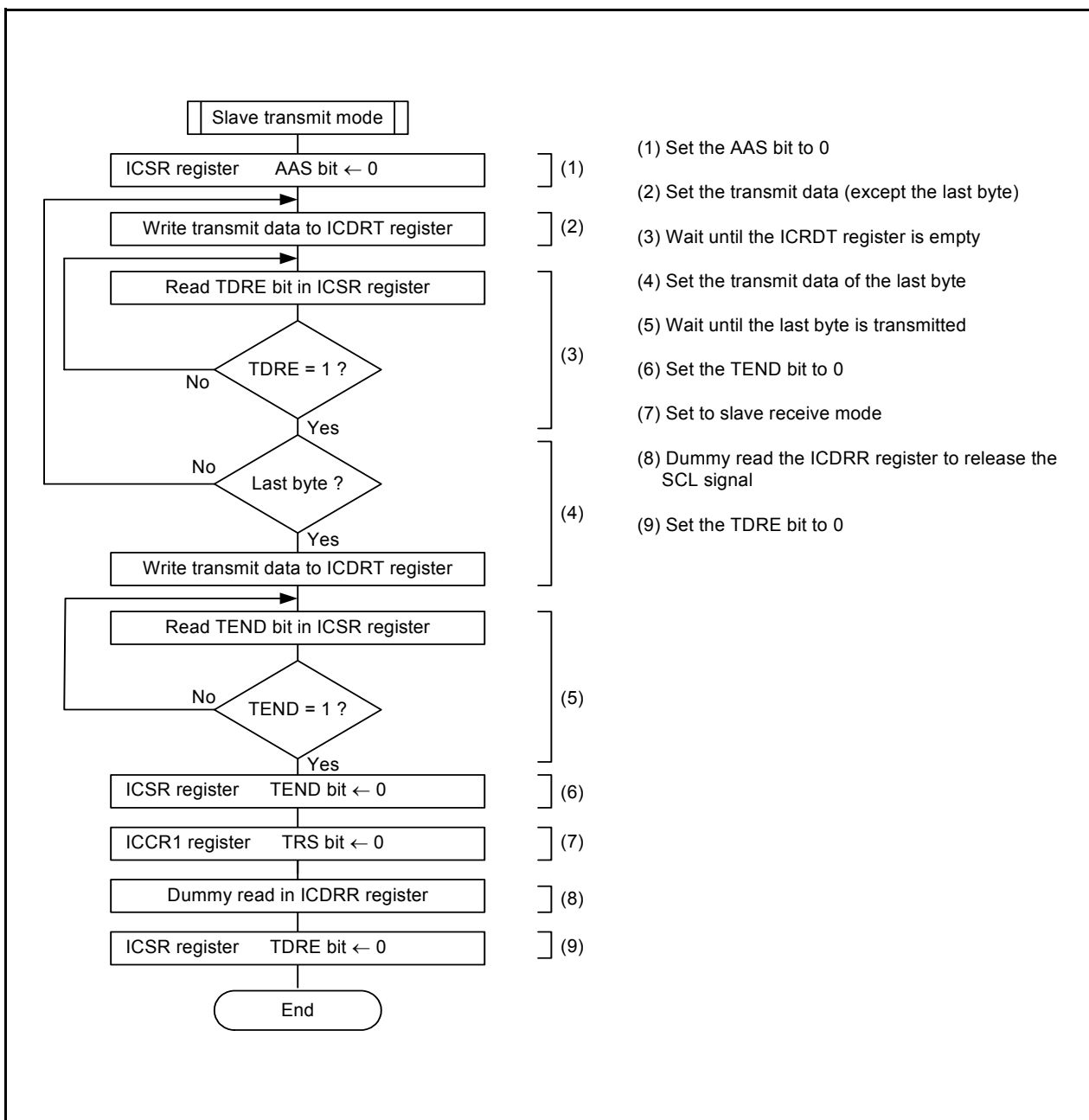


Figure 16.48 Example of Register Setting in Slave Transmit Mode (I²C bus Interface Mode)

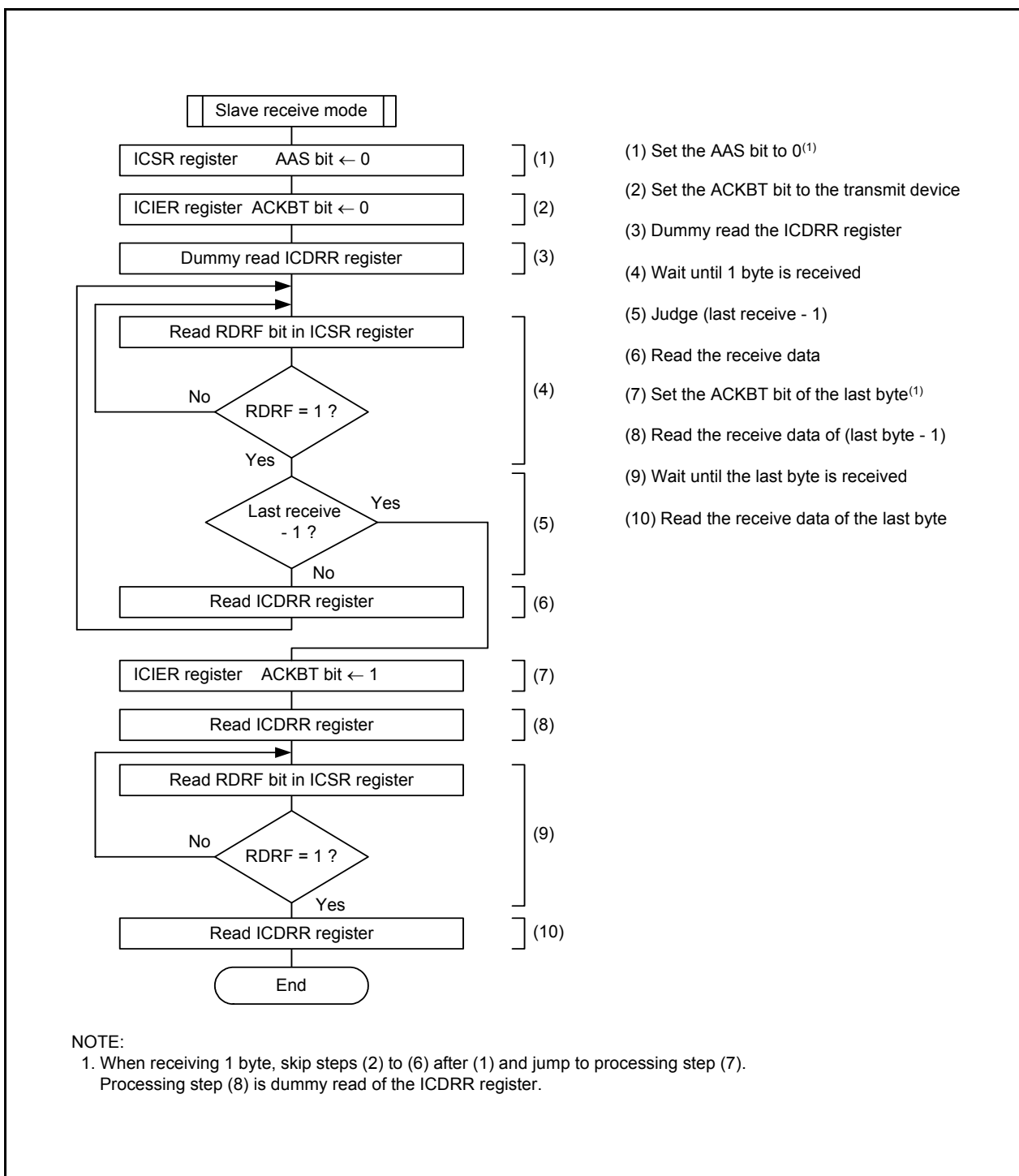


Figure 16.49 Example of Register Setting in Slave Receive Mode (I²C bus Interface Mode)

16.3.6 Noise Canceller

The states of pins SCL and SDA are routed through the noise canceller before being latched internally. Figure 16.50 shows a Block Diagram of Noise Canceller.

The noise canceller consists of two cascaded latch and match detector circuits. When the SCL pin input signal (or SDA pin input signal) is sampled on f1 and two latch outputs match, the level is passed forward to the next circuit. When they do not match, the former value is retained.

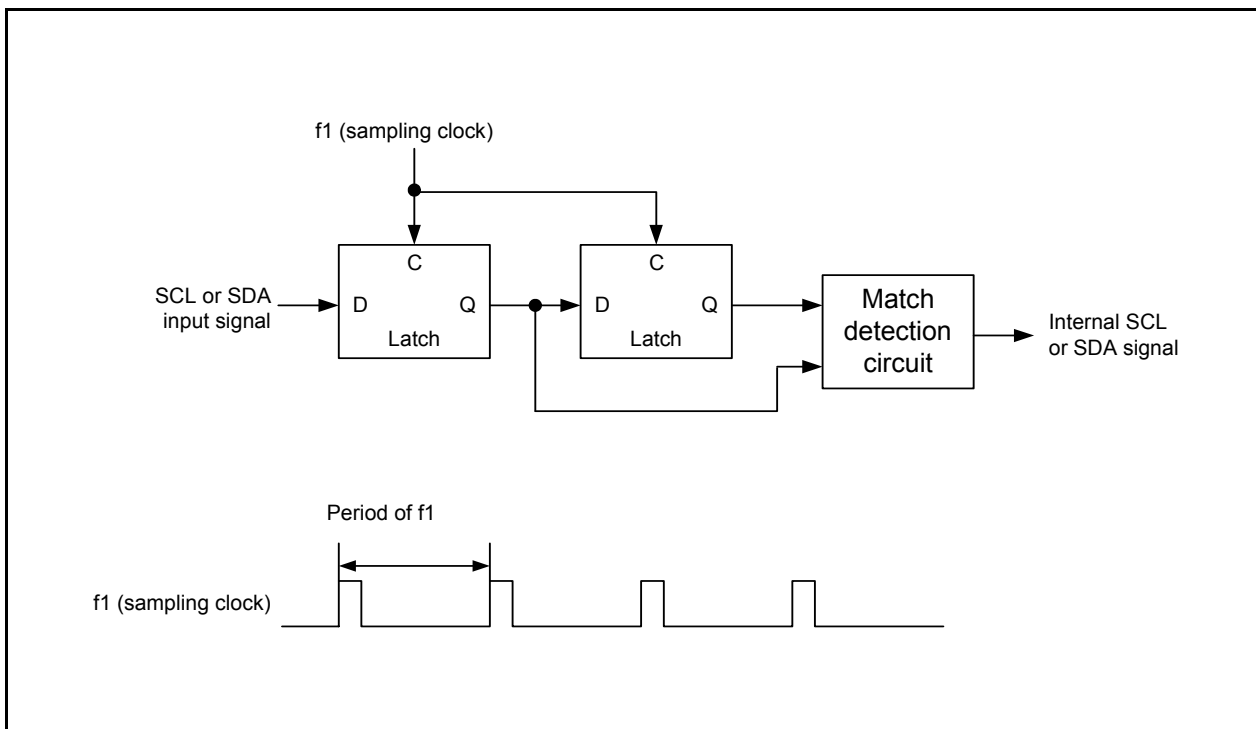


Figure 16.50 Block Diagram of Noise Canceller

16.3.7 Bit Synchronization Circuit

When setting the I²C bus interface to master mode, the high-level period may become shorter in the following two cases:

- If the SCL signal is driven L level by a slave device
- If the rise speed of the SCL signal is reduced by a load (load capacity or pull-up resistor) on the SCL line.

Therefore, the SCL signal is monitored and communication is synchronized bit by bit.

Figure 16.51 shows the Timing of Bit Synchronization Circuit and Table 16.8 lists the Time between Changing SCL Signal from “L” Output to High-Impedance and Monitoring of SCL Signal.

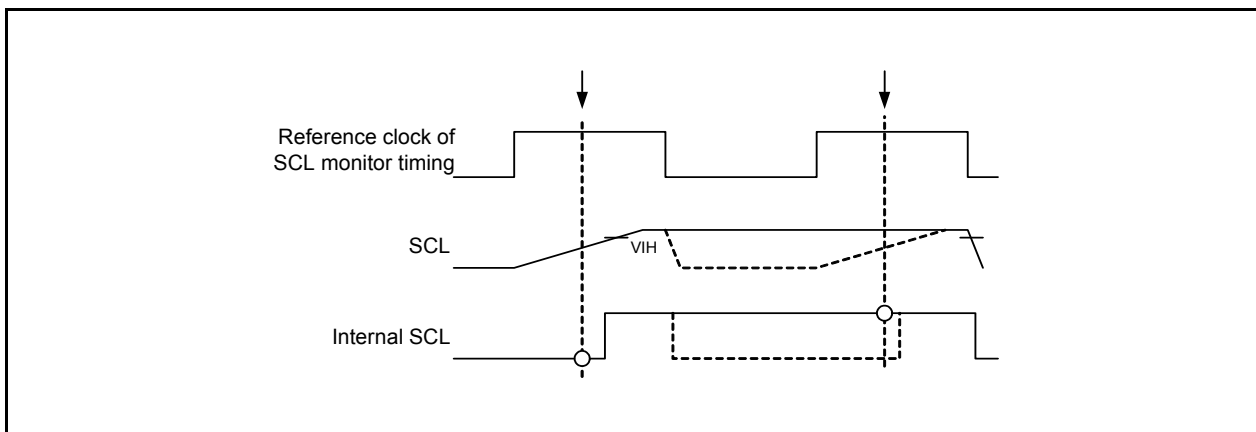


Figure 16.51 Timing of Bit Synchronization Circuit

Table 16.8 Time between Changing SCL Signal from “L” Output to High-Impedance and Monitoring of SCL Signal

| ICCR1 Register | | Time for Monitoring SCL |
|----------------|------|-------------------------|
| CKS3 | CKS2 | |
| 0 | 0 | 7.5Tcyc |
| | 1 | 19.5Tcyc |
| 1 | 0 | 17.5Tcyc |
| | 1 | 41.5Tcyc |

1Tcyc = 1/f1(s)

16.3.8 Notes on I²C bus Interface

Set the IICSEL bit in the PMR register to 1 (select I²C bus interface function) to use the I²C bus interface.

16.3.8.1 Multimaster Operation

The following actions must be performed to use the I²C bus interface in multimaster operation.

- Transfer rate
Set the transfer rate by 1/1.8 or faster than the fastest rate of the other masters. For example, if the fastest transfer rate of the other masters is set to 400 kbps, the I²C-bus transfer rate in this MCU should be set to 223 kbps (= 400/1.8) or more.
- Bits MST and TRS in the ICCR1 register setting
 - (a) Use the MOV instruction to set bits MST and TRS.
 - (b) When arbitration is lost, confirm the contents of bits MST and TRS. If the contents are other than the MST bit set to 0 and the TRS bit set to 0 (slave receive mode), set the MST bit to 0 and the TRS bit to 0 again.

16.3.8.2 Master Receive Mode

Either of the following actions must be performed to use the I²C bus interface in master receive mode.

- (a) In master receive mode while the RDRF bit in the ICSR register is set to 1, read the ICDRR register before the rising edge of the 8th clock.
- (b) In master receive mode, set the RCVD bit in the ICCR1 register to 1 (disables the next receive operation) to perform 1-byte communications.

17. Hardware LIN

The hardware LIN performs LIN communication in cooperation with timer RA and UART0.

17.1 Features

The hardware LIN has the features listed below.

Figure 17.1 shows a Block Diagram of Hardware LIN.

Master mode

- Generates Synch Break
- Detects bus collision

Slave mode

- Detects Synch Break
- Measures Synch Field
- Controls Synch Break and Synch Field signal inputs to UART0
- Detects bus collision

NOTE:

1. The WakeUp function is detected by INT1.

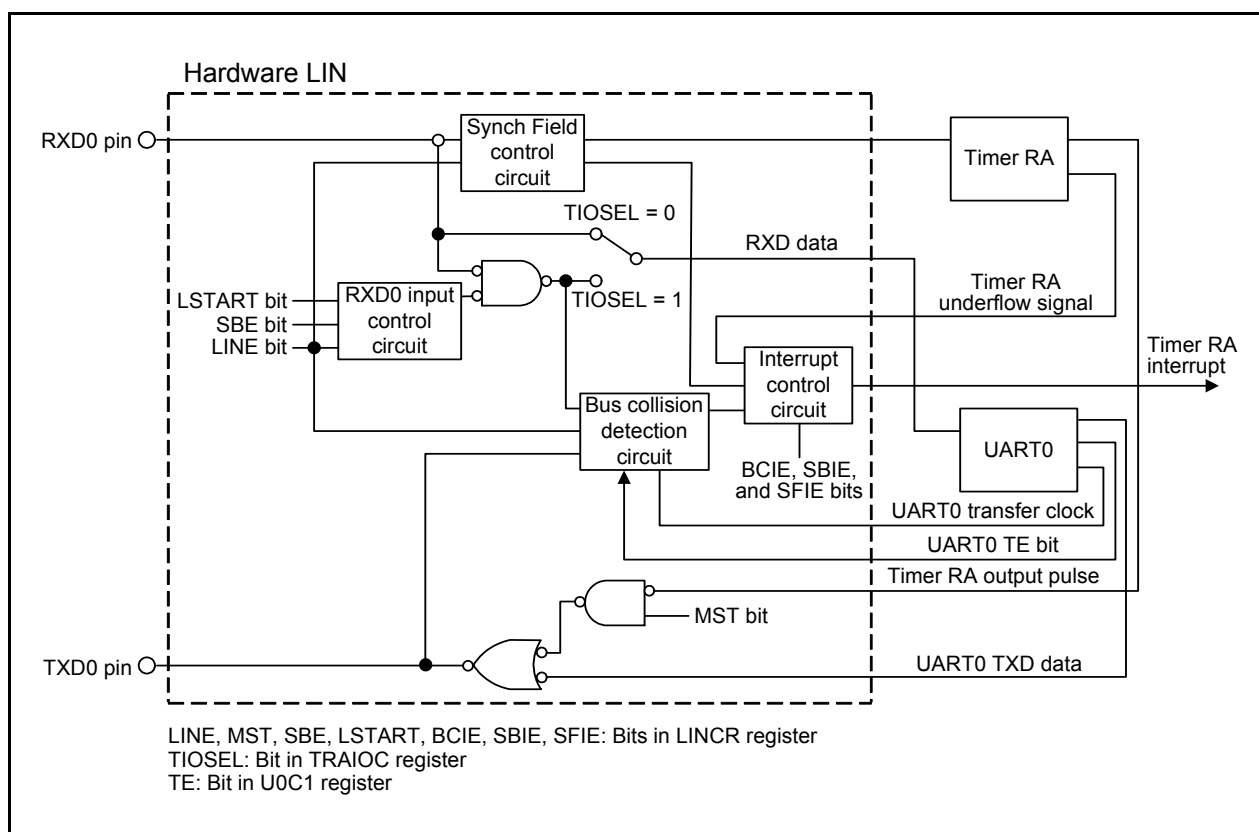


Figure 17.1 Block Diagram of Hardware LIN

17.2 Input/Output Pins

The pin configuration of the hardware LIN is listed in Table 17.1.

Table 17.1 Pin Configuration

| Name | Abbreviation | Input/Output | Function |
|----------------------|--------------|--------------|--|
| Receive data input | RXD0 | Input | Receive data input pin of the hardware LIN |
| Transmit data output | TXD0 | Output | Transmit data output pin of the hardware LIN |

17.3 Register Configuration

The hardware LIN contains the registers listed below.
These registers are detailed in Figures 17.2 and 17.3.

- LIN Control Register 2 (LINCR2)
- LIN Control Register (LINCR)
- LIN Status Register (LINST)

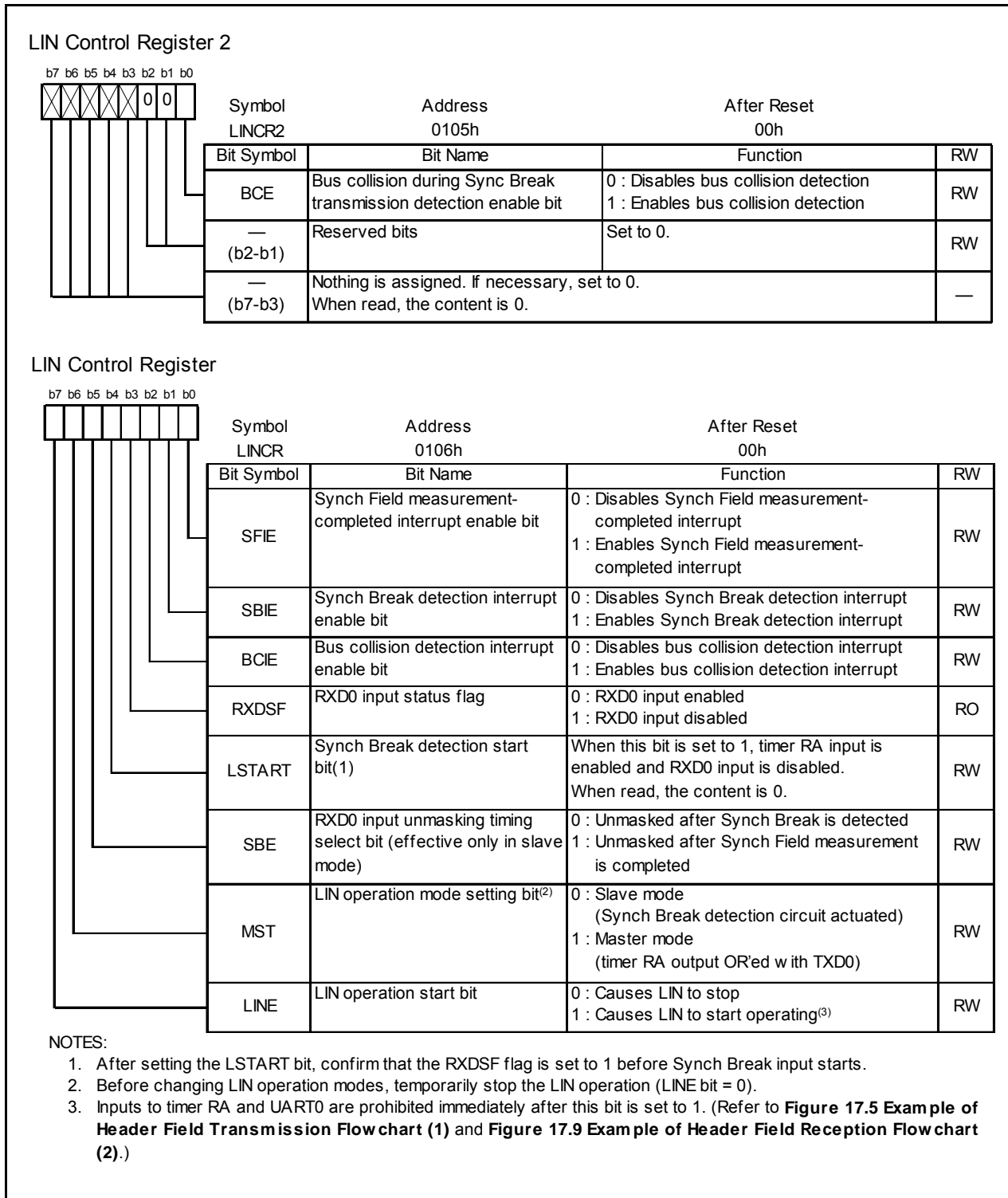


Figure 17.2 Registers LINCR2 and LINCR

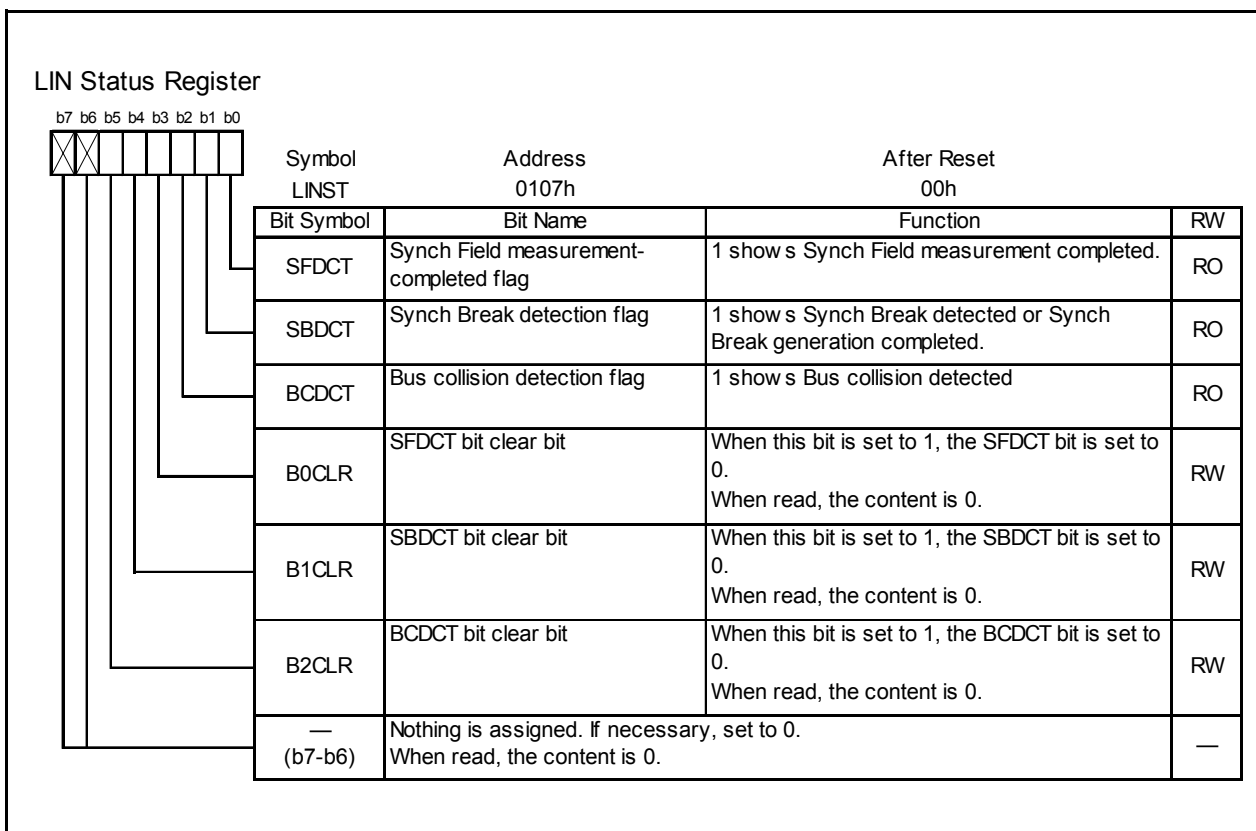


Figure 17.3 LINST Register

17.4 Functional Description

17.4.1 Master Mode

Figure 17.4 shows typical operation of the hardware LIN when transmitting a header field in master mode. Figures 17.5 and 17.6 show an Example of Header Field Transmission Flowchart.

When transmitting a header field, the hardware LIN operates as described below.

- (1) When the TSTART bit in the TRACR register for timer RA is set by writing 1 in software, the hardware LIN outputs “L” level from the TXD0 pin for the period that is set in registers TRAPRE and TRA for timer RA.
- (2) When timer RA underflows upon reaching the terminal count, the hardware LIN reverses the output of the TXD0 pin and sets the SBDCT flag in the LINST register to 1. Furthermore, if the SBIE bit in the LINCR register is set to 1, it generates a timer RA interrupt.
- (3) The hardware LIN transmits 55h via UART0.
- (4) The hardware LIN transmits an ID field via UART0 after it finishes sending 55h.
- (5) The hardware LIN performs communication for a response field after it finishes sending the ID field.

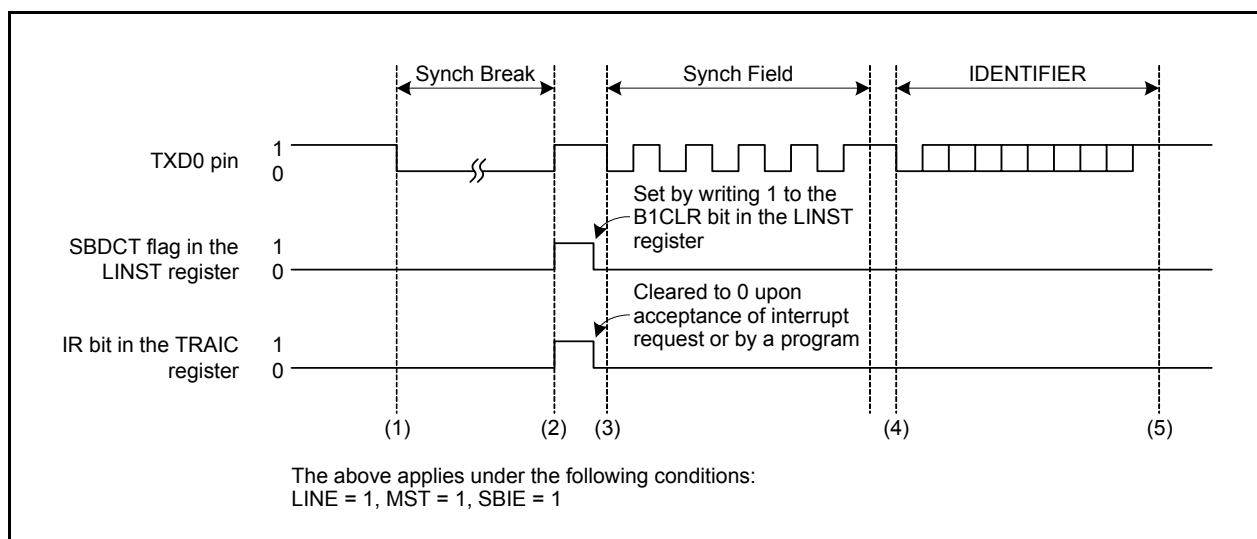


Figure 17.4 Typical Operation when Sending a Header Field

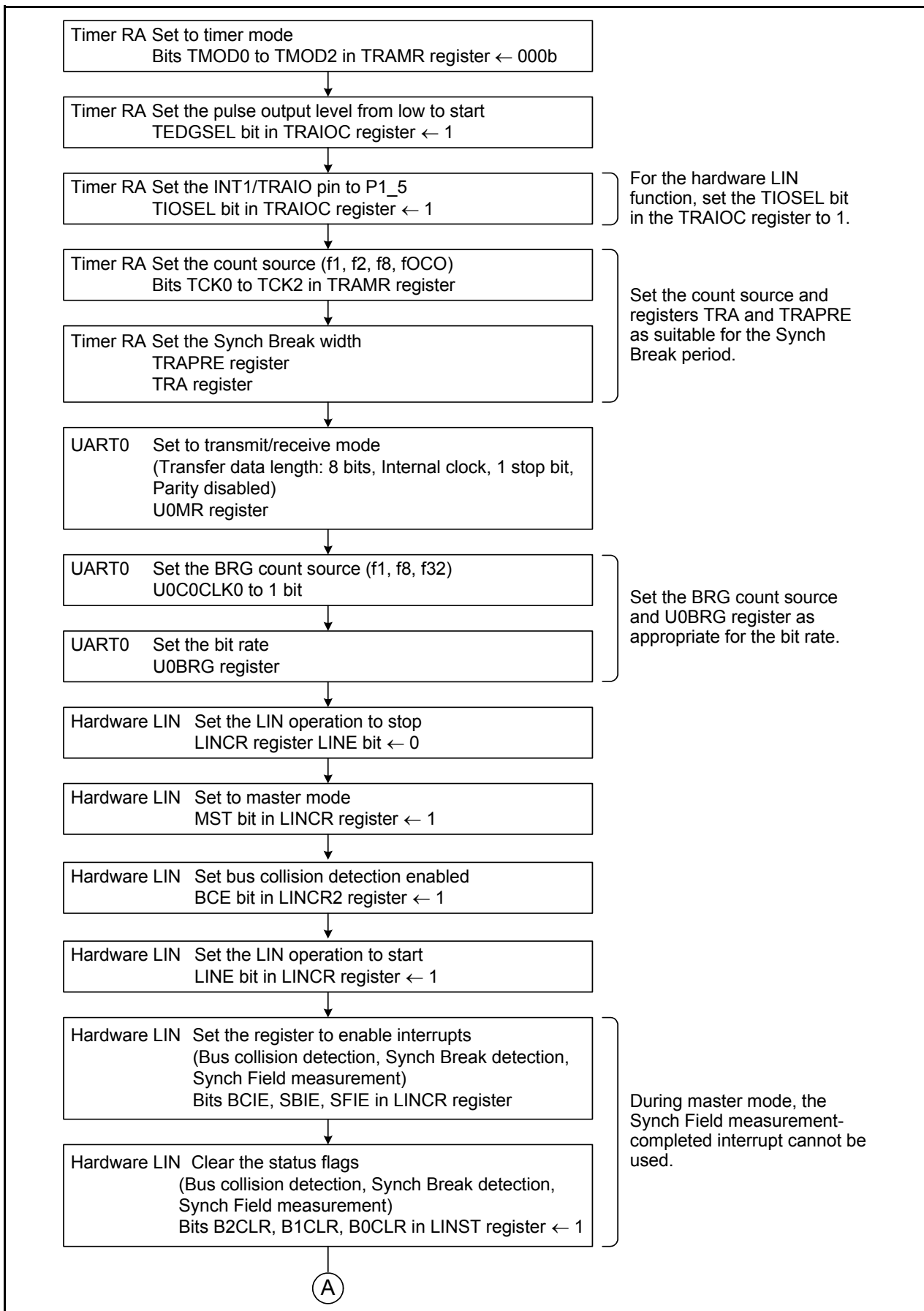


Figure 17.5 Example of Header Field Transmission Flowchart (1)

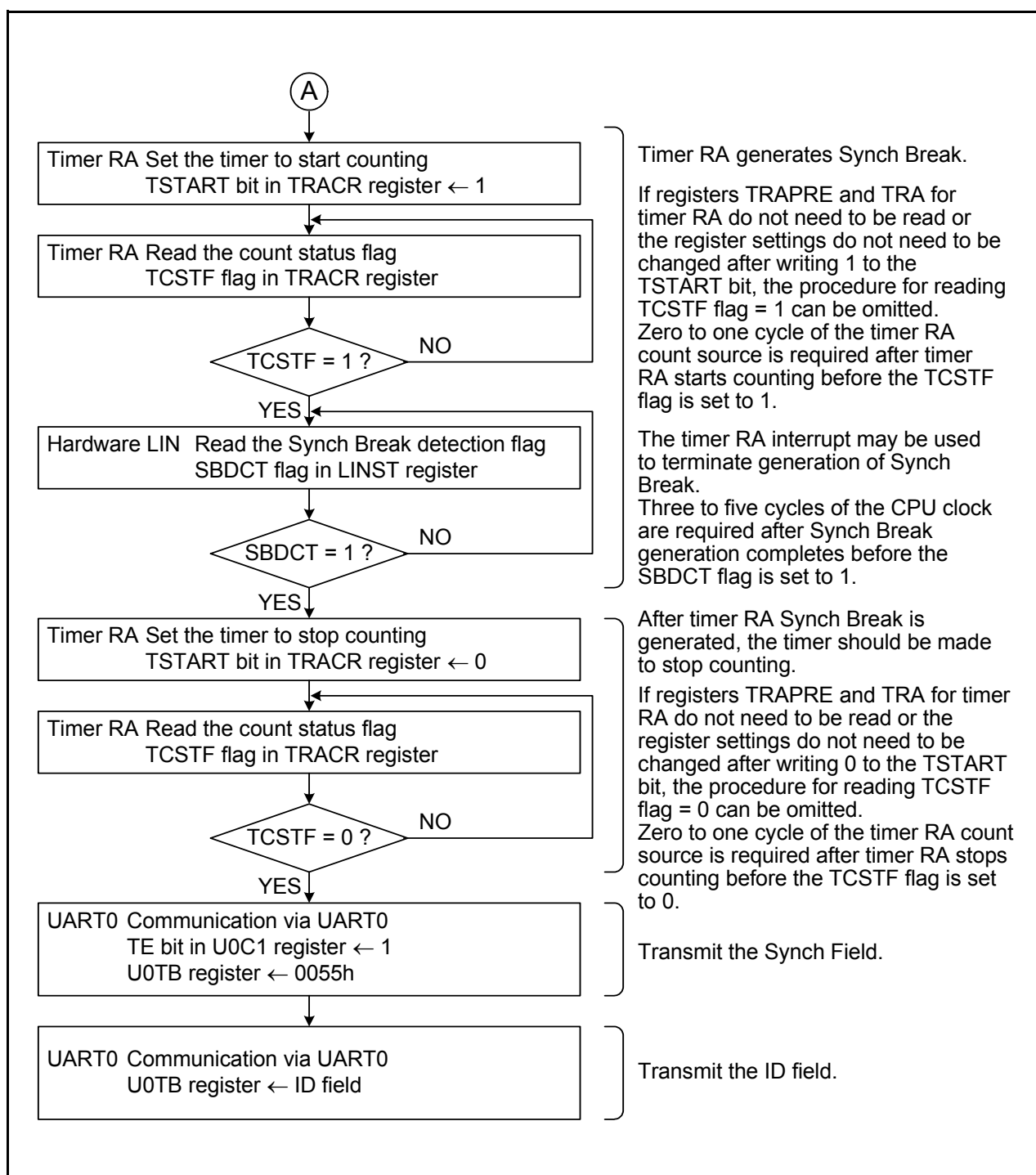


Figure 17.6 Example of Header Field Transmission Flowchart (2)

17.4.2 Slave Mode

Figure 17.7 shows typical operation of the hardware LIN when receiving a header field in slave mode. Figure 17.8 through Figure 17.10 show an Example of Header Field Reception Flowchart.

When receiving a header field, the hardware LIN operates as described below.

- (1) Synch Break detection is enabled by writing 1 to the LSTART bit in the LINCR register of the hardware LIN.
- (2) When “L” level is input for a duration equal to or greater than the period set in timer RA, the hardware LIN detects it as Synch Break. At this time, the SBDCT flag in the LINST register is set to 1. Furthermore, if the SBIE bit in the LINCR register is set to 1, the hardware LIN generates a timer RA interrupt. Then it goes to Synch Field measurement.
- (3) The hardware LIN receives a Synch Field (55h). At this time, it measures the period of the start bit and bits 0 to 6 by using timer RA. In this case, it is possible to select whether to input the Synch Field signal to RXD0 of UART0 by setting the SBE bit in the LINCR register accordingly.
- (4) The hardware LIN sets the SFDCT flag in the LINST register to 1 when it finishes measuring the Synch Field. Furthermore, if the SFIE bit in the LINCR register is set to 1, it generates a timer RA interrupt.
- (5) After it finishes measuring the Synch Field, calculate a transfer rate from the count value of timer RA and set to UART0 and registers TRAPRE and TRA of timer RA again.
- (6) The hardware LIN performs communication for a response field after it finishes receiving the ID field.

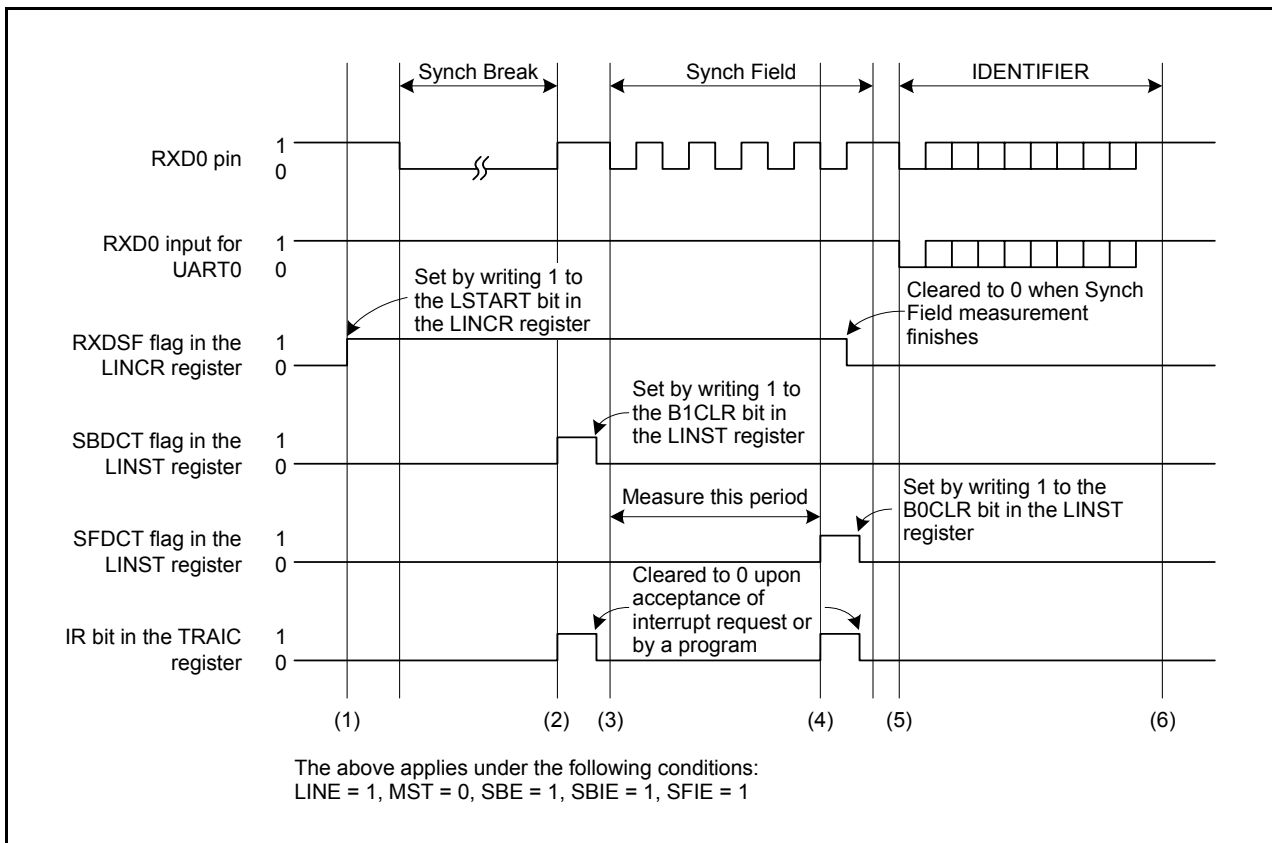


Figure 17.7 Typical Operation when Receiving a Header Field

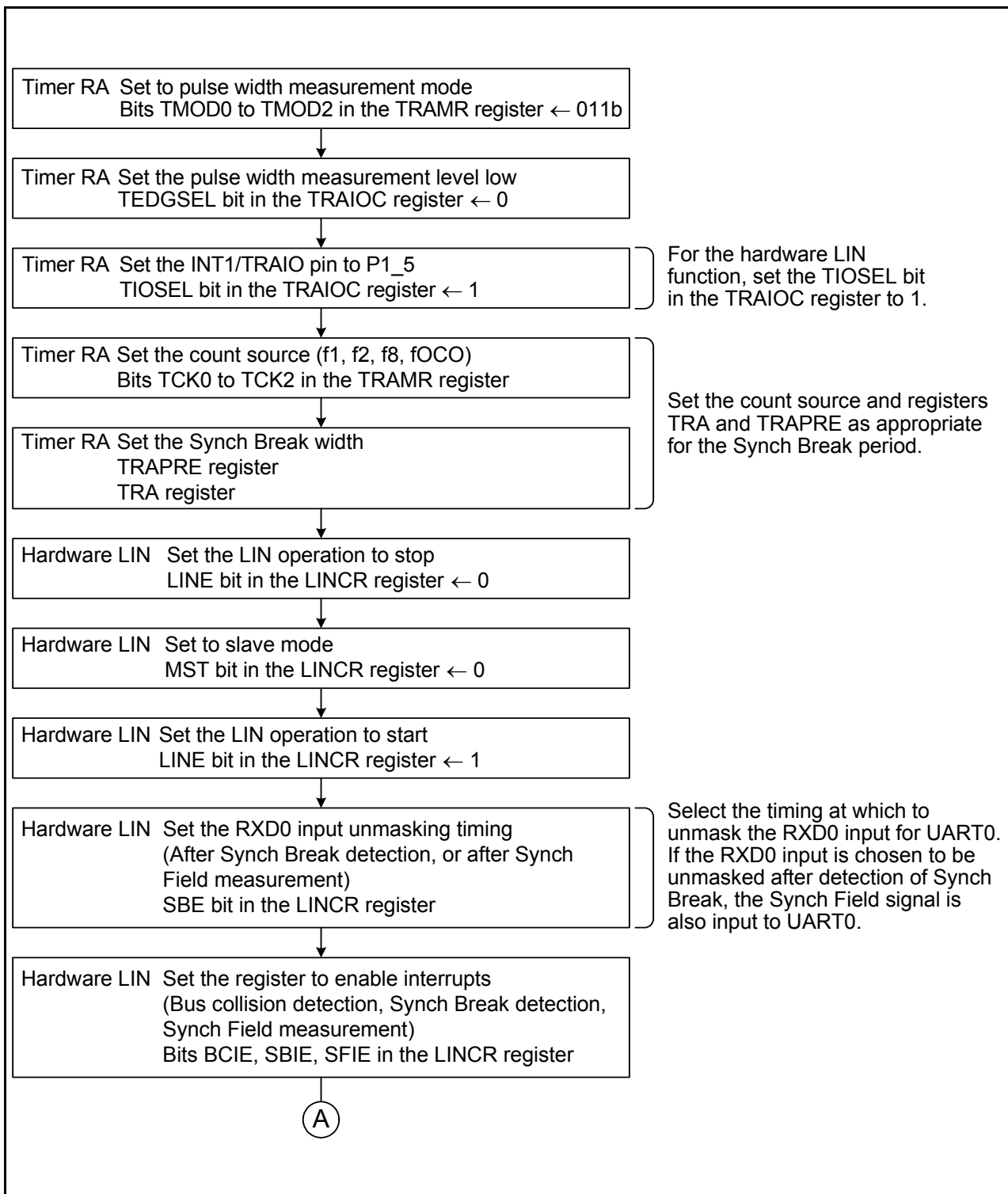


Figure 17.8 Example of Header Field Reception Flowchart (1)

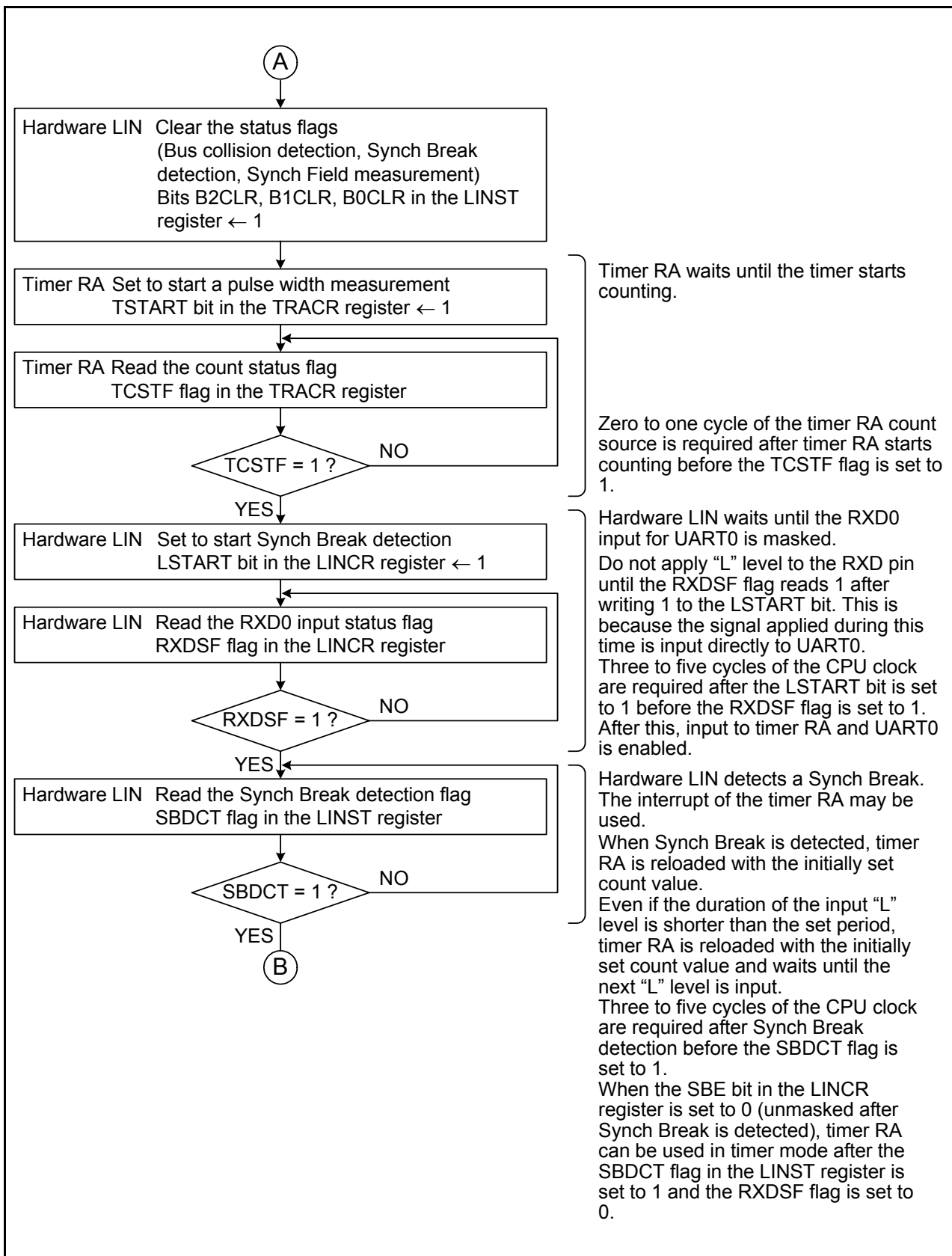


Figure 17.9 Example of Header Field Reception Flowchart (2)

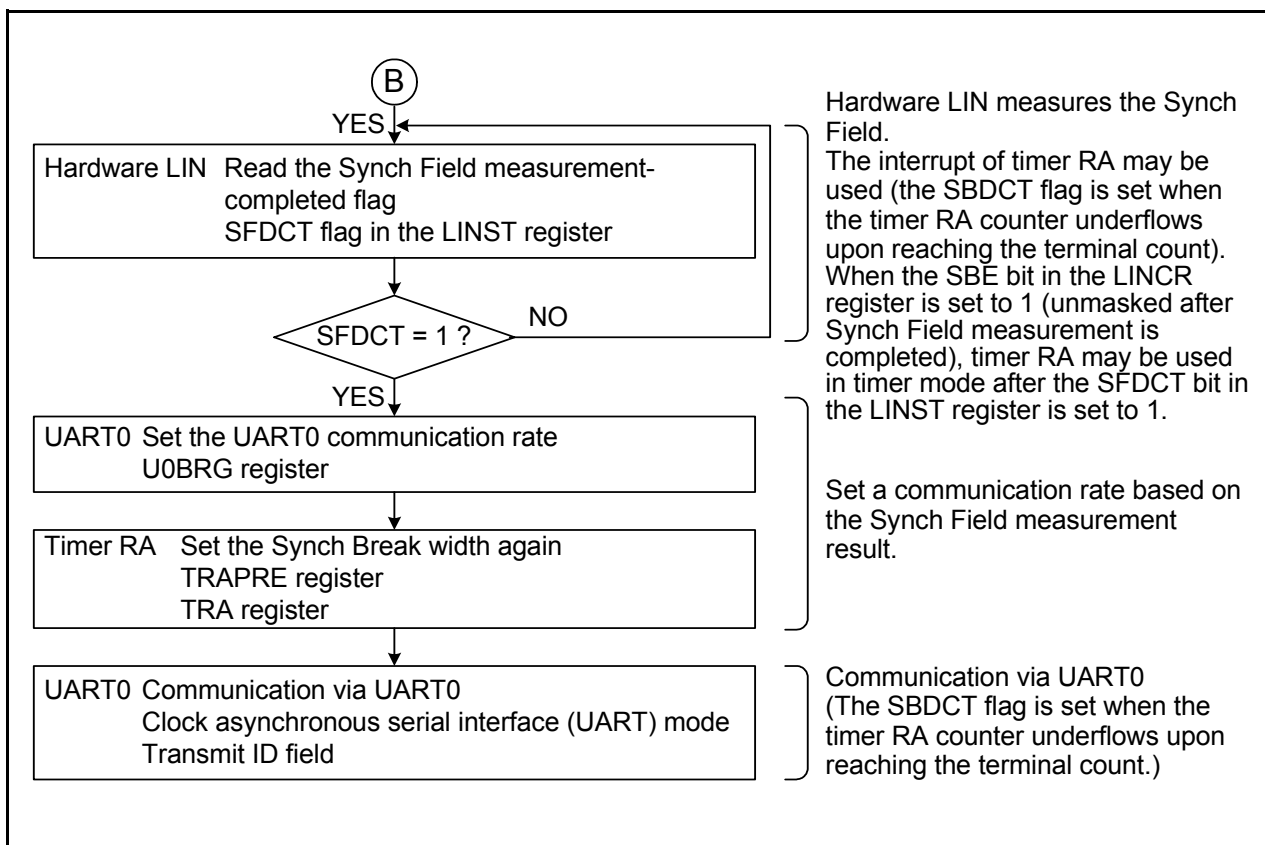


Figure 17.10 Example of Header Field Reception Flowchart (3)

17.4.3 Bus Collision Detection Function

The bus collision detection function can be used when UART0 is enabled for transmission (TE bit in the U0C1 register = 1). To detect a bus collision during Synch Break transmission, set the BCE bit in the LINCR2 register to 1 (bus collision detection enabled).

Figure 17.11 shows typical operation of the hardware LIN when a bus collision is detected.

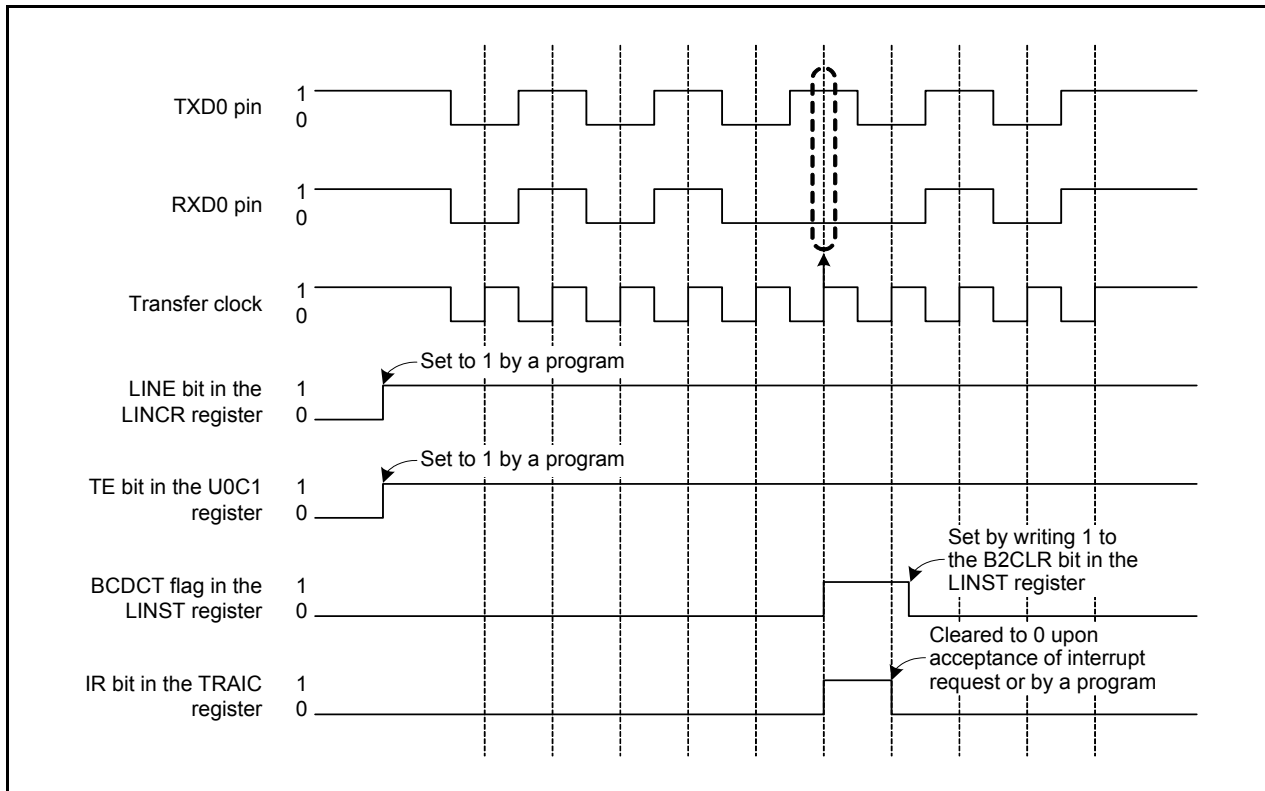


Figure 17.11 Typical Operation when a Bus Collision is Detected

17.4.4 Hardware LIN End Processing

Figure 17.12 shows an Example of Hardware LIN Communication Completion Flowchart. Use the following timing for hardware LIN end processing:

- If the hardware bus collision detection function is used
Perform hardware LIN end processing after checksum transmission completes.
- If the bus collision detection function is not used
Perform hardware LIN end processing after header field transmission and reception complete.

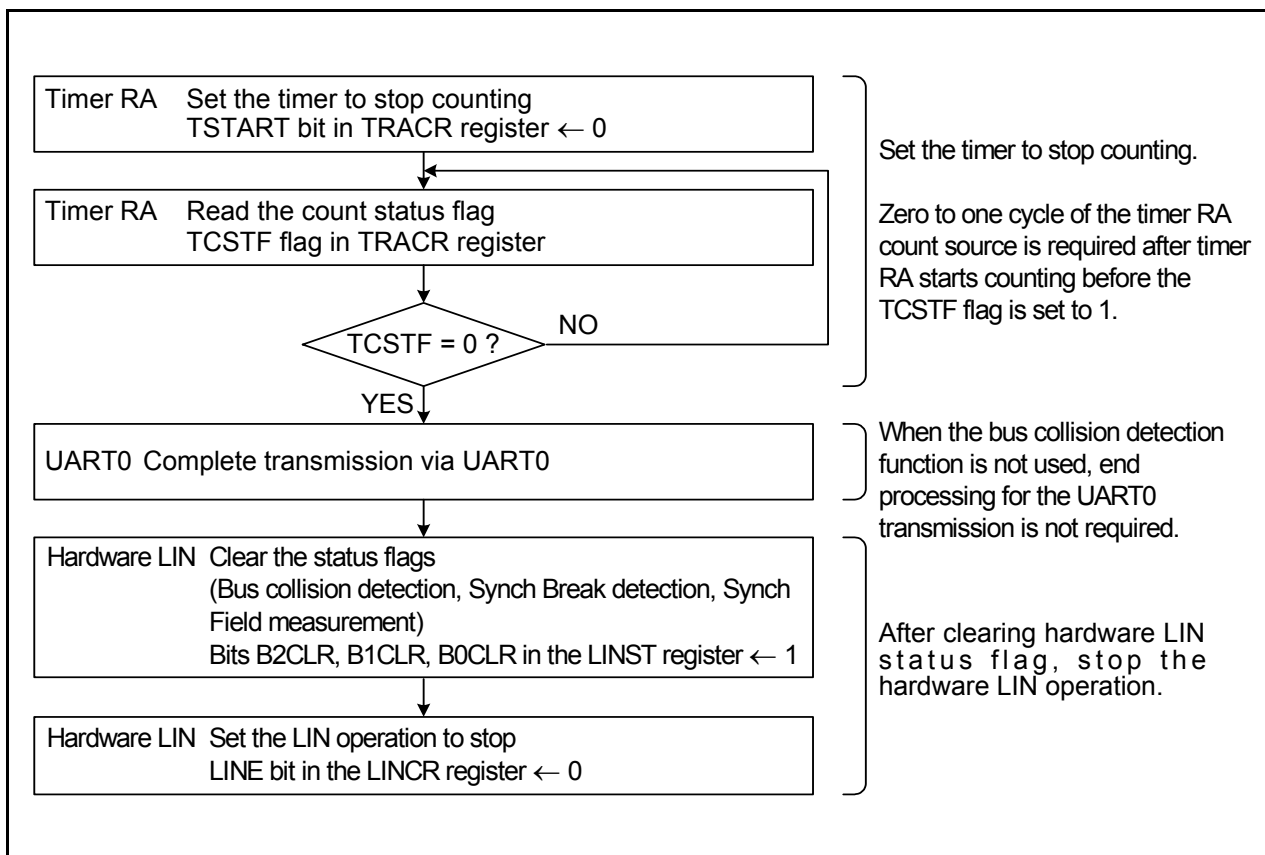


Figure 17.12 Example of Hardware LIN Communication Completion Flowchart

17.5 Interrupt Requests

There are four interrupt requests that are generated by the hardware LIN: Synch Break detection, Synch Break generation completed, Synch Field measurement completed, and bus collision detection. These interrupts are shared with timer RA.

Table 17.2 lists the Interrupt Requests of Hardware LIN.

Table 17.2 Interrupt Requests of Hardware LIN

| Interrupt Request | Status Flag | Cause of Interrupt |
|-----------------------------------|-------------|--|
| Synch Break detection | SBDCT | Generated when timer RA has underflowed after measuring the "L" level duration of RXD0 input, or when a "L" level is input for a duration longer than the Synch Break period during communication. |
| Synch Break generation completed | | Generated when "L" level output to TXD0 for the duration set by timer RA completes. |
| Synch Field measurement completed | SFDCT | Generated when measurement for 6 bits of the Synch Field by timer RA is completed. |
| Bus collision detection | BCDCT | Generated when the RXD0 input and TXD0 output values differed at data latch timing while UART0 is enabled for transmission. |

17.6 Notes on Hardware LIN

For the time-out processing of the header and response fields, use another timer to measure the duration of time with a Synch Break detection interrupt as the starting point.

18. A/D Converter

The A/D converter consists of one 10-bit successive approximation A/D converter circuit with a capacitive coupling amplifier. The analog input shares pins P0_0 to P0_7, and P1_0 to P1_3. Therefore, when using these pins, ensure that the corresponding port direction bits are set to 0 (input mode).

When not using the A/D converter, set the VCUT bit in the ADCON1 register to 0 (VREF unconnected) so that no current will flow from the VREF pin into the resistor ladder. This helps to reduce the power consumption of the chip.

The result of A/D conversion is stored in the AD register.

Table 18.1 lists the Performance of A/D converter. Figure 18.1 shows a Block Diagram of A/D Converter.

Figures 18.2 and 18.4 show the A/D converter-related registers.

Table 18.1 Performance of A/D converter

| Item | Performance |
|--|--|
| A/D conversion method | Successive approximation (with capacitive coupling amplifier) |
| Analog input voltage ⁽¹⁾ | 0 V to AVCC |
| Operating clock ϕ_{AD} ⁽²⁾ | $4.2\text{ V} \leq AVCC \leq 5.5\text{ V}$ f1, f2, f4, fOCO-F $2.2\text{ V} \leq AVCC < 4.2\text{ V}$ f2, f4, fOCO-F |
| Resolution | 8 bits or 10 bits selectable |
| Absolute accuracy | $AVCC = V_{ref} = 5\text{ V}$, $\phi_{AD} = 10\text{ MHz}$ <ul style="list-style-type: none"> • 8-bit resolution $\pm 2\text{ LSB}$ • 10-bit resolution $\pm 3\text{ LSB}$ $AVCC = V_{ref} = 3.3\text{ V}$, $\phi_{AD} = 10\text{ MHz}$ <ul style="list-style-type: none"> • 8-bit resolution $\pm 2\text{ LSB}$ • 10-bit resolution $\pm 5\text{ LSB}$ $AVCC = V_{ref} = 2.2\text{ V}$, $\phi_{AD} = 5\text{ MHz}$ <ul style="list-style-type: none"> • 8-bit resolution $\pm 2\text{ LSB}$ • 10-bit resolution $\pm 5\text{ LSB}$ |
| Operating mode | One-shot mode and repeat mode 0 ⁽³⁾ |
| Analog input pin | 12 pins (AN0 to AN11) |
| A/D conversion start condition | <ul style="list-style-type: none"> • Software trigger Set the ADST bit in the ADCON0 register to 1 (A/D conversion starts) • Capture Timer RD interrupt request is generated while the ADST bit is set to 1 |
| Conversion rate per pin | <ul style="list-style-type: none"> • Without sample and hold function 8-bit resolution: $49\phi_{AD}$ cycles, 10-bit resolution: $59\phi_{AD}$ cycles • With sample and hold function 8-bit resolution: $28\phi_{AD}$ cycles, 10-bit resolution: $33\phi_{AD}$ cycles |

NOTES:

1. The analog input voltage does not depend on use of a sample and hold function.
When the analog input voltage is over the reference voltage, the A/D conversion result will be 3FFh in 10-bit mode and FFh in 8-bit mode.
2. When $2.7\text{ V} \leq AVCC \leq 5.5\text{ V}$, the frequency of ϕ_{AD} must be 10 MHz or below.
When $2.2\text{ V} \leq AVCC < 2.7\text{ V}$, the frequency of ϕ_{AD} must be 5 MHz or below.
Without a sample and hold function, the ϕ_{AD} frequency should be 250 kHz or above.
With a sample and hold function, the ϕ_{AD} frequency should be 1 MHz or above.
3. In repeat mode 0, only 8-bit mode can be used.

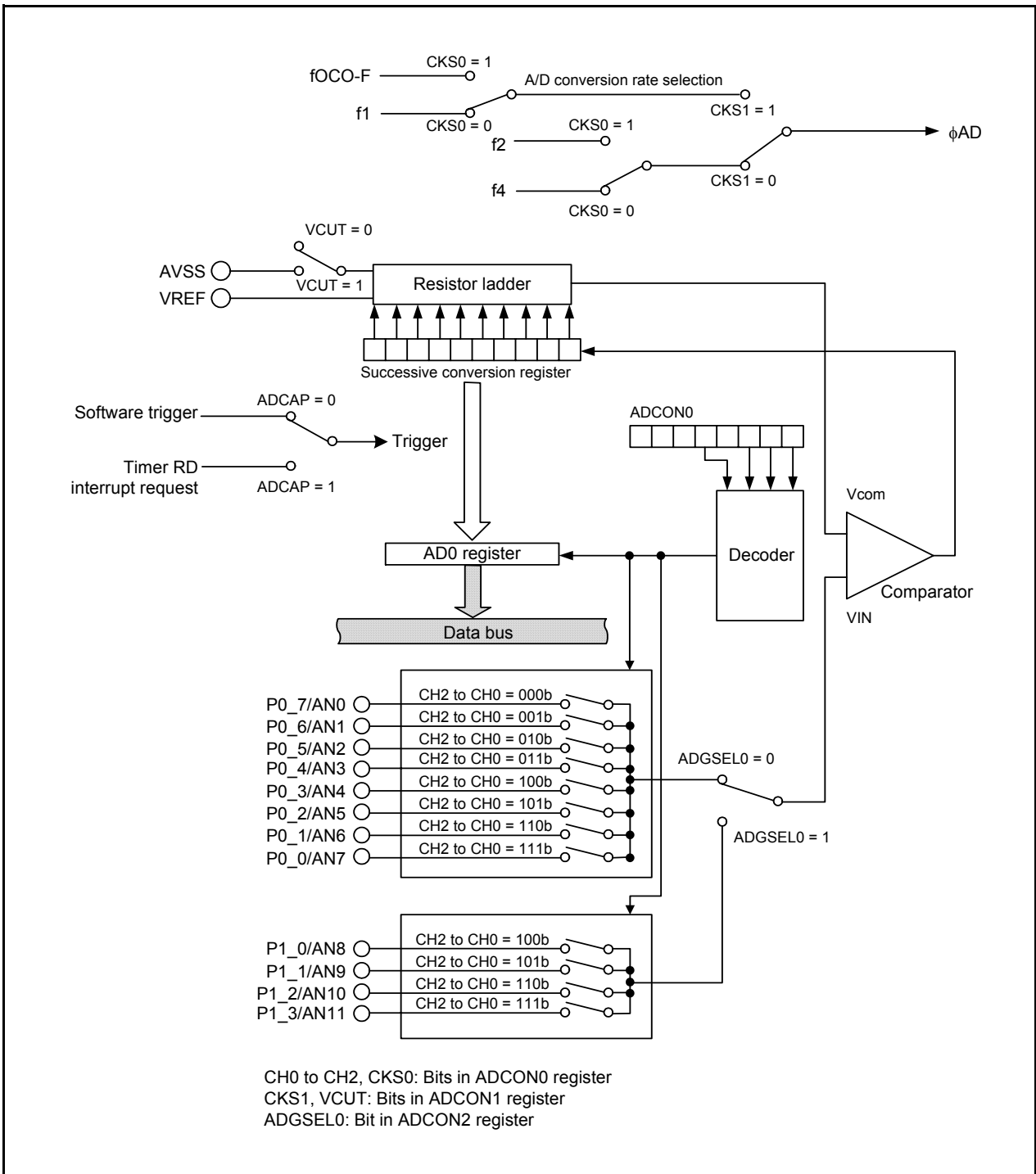


Figure 18.1 Block Diagram of A/D Converter

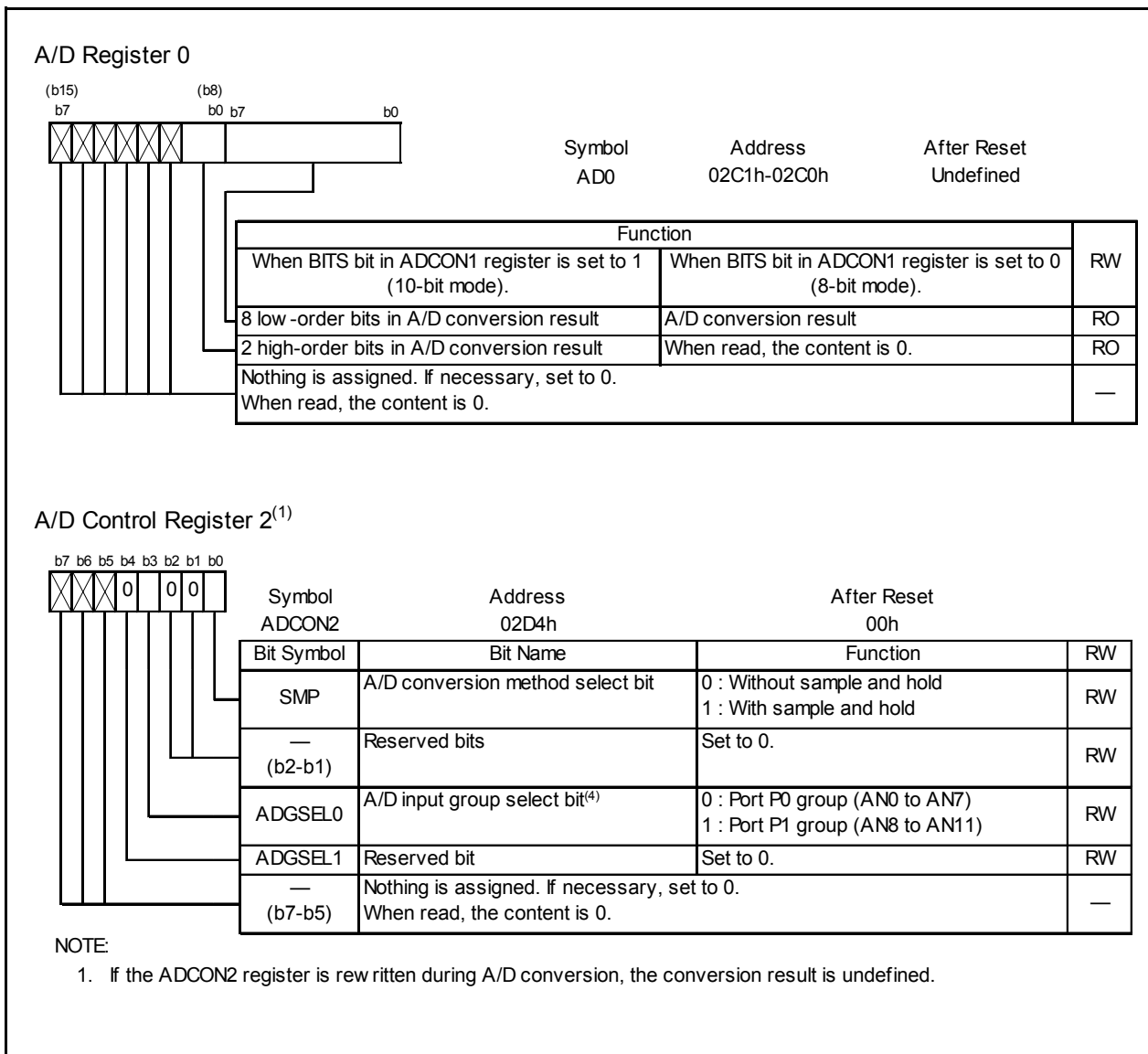


Figure 18.2 Registers AD0 and ADCON2

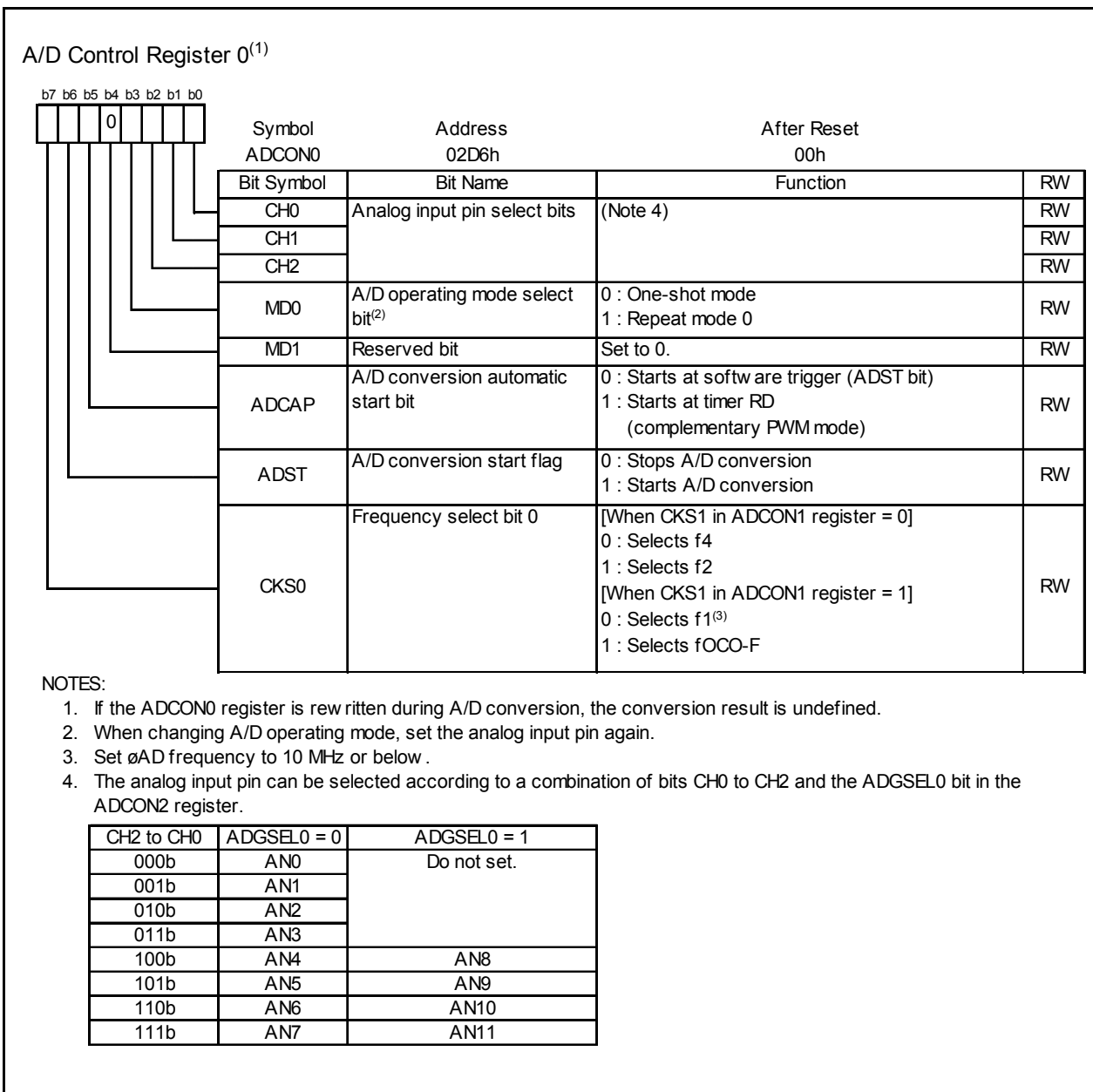


Figure 18.3 ADCON0 Register

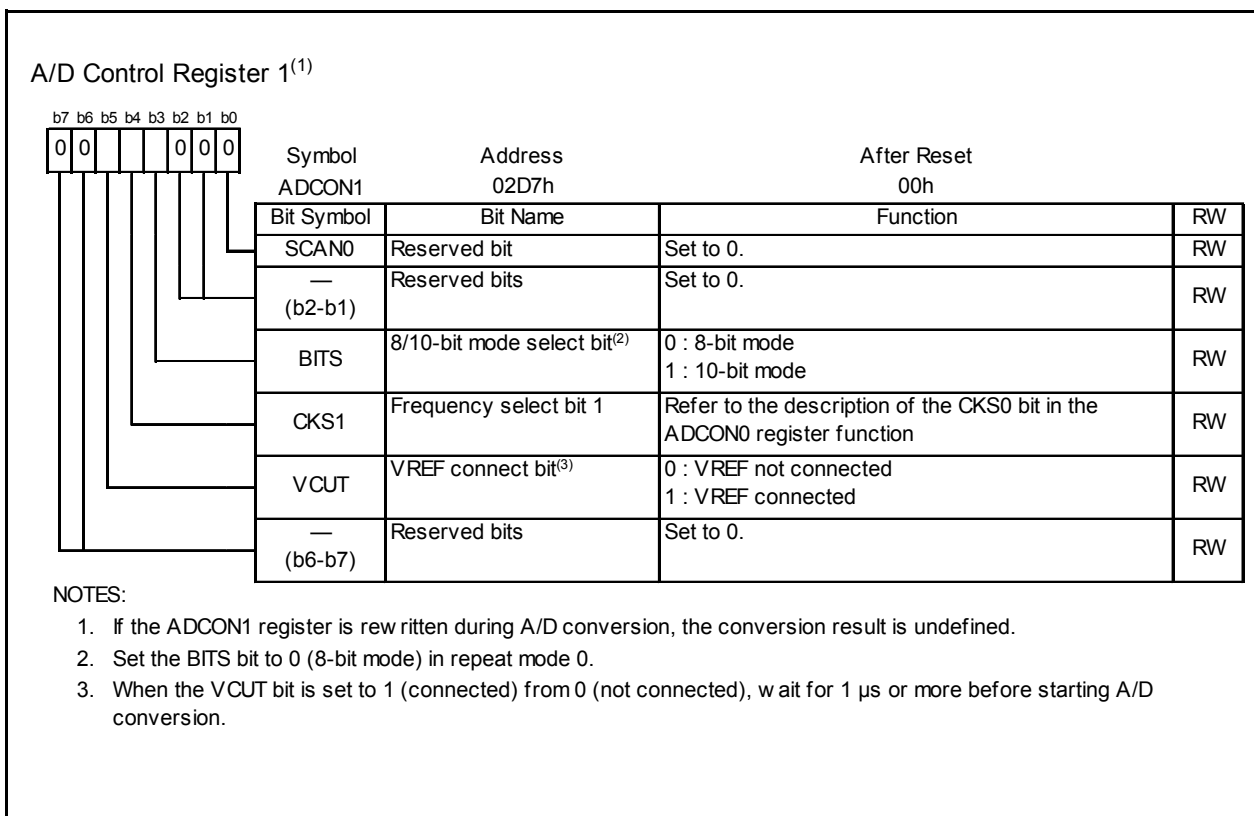


Figure 18.4 ADCON1 Register

18.1 One-Shot Mode

In one-shot mode, the input voltage of one selected pin is A/D converted once.

Table 18.2 lists the One-Shot Mode Specifications. Figure 18.5 shows the ADCON0 Register in One-Shot Mode and Figure 18.6 shows the ADCON1 Register in One-Shot Mode.

Table 18.2 One-Shot Mode Specifications

| Item | Specification |
|-------------------------------------|---|
| Function | The input voltage of one pin selected by bits CH2 to CH0 and ADGSEL0 is A/D converted once |
| Start condition | <ul style="list-style-type: none"> • When the ADCAP bit is set to 0 (software trigger), set the ADST bit to 1 (A/D conversion starts) • When the ADCAP bit is set to 1 (starts in timer RD (complementary PWM mode), A compare match between registers TRD0 and TRDGRA0 or a TRD1 underflow is generated while the ADST bit is set to 1 |
| Stop condition | <ul style="list-style-type: none"> • A/D conversion completes (when the ADCAP bit is set to 0 (software trigger), ADST bit is set to 0) • Set the ADST bit to 0 |
| Interrupt request generation timing | A/D conversion completes |
| Input pin | Select one of AN0 to AN11 |
| Reading of A/D conversion result | Read AD0 register |

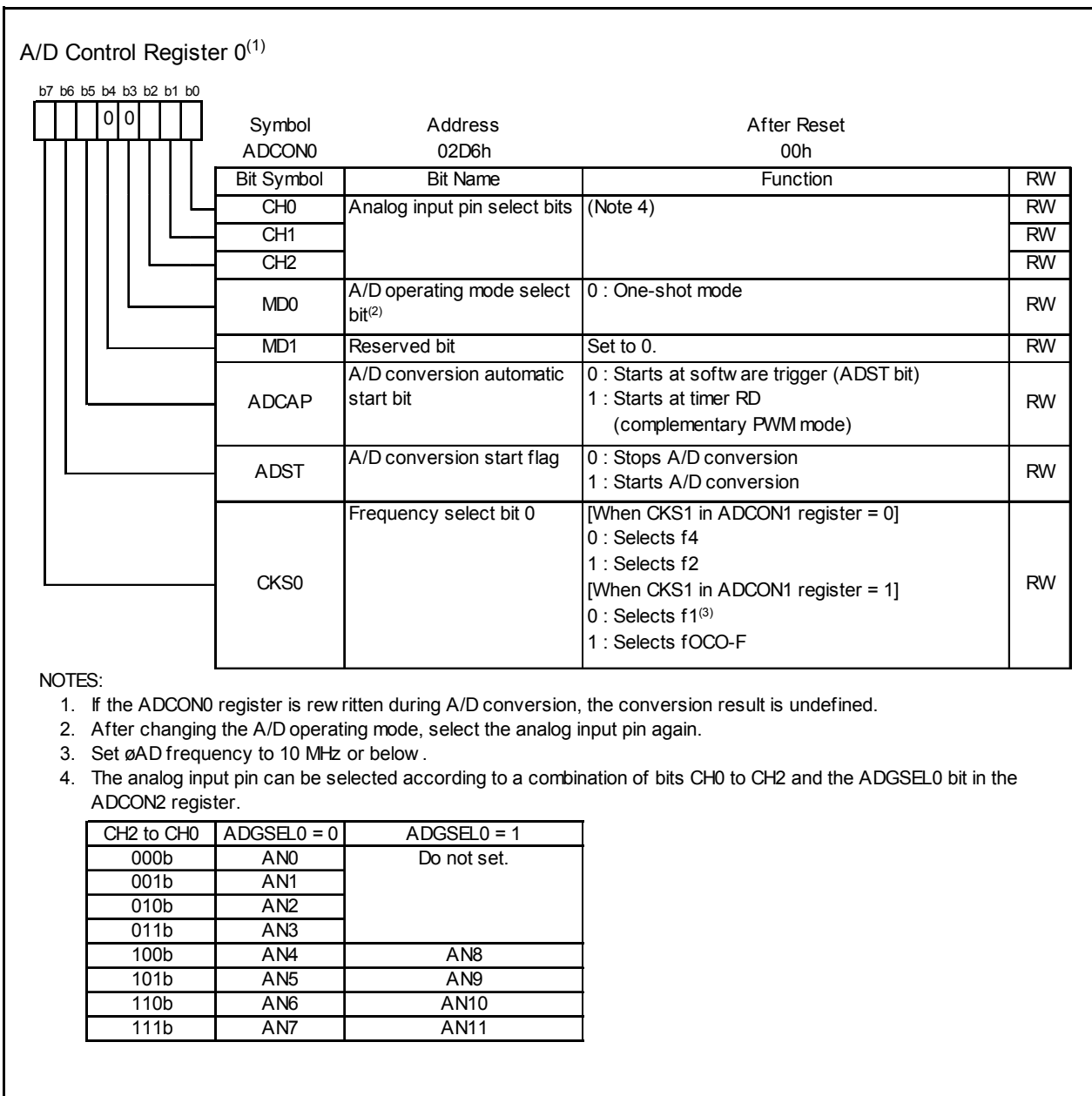


Figure 18.5 ADCON0 Register in One-Shot Mode

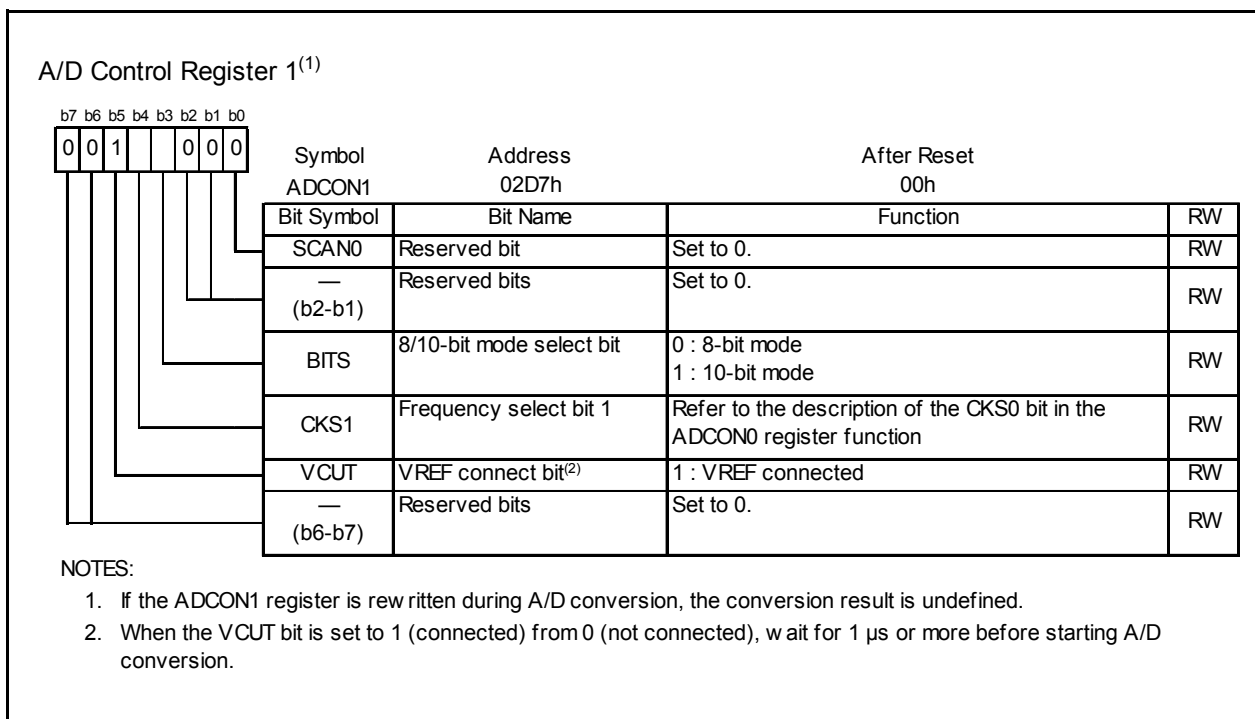


Figure 18.6 ADCON1 Register in One-Shot Mode

18.2 Repeat Mode 0

In repeat mode, the input voltage of one selected pin is A/D converted repeatedly.

Table 18.3 lists the Repeat Mode 0 Specifications. Figure 18.7 shows the ADCON0 Register in Repeat Mode 0 and Figure 18.8 shows the ADCON1 Register in Repeat Mode 0.

Table 18.3 Repeat Mode 0 Specifications

| Item | Specification |
|-------------------------------------|--|
| Function | The Input voltage of one pin selected by bits CH2 to CH0 and ADGSEL0 is A/D converted repeatedly |
| Start conditions | <ul style="list-style-type: none"> • When the ADCAP bit is set to 0 (software trigger), set the ADST bit to 1 (A/D conversion starts) • When the ADCAP bit is set to 1 (starts in timer RD (complementary PWM mode)), a compare match between registers TRD0 and TRDGRA0 or a TRD1 underflow is generated while the ADST bit is set to 1 |
| Stop condition | Set the ADST bit to 0 |
| Interrupt request generation timing | Not generated |
| Input pin | Select one of AN0 to AN11 |
| Reading of result of A/D converter | Read AD0 register |

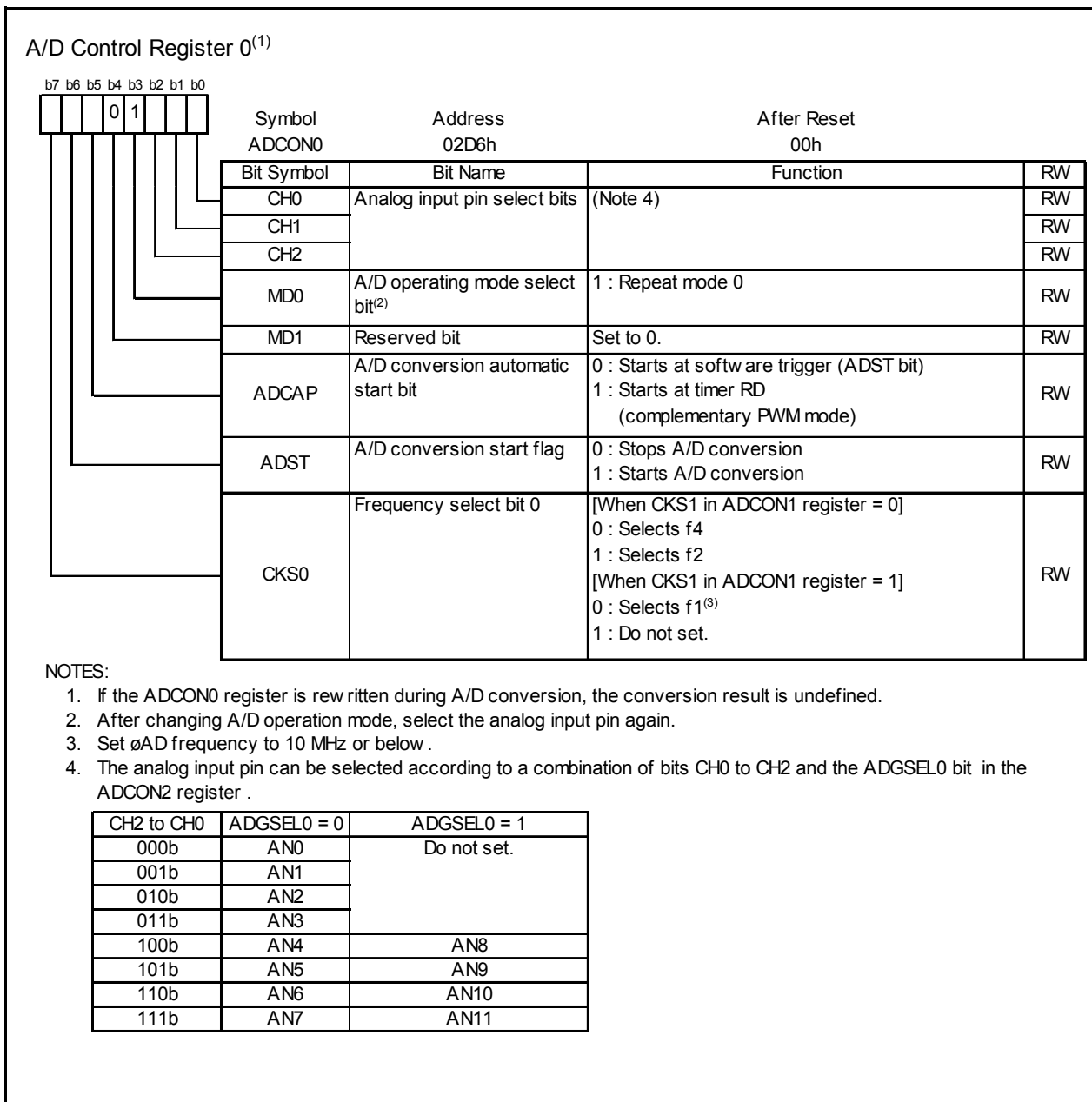


Figure 18.7 ADCON0 Register in Repeat Mode 0

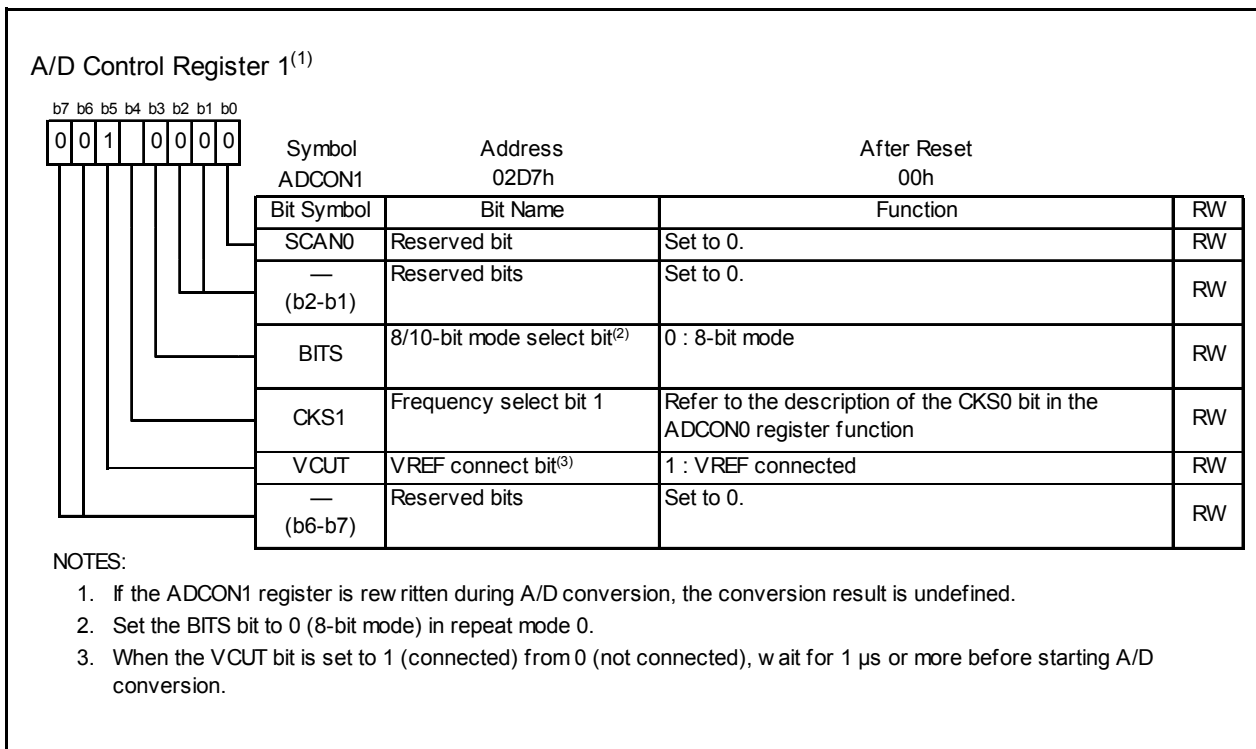


Figure 18.8 ADCON1 Register in Repeat Mode 0

18.3 Sample and Hold

When the SMP bit in the ADCON2 register is set to 1 (sample and hold function enabled), the A/D conversion rate per pin increases. The sample and hold function is available in all operating modes. Start A/D conversion after selecting whether the sample and hold circuit is to be used or not.

Figure 18.9 shows a Timing Diagram of A/D Conversion.

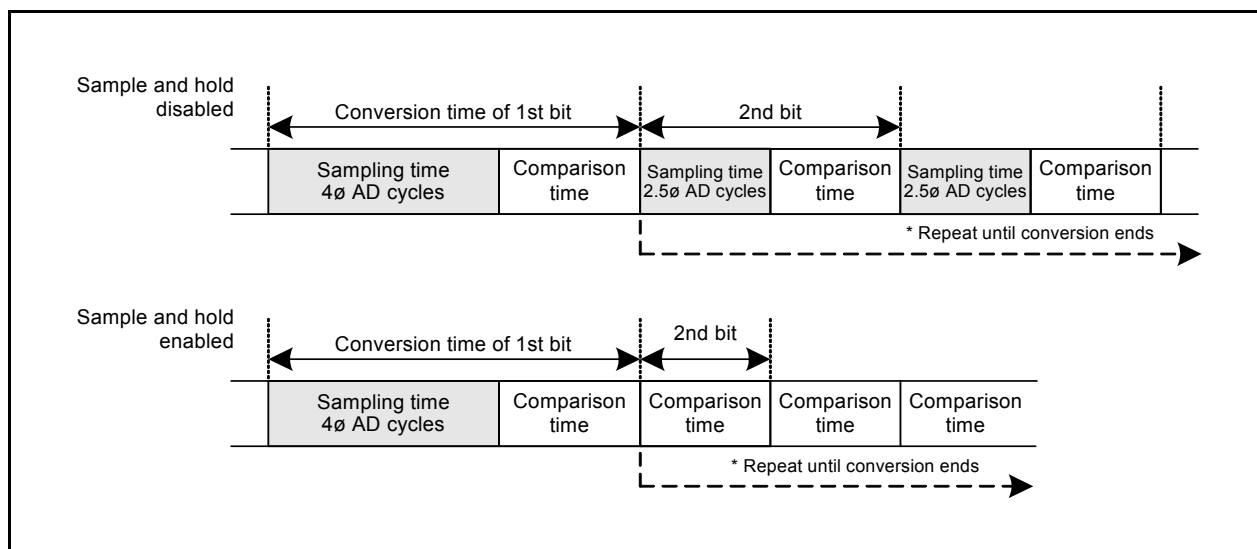


Figure 18.9 Timing Diagram of A/D Conversion

18.4 A/D Conversion Cycles

Figure 18.10 shows the A/D Conversion Cycles.

| A/D Conversion Mode | | Conversion Time | Sampling Time | Comparison Time | Sampling Time | Comparison Time | End process |
|-----------------------|---------|-----------------|---------------|-----------------|---------------|-----------------|--------------|
| Without Sample & Hold | 8 bits | 49ϕ AD | 4ϕ AD | 2.0ϕ AD | 2.5ϕ AD | 2.5ϕ AD | 8.0ϕ AD |
| Without Sample & Hold | 10 bits | 59ϕ AD | 4ϕ AD | 2.0ϕ AD | 2.5ϕ AD | 2.5ϕ AD | 8.0ϕ AD |
| With Sample & Hold | 8 bits | 28ϕ AD | 4ϕ AD | 2.5ϕ AD | 0.0ϕ AD | 2.5ϕ AD | 4.0ϕ AD |
| With Sample & Hold | 10 bits | 33ϕ AD | 4ϕ AD | 2.5ϕ AD | 0.0ϕ AD | 2.5ϕ AD | 4.0ϕ AD |

Figure 18.10 A/D Conversion Cycles

18.5 Internal Equivalent Circuit of Analog Input

Figure 18.11 shows the Internal Equivalent Circuit of Analog Input.

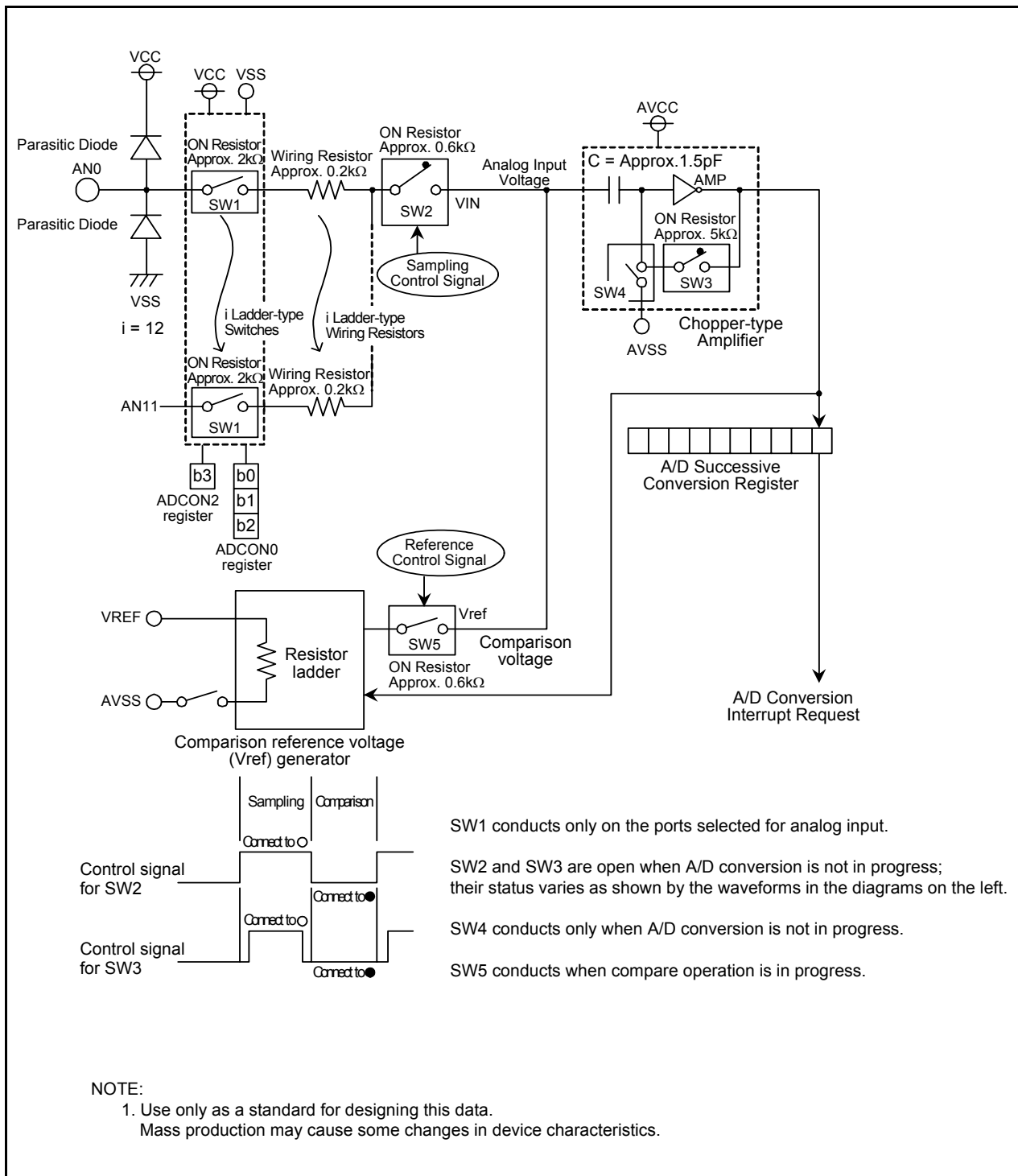


Figure 18.11 Internal Equivalent Circuit of Analog Input

18.6 Output Impedance of Sensor under A/D Conversion

To carry out A/D conversion properly, charging the internal capacitor C shown in Figure 18.12 has to be completed within a specified period of time. T (sampling time) as the specified time. Let output impedance of sensor equivalent circuit be R0, internal resistance of microcomputer be R, precision (error) of the A/D converter be X, and the resolution of A/D converter be Y (Y is 1024 in the 10-bit mode, and 256 in the 8-bit mode).

$$VC \text{ is generally } VC = VIN \left\{ 1 - e^{-\frac{1}{C(R0+R)}t} \right\}$$

$$\text{And when } t = T, \quad VC = VIN - \frac{X}{Y} VIN = VIN \left(1 - \frac{X}{Y} \right)$$

$$e^{-\frac{1}{C(R0+R)}T} = \frac{X}{Y}$$

$$-\frac{1}{C(R0+R)}T = \ln \frac{X}{Y}$$

$$\text{Hence, } R0 = -\frac{T}{C \cdot \ln \frac{X}{Y}} - R$$

Figure 18.12 shows Analog Input Pin and External Sensor Equivalent Circuit. When the difference between VIN and VC becomes 0.1LSB, we find impedance R0 when voltage between pins VC changes from 0 to VIN - (0.1/1024) VIN in time T. (0.1/1024) means that A/D precision drop due to insufficient capacitor charge is held to 0.1LSB at time of A/D conversion in the 10-bit mode. Actual error however is the value of absolute precision added to 0.1LSB.

When $f(XIN) = 10 \text{ MHz}$, $T = 0.25 \mu\text{s}$ in the A/D conversion mode without sample and hold. Output impedance R0 for sufficiently charging capacitor C within time T is determined as follows.

$T = 0.25 \mu\text{s}$, $R = 2.8 \text{ k}\Omega$, $C = 6.0 \text{ pF}$, $X = 0.1$, and $Y = 1024$. Hence,

$$R0 = \frac{0.25 \times 10^{-6}}{6.0 \times 10^{-12} \cdot \ln \frac{0.1}{1024}} - 2.8 \times 10^3 \approx 1.7 \times 10^3$$

Thus, the allowable output impedance of the sensor equivalent circuit, making the precision (error) 0.1LSB or less, is approximately 1.7 kΩ maximum.

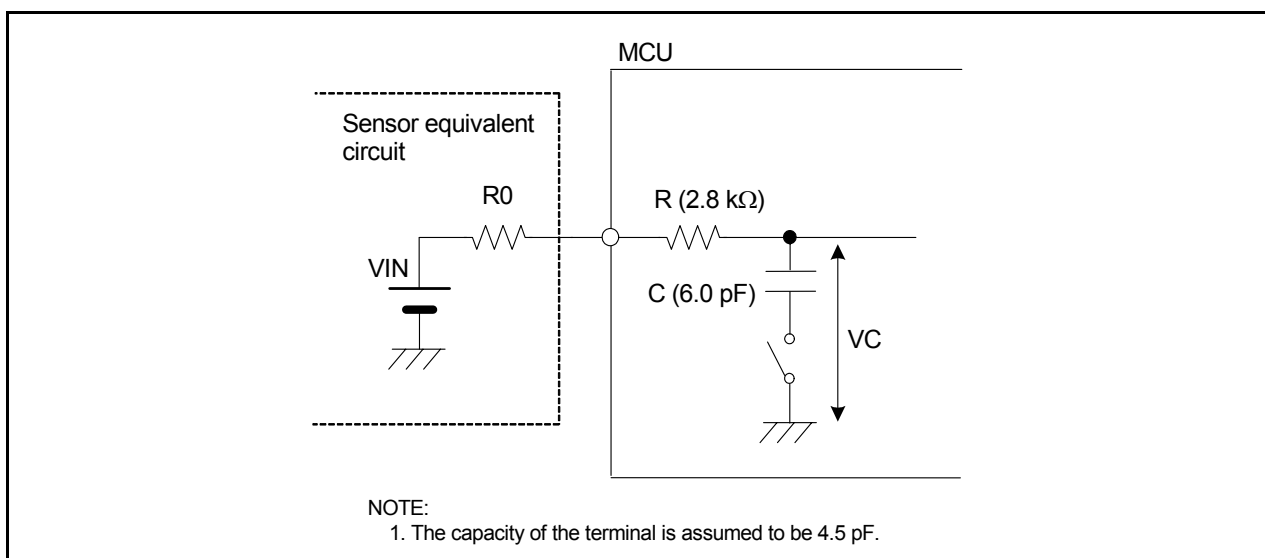


Figure 18.12 Analog Input Pin and External Sensor Equivalent Circuit

18.7 Notes on A/D Converter

- Write to each bit (other than ADST bit) in the ADCON0 register, each bit in the ADCON1 register, or the SMP bit in the ADCON2 register when A/D conversion is stopped (before a trigger occurs).
When the VCUT bit in the ADCON1 register is changed from 0 (VREF not connected) to 1 (VREF connected), wait for at least 1 μ s before starting the A/D conversion.
- After changing the A/D operating mode, select an analog input pin again.
- When using the one-shot mode, ensure that A/D conversion is completed before reading the AD0 register. The IR bit in the ADIC register or the ADST bit in the ADCON0 register can be used to determine whether A/D conversion is completed.
- When using the repeat mode 0, select the frequency of the A/D converter operating clock ϕ AD or more for the CPU clock during A/D conversion.
Do not select the fOCO-F for the ϕ AD.
- If the ADST bit in the ADCON0 register is set to 0 (A/D conversion stops) by a program and A/D conversion is forcibly terminated during an A/D conversion operation, the conversion result of the A/D converter will be undefined. If the ADST bit is set to 0 by a program, do not use the value of the AD0 register.
- Connect 0.1 μ F capacitor between the VREF pin and AVSS pin.
- Do not enter stop mode during A/D conversion.
- Do not enter wait mode when the CM02 bit in the CM0 register is set to 1 (peripheral function clock stops in wait mode) during A/D conversion.

19. D/A Converter

The D/A converters are 8-bit R-2R type units. There are two independent D/A converters.

D/A conversion is performed by writing to the DA_i register (i = 0 or 1). To output the conversion result, set the DA_iE bit in the DACON register to 1 (output enabled). Before using D/A conversion, the corresponding port direction bit must be set to 0 (input mode). Setting the DA_iE bit to 1 removes the pull-up from the corresponding port.

The output analog voltage (V) is determined by the setting value n (n: decimal) of the DA_i register.

$$V = V_{\text{ref}} \times n / 256 \quad (n = 0 \text{ to } 255)$$

V_{ref}: Reference voltage

Table 19.1 lists the D/A Converter Specifications. Figure 19.1 shows the Block Diagram of D/A Converter. Figure 19.2 shows the D/A converter related registers. Figure 19.3 shows the D/A Converter Equivalent Circuit.

Table 19.1 D/A Converter Specifications

| Item | Performance |
|-----------------------|-----------------|
| D/A conversion method | R-2R method |
| Resolution | 8 bits |
| Analog output pins | 2 (DA0 and DA1) |

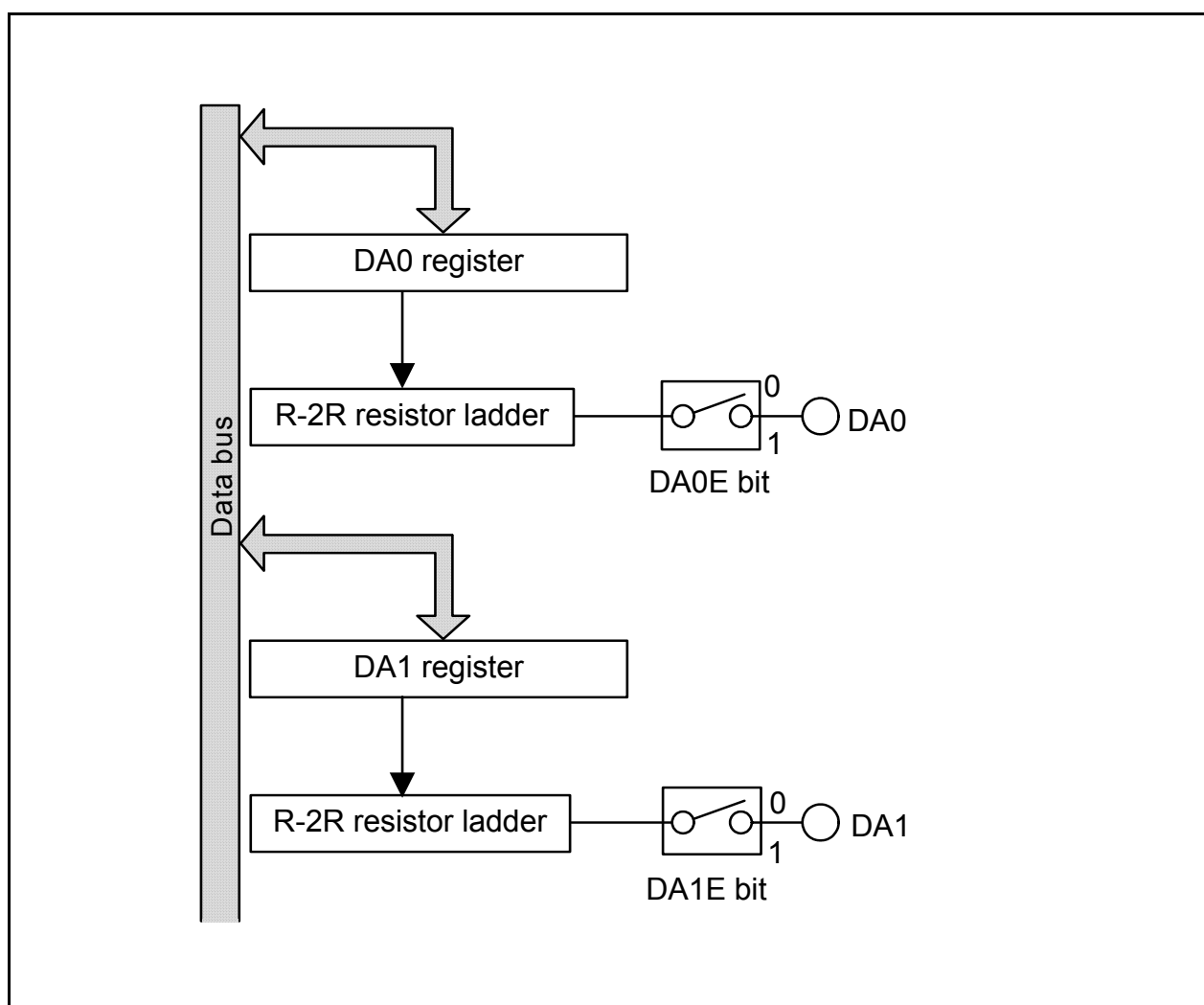


Figure 19.1 Block Diagram of D/A Converter

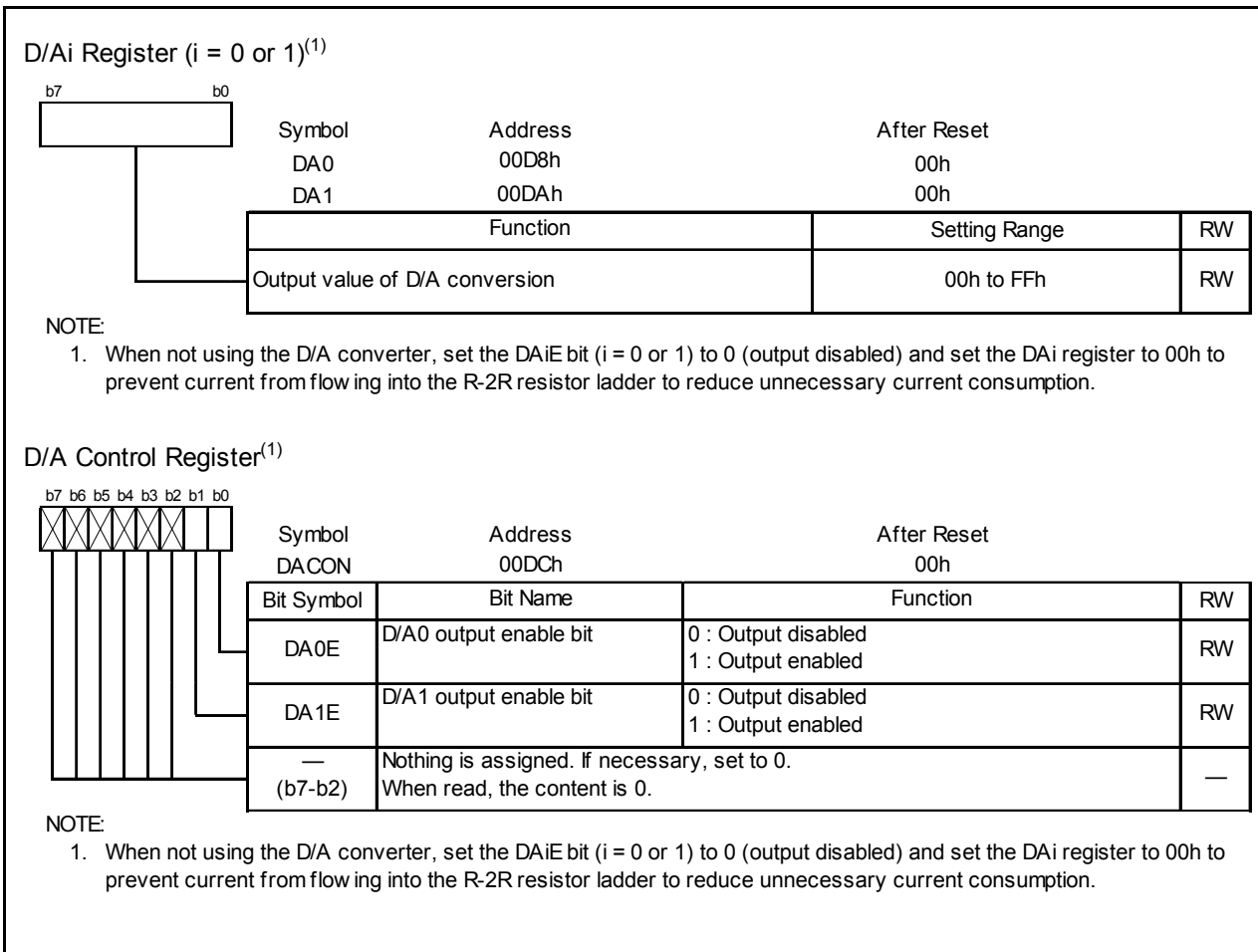


Figure 19.2 Registers DA0 to DA1 and DACON

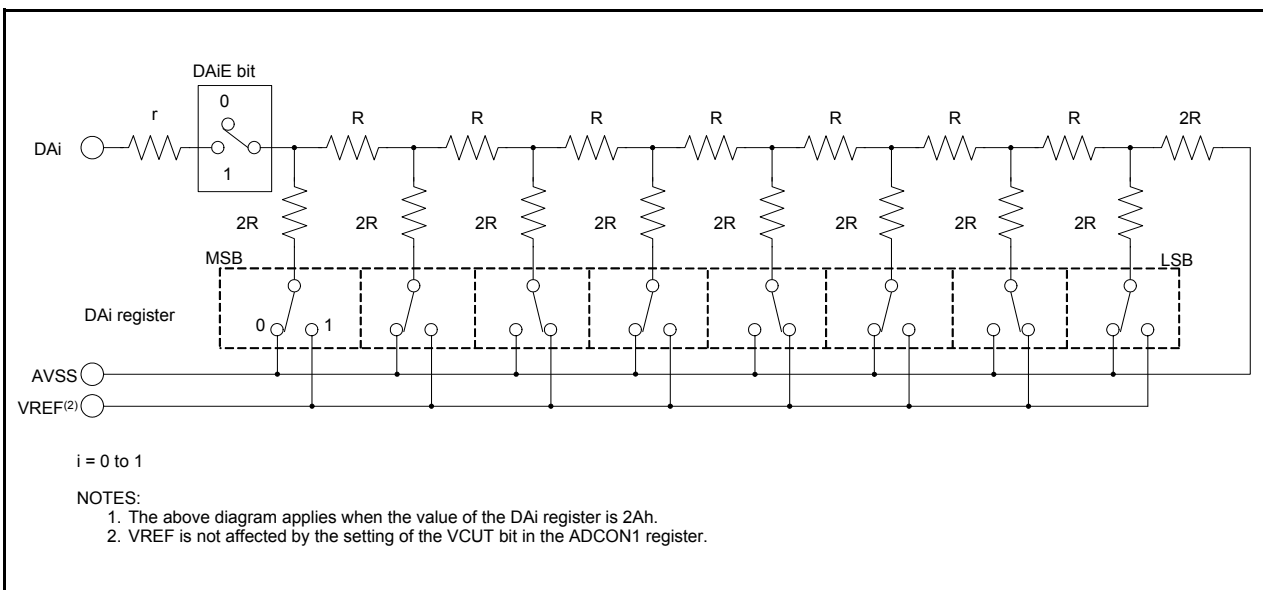


Figure 19.3 D/A Converter Equivalent Circuit

20. Flash Memory

20.1 Overview

In the flash memory, rewrite operations to the flash memory can be performed in three modes: CPU rewrite, standard serial I/O, and parallel I/O.

Table 20.1 lists the Flash Memory Performance.

Table 20.1 Flash Memory Performance

| Item | | Specification |
|---|--|---|
| Flash memory operating mode | | 3 modes (CPU rewrite, standard serial I/O, and parallel I/O) |
| Division of erase block | | Refer to Figure 20.1 and Figure 20.2 |
| Programming method | | Byte unit |
| Erase method | | Block erase |
| Programming and erasure control method ⁽³⁾ | | Program and erase control by software command |
| Rewrite control method | | Rewrite control for blocks 0 to 3 by FMR02 bit in FMR0 register Rewrite control for block 0 by FMR15 bit and Block 1 by FMR16 bit in FMR1 register |
| Number of commands | | 5 commands |
| Programming and erasure endurance ⁽¹⁾ | Blocks 0 to 3 (program ROM) | R8C/2A Group: 100 times; R8C/2B Group: 1,000 times |
| | Blocks A and B (data flash) ⁽²⁾ | 10,000 times |
| ID code check function | | Standard serial I/O mode supported |
| ROM code protect | | Parallel I/O mode supported |

NOTES:

1. Definition of programming and erasure endurance
The programming and erasure endurance is defined on a per-block basis. If the programming and erasure endurance is n (n = 100 or 10,000), each block can be erased n times. For example, if 1,024 1-byte writes are performed to block A, a 1-Kbyte block, and then the block is erased, the erase count stands at one. When performing 100 or more rewrites, the actual erase count can be reduced by executing programming operations in such a way that all blank areas are used before performing an erase operation. Avoid rewriting only particular blocks and try to average out the programming and erasure endurance of the blocks. It is also advisable to retain data on the erase count of each block and limit the number of erase operations to a certain number.
2. Blocks A and B are implemented only in the R8C/2B group.
3. To perform programming and erasure, use VCC = 2.7 V to 5.5 V as the supply voltage. Do not perform programming and erasure at less than 2.7 V.

Table 20.2 Flash Memory Rewrite Modes

| Flash memory Rewrite mode | CPU Rewrite Mode | Standard Serial I/O Mode | Parallel I/O Mode |
|------------------------------|--|--|--|
| Function | User ROM area is rewritten by executing software commands from the CPU. EW0 mode: Rewritable in the RAM EW1 mode: Rewritable in flash memory | User ROM area is rewritten by a dedicated serial programmer. | User ROM area is rewritten by a dedicated parallel programmer. |
| Areas which can be rewritten | User ROM area | User ROM area | User ROM area |
| Operating mode | Single chip mode | Boot mode | Parallel I/O mode |
| ROM Programmer | None | Serial programmer | Parallel programmer |

20.2 Memory Map

The flash memory contains a user ROM area and a boot ROM area (reserved area). Figure 20.1 shows the Flash Memory Block Diagram for R8C/2A Group. Figure 20.2 shows a Flash Memory Block Diagram for R8C/2B Group.

The user ROM area of the R8C/2B Group contains an area (program ROM) which stores MCU operating programs and blocks A and B (data flash) each 1 Kbyte in size.

The user ROM area is divided into several blocks. The user ROM area can be rewritten in CPU rewrite mode and standard serial I/O and parallel I/O modes.

When rewriting blocks 0 and 1 in CPU rewrite mode, set the FMR02 bit in the FMR0 register to 1 (rewrite enabled). When the FMR15 bit in the FMR1 register is set to 0 (rewrite enabled), block 0 is rewritable. When the FMR16 bit is set to 0 (rewrite enabled), block 1 is rewritable. When rewriting blocks 2 and 3 in CPU rewrite mode, FMR02 bit is set to 1 (rewrite enabled), blocks 2 and 3 are rewritable.

The rewrite control program for standard serial I/O mode is stored in the boot ROM area before shipment. The boot ROM area and the user ROM area share the same address, but have separate memory areas.

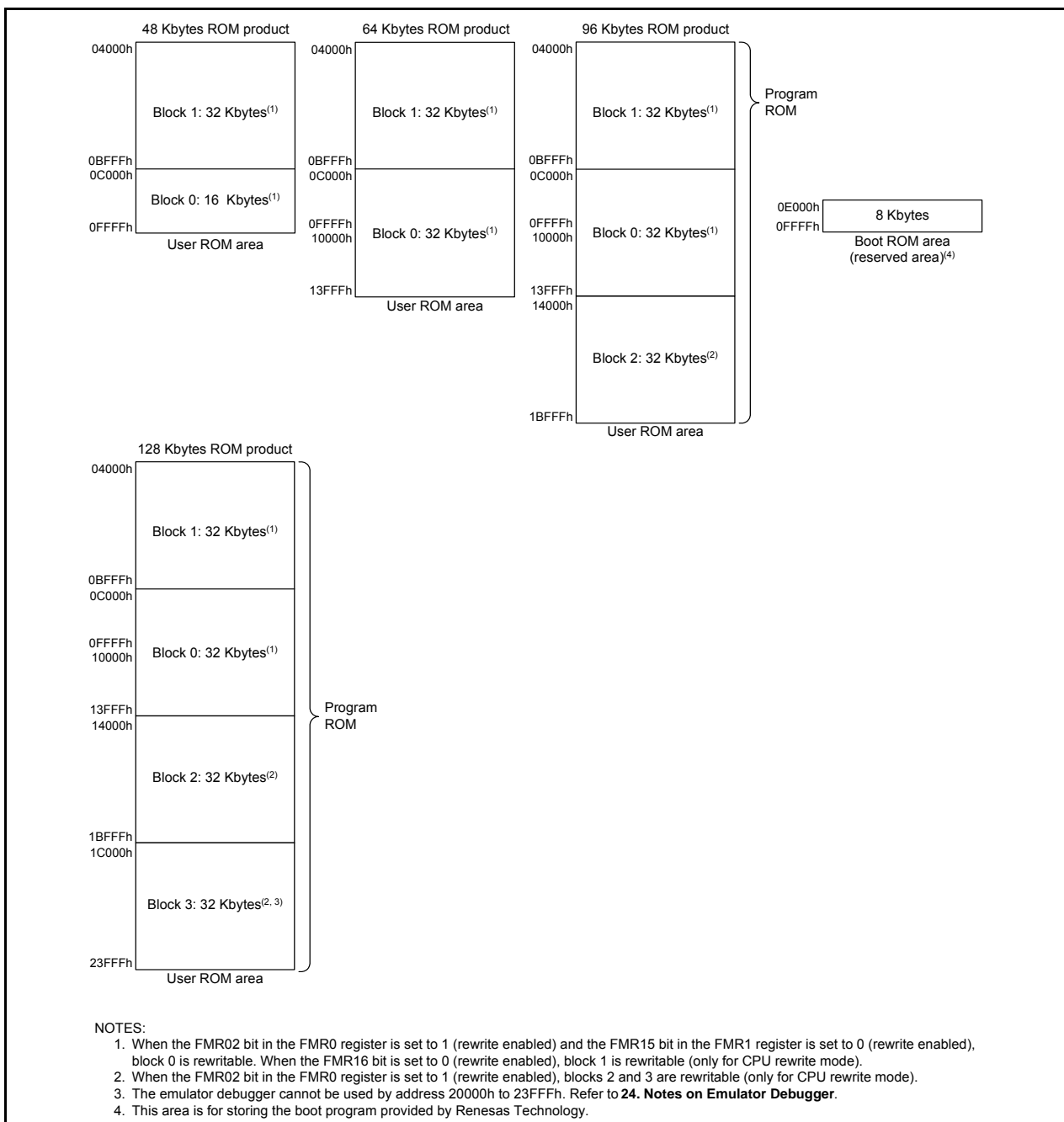


Figure 20.1 Flash Memory Block Diagram for R8C/2A Group

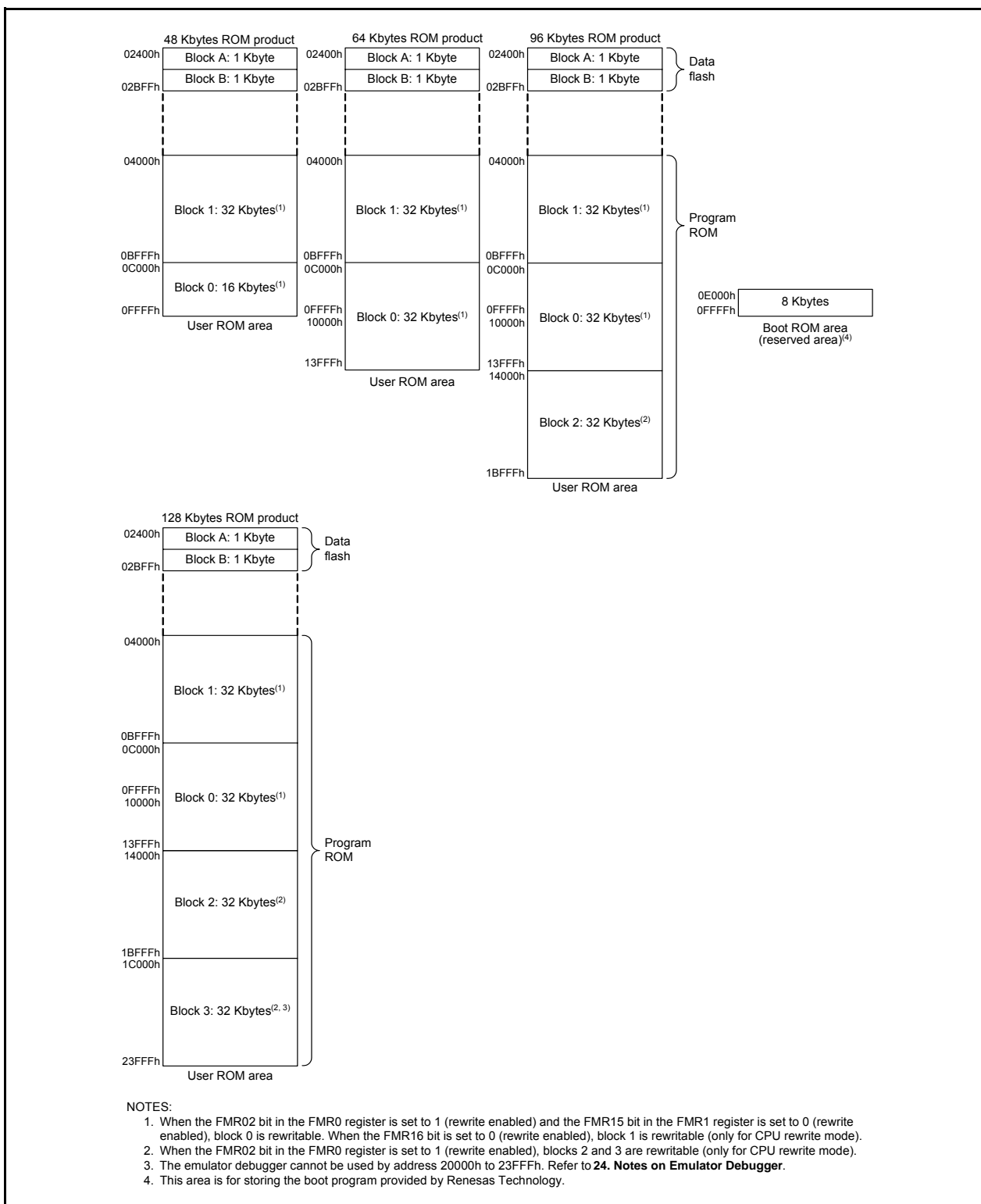


Figure 20.2 Flash Memory Block Diagram for R8C/2B Group

20.3 Functions to Prevent Rewriting of Flash Memory

Standard serial I/O mode has an ID code check function, and parallel I/O mode has a ROM code protect function to prevent the flash memory from being read or rewritten easily.

20.3.1 ID Code Check Function

This function is used in standard serial I/O mode. Unless the flash memory is blank, the ID codes sent from the programmer and the ID codes written in the flash memory are checked to see if they match. If the ID codes do not match, the commands sent from the programmer are not acknowledged. The ID codes consist of 8 bits of data each, the areas of which, beginning with the first byte, are 00FFDFh, 00FFE3h, 00FFEBh, 00FFEFh, 00FFF3h, 00FFF7h, and 00FFFBh. Write programs in which the ID codes are set at these addresses and write them to the flash memory.

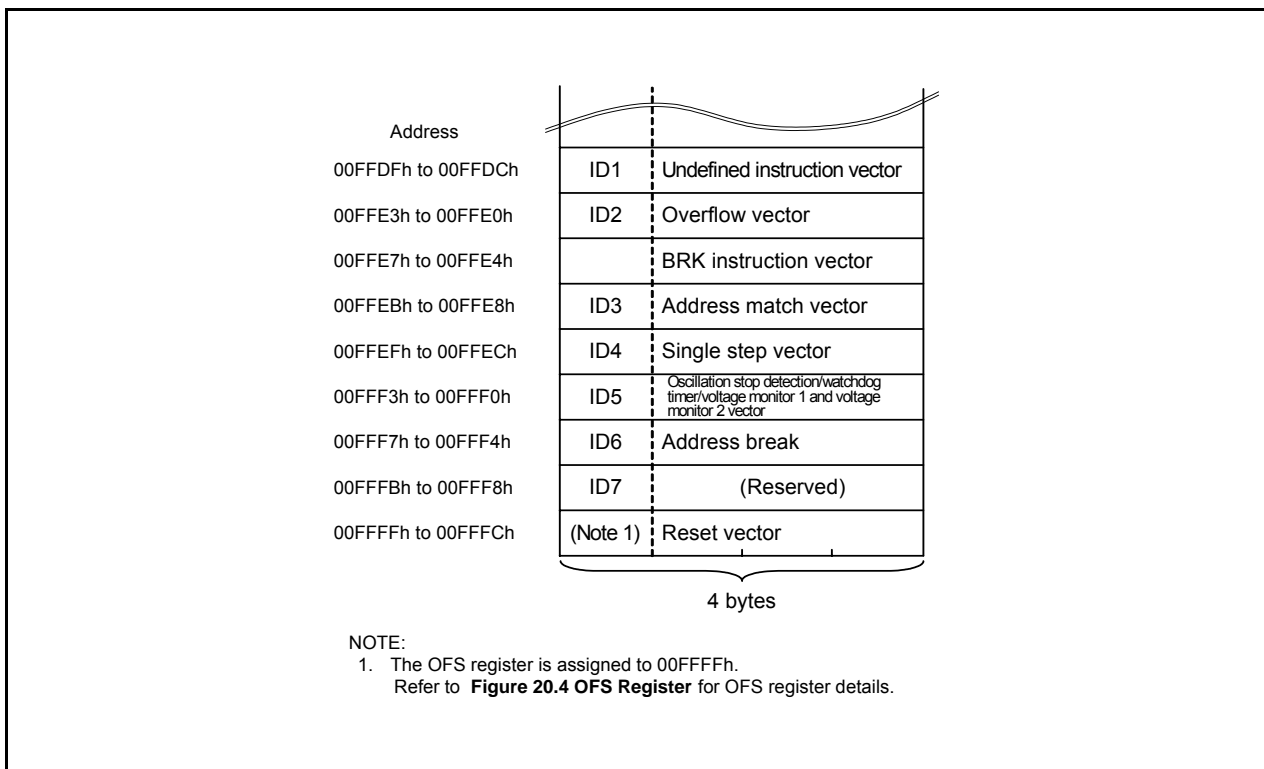


Figure 20.3 Address for Stored ID Code

20.3.2 ROM Code Protect Function

The ROM code protect function disables reading or changing the contents of the on-chip flash memory by the OFS register in parallel I/O mode. Figure 20.4 shows the OFS Register.

The ROM code protect function is enabled by writing 0 to the ROMCP1 bit and 1 to the ROMCR bit. It disables reading or changing the contents of the on-chip flash memory.

Once ROM code protect is enabled, the content in the internal flash memory cannot be rewritten in parallel I/O mode. To disable ROM code protect, erase the block including the OFS register with CPU rewrite mode or standard serial I/O mode.

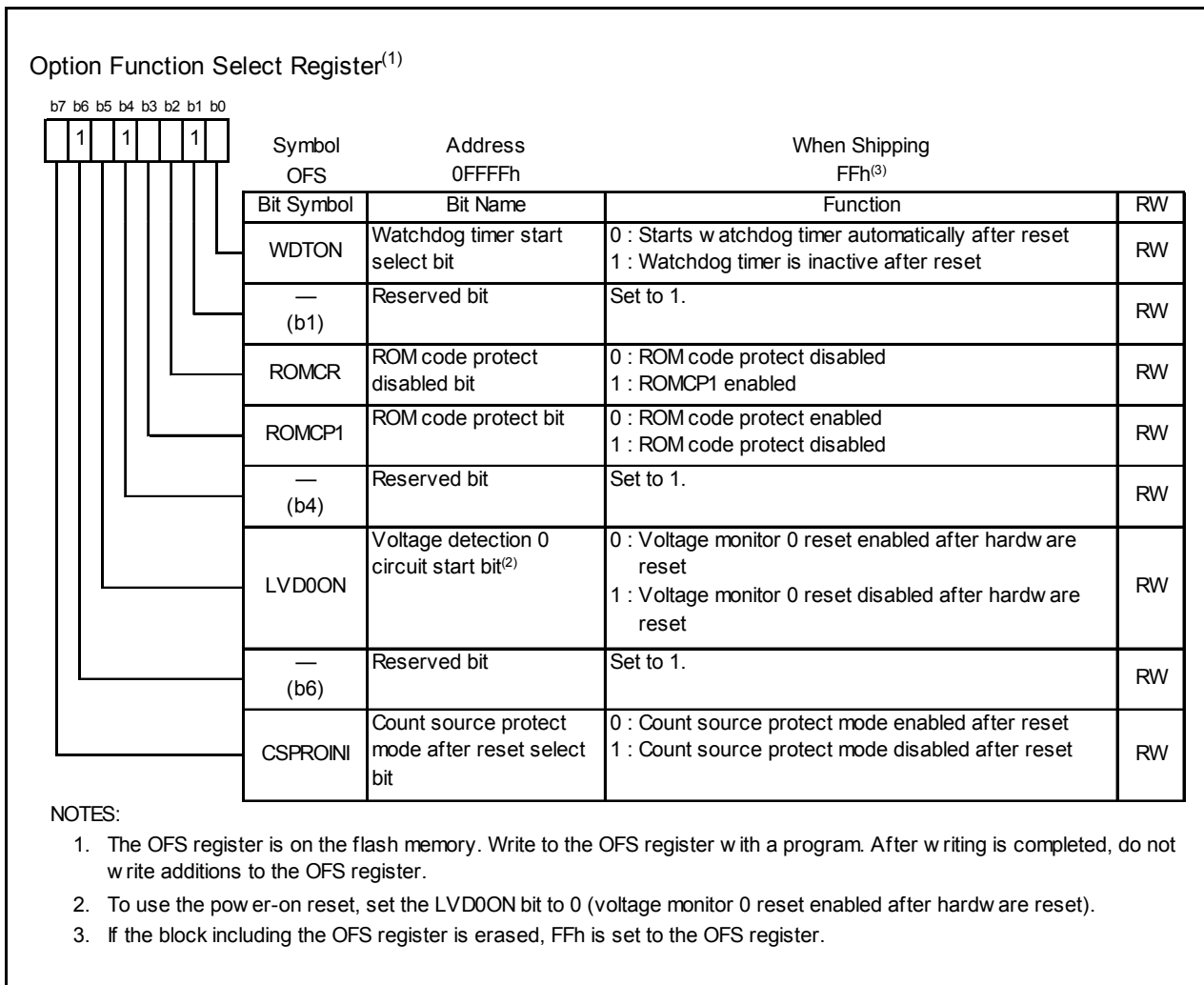


Figure 20.4 OFS Register

20.4 CPU Rewrite Mode

In CPU rewrite mode, the user ROM area can be rewritten by executing software commands from the CPU. Therefore, the user ROM area can be rewritten directly while the MCU is mounted on a board without using a ROM programmer. Execute the program and block erase commands only to blocks in the user ROM area.

The flash module has an erase-suspend function when an interrupt request is generated during an erase operation in CPU rewrite mode. It performs an interrupt process after the erase operation is halted temporarily. During erase-suspend, the user ROM area can be read by a program.

In case an interrupt request is generated during an auto-program operation in CPU rewrite mode, the flash module has a program-suspend function which performs the interrupt process after the auto-program operation is suspended. During program-suspend, the user ROM area can be read by a program.

CPU rewrite mode has an erase write 0 mode (EW0 mode) and an erase write 1 mode (EW1 mode). Table 20.3 lists the Differences between EW0 Mode and EW1 Mode.

Table 20.3 Differences between EW0 Mode and EW1 Mode

| Item | EW0 Mode | EW1 Mode |
|--|---|---|
| Operating mode | Single-chip mode | Single-chip mode |
| Areas in which a rewrite control program can be located | User ROM area | User ROM area |
| Areas in which a rewrite control program can be executed | Necessary to transfer to any area other than the flash memory (e.g., RAM) before executing | Executing directly in user ROM or RAM area possible |
| Areas which can be rewritten | User ROM area | User ROM area However, blocks which contain a rewrite control program are excluded ⁽¹⁾ |
| Software command restrictions | None | <ul style="list-style-type: none"> • Program and block erase commands Cannot be run on any block which contains a rewrite control program • Read status register command Cannot be executed |
| Modes after program or erase | Read status register mode | Read array mode |
| Modes after read status register | Read status register mode | Do not execute this command |
| CPU status during auto-write and auto-erase | Operating | Hold state (I/O ports hold state before the command is executed) |
| Flash memory status detection | <ul style="list-style-type: none"> • Read bits FMR00, FMR06, and FMR07 in the FMR0 register by a program • Execute the read status register command and read bits SR7, SR5, and SR4 in the status register. | Read bits FMR00, FMR06, and FMR07 in the FMR0 register by a program |
| Conditions for transition to erase-suspend | Set bits FMR40 and FMR41 in the FMR4 register to 1 by a program. | The FMR40 bit in the FMR4 register is set to 1 and the interrupt request of the enabled maskable interrupt is generated |
| Conditions for transitions to program-suspend | Set bits FMR40 and FMR42 in the FMR4 register to 1 by a program. | The FMR40 bit in the FMR4 register is set to 1 and the interrupt request of the enabled maskable interrupt is generated |
| CPU clock | 5 MHz or below | No restriction (on clock frequency to be used) |

NOTE:

1. When the FMR02 bit in the FMR0 register is set to 1 (rewrite enabled), rewriting block 0 is enabled by setting the FMR15 bit in the FMR1 register to 0 (rewrite enabled), and rewriting block 1 is enabled by setting the FMR16 bit to 0 (rewrite enabled).

When the FMR02 bit in the FMR0 register is set to 1 (rewrite enabled), blocks 2 and 3 are rewritable.

20.4.1 EW0 Mode

The MCU enters CPU rewrite mode and software commands can be acknowledged by setting the FMR01 bit in the FMR0 register to 1 (CPU rewrite mode enabled). In this case, since the FMR11 bit in the FMR1 register is set to 0, EW0 mode is selected.

Use software commands to control program and erase operations. The FMR0 register or the status register can be used to determine when program and erase operations complete.

During auto-erasure, set the FMR40 bit to 1 (erase-suspend enabled) and the FMR41 bit to 1 (request erase-suspend). Wait for td(SR-SUS) and ensure that the FMR46 bit is set to 1 (read enabled) before accessing the user ROM area. The auto-erase operation can be restarted by setting the FMR41 bit to 0 (erase restarts).

To enter program-suspend during the auto-program operation, set the FMR40 bit to 1 (suspend enabled) and the FMR42 bit to 1 (request program-suspend). Wait for td(SR-SUS) and ensure that the FMR46 bit is set to 1 (read enabled) before accessing the user ROM area. The auto-program operation can be restarted by setting the FMR42 bit to 0 (program restarts).

20.4.2 EW1 Mode

The MCU is switched to EW1 mode by setting the FMR11 bit to 1 (EW1 mode) after setting the FMR01 bit to 1 (CPU rewrite mode enabled).

The FMR0 register can be used to determine when program and erase operations complete. Do not execute commands that use the read status register in EW1 mode.

To enable the erase-suspend function during auto-erasure, execute the block erase command after setting the FMR40 bit to 1 (erase-suspend enabled). The interrupt to enter erase-suspend should be in interrupt enabled status. After waiting for td(SR-SUS) after the block erase command is executed, the interrupt request is acknowledged.

When an interrupt request is generated, the FMR41 bit is automatically set to 1 (requests erase-suspend) and the auto-erase operation suspends. If an auto-erase operation does not complete (FMR00 bit is 0) after an interrupt process completes, the auto-erase operation restarts by setting the FMR41 bit to 0 (erasure restarts)

To enable the program-suspend function during auto-programming, execute the program command after setting the FMR40 bit to 1 (suspend enabled). The interrupt to enter program-suspend should be in interrupt enabled status. After waiting for td(SR-SUS) after the program command is executed, an interrupt request is acknowledged.

When an interrupt request is generated, the FMR42 bit is automatically set to 1 (request program-suspend) and the auto-program operation suspends. When the auto-program operation does not complete (FMR00 bit is 0) after the interrupt process completes, the auto-program operation can be restarted by setting the FMR42 bit to 0 (programming restarts).

Figure 20.5 shows the FMR0 Register, Figure 20.6 shows the FMR1 Register and Figure 20.7 shows the FMR4 Register.

20.4.2.1 FMR00 Bit

This bit indicates the operating status of the flash memory. The bit's value is 0 during programming, erasure (including suspend periods), or erase-suspend mode; otherwise, it is 1.

20.4.2.2 FMR01 Bit

The MCU is made ready to accept commands by setting the FMR01 bit to 1 (CPU rewrite mode).

20.4.2.3 FMR02 Bit

Rewriting of blocks 0 to 3 does not accept program or block erase commands if the FMR02 bit is set to 0 (rewrite disabled).

Rewriting of blocks 2 and 3 are enabled, if the FMR02 bit is set to 1 (rewrite enabled). Rewriting of blocks 0 and 1 is controlled by bits FMR15 and FMR16 if the FMR02 bit is set to 1 (rewrite enabled).

20.4.2.4 FMSTP Bit

This bit is used to initialize the flash memory control circuits, and also to reduce the amount of current consumed by the flash memory. Access to the flash memory is disabled by setting the FMSTP bit to 1. Therefore, the FMSTP bit must be written to by a program transferred to the RAM.

In the following cases, set the FMSTP bit to 1:

- When flash memory access resulted in an error while erasing or programming in EW0 mode (FMR00 bit not reset to 1 (ready))
- To provide lower consumption in high-speed on-chip oscillator mode, low-speed on-chip oscillator mode (XIN clock stops), and low-speed clock mode (XIN clock stops).

Figure 20.11 shows the handling to provide lower consumption in high-speed on-chip oscillator mode, low-speed on-chip oscillator mode (XIN clock stops), and low-speed clock mode (XIN clock stops). Handle according to this flowchart. Note that when going to stop or wait mode while the CPU rewrite mode is disabled, the FMR0 register does not need to be set because the power for the flash memory is automatically turned off and is turned back on again after returning from stop or wait mode.

20.4.2.5 FMR06 Bit

This is a read-only bit indicating the status of an auto-program operation. The bit is set to 1 when a program error occurs; otherwise, it is set to 0. For details, refer to the description in **20.4.5 Full Status Check**.

20.4.2.6 FMR07 Bit

This is a read-only bit indicating the status of an auto-erase operation. The bit is set to 1 when an erase error occurs; otherwise, it is set to 0. Refer to **20.4.5 Full Status Check** for details.

20.4.2.7 FMR11 Bit

Setting this bit to 1 (EW1 mode) places the MCU in EW1 mode.

20.4.2.8 FMR15 Bit

When the FMR02 bit is set to 1 (rewrite enabled) and the FMR15 bit is set to 0 (rewrite enabled), block 0 accepts program and block erase commands.

20.4.2.9 FMR16 Bit

When the FMR02 bit is set to 1 (rewrite enabled) and the FMR16 bit is set to 0 (rewrite enabled), block 1 accepts program and block erase commands.

20.4.2.10 FMR40 Bit

The suspend function is enabled by setting the FMR40 bit to 1 (enable).

20.4.2.11 FMR41 Bit

In EW0 mode, the MCU enters erase-suspend mode when the FMR41 bit is set to 1 by a program. The FMR41 bit is automatically set to 1 (request erase-suspend) when an interrupt request of an enabled interrupt is generated in EW1 mode, and then the MCU enters erase-suspend mode.

Set the FMR41 bit to 0 (erase restarts) when the auto-erase operation restarts.

20.4.2.12 FMR42 Bit

In EW0 mode, the MCU enters program-suspend mode when the FMR42 bit is set to 1 by a program. The FMR42 bit is automatically set to 1 (request program-suspend) when an interrupt request of an enabled interrupt is generated in EW1 mode, and then the MCU enters program-suspend mode.

Set the FMR42 bit to 0 (program restart) when the auto-program operation restarts.

20.4.2.13 FMR43 Bit

When the auto-erase operation starts, the FMR43 bit is set to 1 (erase execution in progress). The FMR43 bit remains set to 1 (erase execution in progress) during erase-suspend operation.

When the auto-erase operation ends, the FMR43 bit is set to 0 (erase not executed).

20.4.2.14 FMR44 Bit

When the auto-program operation starts, the FMR44 bit is set to 1 (program execution in progress). The FMR44 bit remains set to 1 (program execution in progress) during program-suspend operation.

When the auto-program operation ends, the FMR44 bit is set to 0 (program not executed).

20.4.2.15 FMR46 Bit

The FMR46 bit is set to 0 (reading disabled) during auto-program or auto-erase execution and set to 1 (reading enabled) in suspend mode. Do not access the flash memory while this bit is set to 0.

20.4.2.16 FMR47 Bit

Power consumption when reading the flash memory can be reduced by setting the FMR47 bit to 1 (enabled) in low-speed clock mode (XIN clock stops) and low-speed on-chip oscillator mode (XIN clock stops).

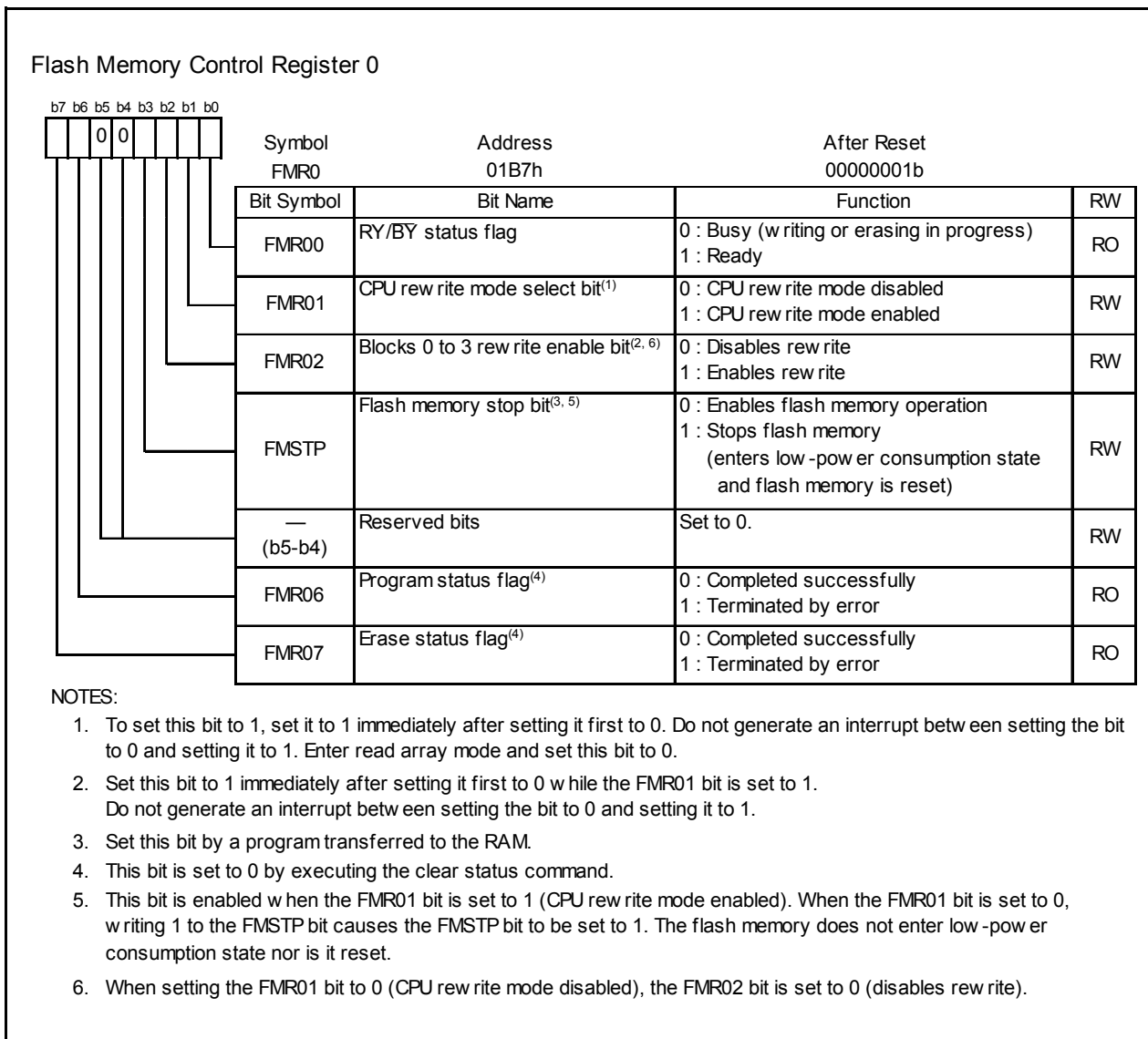


Figure 20.5 FMR0 Register

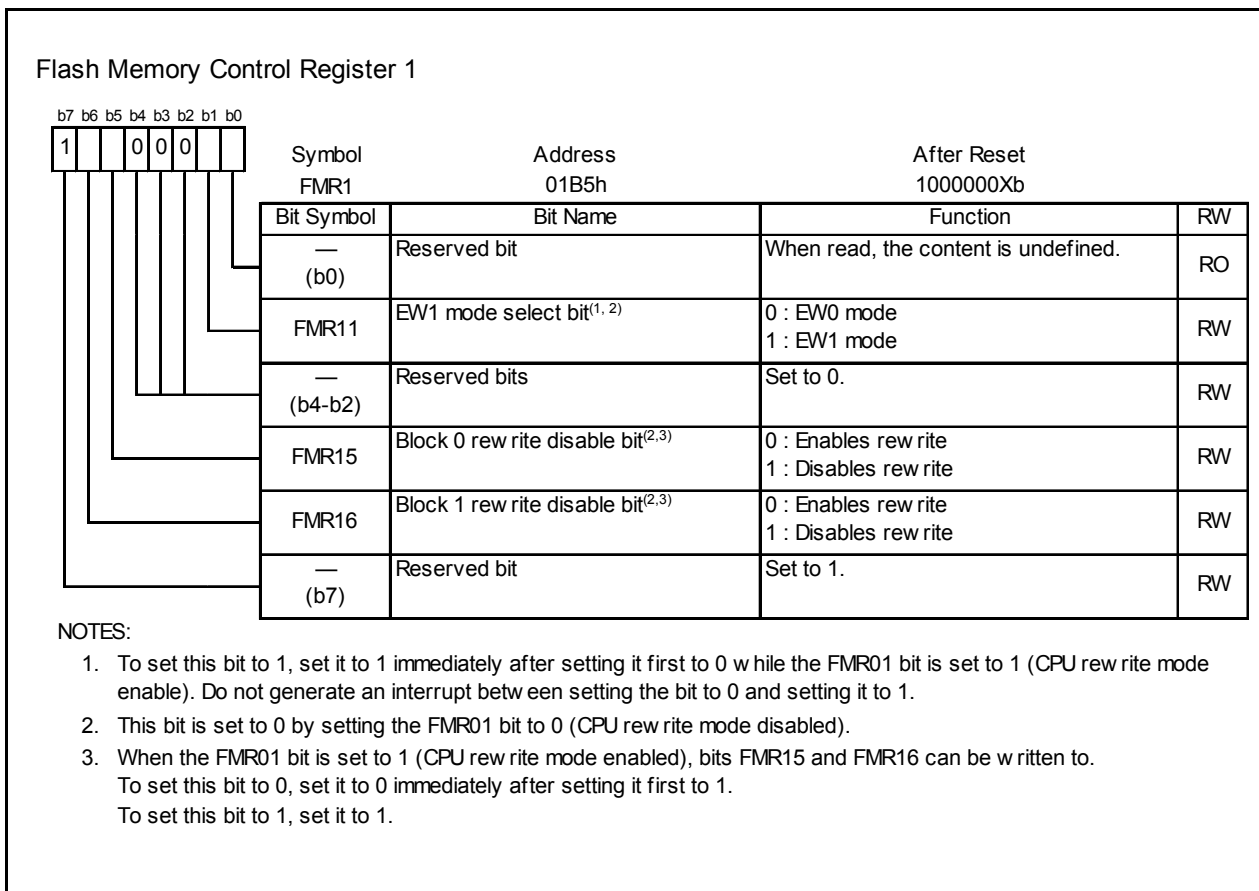


Figure 20.6 FMR1 Register

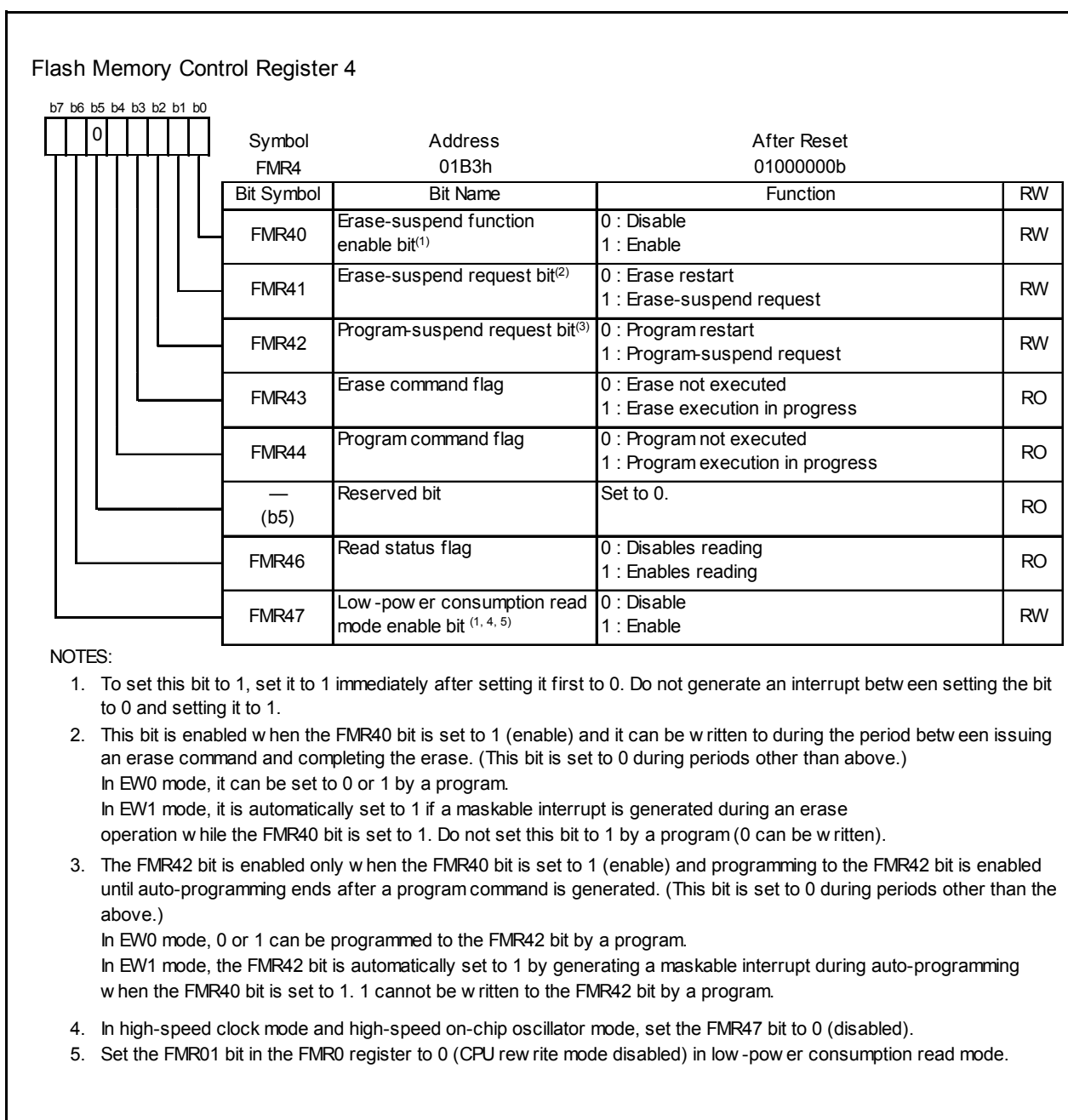


Figure 20.7 FMR4 Register

Figure 20.8 shows the Timing of Suspend Operation.

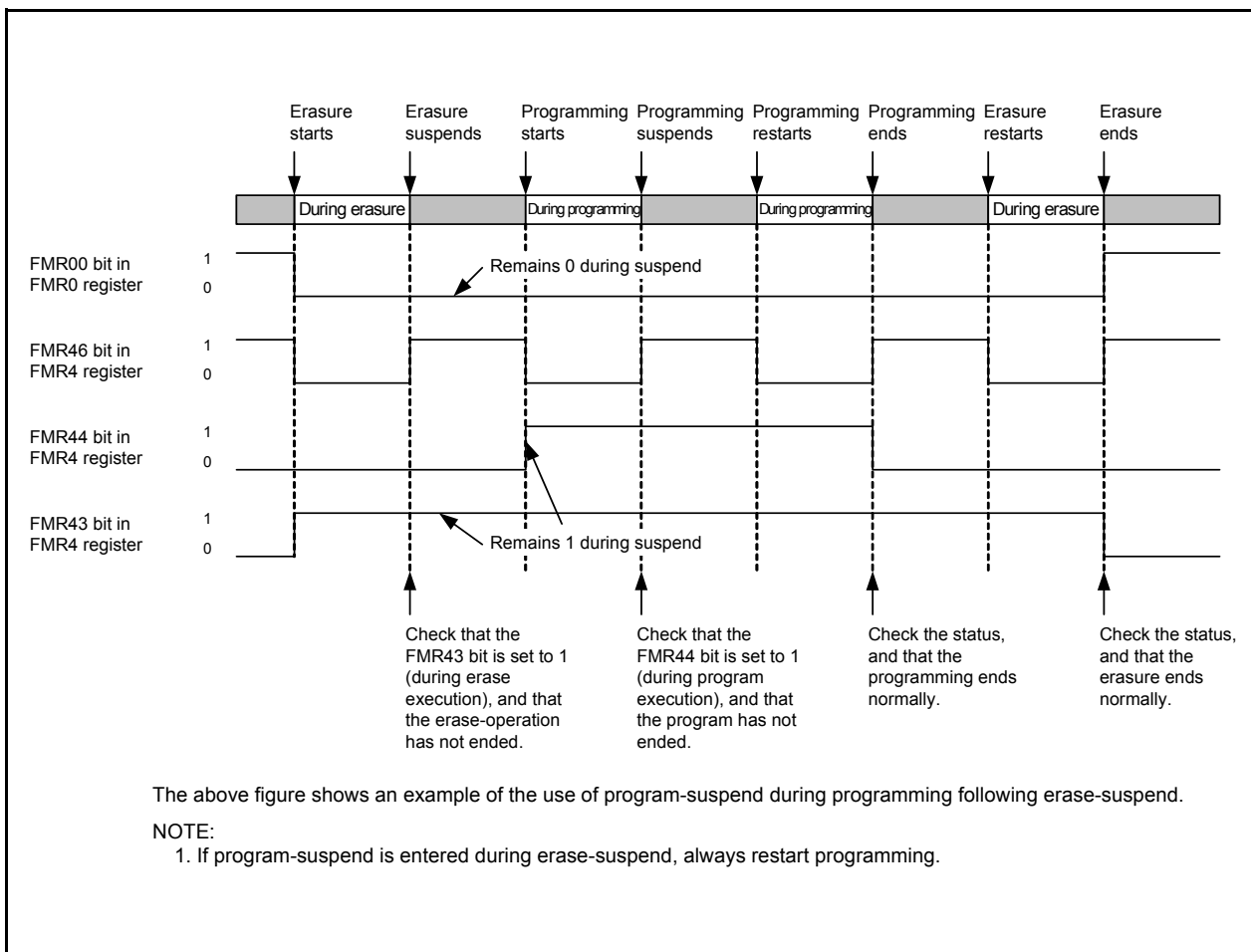


Figure 20.8 Timing of Suspend Operation

Figure 20.9 shows How to Set and Exit EW0 Mode. Figure 20.10 shows How to Set and Exit EW1 Mode.

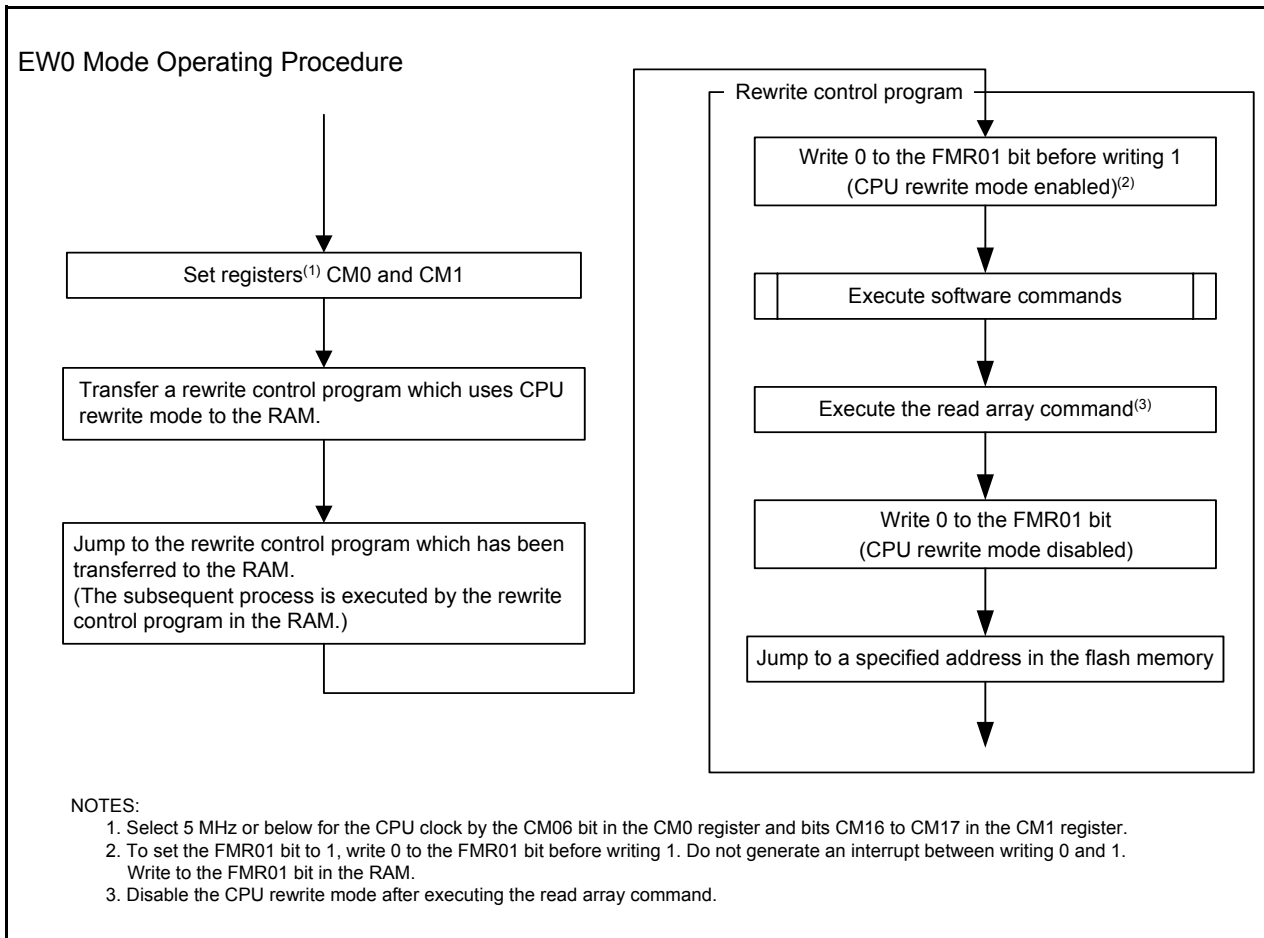


Figure 20.9 How to Set and Exit EW0 Mode

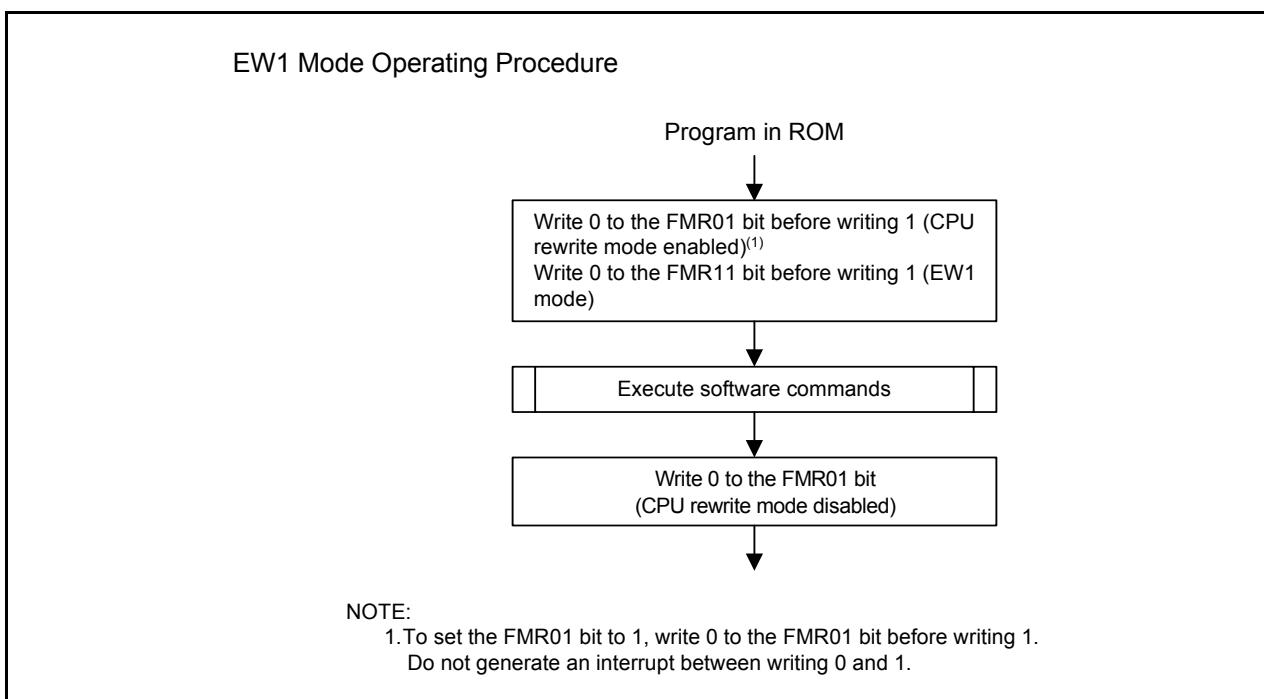


Figure 20.10 How to Set and Exit EW1 Mode

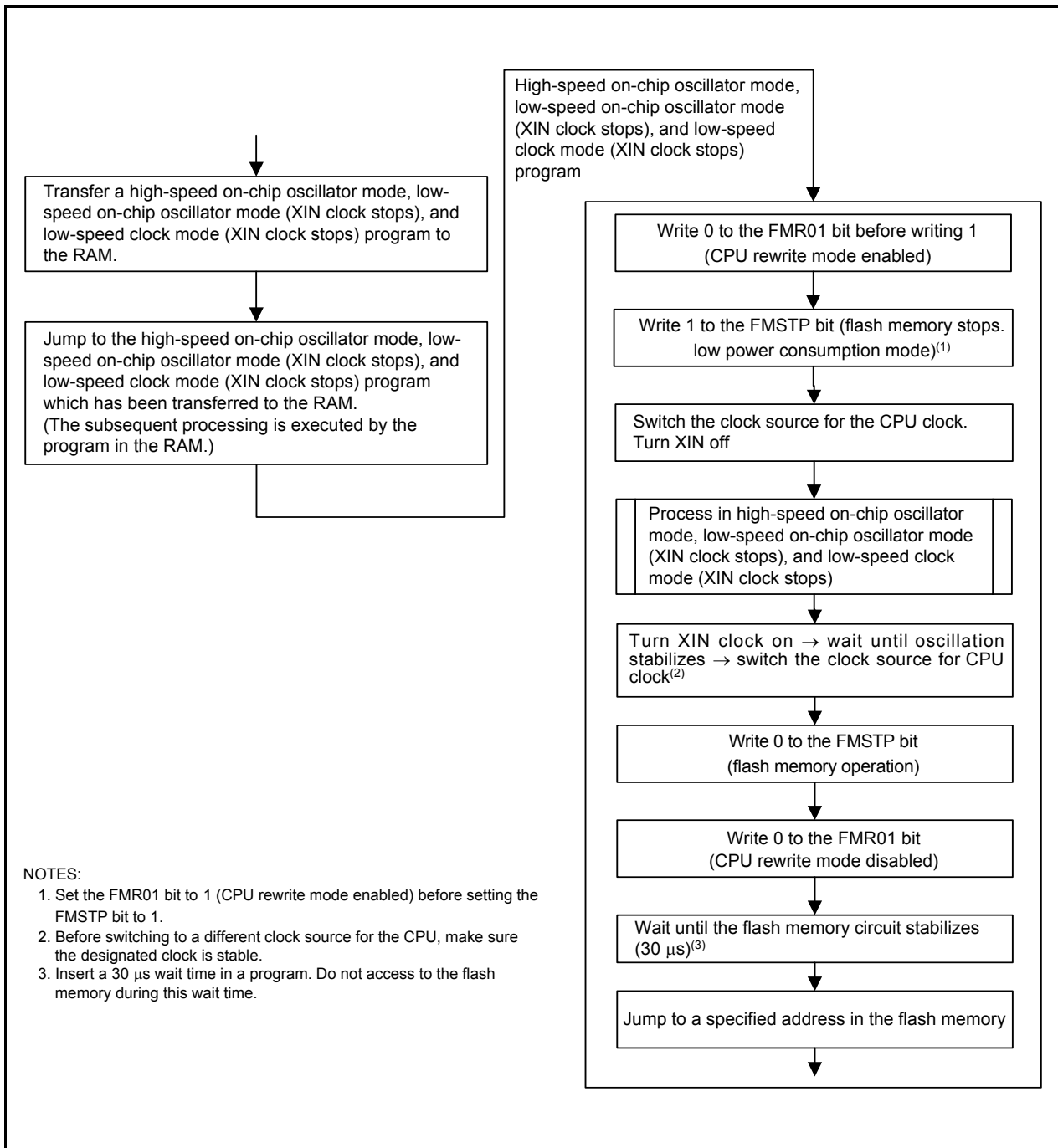


Figure 20.11 Process to Reduce Power Consumption in High-Speed On-Chip Oscillator Mode, Low-Speed On-Chip Oscillator Mode (XIN Clock Stops) and Low-Speed Clock Mode (XIN Clock Stops)

20.4.3 Software Commands

The software commands are described below. Read or write commands and data in 8-bit units.

Table 20.4 Software Commands

| Command | First Bus Cycle | | | Second Bus Cycle | | |
|-----------------------|-----------------|---------|-----------------|------------------|---------|-----------------|
| | Mode | Address | Data (D7 to D0) | Mode | Address | Data (D7 to D0) |
| Read array | Write | x | FFh | | | |
| Read status register | Write | x | 70h | Read | x | SRD |
| Clear status register | Write | x | 50h | | | |
| Program | Write | WA | 40h | Write | WA | WD |
| Block erase | Write | x | 20h | Write | BA | D0h |

SRD: Status register data (D7 to D0)

WA: Write address (ensure the address specified in the first bus cycle is the same address as the write address specified in the second bus cycle.)

WD: Write data (8 bits)

BA: Given block address

x: Any specified address in the user ROM area

20.4.3.1 Read Array Command

The read array command reads the flash memory.

The MCU enters read array mode when FFh is written in the first bus cycle. When the read address is entered in the following bus cycles, the content of the specified address can be read in 8-bit units.

Since the MCU remains in read array mode until another command is written, the contents of multiple addresses can be read continuously.

In addition, the MCU enters read array mode after a reset.

20.4.3.2 Read Status Register Command

The read status register command is used to read the status register.

When 70h is written in the first bus cycle, the status register can be read in the second bus cycle (refer to **20.4.4 Status Registers**). When reading the status register, specify an address in the user ROM area.

Do not execute this command in EW1 mode.

The MCU remains in read status register mode until the next read array command is written.

20.4.3.3 Clear Status Register Command

The clear status register command sets the status register to 0.

When 50h is written in the first bus cycle, bits FMR06 to FMR07 in the FMR0 register and SR4 to SR5 in the status register are set to 0.

20.4.3.4 Program Command

The program command writes data to the flash memory in 1-byte units.

By writing 40h in the first bus cycle and data in the second bus cycle to the write address, an auto-program operation (data program and verify) will start. Make sure the address value specified in the first bus cycle is the same address as the write address specified in the second bus cycle.

The FMR00 bit in the FMR0 register can be used to determine whether auto-programming has completed. When suspend function disabled, the FMR00 bit is set to 0 during auto-programming and set to 1 when auto-programming completes. When suspend function enabled, the FMR44 bit is set to 1 during auto-programming and set to 0 when auto-programming completes.

The FMR06 bit in the FMR0 register can be used to determine the result of auto-programming after it has been finished (refer to **20.4.5 Full Status Check**).

Do not write additions to the already programmed addresses.

When the FMR02 bit in the FMR0 register is set to 0 (rewriting disabled), program commands targeting blocks 0 to 3 are not acknowledged. When the FMR02 bit is set to 1 (rewrite enabled) and the FMR15 bit in the FMR1 register is set to 1 (rewriting disabled), program commands targeting block 0 are not acknowledged. When the FMR16 bit is set to 1 (rewriting disabled), program commands targeting block 1 are not acknowledged.

Figure 20.12 shows the Program Command (When Suspend Function Disabled). Figure 20.13 shows the Program Command (When Suspend Function Enabled).

In EW1 mode, do not execute this command for any address which a rewrite control program is allocated.

In EW0 mode, the MCU enters read status register mode at the same time auto-programming starts and the status register can be read. The status register bit 7 (SR7) is set to 0 at the same time auto-programming starts and set back to 1 when auto-programming completes. In this case, the MCU remains in read status register mode until the next read array command is written. The status register can be read to determine the result of auto-programming after auto-programming has completed.

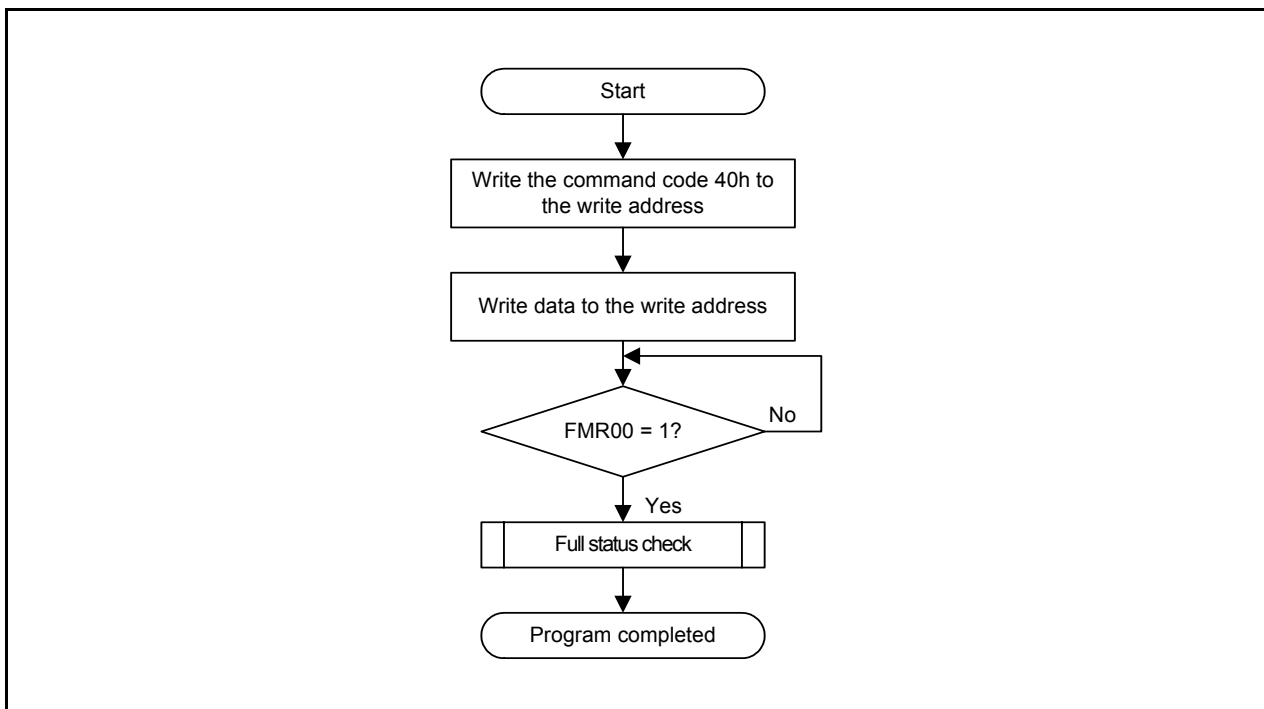


Figure 20.12 Program Command (When Suspend Function Disabled)

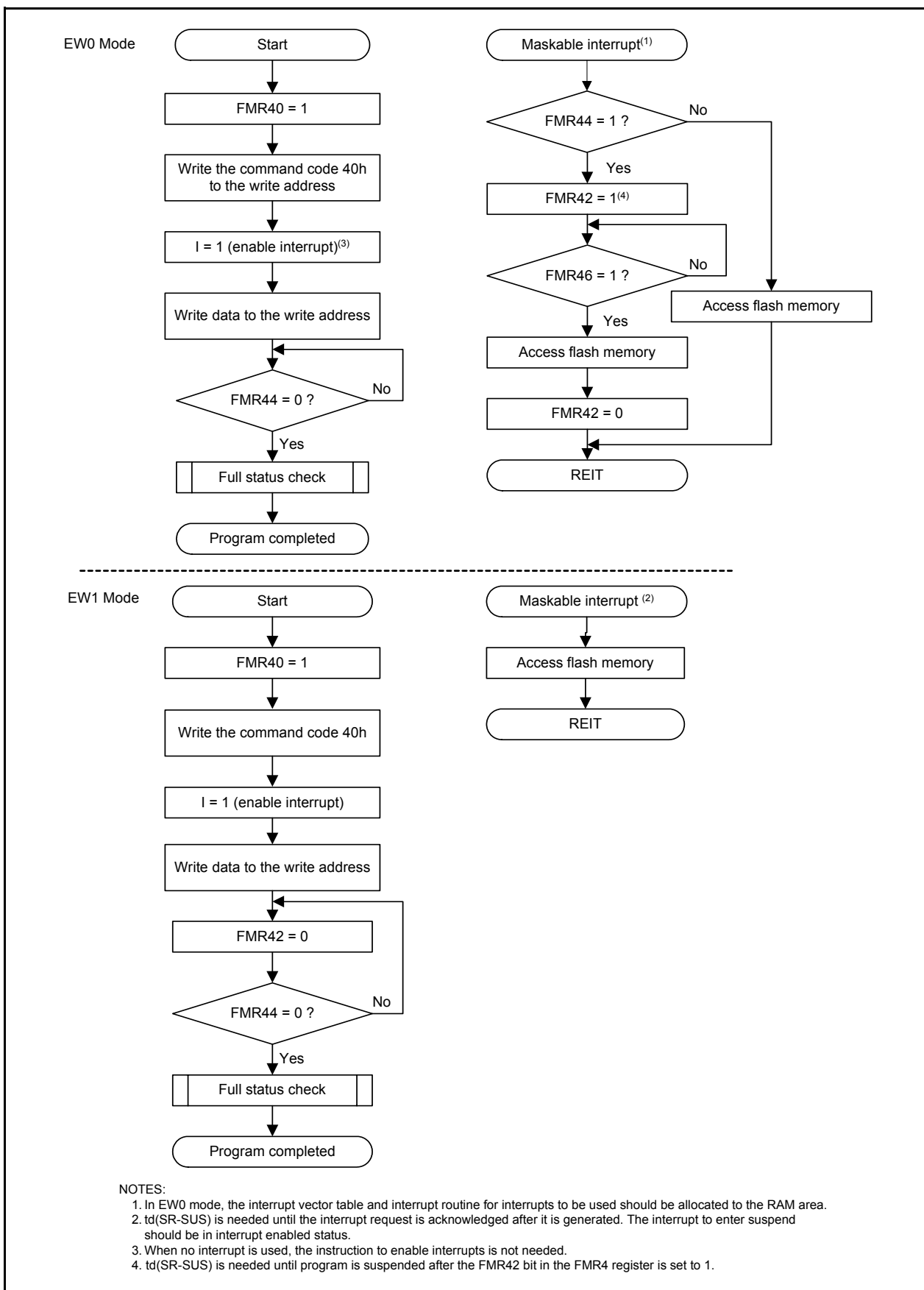


Figure 20.13 Program Command (When Suspend Function Enabled)

20.4.3.5 Block Erase

When 20h is written in the first bus cycle and D0h is written to a given address of a block in the second bus cycle, an auto-erase operation (erase and verify) of the specified block starts.

The FMR00 bit in the FMR0 register can be used to determine whether auto-erase has completed.

The FMR00 bit is set to 0 during auto-erase and set to 1 when auto-erase completes.

The FMR07 bit in the FMR0 register can be used to determine the result of auto-erase after auto-erase has completed (refer to **20.4.5 Full Status Check**).

When the FMR02 bit in the FMR0 register is set to 0 (rewriting disabled), the block erase commands targeting blocks 0 to 3 are not acknowledged. When the FMR02 bit is set to 1 (rewriting enabled) and the FMR15 bit in the FMR1 register is set to 1 (rewriting disabled), the block erase commands targeting block 0 are not acknowledged. When the FMR16 bit is set to 1 (rewriting disabled), block erase commands targeting block 1 are not acknowledged.

Do not use the block erase command during program-suspend.

Figure 20.14 shows the Block Erase Command (When Erase-Suspend Function Disabled). Figure 20.15 shows the Block Erase Command (When Erase-Suspend Function Enabled).

In EW1 mode, do not execute this command for any address to which a rewrite control program is allocated.

In EW0 mode, the MCU enters read status register mode at the same time auto-erase starts and the status register can be read. The status register bit 7 (SR7) is set to 0 at the same time auto-erase starts and set back to 1 when auto-erase completes. In this case, the MCU remains in read status register mode until the next read array command is written.

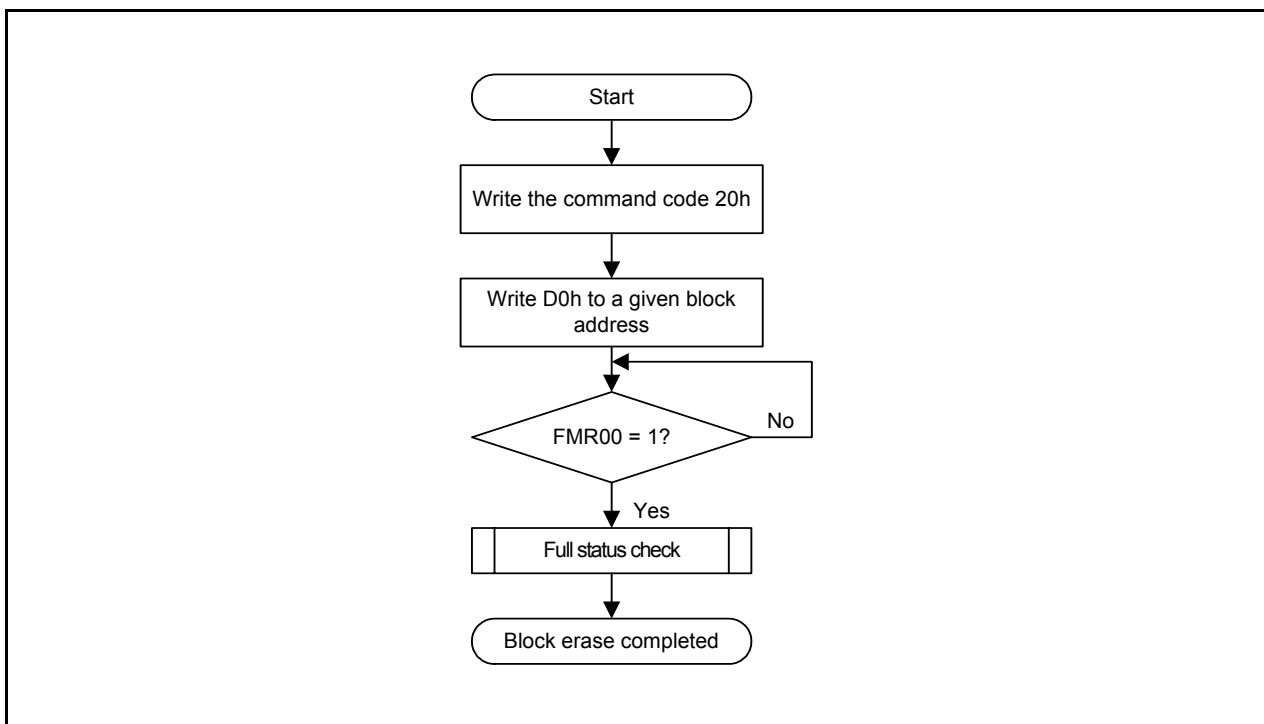


Figure 20.14 Block Erase Command (When Erase-Suspend Function Disabled)

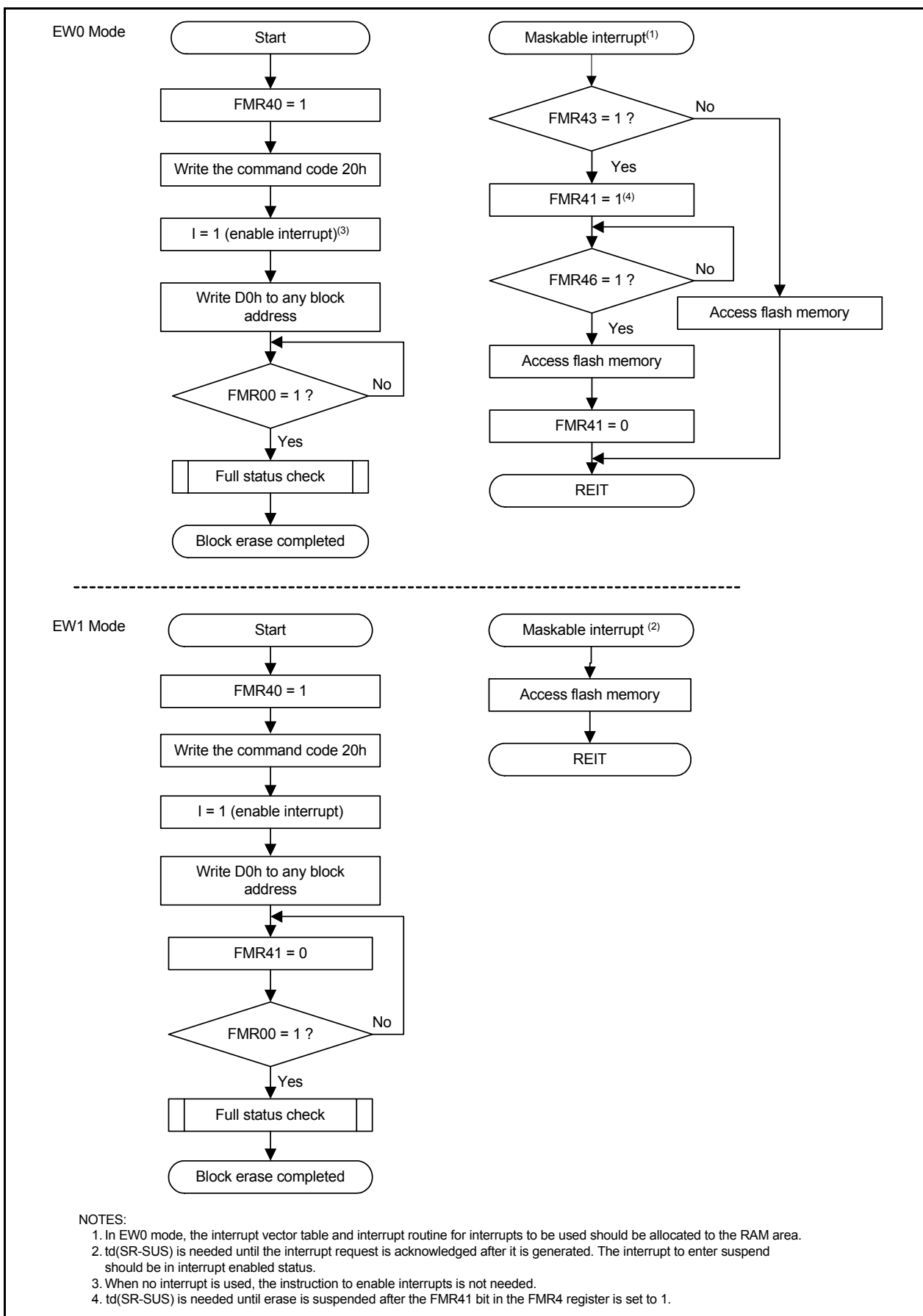


Figure 20.15 Block Erase Command (When Erase-Suspend Function Enabled)

20.4.4 Status Registers

The status register indicates the operating status of the flash memory and whether an erase or program operation has completed normally or in error. Status of the status register can be read by bits FMR00, FMR06, and FMR07 in the FMR0 register.

Table 20.5 lists the Status Register Bits.

In EW0 mode, the status register can be read in the following cases:

- When a given address in the user ROM area is read after writing the read status register command
- When a given address in the user ROM area is read after executing program or block erase command but before executing the read array command.

20.4.4.1 Sequencer Status (Bits SR7 and FMR00)

The sequencer status bits indicate the operating status of the flash memory. SR7 is set to 0 (busy) during auto-programming and auto-erasure, and is set to 1 (ready) at the same time the operation completes.

20.4.4.2 Erase Status (Bits SR5 and FMR07)

Refer to 20.4.5 Full Status Check.

20.4.4.3 Program Status (Bits SR4 and FMR06)

Refer to 20.4.5 Full Status Check.

Table 20.5 Status Register Bits

| Status Register Bit | FMR0 Register Bit | Status Name | Description | | Value After Reset |
|---------------------|-------------------|------------------|--------------------|-------|-------------------|
| | | | 0 | 1 | |
| SR0 (D0) | – | Reserved | – | – | – |
| SR1 (D1) | – | Reserved | – | – | – |
| SR2 (D2) | – | Reserved | – | – | – |
| SR3 (D3) | – | Reserved | – | – | – |
| SR4 (D4) | FMR06 | Program status | Completed normally | Error | 0 |
| SR5 (D5) | FMR07 | Erase status | Completed normally | Error | 0 |
| SR6 (D6) | – | Reserved | – | – | – |
| SR7 (D7) | FMR00 | Sequencer status | Busy | Ready | 1 |

D0 to D7: Indicate the data bus which is read when the read status register command is executed.

Bits FMR07 (SR5) to FMR06 (SR4) are set to 0 by executing the clear status register command.

When the FMR07 bit (SR5) or FMR06 bit (SR4) is set to 1, the program and block erase commands cannot be accepted.

20.4.5 Full Status Check

When an error occurs, bits FMR06 to FMR07 in the FMR0 register are set to 1, indicating the occurrence of an error. Therefore, checking these status bits (full status check) can be used to determine the execution result.

Table 20.6 lists the Errors and FMR0 Register Status. Figure 20.16 shows the Full Status Check and Handling Procedure for Individual Errors.

Table 20.6 Errors and FMR0 Register Status

| FRM0 Register (Status Register) Status | | Error | Error Occurrence Condition |
|--|------------|------------------------|---|
| FMR07(SR5) | FMR06(SR4) | | |
| 1 | 1 | Command sequence error | <ul style="list-style-type: none"> • When a command is not written correctly • When invalid data other than that which can be written in the second bus cycle of the block erase command is written (i.e., other than D0h or FFh)⁽¹⁾ • When the program command or block erase command is executed while rewriting is disabled by the FMR02 bit in the FMR0 register, or the FMR15 or FMR16 bit in the FMR1 register. • When an address not allocated in flash memory is input during erase command input • When attempting to erase the block for which rewriting is disabled during erase command input. • When an address not allocated in flash memory is input during write command input. • When attempting to write to a block for which rewriting is disabled during write command input. |
| 1 | 0 | Erase error | <ul style="list-style-type: none"> • When the block erase command is executed but auto-erasure does not complete correctly |
| 0 | 1 | Program error | <ul style="list-style-type: none"> • When the program command is executed but not auto-programming does not complete. |

NOTE:

1. The MCU enters read array mode when FFh is written in the second bus cycle of these commands. At the same time, the command code written in the first bus cycle is disabled.

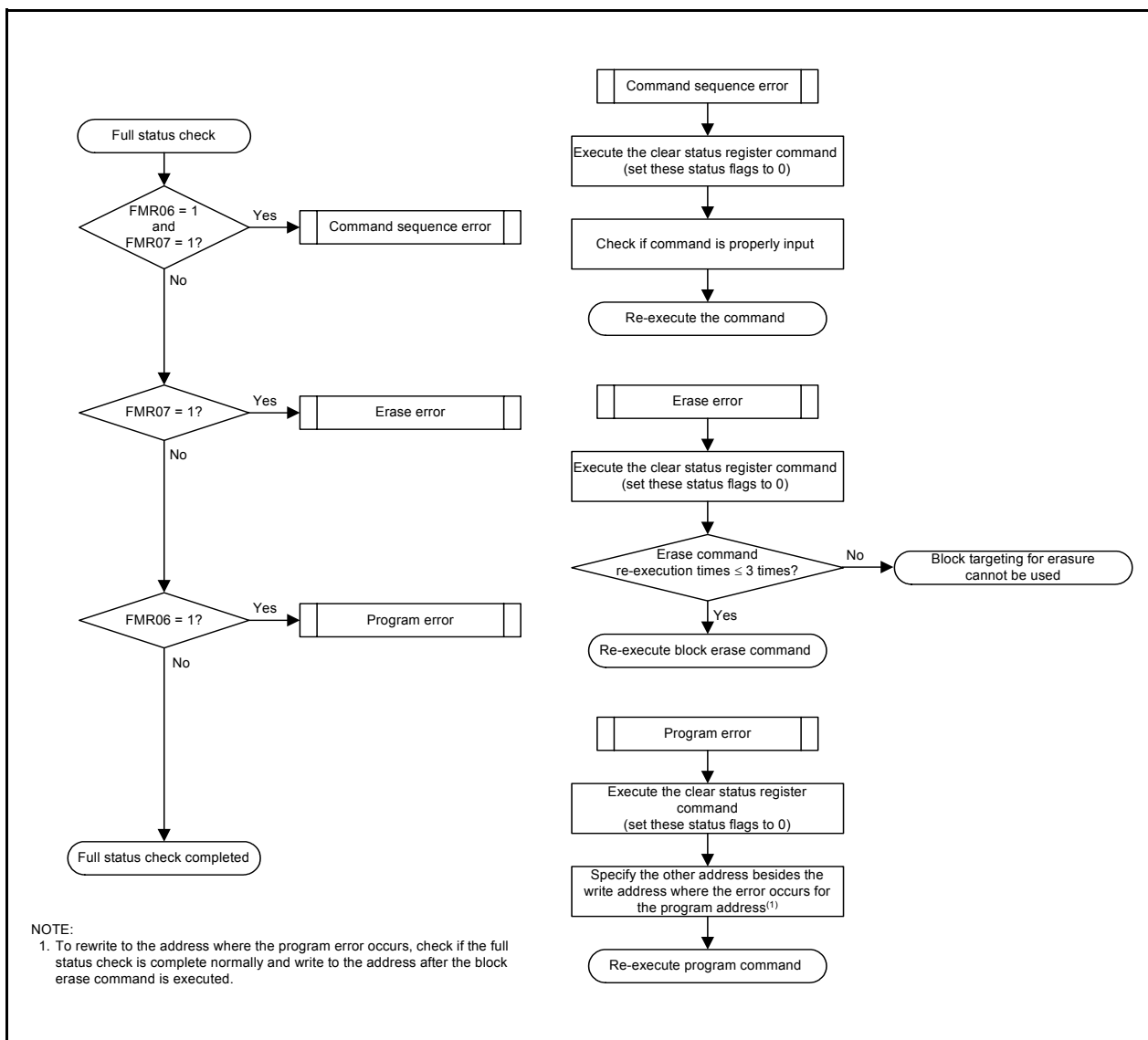


Figure 20.16 Full Status Check and Handling Procedure for Individual Errors

20.5 Standard Serial I/O Mode

In standard serial I/O mode, the user ROM area can be rewritten while the MCU is mounted on-board by using a serial programmer which is suitable for the MCU.

There are three types of Standard serial I/O modes:

- Standard serial I/O mode 1Clock synchronous serial I/O used to connect with a serial programmer
- Standard serial I/O mode 2Clock asynchronous serial I/O used to connect with a serial programmer
- Standard serial I/O mode 3Special clock asynchronous serial I/O used to connect with a serial programmer

This MCU uses Standard serial I/O mode 2 and Standard serial I/O mode 3.

Refer to **Appendix 2. Connection Examples between Serial Writer and On-Chip Debugging Emulator.** Contact the manufacturer of your serial programmer for details. Refer to the user's manual of your serial programmer for instructions on how to use it.

Table 20.7 lists the Pin Functions (Flash Memory Standard Serial I/O Mode 2), Table 20.8 lists the Pin Functions (Flash Memory Standard Serial I/O Mode 3), and Figure 20.17 shows Pin Connections for Standard Serial I/O Mode 3.

After processing the pins shown in Table 20.8 and rewriting the flash memory using the programmer, apply "H" to the MODE pin and reset the hardware to run a program in the flash memory in single-chip mode.

20.5.1 ID Code Check Function

The ID code check function determines whether the ID codes sent from the serial programmer and those written in the flash memory match (refer to **20.3 Functions to Prevent Rewriting of Flash Memory**).

Table 20.7 Pin Functions (Flash Memory Standard Serial I/O Mode 2)

| Pin | Name | I/O | Description |
|---------------------------|-------------------------|-----|---|
| VCC,VSS | Power input | | Apply the voltage guaranteed for programming and erasure to the VCC pin and 0 V to the VSS pin. |
| VREF | Reference voltage input | I | Reference voltage input pin to A/D converter and D/A converter. |
| $\overline{\text{RESET}}$ | Reset input | I | Reset input pin. |
| P4_6/XIN | P4_6 input/clock input | I | Connect a ceramic resonator or crystal oscillator between the XIN and XOUT pins. |
| P4_7/XOUT | P4_7 input/clock output | I/O | |
| P4_3/XCIN | P4_3 input/clock input | I | Connect crystal oscillator between pins XCIN and XCOUT. |
| P4_4/XCOUT | P4_4 input/clock output | I/O | |
| P0_0 to P0_7 | Input port P0 | I | Input "H" or "L" level signal or leave the pin open. |
| P1_0 to P1_7 | Input port P1 | I | |
| P2_0 to P2_7 | Input port P2 | I | |
| P3_0 to P3_7 | Input port P3 | I | |
| P4_5 | Input port P4 | I | |
| P5_0 to P5_4 | Input port P5 | I | |
| P6_0 to P6_5 | Input port P6 | I | |
| P8_0 to P8_6 | Input port P8 | I | |
| P6_6 | TXD output | O | Serial data output pin. |
| P6_7 | RXD input | I | Serial data input pin. |
| MODE | MODE | I | Input "L" level signal. |

Table 20.8 Pin Functions (Flash Memory Standard Serial I/O Mode 3)

| Pin | Name | I/O | Description |
|---------------------------|-------------------------|-----|--|
| VCC,VSS | Power input | | Apply the voltage guaranteed for programming and erasure to the VCC pin and 0 V to the VSS pin. |
| VREF | Reference voltage input | I | Reference voltage input pin to A/D converter and D/A converter. |
| $\overline{\text{RESET}}$ | Reset input | I | Reset input pin. |
| P4_6/XIN | P4_6 input/clock input | I | Connect a ceramic resonator or crystal oscillator between the XIN and XOUT pins when connecting external oscillator. Apply "H" and "L" or leave the pin open when using as input port. |
| P4_7/XOUT | P4_7 input/clock output | I/O | |
| P4_3/XCIN | P4_3 input/clock input | I | Connect crystal oscillator between pins XCIN and XCOOUT when connecting external oscillator. Apply "H" and "L" or leave the pin open when using as a port. |
| P4_4/XCOOUT | P4_4 input/clock output | I/O | |
| P0_0 to P0_7 | Input port P0 | I | Input "H" or "L" level signal or leave the pin open. |
| P1_0 to P1_7 | Input port P1 | I | |
| P2_0 to P2_7 | Input port P2 | I | |
| P3_0 to P3_7 | Input port P3 | I | |
| P4_5 | Input port P4 | I | |
| P5_0 to P5_4 | Input port P5 | I | |
| P6_0 to P6_7 | Input port P6 | I | |
| P8_0 to P8_6 | Input port P8 | I | |
| MODE | MODE | I/O | |

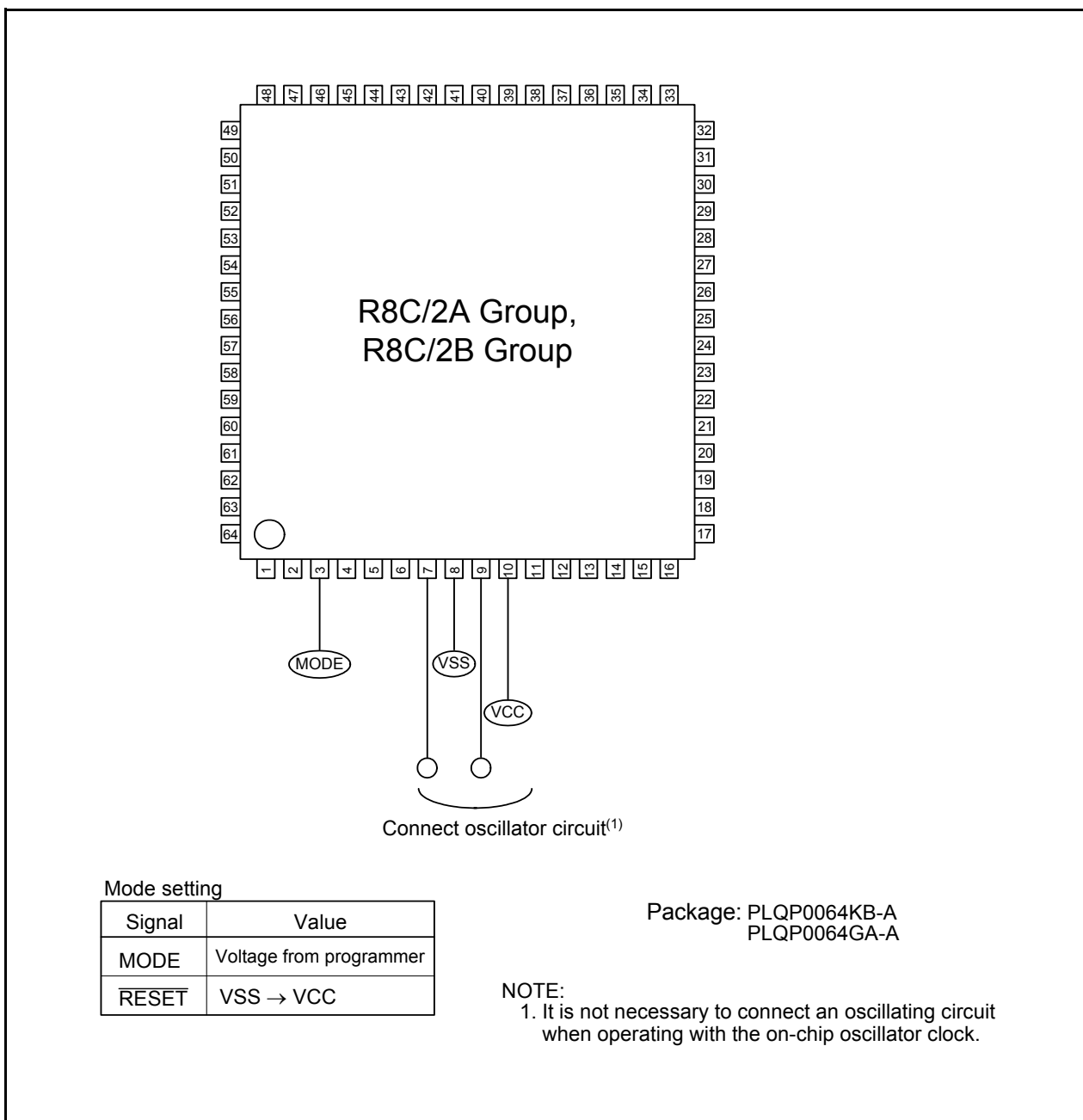


Figure 20.17 Pin Connections for Standard Serial I/O Mode 3

20.5.1.1 Example of Circuit Application in Standard Serial I/O Mode

Figure 20.18 shows an example of Pin Processing in Standard Serial I/O Mode 2 and Figure 20.19 shows an example of Pin Processing in Standard Serial I/O Mode 3. Since the controlled pins vary depending on the programmer, refer to the manual of your serial programmer for details.

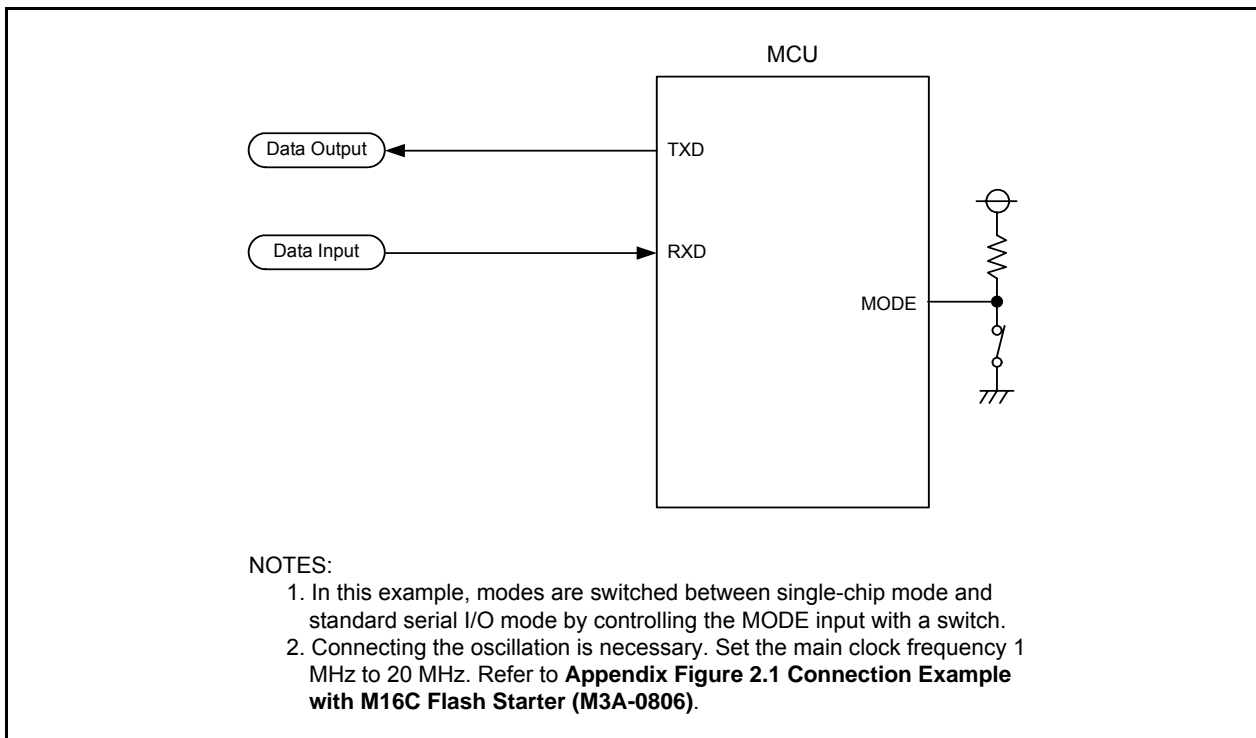


Figure 20.18 Pin Processing in Standard Serial I/O Mode 2

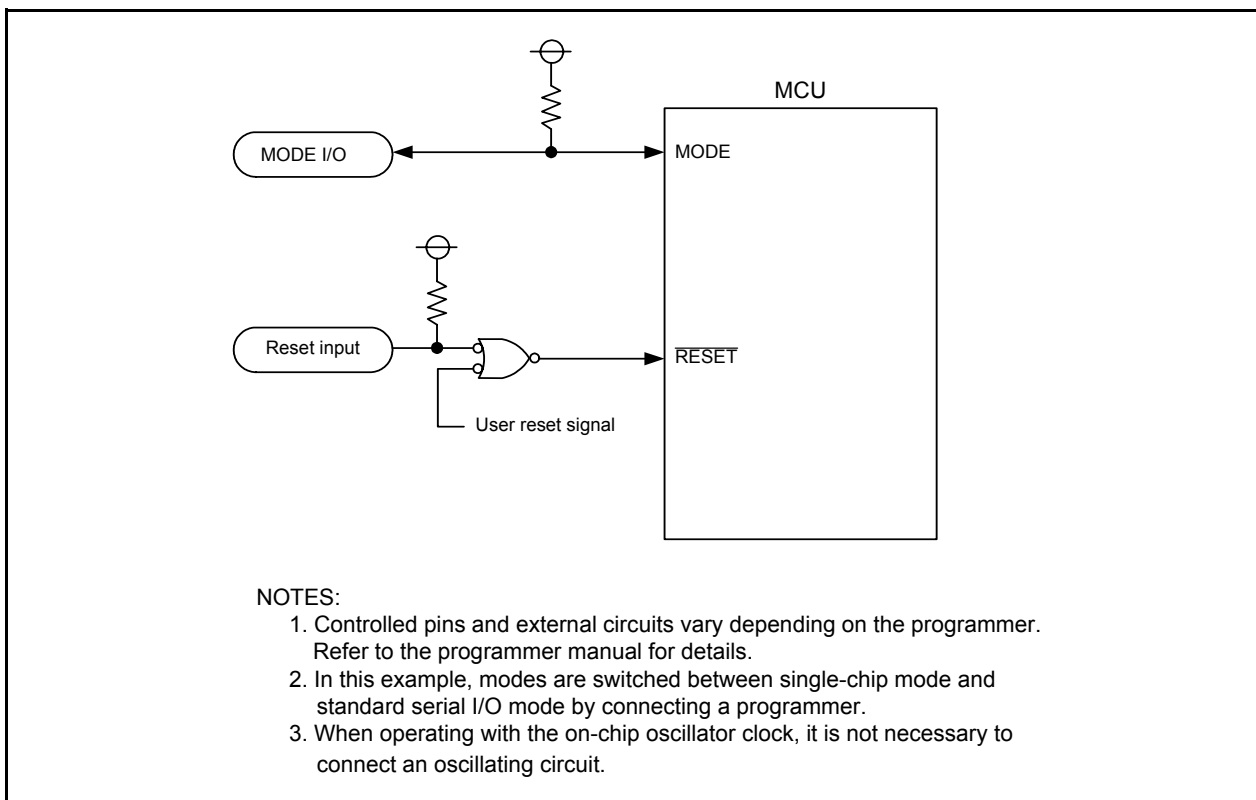


Figure 20.19 Pin Processing in Standard Serial I/O Mode 3

20.6 Parallel I/O Mode

Parallel I/O mode is used to input and output software commands, addresses and data necessary to control (read, program, and erase) the on-chip flash memory. Use a parallel programmer which supports this MCU. Contact the manufacturer of the parallel programmer for more information, and refer to the user's manual of the parallel programmer for details on how to use it.

ROM areas shown in Figures 20.1 and 20.2 can be rewritten in parallel I/O mode.

20.6.1 ROM Code Protect Function

The ROM code protect function disables the reading and rewriting of the flash memory. (Refer to the **20.3 Functions to Prevent Rewriting of Flash Memory.**)

20.7 Notes on Flash Memory

20.7.1 CPU Rewrite Mode

20.7.1.1 Operating Speed

Before entering CPU rewrite mode (EW0 mode), select 5 MHz or below for the CPU clock using the CM06 bit in the CM0 register and bits CM16 to CM17 in the CM1 register. This does not apply to EW1 mode.

20.7.1.2 Prohibited Instructions

The following instructions cannot be used in EW0 mode because they reference data in the flash memory: UND, INTO, and BRK.

20.7.1.3 Interrupts

Table 20.9 lists the EW0 Mode Interrupts and Table 20.10 lists the EW1 Mode Interrupts.

Table 20.9 EW0 Mode Interrupts

| Mode | Status | When Maskable Interrupt Request is Acknowledged | When Watchdog Timer, Oscillation Stop Detection, Voltage Monitor 1, or Voltage Monitor 2 Interrupt Request is Acknowledged |
|------|---------------------|---|--|
| EW0 | During auto-erasure | Any interrupt can be used by allocating a vector in RAM | Once an interrupt request is acknowledged, auto-programming or auto-erasure is forcibly stopped immediately and the flash memory is reset. Interrupt handling starts after the fixed period and the flash memory restarts. Since the block during auto-erasure or the address during auto-programming is forcibly stopped, the normal value may not be read. Execute auto-erasure again and ensure it completes normally. Since the watchdog timer does not stop during the command operation, interrupt requests may be generated. Reset the watchdog timer regularly. |
| | Auto-programming | | |

NOTES:

1. Do not use the address match interrupt while a command is being executed because the vector of the address match interrupt is allocated in ROM.
2. Do not use a non-maskable interrupt while block 0 is being automatically erased because the fixed vector is allocated in block 0.

Table 20.10 EW1 Mode Interrupts

| Mode | Status | When Maskable Interrupt Request is Acknowledged | When Watchdog Timer, Oscillation Stop Detection, Voltage Monitor 1, or Voltage Monitor 2 Interrupt Request is Acknowledged |
|------|---|--|--|
| EW1 | During auto-erasure (erase-suspend function enabled) | Auto-erasure is suspended after $t_d(SR-SUS)$ and interrupt handling is executed. Auto-erasure can be restarted by setting the FMR41 bit in the FMR4 register to 0 (erase restart) after interrupt handling completes. | Once an interrupt request is acknowledged, auto-programming or auto-erasure is forcibly stopped immediately and the flash memory is reset. Interrupt handling starts after the fixed period and the flash memory restarts. Since the block during auto-erasure or the address during auto-programming is forcibly stopped, the normal value may not be read. Execute auto-erasure again and ensure it completes normally. Since the watchdog timer does not stop during the command operation, interrupt requests may be generated. Reset the watchdog timer regularly using the erase-suspend function. |
| | During auto-erasure (erase-suspend function disabled) | Auto-erasure has priority and the interrupt request acknowledgement is put on standby. Interrupt handling is executed after auto-erasure completes. | |
| | During auto-programming (program suspend function enabled) | Auto-programming is suspended after $t_d(SR-SUS)$ and interrupt handling is executed. Auto-programming can be restarted by setting the FMR42 bit in the FMR4 register to 0 (program restart) after interrupt handling completes. | |
| | During auto-programming (program suspend function disabled) | Auto-programming has priority and the interrupt request acknowledgement is put on standby. Interrupt handling is executed after auto-programming completes. | |

NOTES:

1. Do not use the address match interrupt while a command is executing because the vector of the address match interrupt is allocated in ROM.
2. Do not use a non-maskable interrupt while block 0 is being automatically erased because the fixed vector is allocated in block 0.

20.7.1.4 How to Access

Write 0 before writing 1 when setting the FMR01, FMR02, or FMR11 bit to 1. Do not generate an interrupt between writing 0 and 1.

20.7.1.5 Rewriting User ROM Area

In EW0 Mode, if the supply voltage drops while rewriting any block in which a rewrite control program is stored, it may not be possible to rewrite the flash memory because the rewrite control program cannot be rewritten correctly. In this case, use standard serial I/O mode.

20.7.1.6 Program

Do not write additions to the already programmed address.

20.7.1.7 Entering Stop Mode or Wait Mode

Do not enter stop mode or wait mode during erase-suspend.

20.7.1.8 Program and Erase Voltage for Flash Memory

To perform programming and erasure, use $V_{CC} = 2.7\text{ V}$ to 5.5 V as the supply voltage. Do not perform programming and erasure at less than 2.7 V .

21. Electrical Characteristics

The electrical characteristics of N version ($T_{opr} = -20^{\circ}\text{C}$ to 85°C) and D version ($T_{opr} = -40^{\circ}\text{C}$ to 85°C) are listed below.

Please contact Renesas Technology sales offices for the electrical characteristics in the Y version ($T_{opr} = -20^{\circ}\text{C}$ to 105°C).

Table 21.1 Absolute Maximum Ratings

| Symbol | Parameter | Condition | Rated Value | Unit |
|------------------|-------------------------------|--------------------------------|--|--------------------|
| V_{CC}/AV_{CC} | Supply voltage | | -0.3 to 6.5 | V |
| V_I | Input voltage | | -0.3 to $V_{CC} + 0.3$ | V |
| V_O | Output voltage | | -0.3 to $V_{CC} + 0.3$ | V |
| P_d | Power dissipation | $T_{opr} = 25^{\circ}\text{C}$ | 700 | mW |
| T_{opr} | Operating ambient temperature | | -20 to 85 (N version) / -40 to 85 (D version) | $^{\circ}\text{C}$ |
| T_{stg} | Storage temperature | | -65 to 150 | $^{\circ}\text{C}$ |

Table 21.2 Recommended Operating Conditions

| Symbol | Parameter | | Conditions | Standard | | | Unit | |
|-----------------------------------|--|---|--|---------------------|------|---------------------|------|-----|
| | | | | Min. | Typ. | Max. | | |
| V _{CC} /AV _{CC} | Supply voltage | | | 2.2 | – | 5.5 | V | |
| V _{SS} /AV _{SS} | Supply voltage | | | – | 0 | – | V | |
| V _{IH} | Input “H” voltage | | | 0.8 V _{CC} | – | V _{CC} | V | |
| V _{IL} | Input “L” voltage | | | 0 | – | 0.2 V _{CC} | V | |
| I _{OH} (sum) | Peak sum output “H” current | Sum of all pins I _{OH} (peak) | | – | – | –240 | mA | |
| I _{OH} (sum) | Average sum output “H” current | Sum of all pins I _{OH} (avg) | | – | – | –120 | mA | |
| I _{OH} (peak) | Peak output “H” current | Except P2_0 to P2_7 | | – | – | –10 | mA | |
| | | P2_0 to P2_7 | | – | – | –40 | mA | |
| I _{OH} (avg) | Average output “H” current | Except P2_0 to P2_7 | | – | – | –5 | mA | |
| | | P2_0 to P2_7 | | – | – | –20 | mA | |
| I _{OL} (sum) | Peak sum output “L” current | Sum of all pins I _{OL} (peak) | | – | – | 240 | mA | |
| I _{OL} (sum) | Average sum output “L” current | Sum of all pins I _{OL} (avg) | | – | – | 120 | mA | |
| I _{OL} (peak) | Peak output “L” current | Except P2_0 to P2_7 | | – | – | 10 | mA | |
| | | P2_0 to P2_7 | | – | – | 40 | mA | |
| I _{OL} (avg) | Average output “L” current | Except P2_0 to P2_7 | | – | – | 5 | mA | |
| | | P2_0 to P2_7 | | – | – | 20 | mA | |
| f _(XIN) | XIN clock input oscillation frequency | | 3.0 V ≤ V _{CC} ≤ 5.5 V | 0 | – | 20 | MHz | |
| | | | 2.7 V ≤ V _{CC} < 3.0 V | 0 | – | 10 | MHz | |
| | | | 2.2 V ≤ V _{CC} < 2.7 V | 0 | – | 5 | MHz | |
| f _(XCIN) | XCIN clock input oscillation frequency | | 2.2 V ≤ V _{CC} ≤ 5.5 V | 0 | – | 70 | kHz | |
| | | | | | | | | |
| – | System clock | OCD2 = 0 XIN clock selected | 3.0 V ≤ V _{CC} ≤ 5.5 V | 0 | – | 20 | MHz | |
| | | | 2.7 V ≤ V _{CC} < 3.0 V | 0 | – | 10 | MHz | |
| | | | 2.2 V ≤ V _{CC} < 2.7 V | 0 | – | 5 | MHz | |
| | | OCD2 = 1 On-chip oscillator clock selected | FRA01 = 0 Low-speed on-chip oscillator clock selected | – | 125 | – | – | kHz |
| | | | FRA01 = 1 High-speed on-chip oscillator clock selected 3.0 V ≤ V _{CC} ≤ 5.5 V | – | – | 20 | MHz | |
| | | | FRA01 = 1 High-speed on-chip oscillator clock selected 2.7 V ≤ V _{CC} ≤ 5.5 V | – | – | 10 | MHz | |
| | | | FRA01 = 1 High-speed on-chip oscillator clock selected 2.2 V ≤ V _{CC} ≤ 5.5 V | – | – | 5 | MHz | |

NOTES:

- V_{CC} = 2.2 to 5.5 V at T_{opr} = -20 to 85°C (N version) / -40 to 85°C (D version), unless otherwise specified.
- The average output current indicates the average value of current measured during 100 ms.

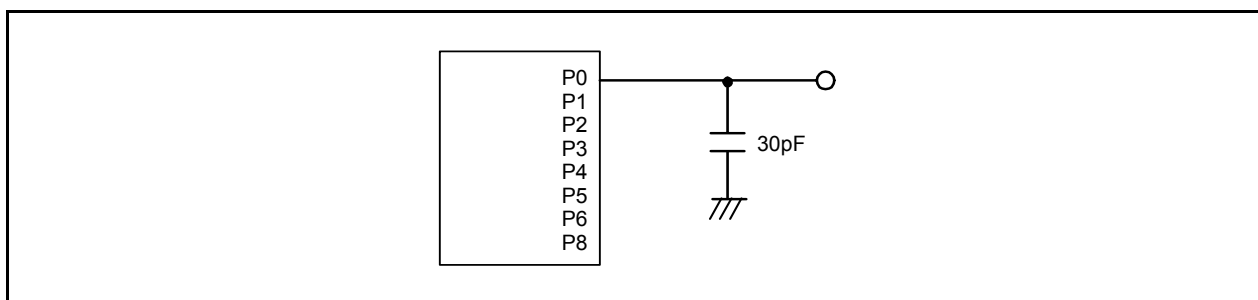
**Figure 21.1 Ports P0 to P6, P8 Timing Measurement Circuit**

Table 21.3 A/D Converter Characteristics(1)

| Symbol | Parameter | | Conditions | Standard | | | Unit |
|--------------|-------------------------------|-------------------------|---|----------|------|-----------|---------------|
| | | | | Min. | Typ. | Max. | |
| – | Resolution | | $V_{ref} = AV_{CC}$ | – | – | 10 | Bit |
| – | Absolute accuracy | 10-bit mode | $\phi_{AD} = 10 \text{ MHz}, V_{ref} = AV_{CC} = 5.0 \text{ V}$ | – | – | ± 3 | LSB |
| | | 8-bit mode | $\phi_{AD} = 10 \text{ MHz}, V_{ref} = AV_{CC} = 5.0 \text{ V}$ | – | – | ± 2 | LSB |
| | | 10-bit mode | $\phi_{AD} = 10 \text{ MHz}, V_{ref} = AV_{CC} = 3.3 \text{ V}$ | – | – | ± 5 | LSB |
| | | 8-bit mode | $\phi_{AD} = 10 \text{ MHz}, V_{ref} = AV_{CC} = 3.3 \text{ V}$ | – | – | ± 2 | LSB |
| | | 10-bit mode | $\phi_{AD} = 5 \text{ MHz}, V_{ref} = AV_{CC} = 2.2 \text{ V}$ | – | – | ± 5 | LSB |
| | | 8-bit mode | $\phi_{AD} = 5 \text{ MHz}, V_{ref} = AV_{CC} = 2.2 \text{ V}$ | – | – | ± 2 | LSB |
| R_{ladder} | Resistor ladder | | $V_{ref} = AV_{CC}$ | 10 | – | 40 | $k\Omega$ |
| t_{conv} | Conversion time | 10-bit mode | $\phi_{AD} = 10 \text{ MHz}, V_{ref} = AV_{CC} = 5.0 \text{ V}$ | 3.3 | – | – | μs |
| | | 8-bit mode | $\phi_{AD} = 10 \text{ MHz}, V_{ref} = AV_{CC} = 5.0 \text{ V}$ | 2.8 | – | – | μs |
| V_{ref} | Reference voltage | | | 2.2 | – | AV_{CC} | V |
| V_{IA} | Analog input voltage(2) | | | 0 | – | AV_{CC} | V |
| – | A/D operating clock frequency | Without sample and hold | $V_{ref} = AV_{CC} = 2.7 \text{ to } 5.5 \text{ V}$ | 0.25 | – | 10 | MHz |
| | | With sample and hold | $V_{ref} = AV_{CC} = 2.7 \text{ to } 5.5 \text{ V}$ | 1 | – | 10 | MHz |
| | | Without sample and hold | $V_{ref} = AV_{CC} = 2.2 \text{ to } 5.5 \text{ V}$ | 0.25 | – | 5 | MHz |
| | | With sample and hold | $V_{ref} = AV_{CC} = 2.2 \text{ to } 5.5 \text{ V}$ | 1 | – | 5 | MHz |

NOTES:

- $V_{CC}/AV_{CC} = V_{ref} = 2.2 \text{ to } 5.5 \text{ V}$ at $T_{opr} = -20 \text{ to } 85^\circ\text{C}$ (N version) / $-40 \text{ to } 85^\circ\text{C}$ (D version), unless otherwise specified.
- When the analog input voltage is over the reference voltage, the A/D conversion result will be 3FFh in 10-bit mode and FFh in 8-bit mode.

Table 21.4 D/A Converter Characteristics(1)

| Symbol | Parameter | | Conditions | Standard | | | Unit |
|------------|-------------------------------|--|------------|----------|------|------|---------------|
| | | | | Min. | Typ. | Max. | |
| – | Resolution | | | – | – | 8 | Bit |
| – | Absolute accuracy | | | – | – | 1.0 | % |
| t_{su} | Setup time | | | – | – | 3 | μs |
| R_o | Output resistor | | | 4 | 10 | 20 | $k\Omega$ |
| I_{vref} | Reference power input current | | (NOTE 2) | – | – | 1.5 | mA |

NOTES:

- $V_{CC}/AV_{CC} = V_{ref} = 2.7 \text{ to } 5.5 \text{ V}$ at $T_{opr} = -20 \text{ to } 85^\circ\text{C}$ (N version) / $-40 \text{ to } 85^\circ\text{C}$ (D version), unless otherwise specified.
- This applies when one D/A converter is used and the value of the DAI register ($i = 0$ or 1) for the unused D/A converter is 00h. The resistor ladder of the A/D converter is not included. Also, even if the VCUT bit in the ADCON1 register is set to 0 (V_{REF} not connected), I_{vref} flows into the D/A converters.

Table 21.5 Flash Memory (Program ROM) Electrical Characteristics

| Symbol | Parameter | Conditions | Standard | | | Unit |
|-------------------------|---|----------------------------|----------------------|------|----------------------------|-------|
| | | | Min. | Typ. | Max. | |
| – | Program/erase endurance ⁽²⁾ | R8C/2A Group | 100 ⁽³⁾ | – | – | times |
| | | R8C/2B Group | 1,000 ⁽³⁾ | – | – | times |
| – | Byte program time | | – | 50 | 400 | μs |
| – | Block erase time | | – | 0.4 | 9 | s |
| t _d (SR-SUS) | Time delay from suspend request until suspend | | – | – | 97+CPU clock × 6 cycles | μs |
| – | Interval from erase start/restart until following suspend request | | 650 | – | – | μs |
| – | Interval from program start/restart until following suspend request | | 0 | – | – | ns |
| – | Time from suspend until program/erase restart | | – | – | 3+CPU clock × 4 cycles | μs |
| – | Program, erase voltage | | 2.7 | – | 5.5 | V |
| – | Read voltage | | 2.2 | – | 5.5 | V |
| – | Program, erase temperature | | 0 | – | 60 | °C |
| – | Data hold time ⁽⁷⁾ | Ambient temperature = 55°C | 20 | – | – | year |

NOTES:

1. V_{CC} = 2.7 to 5.5 V at T_{opr} = 0 to 60°C, unless otherwise specified.
2. 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 = 100 or 10,000), each block can be erased n times. For example, if 1,024 1-byte writes are performed to block A, a 1 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).
3. Endurance to guarantee all electrical characteristics after program and erase. (1 to Min. value can be guaranteed).
4. In a system that executes multiple programming operations, the actual erasure count can be reduced by writing to sequential addresses in turn so that as much of the block as possible is used up before performing an erase operation. For example, when programming groups of 16 bytes, the effective number of rewrites can be minimized by programming up to 128 groups before erasing them all in one operation. It is also advisable to retain data on the erase count of each block and limit the number of erase operations to a certain number.
5. If an error occurs during block erase, attempt to execute the clear status register command, then execute the block erase command at least three times until the erase error does not occur.
6. Customers desiring program/erase failure rate information should contact their Renesas technical support representative.
7. The data hold time includes time that the power supply is off or the clock is not supplied.

Table 21.6 Flash Memory (Data flash Block A, Block B) Electrical Characteristics⁽⁴⁾

| Symbol | Parameter | Conditions | Standard | | | Unit |
|------------------------|---|-----------------------------|-----------------------|------|----------------------------|-------|
| | | | Min. | Typ. | Max. | |
| – | Program/erase endurance ⁽²⁾ | | 10,000 ⁽³⁾ | – | – | times |
| – | Byte program time (program/erase endurance ≤ 1,000 times) | | – | 50 | 400 | μs |
| – | Byte program time (program/erase endurance > 1,000 times) | | – | 65 | – | μs |
| – | Block erase time (program/erase endurance ≤ 1,000 times) | | – | 0.2 | 9 | s |
| – | Block erase time (program/erase endurance > 1,000 times) | | – | 0.3 | – | s |
| t _{d(SR-SUS)} | Time delay from suspend request until suspend | | – | – | 97+CPU clock × 6 cycles | μs |
| – | Interval from erase start/restart until following suspend request | | 650 | – | – | μs |
| – | Interval from program start/restart until following suspend request | | 0 | – | – | ns |
| – | Time from suspend until program/erase restart | | – | – | 3+CPU clock × 4 cycles | μs |
| – | Program, erase voltage | | 2.7 | – | 5.5 | V |
| – | Read voltage | | 2.2 | – | 5.5 | V |
| – | Program, erase temperature | | -20 ⁽⁸⁾ | – | 85 | °C |
| – | Data hold time ⁽⁹⁾ | Ambient temperature = 55 °C | 20 | – | – | year |

NOTES:

- V_{CC} = 2.7 to 5.5 V at T_{opr} = -20 to 85°C (N version) / -40 to 85°C (D version), unless otherwise specified.
- 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 = 100 or 10,000), each block can be erased n times. For example, if 1,024 1-byte writes are performed to block A, a 1 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).
- Endurance to guarantee all electrical characteristics after program and erase. (1 to Min. value can be guaranteed).
- Standard of block A and block B when program and erase endurance exceeds 1,000 times. Byte program time to 1,000 times is the same as that in program ROM.
- In a system that executes multiple programming operations, the actual erasure count can be reduced by writing to sequential addresses in turn so that as much of the block as possible is used up before performing an erase operation. For example, when programming groups of 16 bytes, the effective number of rewrites can be minimized by programming up to 128 groups before erasing them all in one operation. It is also advisable to retain data on the erase count of each block and limit the number of erase operations to a certain number.
- If an error occurs during block erase, attempt to execute the clear status register command, then execute the block erase command at least three times until the erase error does not occur.
- Customers desiring program/erase failure rate information should contact their Renesas technical support representative.
- 40°C for D version.
- The data hold time includes time that the power supply is off or the clock is not supplied.

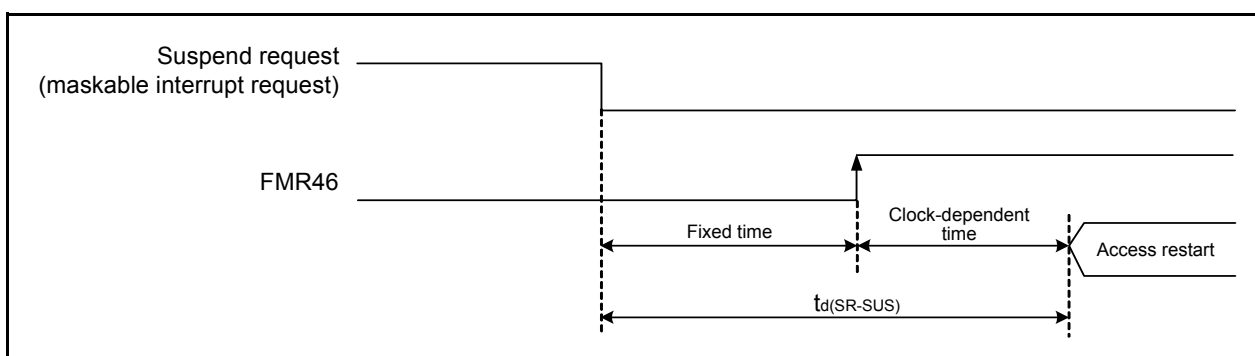


Figure 21.2 Time delay until Suspend

Table 21.7 Voltage Detection 0 Circuit Electrical Characteristics

| Symbol | Parameter | Condition | Standard | | | Unit |
|---------------------|--|------------------------------------|----------|------|------|------|
| | | | Min. | Typ. | Max. | |
| V _{det0} | Voltage detection level | | 2.2 | 2.3 | 2.4 | V |
| – | Voltage detection circuit self power consumption | VCA25 = 1, V _{CC} = 5.0 V | – | 0.9 | – | μA |
| t _{d(E-A)} | Waiting time until voltage detection circuit operation starts ⁽²⁾ | | – | – | 300 | μs |
| V _{ccmin} | MCU operating voltage minimum value | | 2.2 | – | – | V |

NOTES:

1. The measurement condition is V_{CC} = 2.2 V to 5.5 V and T_{opr} = -20 to 85°C (N version) / -40 to 85°C (D version).
2. Necessary time until the voltage detection circuit operates when setting to 1 again after setting the VCA25 bit in the VCA2 register to 0.

Table 21.8 Voltage Detection 1 Circuit Electrical Characteristics

| Symbol | Parameter | Condition | Standard | | | Unit |
|---------------------|--|------------------------------------|----------|------|------|------|
| | | | Min. | Typ. | Max. | |
| V _{det1} | Voltage detection level | | 2.70 | 2.85 | 3.00 | V |
| – | Voltage monitor 1 interrupt request generation time ⁽²⁾ | | – | 40 | – | μs |
| – | Voltage detection circuit self power consumption | VCA26 = 1, V _{CC} = 5.0 V | – | 0.6 | – | μA |
| t _{d(E-A)} | Waiting time until voltage detection circuit operation starts ⁽³⁾ | | – | – | 100 | μs |

NOTES:

1. The measurement condition is V_{CC} = 2.2 V to 5.5 V and T_{opr} = -20 to 85°C (N version) / -40 to 85°C (D version).
2. Time until the voltage monitor 1 interrupt request is generated after the voltage passes V_{det1}.
3. Necessary time until the voltage detection circuit operates when setting to 1 again after setting the VCA26 bit in the VCA2 register to 0.

Table 21.9 Voltage Detection 2 Circuit Electrical Characteristics

| Symbol | Parameter | Condition | Standard | | | Unit |
|---------------------|--|------------------------------------|----------|------|------|------|
| | | | Min. | Typ. | Max. | |
| V _{det2} | Voltage detection level | | 3.3 | 3.6 | 3.9 | V |
| – | Voltage monitor 2 interrupt request generation time ⁽²⁾ | | – | 40 | – | μs |
| – | Voltage detection circuit self power consumption | VCA27 = 1, V _{CC} = 5.0 V | – | 0.6 | – | μA |
| t _{d(E-A)} | Waiting time until voltage detection circuit operation starts ⁽³⁾ | | – | – | 100 | μs |

NOTES:

1. The measurement condition is V_{CC} = 2.2 V to 5.5 V and T_{opr} = -20 to 85°C (N version) / -40 to 85°C (D version).
2. Time until the voltage monitor 2 interrupt request is generated after the voltage passes V_{det2}.
3. Necessary time until the voltage detection circuit operates after setting to 1 again after setting the VCA27 bit in the VCA2 register to 0.

Table 21.10 Power-on Reset Circuit, Voltage Monitor 0 Reset Electrical Characteristics⁽³⁾

| Symbol | Parameter | Condition | Standard | | | Unit |
|-------------------|---|-----------|----------|------|-------------------|---------|
| | | | Min. | Typ. | Max. | |
| V _{por1} | Power-on reset valid voltage ⁽⁴⁾ | | – | – | 0.1 | V |
| V _{por2} | Power-on reset or voltage monitor 0 reset valid voltage | | 0 | – | V _{det0} | V |
| t _{rth} | External power V _{CC} rise gradient ⁽²⁾ | | 20 | – | – | mV/msec |

NOTES:

- The measurement condition is T_{opr} = -20 to 85°C (N version) / -40 to 85°C (D version), unless otherwise specified.
- This condition (external power V_{CC} rise gradient) does not apply if V_{CC} ≥ 1.0 V.
- To use the power-on reset function, enable voltage monitor 0 reset by setting the LVD0ON bit in the OFS register to 0, the VW0C0 and VW0C6 bits in the VW0C register to 1 respectively, and the VCA25 bit in the VCA2 register to 1.
- t_{w(por1)} indicates the duration the external power V_{CC} must be held below the effective voltage (V_{por1}) to enable a power on reset. When turning on the power for the first time, maintain t_{w(por1)} for 30 s or more if -20°C ≤ T_{opr} ≤ 85°C, maintain t_{w(por1)} for 3,000 s or more if -40°C ≤ T_{opr} < -20°C.

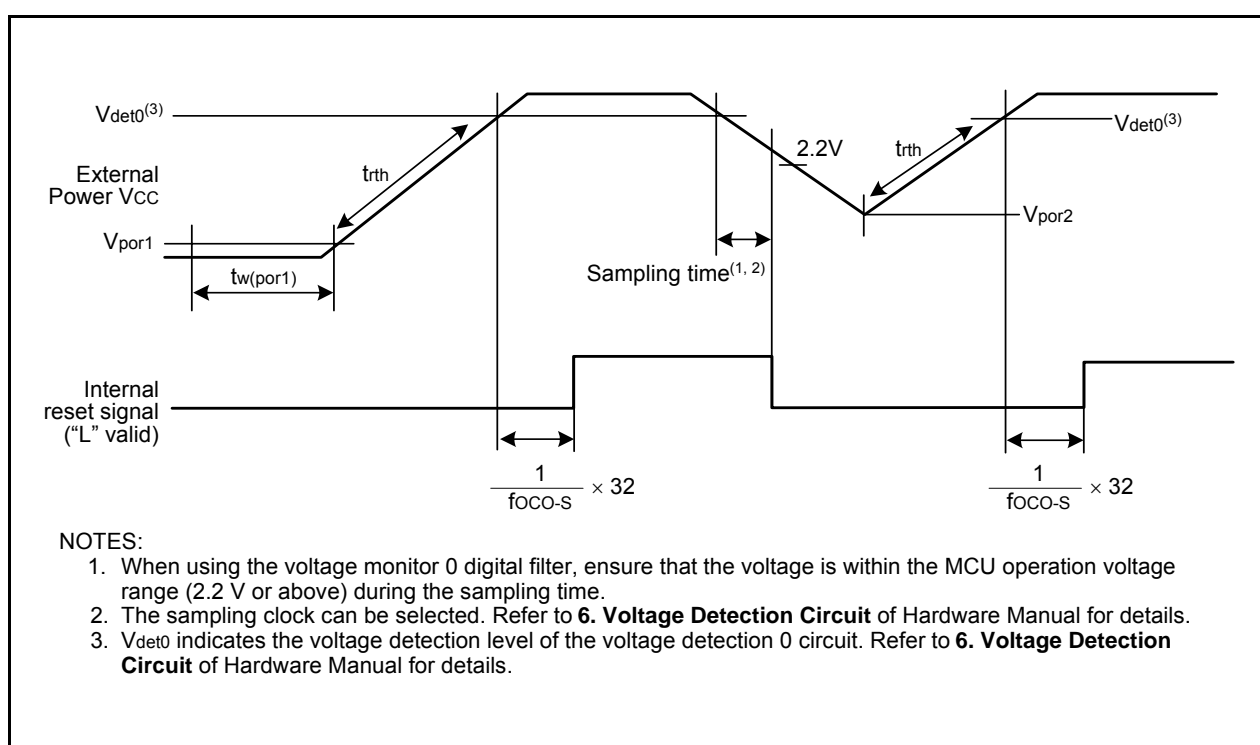
**Figure 21.3 Power-on Reset Circuit Electrical Characteristics**

Table 21.11 High-speed On-Chip Oscillator Circuit Electrical Characteristics

| Symbol | Parameter | Condition | Standard | | | Unit |
|---------|--|--|----------|--------|------|------|
| | | | Min. | Typ. | Max. | |
| fOCO40M | High-speed on-chip oscillator frequency temperature • supply voltage dependence | V _{CC} = 2.7 V to 5.5 V -20°C ≤ T _{opr} ≤ 85°C ⁽²⁾ | 39.2 | 40 | 40.8 | MHz |
| | | V _{CC} = 2.7 V to 5.5 V -40°C ≤ T _{opr} ≤ 85°C ⁽²⁾ | 39.0 | 40 | 41.0 | MHz |
| | | V _{CC} = 2.2 V to 5.5 V -20°C ≤ T _{opr} ≤ 85°C ⁽³⁾ | 35.2 | 40 | 44.8 | MHz |
| | | V _{CC} = 2.2 V to 5.5 V -40°C ≤ T _{opr} ≤ 85°C ⁽³⁾ | 34.0 | 40 | 46.0 | MHz |
| - | High-speed on-chip oscillator frequency when correction value in FRA7 register is written to FRA1 register | V _{CC} = 5.0 V, T _{opr} = 25°C | - | 36.864 | - | MHz |
| | | V _{CC} = 2.7 V to 5.5 V -20°C ≤ T _{opr} ≤ 85°C | -3% | - | 3% | % |
| - | Value in FRA1 register after reset | | 08h | - | F7h | - |
| - | Oscillation frequency adjustment unit of high- speed on-chip oscillator | Adjust FRA1 register (value after reset) to -1 | - | +0.3 | - | MHz |
| - | Oscillation stability time | V _{CC} = 5.0 V, T _{opr} = 25°C | - | 10 | 100 | μs |
| - | Self power consumption at oscillation | V _{CC} = 5.0 V, T _{opr} = 25°C | - | 550 | - | μA |

NOTES:

- V_{CC} = 2.2 to 5.5 V, T_{opr} = -20 to 85°C (N version) / -40 to 85°C (D version), unless otherwise specified.
- These standard values show when the FRA1 register value after reset is assumed.
- These standard values show when the correction value in the FRA6 register is written to the FRA1 register.

Table 21.12 Low-speed On-Chip Oscillator Circuit Electrical Characteristics

| Symbol | Parameter | Condition | Standard | | | Unit |
|--------|--|--|----------|------|------|------|
| | | | Min. | Typ. | Max. | |
| fOCO-S | Low-speed on-chip oscillator frequency | | 30 | 125 | 250 | kHz |
| - | Oscillation stability time | V _{CC} = 5.0 V, T _{opr} = 25°C | - | 10 | 100 | μs |
| - | Self power consumption at oscillation | V _{CC} = 5.0 V, T _{opr} = 25°C | - | 15 | - | μA |

NOTE:

- V_{CC} = 2.2 to 5.5 V, T_{opr} = -20 to 85°C (N version) / -40 to 85°C (D version), unless otherwise specified.

Table 21.13 Power Supply Circuit Timing Characteristics

| Symbol | Parameter | Condition | Standard | | | Unit |
|----------------------|--|-----------|----------|------|------|------|
| | | | Min. | Typ. | Max. | |
| t _d (P-R) | Time for internal power supply stabilization during power-on ⁽²⁾ | | 1 | - | 2000 | μs |
| t _d (R-S) | STOP exit time ⁽³⁾ | | - | - | 150 | μs |

NOTES:

- The measurement condition is V_{CC} = 2.2 to 5.5 V and T_{opr} = 25°C.
- Waiting time until the internal power supply generation circuit stabilizes during power-on.
- Time until system clock supply starts after the interrupt is acknowledged to exit stop mode.

Table 21.14 Timing Requirements of Clock Synchronous Serial I/O with Chip Select⁽¹⁾

| Symbol | Parameter | | Conditions | Standard | | | Unit |
|--------|------------------------------------|--------|---|------------|------|------------------------|---------------------|
| | | | | Min. | Typ. | Max. | |
| tsucyc | SSCK clock cycle time | | | 4 | – | – | tcyc ⁽²⁾ |
| tHI | SSCK clock "H" width | | | 0.4 | – | 0.6 | tsucyc |
| tLO | SSCK clock "L" width | | | 0.4 | – | 0.6 | tsucyc |
| tRISE | SSCK clock rising time | Master | | – | – | 1 | tcyc ⁽²⁾ |
| | | Slave | | – | – | 1 | μs |
| tFALL | SSCK clock falling time | Master | | – | – | 1 | tcyc ⁽²⁾ |
| | | Slave | | – | – | 1 | μs |
| tsu | SSO, SSI data input setup time | | | 100 | – | – | ns |
| tH | SSO, SSI data input hold time | | | 1 | – | – | tcyc ⁽²⁾ |
| tLEAD | $\overline{\text{SCS}}$ setup time | Slave | | 1tcyc + 50 | – | – | ns |
| tLAG | $\overline{\text{SCS}}$ hold time | Slave | | 1tcyc + 50 | – | – | ns |
| tOD | SSO, SSI data output delay time | | | – | – | 1 | tcyc ⁽²⁾ |
| tsa | SSI slave access time | | $2.7 \text{ V} \leq V_{\text{CC}} \leq 5.5 \text{ V}$ | – | – | $1.5\text{tcyc} + 100$ | ns |
| | | | $2.2 \text{ V} \leq V_{\text{CC}} < 2.7 \text{ V}$ | – | – | $1.5\text{tcyc} + 200$ | ns |
| tor | SSI slave out open time | | $2.7 \text{ V} \leq V_{\text{CC}} \leq 5.5 \text{ V}$ | – | – | $1.5\text{tcyc} + 100$ | ns |
| | | | $2.2 \text{ V} \leq V_{\text{CC}} < 2.7 \text{ V}$ | – | – | $1.5\text{tcyc} + 200$ | ns |

NOTES:

1. $V_{\text{CC}} = 2.2$ to 5.5 V , $V_{\text{SS}} = 0 \text{ V}$ at $T_{\text{opr}} = -20$ to 85°C (N version) / -40 to 85°C (D version), unless otherwise specified.
2. $1\text{tcyc} = 1/f_1(\text{s})$

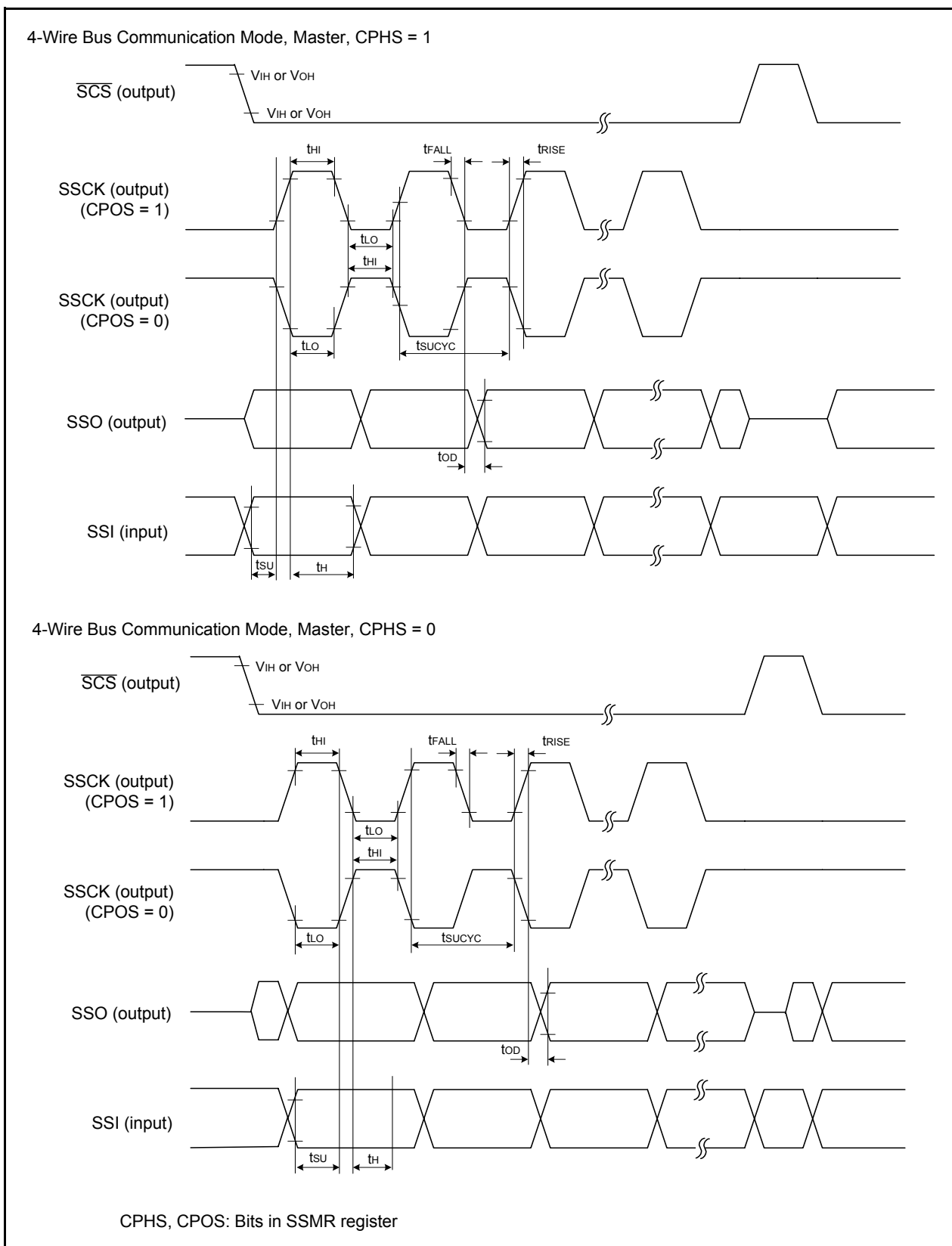


Figure 21.4 I/O Timing of Clock Synchronous Serial I/O with Chip Select (Master)

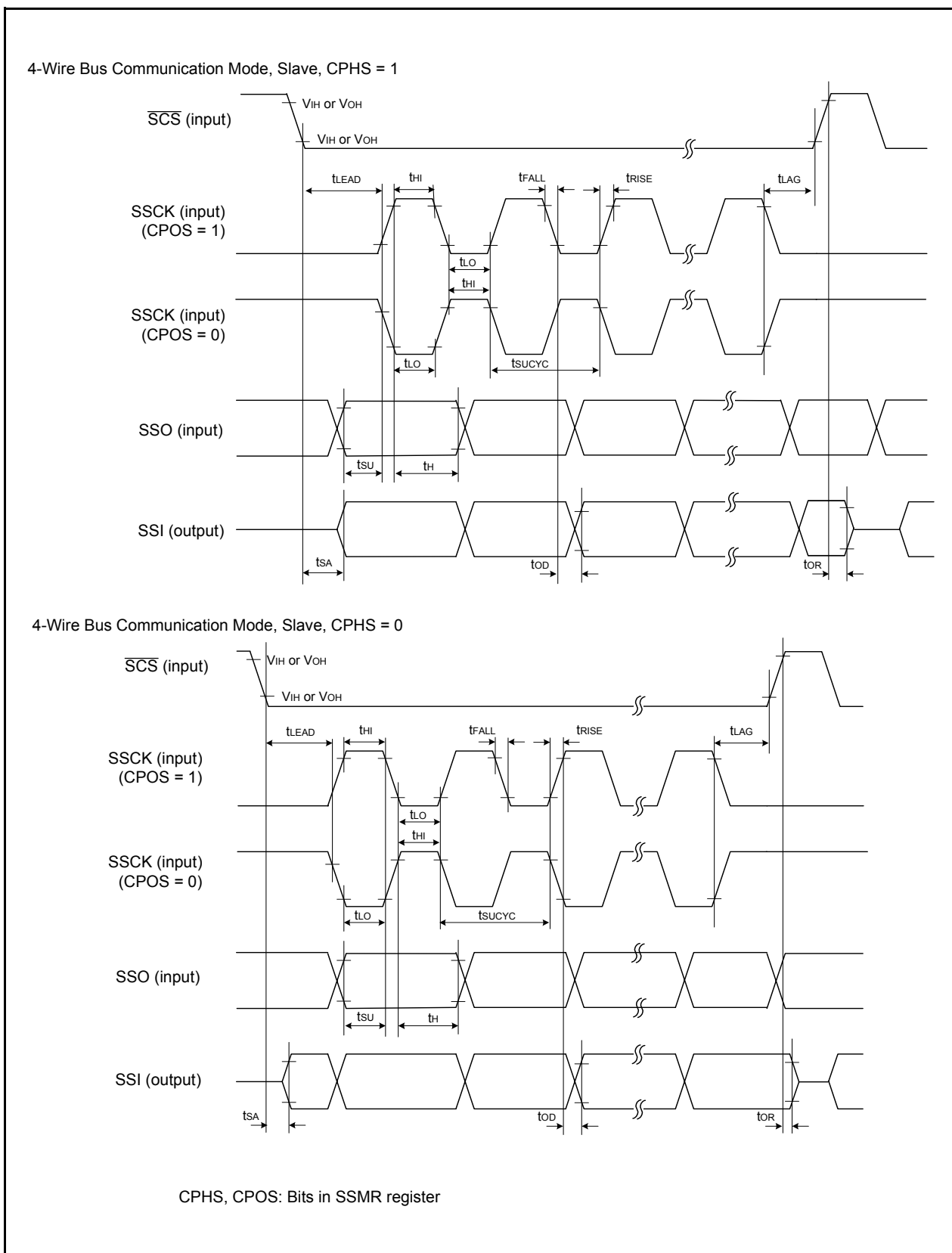


Figure 21.5 I/O Timing of Clock Synchronous Serial I/O with Chip Select (Slave)

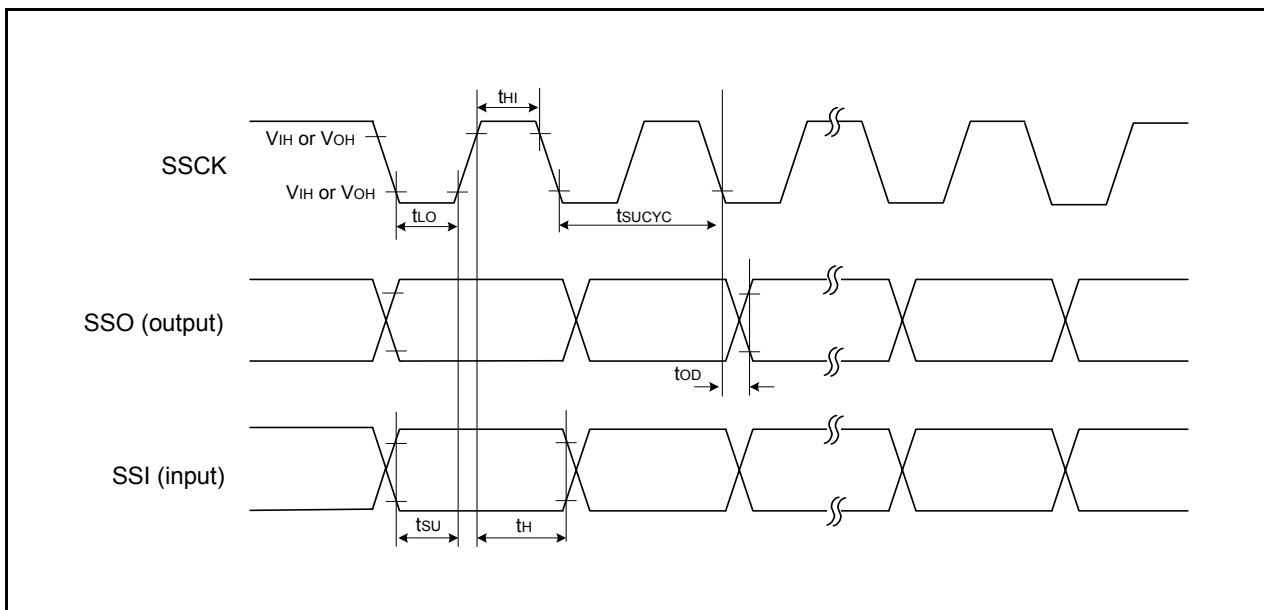


Figure 21.6 I/O Timing of Clock Synchronous Serial I/O with Chip Select (Clock Synchronous Communication Mode)

Table 21.15 Timing Requirements of I²C bus Interface (1)

| Symbol | Parameter | Condition | Standard | | | Unit |
|-------------------|---|-----------|-----------------------------|------|----------------------|------|
| | | | Min. | Typ. | Max. | |
| t _{SCL} | SCL input cycle time | | 12tcyc + 600 ⁽²⁾ | – | – | ns |
| t _{SCLH} | SCL input “H” width | | 3tcyc + 300 ⁽²⁾ | – | – | ns |
| t _{SCLL} | SCL input “L” width | | 5tcyc + 500 ⁽²⁾ | – | – | ns |
| t _{sf} | SCL, SDA input fall time | | – | – | 300 | ns |
| t _{SP} | SCL, SDA input spike pulse rejection time | | – | – | 1tcyc ⁽²⁾ | ns |
| t _{BUF} | SDA input bus-free time | | 5tcyc ⁽²⁾ | – | – | ns |
| t _{STAH} | Start condition input hold time | | 3tcyc ⁽²⁾ | – | – | ns |
| t _{STAS} | Retransmit start condition input setup time | | 3tcyc ⁽²⁾ | – | – | ns |
| t _{STOP} | Stop condition input setup time | | 3tcyc ⁽²⁾ | – | – | ns |
| t _{SDAS} | Data input setup time | | 1tcyc + 20 ⁽²⁾ | – | – | ns |
| t _{SDAH} | Data input hold time | | 0 | – | – | ns |

NOTES:

1. V_{CC} = 2.2 to 5.5 V, V_{SS} = 0 V and T_{opr} = -20 to 85°C (N version) / -40 to 85°C (D version), unless otherwise specified.
2. 1tcyc = 1/f₁(s)

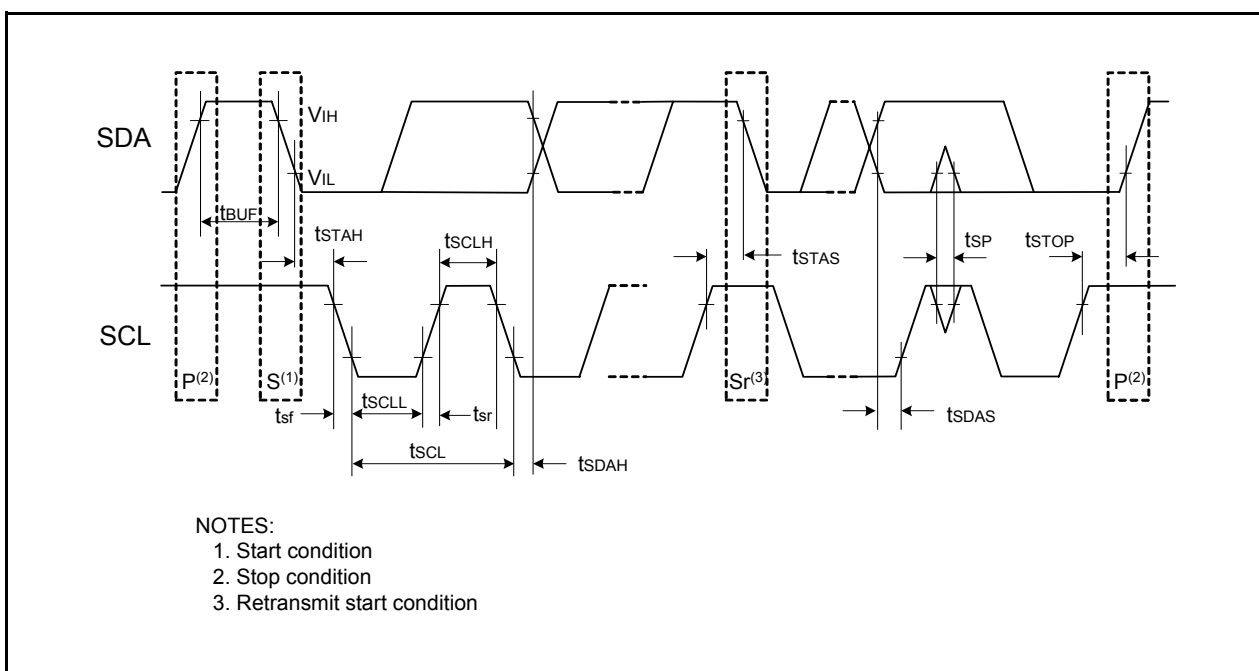


Figure 21.7 I/O Timing of I²C bus Interface

Table 21.16 Electrical Characteristics (1) [V_{CC} = 5 V]

| Symbol | Parameter | | Condition | Standard | | | Unit | |
|----------------------------------|---------------------|---|---------------------------|---------------------------|-----------------------|------|-----------------|----|
| | | | | Min. | Typ. | Max. | | |
| V _{OH} | Output "H" voltage | Except P2_0 to P2_7, XOUT | I _{OH} = -5 mA | | V _{CC} - 2.0 | - | V _{CC} | V |
| | | | I _{OH} = -200 μA | | V _{CC} - 0.5 | - | V _{CC} | V |
| | P2_0 to P2_7 | | Drive capacity HIGH | I _{OH} = -20 mA | V _{CC} - 2.0 | - | V _{CC} | V |
| | | | Drive capacity LOW | I _{OH} = -5 mA | V _{CC} - 2.0 | - | V _{CC} | V |
| | XOUT | | Drive capacity HIGH | I _{OH} = -1 mA | V _{CC} - 2.0 | - | V _{CC} | V |
| | | | Drive capacity LOW | I _{OH} = -500 μA | V _{CC} - 2.0 | - | V _{CC} | V |
| V _{OL} | Output "L" voltage | Except P2_0 to P2_7, XOUT | I _{OL} = 5 mA | | - | - | 2.0 | V |
| | | | I _{OL} = 200 μA | | - | - | 0.45 | V |
| | P2_0 to P2_7 | | Drive capacity HIGH | I _{OL} = 20 mA | - | - | 2.0 | V |
| | | | Drive capacity LOW | I _{OL} = 5 mA | - | - | 2.0 | V |
| | XOUT | | Drive capacity HIGH | I _{OL} = 1 mA | - | - | 2.0 | V |
| | | | Drive capacity LOW | I _{OL} = 500 μA | - | - | 2.0 | V |
| V _{T+} -V _{T-} | Hysteresis | INT0, INT1, INT2, INT3, KI0, KI1, KI2, KI3, TRAIO, TRFI, RXD0, RXD1, CLK0, CLK1, CLK2, SSI, SCL, SDA, SSO | | | 0.1 | 0.5 | - | V |
| | | RESET | | | 0.1 | 1.0 | - | V |
| I _{IH} | Input "H" current | | V _I = 5 V | | - | - | 5.0 | μA |
| I _{IL} | Input "L" current | | V _I = 0 V | | - | - | -5.0 | μA |
| R _{PULLUP} | Pull-up resistance | | V _I = 0 V | | 30 | 50 | 167 | kΩ |
| R _{FXIN} | Feedback resistance | XIN | | | - | 1.0 | - | MΩ |
| R _{FXCIN} | Feedback resistance | XCIN | | | - | 18 | - | MΩ |
| V _{RAM} | RAM hold voltage | | During stop mode | | 1.8 | - | - | V |

NOTE:

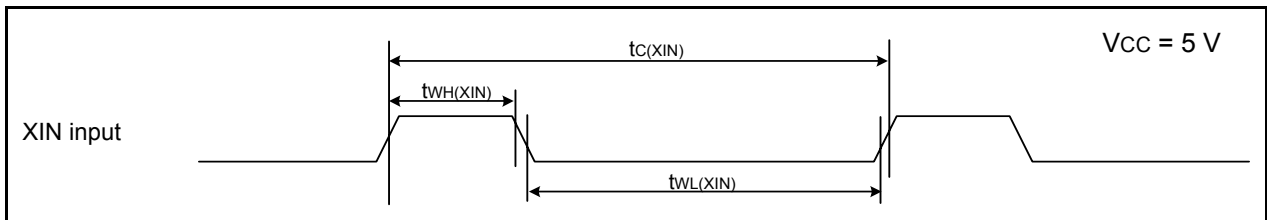
- V_{CC} = 4.2 to 5.5 V at T_{opr} = -20 to 85°C (N version) / -40 to 85°C (D version), f(XIN) = 20 MHz, unless otherwise specified.

**Table 21.17 Electrical Characteristics (2) [V_{CC} = 5 V]
(T_{opr} = -20 to 85°C (N version) / -40 to 85°C (D version), unless otherwise specified.)**

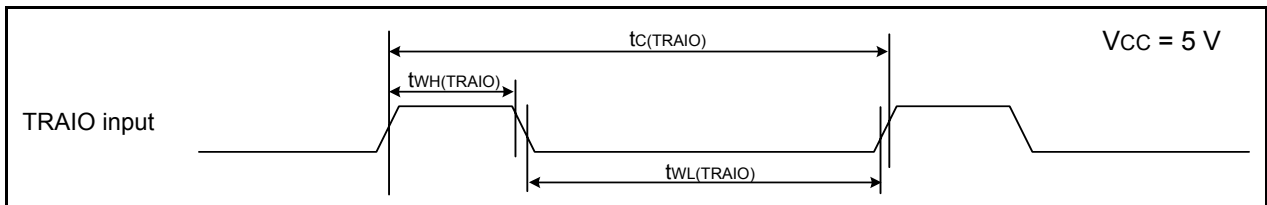
| Symbol | Parameter | Condition | Standard | | | Unit | |
|-----------------|---|--|--|------|------|------|----|
| | | | Min. | Typ. | Max. | | |
| I _{CC} | Power supply current (V _{CC} = 3.3 to 5.5 V) Single-chip mode, output pins are open, other pins are V _{SS} | High-speed clock mode | XIN = 20 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz No division | – | 12 | 20 | mA |
| | | | XIN = 16 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz No division | – | 10 | 16 | mA |
| | | | XIN = 10 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz No division | – | 7 | – | mA |
| | | | XIN = 20 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz Divide-by-8 | – | 5.5 | – | mA |
| | | | XIN = 16 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz Divide-by-8 | – | 4.5 | – | mA |
| | | | XIN = 10 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz Divide-by-8 | – | 3 | – | mA |
| | | High-speed on-chip oscillator mode | XIN clock off High-speed on-chip oscillator on f _{OCO} = 10 MHz Low-speed on-chip oscillator on = 125 kHz No division | – | 6 | 12 | mA |
| | | | XIN clock off High-speed on-chip oscillator on f _{OCO} = 10 MHz Low-speed on-chip oscillator on = 125 kHz Divide-by-8 | – | 2.5 | – | mA |
| | | Low-speed on-chip oscillator mode | XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz Divide-by-8, FMR47 = 1 | – | 150 | 400 | μA |
| | | Low-speed clock mode | XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator off XCIN clock oscillator on = 32 kHz FMR47 = 1 | – | 150 | 400 | μA |
| | | | XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator off XCIN clock oscillator on = 32 kHz Program operation on RAM Flash memory off, FMSTP = 1 | – | 35 | – | μA |
| | | Wait mode | XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz While a WAIT instruction is executed Peripheral clock operation VCA27 = VCA26 = VCA25 = 0 VCA20 = 1 | – | 30 | 90 | μA |
| | | | XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz While a WAIT instruction is executed Peripheral clock off VCA27 = VCA26 = VCA25 = 0 VCA20 = 1 | – | 18 | 55 | μA |
| | | | XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator off XCIN clock oscillator on = 32 kHz (high drive) While a WAIT instruction is executed VCA27 = VCA26 = VCA25 = 0 VCA20 = 1 | – | 3.5 | – | μA |
| | | | XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator off XCIN clock oscillator on = 32 kHz (low drive) While a WAIT instruction is executed VCA27 = VCA26 = VCA25 = 0 VCA20 = 1 | – | 2.3 | – | μA |
| | | | XIN clock off, T _{opr} = 25°C High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA27 = VCA26 = VCA25 = 0 | – | 0.7 | 3.0 | μA |
| | | Stop mode | XIN clock off, T _{opr} = 85°C High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA27 = VCA26 = VCA25 = 0 | – | 1.7 | – | μA |

Timing Requirements**(Unless Otherwise Specified: $V_{CC} = 5\text{ V}$, $V_{SS} = 0\text{ V}$ at $T_{op} = 25^\circ\text{C}$) [$V_{CC} = 5\text{ V}$]****Table 21.18 XIN Input, XCIN Input**

| Symbol | Parameter | Standard | | Unit |
|----------------|-----------------------|----------|------|---------------|
| | | Min. | Max. | |
| $t_{c(XIN)}$ | XIN input cycle time | 50 | – | ns |
| $t_{WH(XIN)}$ | XIN input “H” width | 25 | – | ns |
| $t_{WL(XIN)}$ | XIN input “L” width | 25 | – | ns |
| $t_{c(XCIN)}$ | XCIN input cycle time | 14 | – | μs |
| $t_{WH(XCIN)}$ | XCIN input “H” width | 7 | – | μs |
| $t_{WL(XCIN)}$ | XCIN input “L” width | 7 | – | μs |

**Figure 21.8 XIN Input and XCIN Input Timing Diagram when $V_{CC} = 5\text{ V}$** **Table 21.19 TRAIO Input, $\overline{\text{INT1}}$ Input**

| Symbol | Parameter | Standard | | Unit |
|------------------------|------------------------|----------|------|------|
| | | Min. | Max. | |
| $t_{c(\text{TRAIO})}$ | TRAIO input cycle time | 100 | – | ns |
| $t_{WH(\text{TRAIO})}$ | TRAIO input “H” width | 40 | – | ns |
| $t_{WL(\text{TRAIO})}$ | TRAIO input “L” width | 40 | – | ns |

**Figure 21.9 TRAIO Input and $\overline{\text{INT1}}$ Input Timing Diagram when $V_{CC} = 5\text{ V}$** **Table 21.20 TRFI Input**

| Symbol | Parameter | Standard | | Unit |
|-----------------------|-----------------------|--------------------|------|------|
| | | Min. | Max. | |
| $t_{c(\text{TRFI})}$ | TRFI input cycle time | 400 ⁽¹⁾ | – | ns |
| $t_{WH(\text{TRFI})}$ | TRFI input “H” width | 200 ⁽²⁾ | – | ns |
| $t_{WL(\text{TRFI})}$ | TRFI input “L” width | 200 ⁽²⁾ | – | ns |

NOTES:

1. When using timer RF input capture mode, adjust the cycle time to $(1/\text{timer RF count source frequency} \times 3)$ or above.
2. When using timer RF input capture mode, adjust the pulse width to $(1/\text{timer RF count source frequency} \times 1.5)$ or above.

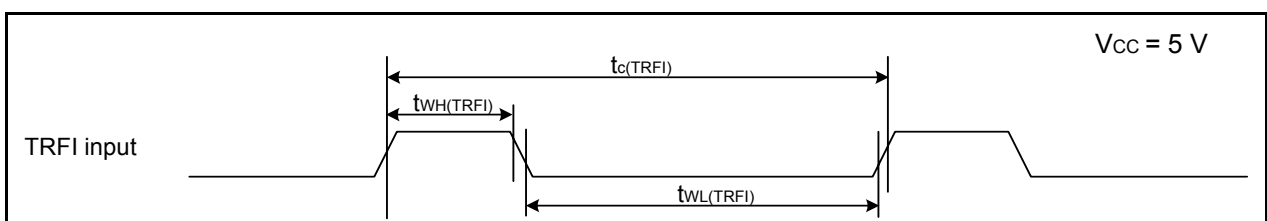
**Figure 21.10 TRFI Input Timing Diagram when $V_{CC} = 5\text{ V}$**

Table 21.21 Serial Interface

| Symbol | Parameter | Standard | | Unit |
|---------------|------------------------|----------|------|------|
| | | Min. | Max. | |
| $t_{c(CK)}$ | CLKi input cycle time | 200 | – | ns |
| $t_{w(CKH)}$ | CLKi input “H” width | 100 | – | ns |
| $t_{w(CKL)}$ | CLKi input “L” width | 100 | – | ns |
| $t_{d(C-Q)}$ | TXDi output delay time | – | 50 | ns |
| $t_{h(C-Q)}$ | TXDi hold time | 0 | – | ns |
| $t_{su(D-C)}$ | RXDi input setup time | 50 | – | ns |
| $t_{h(C-D)}$ | RXDi input hold time | 90 | – | ns |

$i = 0$ to 2

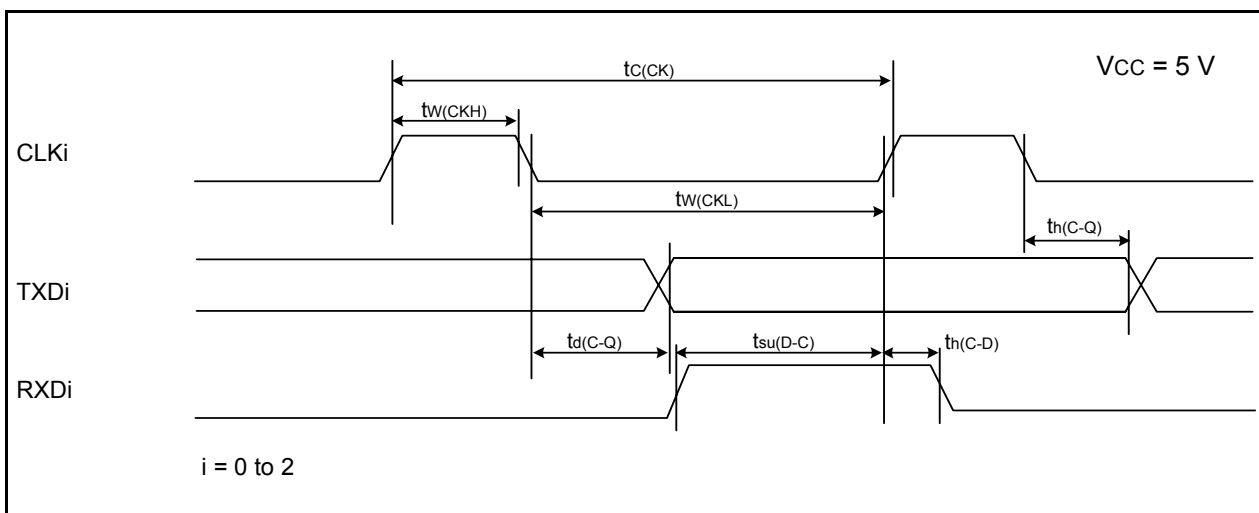


Figure 21.11 Serial Interface Timing Diagram when Vcc = 5 V

Table 21.22 External Interrupt \overline{INTi} ($i = 0, 2, 3$) Input

| Symbol | Parameter | Standard | | Unit |
|--------------|-----------------------------------|--------------------|------|------|
| | | Min. | Max. | |
| $t_{w(INH)}$ | $\overline{INT0}$ input “H” width | 250 ⁽¹⁾ | – | ns |
| $t_{w(INL)}$ | $\overline{INT0}$ input “L” width | 250 ⁽²⁾ | – | ns |

NOTES:

1. When selecting the digital filter by the \overline{INTi} input filter select bit, use an \overline{INTi} input HIGH width of either (1/digital filter clock frequency × 3) or the minimum value of standard, whichever is greater.
2. When selecting the digital filter by the \overline{INTi} input filter select bit, use an \overline{INTi} input LOW width of either (1/digital filter clock frequency × 3) or the minimum value of standard, whichever is greater.

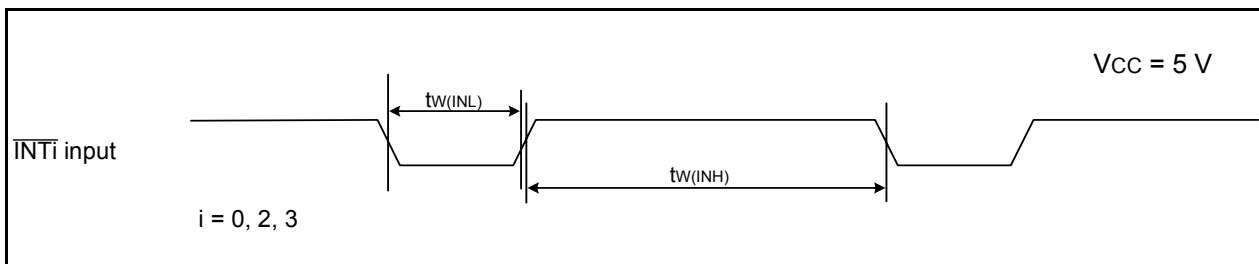


Figure 21.12 External Interrupt \overline{INTi} Input Timing Diagram when Vcc = 5 V

Table 21.23 Electrical Characteristics (3) [V_{CC} = 3 V]

| Symbol | Parameter | | Condition | | Standard | | | Unit |
|----------------------------------|---------------------|--|-------------------------|---------------------------|-----------------------|------|-----------------|------|
| | | | | | Min. | Typ. | Max. | |
| V _{OH} | Output "H" voltage | Except P2_0 to P2_7, XOUT | I _{OH} = -1 mA | | V _{CC} - 0.5 | - | V _{CC} | V |
| | | P2_0 to P2_7 | Drive capacity HIGH | I _{OH} = -5 mA | V _{CC} - 0.5 | - | V _{CC} | V |
| | | | Drive capacity LOW | I _{OH} = -1 mA | V _{CC} - 0.5 | - | V _{CC} | V |
| | | XOUT | Drive capacity HIGH | I _{OH} = -0.1 mA | V _{CC} - 0.5 | - | V _{CC} | V |
| | | | Drive capacity LOW | I _{OH} = -50 μA | V _{CC} - 0.5 | - | V _{CC} | V |
| V _{OL} | Output "L" voltage | Except P2_0 to P2_7, XOUT | I _{OL} = 1 mA | | - | - | 0.5 | V |
| | | P2_0 to P2_7 | Drive capacity HIGH | I _{OL} = 5 mA | - | - | 0.5 | V |
| | | | Drive capacity LOW | I _{OL} = 1 mA | - | - | 0.5 | V |
| | | XOUT | Drive capacity HIGH | I _{OL} = 0.1 mA | - | - | 0.5 | V |
| | | | Drive capacity LOW | I _{OL} = 50 μA | - | - | 0.5 | V |
| V _{T+} -V _{T-} | Hysteresis | $\overline{\text{INT0}}, \overline{\text{INT1}}, \overline{\text{INT2}}, \overline{\text{INT3}}, \text{KI0}, \text{KI1}, \text{KI2}, \text{KI3}, \text{TRAIO}, \text{TRFI}, \text{RXD0}, \text{RXD1}, \text{CLK0}, \text{CLK1}, \text{CLK2}, \text{SSI}, \text{SCL}, \text{SDA}, \text{SSO}$ | | | 0.1 | 0.3 | - | V |
| | | $\overline{\text{RESET}}$ | | | 0.1 | 0.4 | - | V |
| I _{IH} | Input "H" current | | V _I = 3 V | | - | - | 4.0 | μA |
| I _{IL} | Input "L" current | | V _I = 0 V | | - | - | -4.0 | μA |
| R _{PULLUP} | Pull-up resistance | | V _I = 0 V | | 66 | 160 | 500 | kΩ |
| R _{fXIN} | Feedback resistance | XIN | | | - | 3.0 | - | MΩ |
| R _{fXCIN} | Feedback resistance | XCIN | | | - | 18 | - | MΩ |
| V _{RAM} | RAM hold voltage | | During stop mode | | 1.8 | - | - | V |

NOTE:

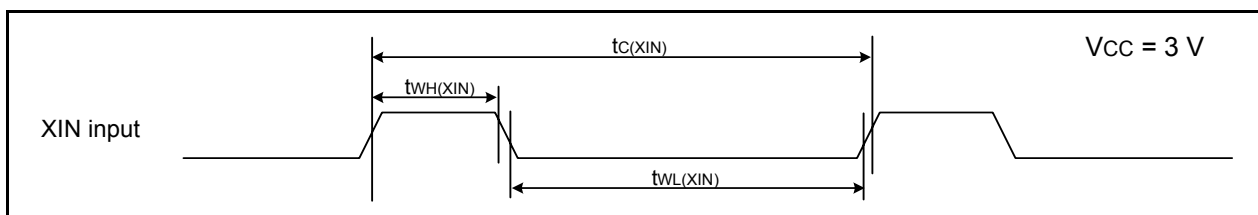
- V_{CC} = 2.7 to 3.3 V at T_{opr} = -20 to 85°C (N version) / -40 to 85°C (D version), f(XIN) = 10 MHz, unless otherwise specified.

**Table 21.24 Electrical Characteristics (4) [V_{CC} = 3 V]
(T_{opr} = -20 to 85°C (N version) / -40 to 85°C (D version), unless otherwise specified.)**

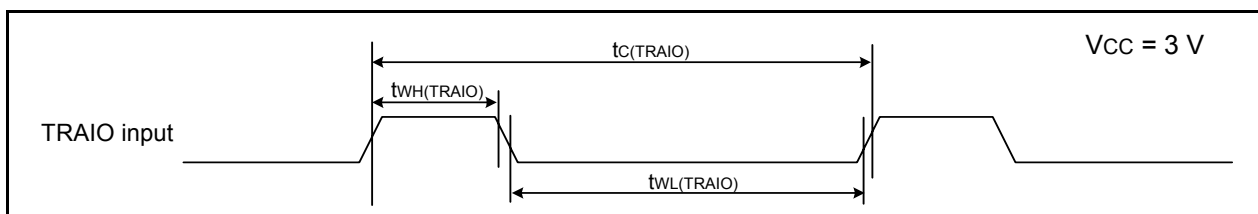
| Symbol | Parameter | Condition | Standard | | | Unit | |
|-----------------|--|---|--|------|------|------|----|
| | | | Min. | Typ. | Max. | | |
| I _{CC} | Power supply current (V _{CC} = 2.7 to 3.3 V) Single-chip mode, output pins are open, other pins are V _{SS} | High-speed clock mode | XIN = 10 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz No division | – | 5.5 | – | mA |
| | | | XIN = 10 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz Divide-by-8 | – | 2 | – | mA |
| | | High-speed on-chip oscillator mode | XIN clock off High-speed on-chip oscillator on fOCO = 10 MHz Low-speed on-chip oscillator on = 125 kHz No division | – | 5.5 | 11 | mA |
| | | | XIN clock off High-speed on-chip oscillator on fOCO = 10 MHz Low-speed on-chip oscillator on = 125 kHz Divide-by-8 | – | 2.2 | – | mA |
| | | Low-speed on-chip oscillator mode | XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz Divide-by-8, FMR47 = 1 | – | 145 | 400 | μA |
| | | Low-speed clock mode | XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator off XCIN clock oscillator on = 32 kHz FMR47 = 1 | – | 145 | 400 | μA |
| | | | XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator off XCIN clock oscillator on = 32 kHz Program operation on RAM Flash memory off, FMSTP = 1 | – | 30 | – | μA |
| | | Wait mode | XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz While a WAIT instruction is executed Peripheral clock operation VCA27 = VCA26 = VCA25 = 0 VCA20 = 1 | – | 28 | 85 | μA |
| | | | XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz While a WAIT instruction is executed Peripheral clock off VCA27 = VCA26 = VCA25 = 0 VCA20 = 1 | – | 17 | 50 | μA |
| | | | XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator off XCIN clock oscillator on = 32 kHz (high drive) While a WAIT instruction is executed VCA27 = VCA26 = VCA25 = 0 VCA20 = 1 | – | 3.3 | – | μA |
| | | | XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator off XCIN clock oscillator on = 32 kHz (low drive) While a WAIT instruction is executed VCA27 = VCA26 = VCA25 = 0 VCA20 = 1 | – | 2.1 | – | μA |
| | | Stop mode | XIN clock off, T _{opr} = 25°C High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA27 = VCA26 = VCA25 = 0 | – | 0.65 | 3.0 | μA |
| | | | XIN clock off, T _{opr} = 85°C High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA27 = VCA26 = VCA25 = 0 | – | 1.65 | – | μA |

Timing requirements**(Unless Otherwise Specified: $V_{CC} = 3\text{ V}$, $V_{SS} = 0\text{ V}$ at $T_{\text{opr}} = 25^\circ\text{C}$) [$V_{CC} = 3\text{ V}$]****Table 21.25 XIN Input, XCIN Input**

| Symbol | Parameter | Standard | | Unit |
|------------------------------|-----------------------|----------|------|---------------|
| | | Min. | Max. | |
| $t_{c(\text{XIN})}$ | XIN input cycle time | 100 | – | ns |
| $t_{\text{WH}(\text{XIN})}$ | XIN input “H” width | 40 | – | ns |
| $t_{\text{WL}(\text{XIN})}$ | XIN input “L” width | 40 | – | ns |
| $t_{c(\text{XCIN})}$ | XCIN input cycle time | 14 | – | μs |
| $t_{\text{WH}(\text{XCIN})}$ | XCIN input “H” width | 7 | – | μs |
| $t_{\text{WL}(\text{XCIN})}$ | XCIN input “L” width | 7 | – | μs |

**Figure 21.13 XIN Input and XCIN Input Timing Diagram when $V_{CC} = 3\text{ V}$** **Table 21.26 TRAIO Input, $\overline{\text{INT1}}$ Input**

| Symbol | Parameter | Standard | | Unit |
|-------------------------------|------------------------|----------|------|------|
| | | Min. | Max. | |
| $t_{c(\text{TRAIO})}$ | TRAIO input cycle time | 300 | – | ns |
| $t_{\text{WH}(\text{TRAIO})}$ | TRAIO input “H” width | 120 | – | ns |
| $t_{\text{WL}(\text{TRAIO})}$ | TRAIO input “L” width | 120 | – | ns |

**Figure 21.14 TRAIO Input and $\overline{\text{INT1}}$ Input Timing Diagram when $V_{CC} = 3\text{ V}$** **Table 21.27 TRFI Input**

| Symbol | Parameter | Standard | | Unit |
|------------------------------|-----------------------|----------|------|------|
| | | Min. | Max. | |
| $t_{c(\text{TRFI})}$ | TRFI input cycle time | 1200(1) | – | ns |
| $t_{\text{WH}(\text{TRFI})}$ | TRFI input “H” width | 600(2) | – | ns |
| $t_{\text{WL}(\text{TRFI})}$ | TRFI input “L” width | 600(2) | – | ns |

NOTES:

1. When using timer RF input capture mode, adjust the cycle time to $(1/\text{timer RF count source frequency} \times 3)$ or above.
2. When using timer RF input capture mode, adjust the pulse width to $(1/\text{timer RF count source frequency} \times 1.5)$ or above.

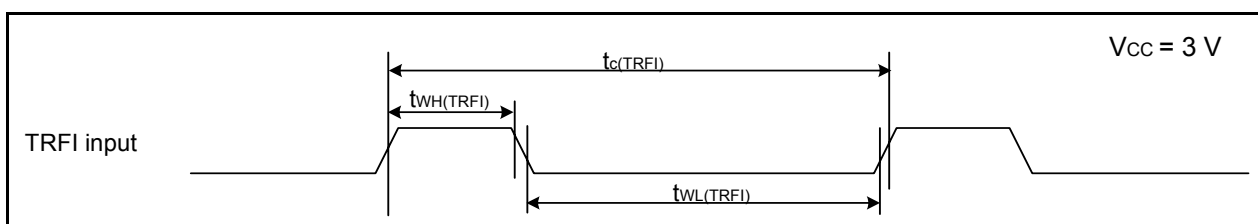
**Figure 21.15 TRFI Input Timing Diagram when $V_{CC} = 3\text{ V}$**

Table 21.28 Serial Interface

| Symbol | Parameter | Standard | | Unit |
|---------------|------------------------|----------|------|------|
| | | Min. | Max. | |
| $t_{c(CK)}$ | CLKi input cycle time | 300 | – | ns |
| $t_{w(CKH)}$ | CLKi input “H” width | 150 | – | ns |
| $t_{w(CKL)}$ | CLKi Input “L” width | 150 | – | ns |
| $t_{d(C-Q)}$ | TXDi output delay time | – | 80 | ns |
| $t_{h(C-Q)}$ | TXDi hold time | 0 | – | ns |
| $t_{su(D-C)}$ | RXDi input setup time | 70 | – | ns |
| $t_{h(C-D)}$ | RXDi input hold time | 90 | – | ns |

$i = 0$ to 2

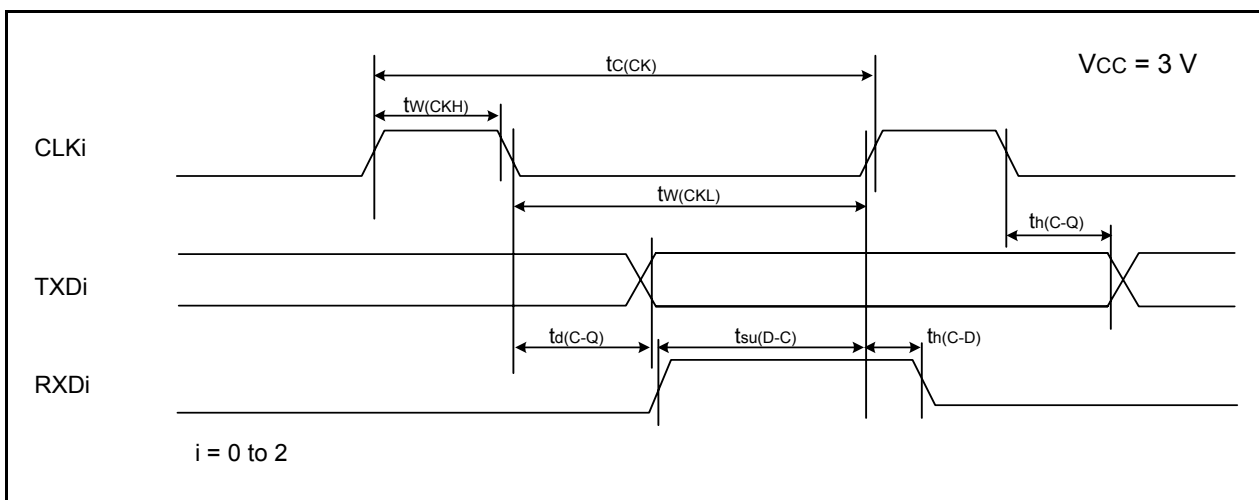


Figure 21.16 Serial Interface Timing Diagram when Vcc = 3 V

Table 21.29 External Interrupt \overline{INTi} ($i = 0, 2, 3$) Input

| Symbol | Parameter | Standard | | Unit |
|--------------|-----------------------------------|--------------------|------|------|
| | | Min. | Max. | |
| $t_{w(INH)}$ | $\overline{INT0}$ input “H” width | 380 ⁽¹⁾ | – | ns |
| $t_{w(INL)}$ | $\overline{INT0}$ input “L” width | 380 ⁽²⁾ | – | ns |

NOTES:

1. When selecting the digital filter by the \overline{INTi} input filter select bit, use an \overline{INTi} input HIGH width of either (1/digital filter clock frequency × 3) or the minimum value of standard, whichever is greater.
2. When selecting the digital filter by the \overline{INTi} input filter select bit, use an \overline{INTi} input LOW width of either (1/digital filter clock frequency × 3) or the minimum value of standard, whichever is greater.

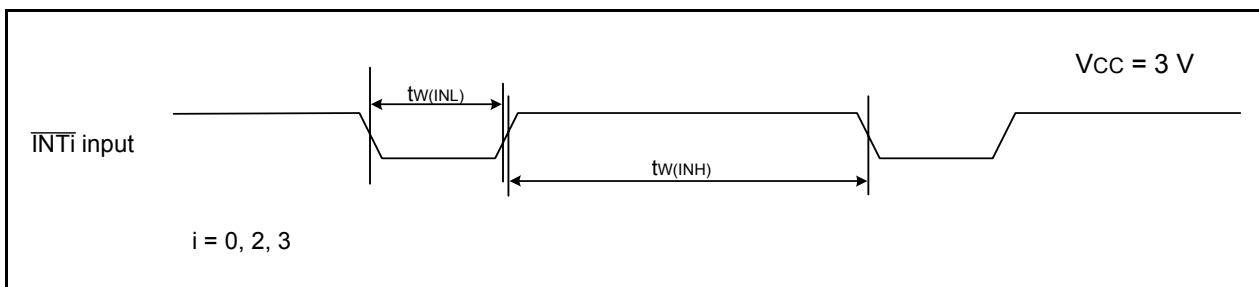


Figure 21.17 External Interrupt \overline{INTi} Input Timing Diagram when Vcc = 3 V

Table 21.30 Electrical Characteristics (5) [V_{CC} = 2.2 V]

| Symbol | Parameter | | Condition | | Standard | | | Unit |
|----------------------------------|---------------------|--|-------------------------|---------------------------|-----------------------|------|-----------------|------|
| | | | | | Min. | Typ. | Max. | |
| V _{OH} | Output "H" voltage | Except P2_0 to P2_7, XOUT | I _{OH} = -1 mA | | V _{CC} - 0.5 | – | V _{CC} | V |
| | | P2_0 to P2_7 | Drive capacity HIGH | I _{OH} = -2 mA | V _{CC} - 0.5 | – | V _{CC} | V |
| | | | Drive capacity LOW | I _{OH} = -1 mA | V _{CC} - 0.5 | – | V _{CC} | V |
| | | XOUT | Drive capacity HIGH | I _{OH} = -0.1 mA | V _{CC} - 0.5 | – | V _{CC} | V |
| | | | Drive capacity LOW | I _{OH} = -50 μA | V _{CC} - 0.5 | – | V _{CC} | V |
| V _{OL} | Output "L" voltage | Except P2_0 to P2_7, XOUT | I _{OL} = 1 mA | | – | – | 0.5 | V |
| | | P2_0 to P2_7 | Drive capacity HIGH | I _{OL} = 2 mA | – | – | 0.5 | V |
| | | | Drive capacity LOW | I _{OL} = 1 mA | – | – | 0.5 | V |
| | | XOUT | Drive capacity HIGH | I _{OL} = 0.1 mA | – | – | 0.5 | V |
| | | | Drive capacity LOW | I _{OL} = 50 μA | – | – | 0.5 | V |
| V _{T+} -V _{T-} | Hysteresis | $\overline{\text{INT0}}, \overline{\text{INT1}}, \overline{\text{INT2}}, \overline{\text{INT3}}, \text{KI0}, \text{KI1}, \text{KI2}, \text{KI3}, \text{TRAIO}, \text{TRFI}, \text{RXD0}, \text{RXD1}, \text{CLK0}, \text{CLK1}, \text{CLK2}, \text{SSI}, \text{SCL}, \text{SDA}, \text{SSO}$ | | | 0.05 | 0.3 | – | V |
| | | $\overline{\text{RESET}}$ | | | 0.05 | 0.15 | – | V |
| I _{IH} | Input "H" current | | V _I = 2.2 V | | – | – | 4.0 | μA |
| I _{IL} | Input "L" current | | V _I = 0 V | | – | – | -4.0 | μA |
| R _{PULLUP} | Pull-up resistance | | V _I = 0 V | | 100 | 200 | 600 | kΩ |
| R _{fXIN} | Feedback resistance | XIN | | | – | 5 | – | MΩ |
| R _{fXCIN} | Feedback resistance | XCIN | | | – | 35 | – | MΩ |
| V _{RAM} | RAM hold voltage | | During stop mode | | 1.8 | – | – | V |

NOTE:

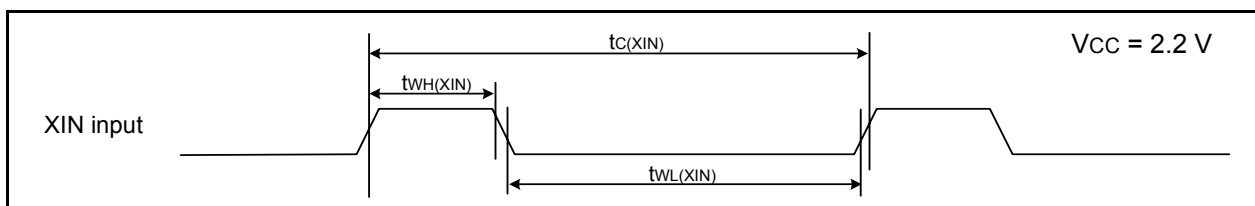
- V_{CC} = 2.2 V at T_{opr} = -20 to 85°C (N version) / -40 to 85°C (D version), f(XIN) = 5 MHz, unless otherwise specified.

**Table 21.31 Electrical Characteristics (6) [V_{CC} = 2.2 V]
(T_{opr} = -20 to 85°C (N version) / -40 to 85°C (D version), unless otherwise specified.)**

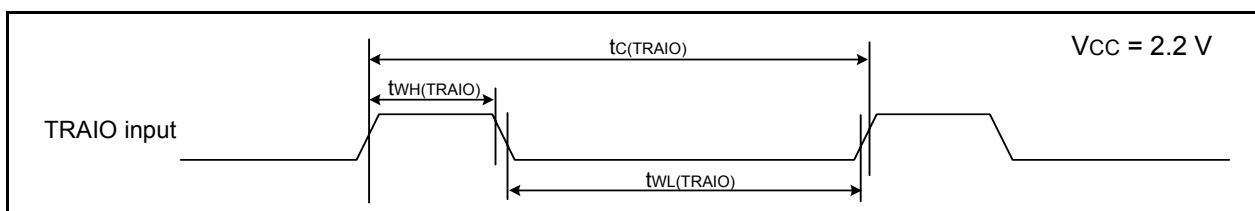
| Symbol | Parameter | Condition | Standard | | | Unit | |
|-----------------|---|------------------------------------|--|------|------|------|----|
| | | | Min. | Typ. | Max. | | |
| I _{CC} | Power supply current (V _{CC} = 2.2 to 2.7 V) Single-chip mode, output pins are open, other pins are V _{SS} | High-speed clock mode | XIN = 5 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz No division | – | 2.5 | – | mA |
| | | | XIN = 5 MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz Divide-by-8 | – | 1 | – | mA |
| | | High-speed on-chip oscillator mode | XIN clock off High-speed on-chip oscillator on f _{OCO} = 5 MHz Low-speed on-chip oscillator on = 125 kHz No division | – | 4 | – | mA |
| | | | XIN clock off High-speed on-chip oscillator on f _{OCO} = 5 MHz Low-speed on-chip oscillator on = 125 kHz Divide-by-8 | – | 1.7 | – | mA |
| | | Low-speed on-chip oscillator mode | XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz Divide-by-8, FMR47 = 1 | – | 110 | 300 | μA |
| | | Low-speed clock mode | XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator off XCIN clock oscillator on = 32 kHz FMR47 = 1 | – | 125 | 350 | μA |
| | | | XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator off XCIN clock oscillator on = 32 kHz Program operation on RAM Flash memory off, FMSTP = 1 | – | 27 | – | μA |
| | | Wait mode | XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz While a WAIT instruction is executed Peripheral clock operation VCA27 = VCA26 = VCA25 = 0 VCA20 = 1 | – | 20 | 60 | μA |
| | | | XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on = 125 kHz While a WAIT instruction is executed Peripheral clock off VCA27 = VCA26 = VCA25 = 0 VCA20 = 1 | – | 12 | 40 | μA |
| | | | XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator off XCIN clock oscillator on = 32 kHz (high drive) While a WAIT instruction is executed VCA27 = VCA26 = VCA25 = 0 VCA20 = 1 | – | 2.8 | – | μA |
| | | | XIN clock off High-speed on-chip oscillator off Low-speed on-chip oscillator off XCIN clock oscillator on = 32 kHz (low drive) While a WAIT instruction is executed VCA27 = VCA26 = VCA25 = 0 VCA20 = 1 | – | 1.9 | – | μA |
| | | Stop mode | XIN clock off, T _{opr} = 25°C High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA27 = VCA26 = VCA25 = 0 | – | 0.6 | 3.0 | μA |
| | | | XIN clock off, T _{opr} = 85°C High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA27 = VCA26 = VCA25 = 0 | – | 1.60 | – | μA |

Timing requirements**(Unless Otherwise Specified: $V_{CC} = 2.2\text{ V}$, $V_{SS} = 0\text{ V}$ at $T_{opr} = 25^\circ\text{C}$) [$V_{CC} = 2.2\text{ V}$]****Table 21.32 XIN Input, XCIN Input**

| Symbol | Parameter | Standard | | Unit |
|----------------|-----------------------|----------|------|---------------|
| | | Min. | Max. | |
| $t_{c(XIN)}$ | XIN input cycle time | 200 | – | ns |
| $t_{WH(XIN)}$ | XIN input “H” width | 90 | – | ns |
| $t_{WL(XIN)}$ | XIN input “L” width | 90 | – | ns |
| $t_{c(XCIN)}$ | XCIN input cycle time | 14 | – | μs |
| $t_{WH(XCIN)}$ | XCIN input “H” width | 7 | – | μs |
| $t_{WL(XCIN)}$ | XCIN input “L” width | 7 | – | μs |

**Figure 21.18 XIN Input and XCIN Input Timing Diagram when $V_{CC} = 2.2\text{ V}$** **Table 21.33 TRAIO Input, $\overline{\text{INT1}}$ Input**

| Symbol | Parameter | Standard | | Unit |
|------------------------|------------------------|----------|------|------|
| | | Min. | Max. | |
| $t_{c(\text{TRAIO})}$ | TRAIO input cycle time | TBD | – | ns |
| $t_{WH(\text{TRAIO})}$ | TRAIO input “H” width | TBD | – | ns |
| $t_{WL(\text{TRAIO})}$ | TRAIO input “L” width | TBD | – | ns |

**Figure 21.19 TRAIO Input and $\overline{\text{INT1}}$ Input Timing Diagram when $V_{CC} = 2.2\text{ V}$** **Table 21.34 TRFI Input**

| Symbol | Parameter | Standard | | Unit |
|-----------------------|-----------------------|---------------------|------|------|
| | | Min. | Max. | |
| $t_{c(\text{TRFI})}$ | TRFI input cycle time | 2000 ⁽¹⁾ | – | ns |
| $t_{WH(\text{TRFI})}$ | TRFI input “H” width | 1000 ⁽²⁾ | – | ns |
| $t_{WL(\text{TRFI})}$ | TRFI input “L” width | 1000 ⁽²⁾ | – | ns |

NOTES:

1. When using timer RF input capture mode, adjust the cycle time to $(1/\text{timer RF count source frequency} \times 3)$ or above.
2. When using timer RF input capture mode, adjust the pulse width to $(1/\text{timer RF count source frequency} \times 1.5)$ or above.

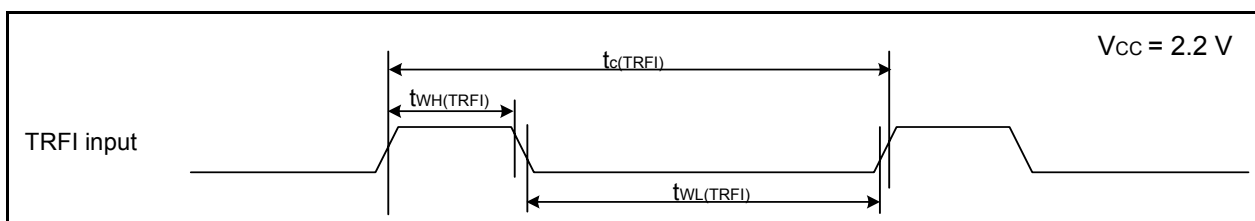
**Figure 21.20 TRFI Input Timing Diagram when $V_{CC} = 2.2\text{ V}$**

Table 21.35 Serial Interface

| Symbol | Parameter | Standard | | Unit |
|---------------|------------------------|----------|------|------|
| | | Min. | Max. | |
| $t_{c(CK)}$ | CLKi input cycle time | 800 | – | ns |
| $t_{w(CKH)}$ | CLKi input “H” width | 400 | – | ns |
| $t_{w(CKL)}$ | CLKi input “L” width | 400 | – | ns |
| $t_{d(C-Q)}$ | TXDi output delay time | – | 200 | ns |
| $t_{h(C-Q)}$ | TXDi hold time | 0 | – | ns |
| $t_{su(D-C)}$ | RXDi input setup time | 150 | – | ns |
| $t_{h(C-D)}$ | RXDi input hold time | 90 | – | ns |

$i = 0$ to 2

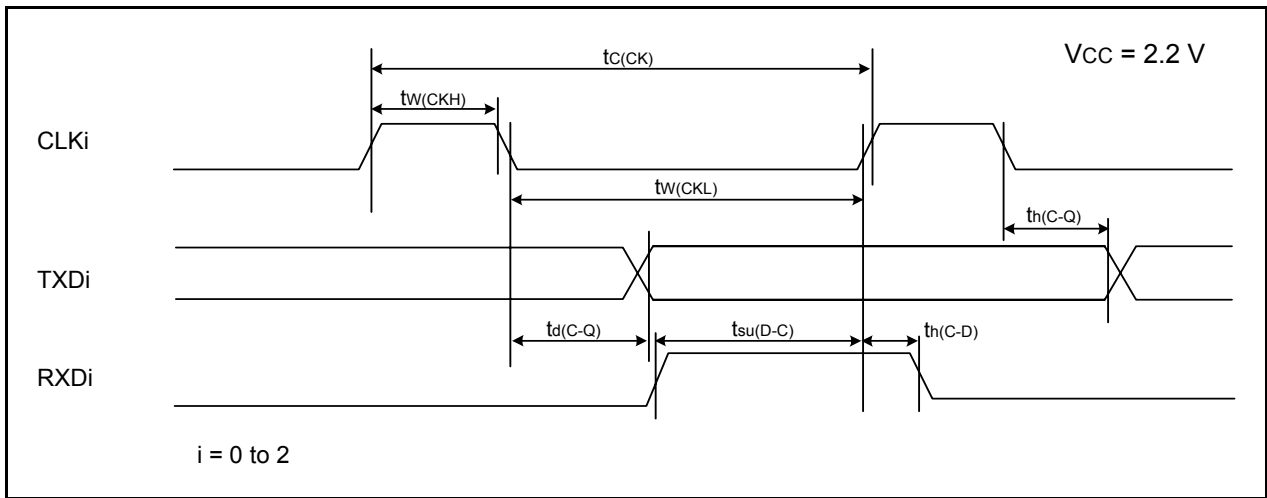


Figure 21.21 Serial Interface Timing Diagram when $V_{cc} = 2.2 V$

Table 21.36 External Interrupt \overline{INTi} ($i = 0, 2, 3$) Input

| Symbol | Parameter | Standard | | Unit |
|--------------|-----------------------------------|---------------------|------|------|
| | | Min. | Max. | |
| $t_{w(INH)}$ | $\overline{INT0}$ input “H” width | 1000 ⁽¹⁾ | – | ns |
| $t_{w(INL)}$ | $\overline{INT0}$ input “L” width | 1000 ⁽²⁾ | – | ns |

NOTES:

1. When selecting the digital filter by the \overline{INTi} input filter select bit, use an \overline{INTi} input HIGH width of either (1/digital filter clock frequency \times 3) or the minimum value of standard, whichever is greater.
2. When selecting the digital filter by the \overline{INTi} input filter select bit, use an \overline{INTi} input LOW width of either (1/digital filter clock frequency \times 3) or the minimum value of standard, whichever is greater.

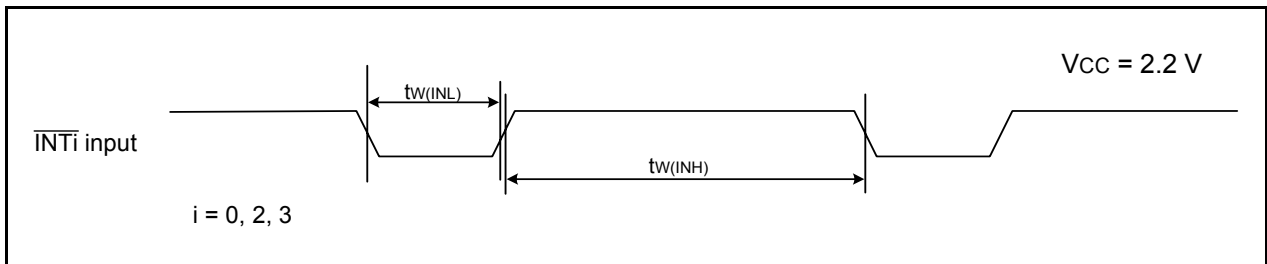


Figure 21.22 External Interrupt \overline{INTi} Input Timing Diagram when $V_{cc} = 2.2 V$

22. Usage Notes

22.1 Notes on Clock Generation Circuit

22.1.1 Stop Mode

When entering stop mode, set the FMR01 bit in the FMR0 register to 0 (CPU rewrite mode disabled) and the CM10 bit in the CM1 register to 1 (stop mode). An instruction queue pre-reads 4 bytes from the instruction which sets the CM10 bit to 1 (stop mode) and the program stops.

Insert at least 4 NOP instructions following the JMP.B instruction after the instruction which sets the CM10 bit to 1.

- Program example to enter stop mode

```

BCLR      1,FMR0      ; CPU rewrite mode disabled
BSET      0,PRCR      ; Protect disabled
FSET      I           ; Enable interrupt
BSET      0,CM1       ; Stop mode
JMP.B     LABEL_001
LABEL_001 :
NOP
NOP
NOP
NOP

```

22.1.2 Wait Mode

When entering wait mode, set the FMR01 bit in the FMR0 register to 0 (CPU rewrite mode disabled) and execute the WAIT instruction. An instruction queue pre-reads 4 bytes from the WAIT instruction and the program stops. Insert at least 4 NOP instructions after the WAIT instruction.

- Program example to execute the WAIT instruction

```

BCLR      1,FMR0      ; CPU rewrite mode disabled
FSET      I           ; Enable interrupt
WAIT      ; Wait mode
NOP
NOP
NOP
NOP

```

22.1.3 Oscillation Stop Detection Function

Since the oscillation stop detection function cannot be used if the XIN clock frequency is 2 MHz or below, set bits OCD1 to OCD0 to 00b.

22.1.4 Oscillation Circuit Constants

Ask the manufacturer of the oscillator to specify the best oscillation circuit constants for your system.

To use this MCU with supply voltage below $VCC = 2.7$ V, it is recommended to set the CM11 bit in the CM1 register to 1 (on-chip feedback resistor disabled), the CM15 bit to 1 (high drive capacity), and connect the feedback resistor to the chip externally.

22.2 Notes on Interrupts

22.2.1 Reading Address 00000h

Do not read address 00000h by a program. When a maskable interrupt request is acknowledged, the CPU reads interrupt information (interrupt number and interrupt request level) from 00000h in the interrupt sequence. At this time, the acknowledged interrupt IR bit is set to 0.

If address 00000h is read by a program, the IR bit for the interrupt which has the highest priority among the enabled interrupts is set to 0. This may cause the interrupt to be canceled, or an unexpected interrupt to be generated.

22.2.2 SP Setting

Set any value in the SP before an interrupt is acknowledged. The SP is set to 0000h after reset. Therefore, if an interrupt is acknowledged before setting a value in the SP, the program may run out of control.

22.2.3 External Interrupt and Key Input Interrupt

Either “L” level or an “H” level of width shown in the Electrical Characteristics is necessary for the signal input to pins $\overline{\text{INT0}}$ to $\overline{\text{INT3}}$ and pins $\overline{\text{KI0}}$ to $\overline{\text{KI3}}$, regardless of the CPU clock.

For details, refer to **Table 21.22** (VCC = 5V), **Table 21.29** (VCC = 3V), **Table 21.36** (VCC = 2.2V) **External Interrupt INT_i (i = 0, 2, 3) Input** and **Table 21.19** (VCC = 5V), **Table 21.26** (VCC = 3V), **Table 21.33** (VCC = 2.2V) **TRAI0 Input, INT1 Input**.

22.2.4 Changing Interrupt Sources

The IR bit in the interrupt control register may be set to 1 (interrupt requested) when the interrupt source changes. When using an interrupt, set the IR bit to 0 (no interrupt requested) after changing the interrupt source. In addition, changes of interrupt sources include all factors that change the interrupt sources assigned to individual software interrupt numbers, polarities, and timing. Therefore, if a mode change of a peripheral function involves interrupt sources, edge polarities, and timing, set the IR bit to 0 (no interrupt requested) after the change. Refer to the individual peripheral function for its related interrupts.

Figure 22.1 shows an Example of Procedure for Changing Interrupt Sources.

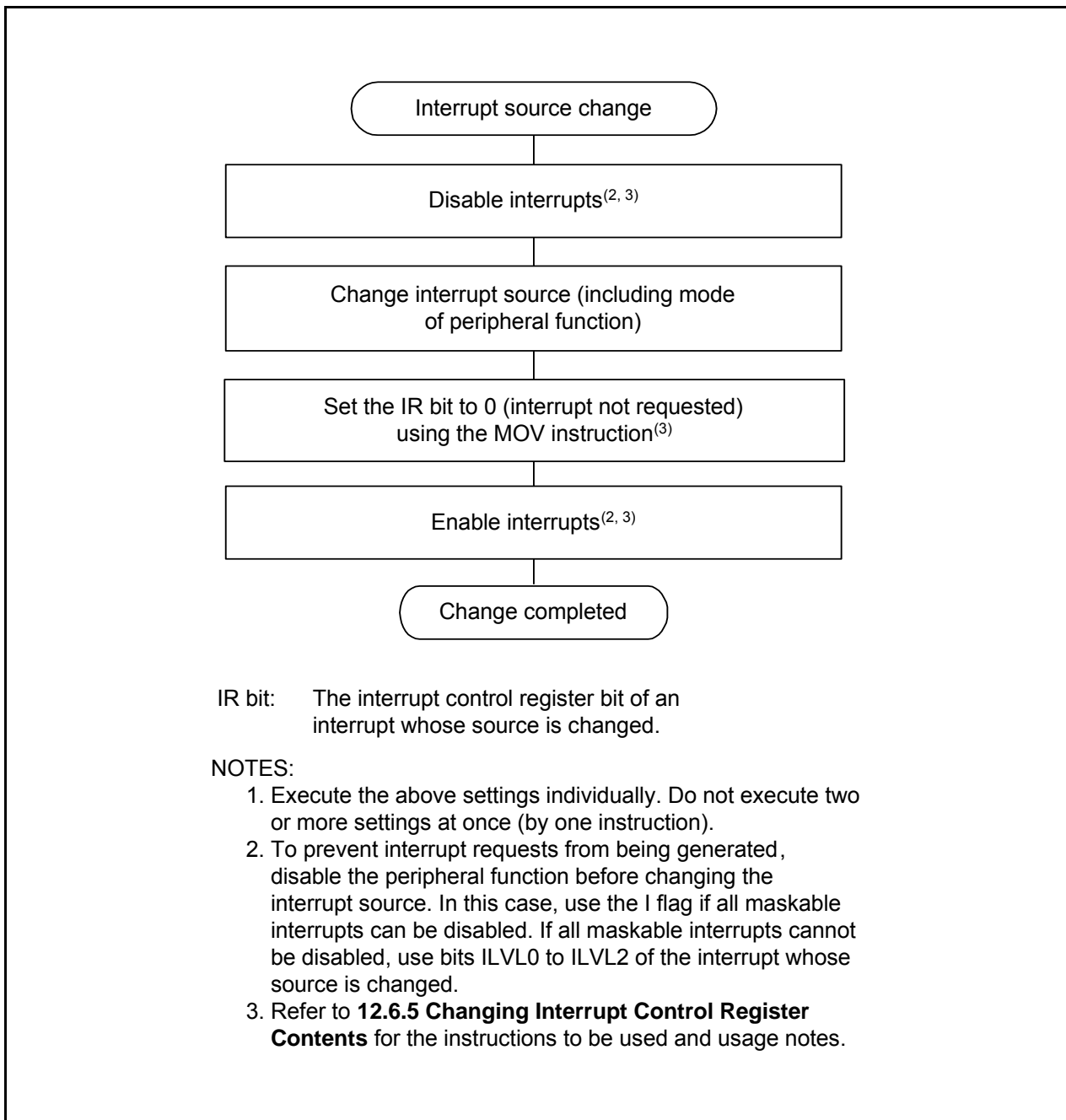


Figure 22.1 Example of Procedure for Changing Interrupt Sources

22.2.5 Changing Interrupt Control Register Contents

- (a) The contents of an interrupt control register can only be changed while no interrupt requests corresponding to that register are generated. If interrupt requests may be generated, disable interrupts before changing the interrupt control register contents.
- (b) When changing the contents of an interrupt control register after disabling interrupts, be careful to choose appropriate instructions.

Changing any bit other than IR bit

If an interrupt request corresponding to a register is generated while executing the instruction, the IR bit may not be set to 1 (interrupt requested), and the interrupt request may be ignored. If this causes a problem, use the following instructions to change the register: AND, OR, BCLR, BSET

Changing IR bit

If the IR bit is set to 0 (interrupt not requested), it may not be set to 0 depending on the instruction used. Therefore, use the MOV instruction to set the IR bit to 0.

- (c) When disabling interrupts using the I flag, set the I flag as shown in the sample programs below. Refer to (b) regarding changing the contents of interrupt control registers by the sample programs.

Sample programs 1 to 3 are for preventing the I flag from being set to 1 (interrupts enabled) before the interrupt control register is changed for reasons of the internal bus or the instruction queue buffer.

Example 1: Use NOP instructions to prevent I flag from being set to 1 before interrupt control register is changed

```
INT_SWITCH1:
  FCLR   I           ; Disable interrupts
  AND.B  #00H,0056H ; Set TRAIC register to 00h
  NOP                    ;
  NOP                    ;
  FSET   I           ; Enable interrupts
```

Example 2: Use dummy read to delay FSET instruction

```
INT_SWITCH2:
  FCLR   I           ; Disable interrupts
  AND.B  #00H,0056H ; Set TRAIC register to 00h
  MOV.W  MEM,R0      ; Dummy read
  FSET   I           ; Enable interrupts
```

Example 3: Use POPC instruction to change I flag

```
INT_SWITCH3:
  PUSHC  FLG
  FCLR   I           ; Disable interrupts
  AND.B  #00H,0056H ; Set TRAIC register to 00h
  POPC   FLG        ; Enable interrupts
```

22.3 Notes on Timers

22.3.1 Notes on Timer RA

- Timer RA stops counting after a reset. Set the values in the timer RA and timer RA prescalers before the count starts.
- Even if the prescaler and timer RA are read out in 16-bit units, these registers are read 1 byte at a time by the MCU. Consequently, the timer value may be updated during the period when these two registers are being read.
- In pulse period measurement mode, bits TEDGF and TUNDF in the TRACR register can be set to 0 by writing 0 to these bits by a program. However, these bits remain unchanged if 1 is written. When using the READ-MODIFY-WRITE instruction for the TRACR register, the TEDGF or TUNDF bit may be set to 0 although these bits are set to 1 while the instruction is being executed. In this case, write 1 to the TEDGF or TUNDF bit which is not supposed to be set to 0 with the MOV instruction.
- When changing to pulse period measurement mode from another mode, the contents of bits TEDGF and TUNDF are undefined. Write 0 to bits TEDGF and TUNDF before the count starts.
- The TEDGF bit may be set to 1 by the first timer RA prescaler underflow generated after the count starts.
- When using the pulse period measurement mode, leave two or more periods of the timer RA prescaler immediately after the count starts, then set the TEDGF bit to 0.
- The TCSTF bit retains 0 (count stops) for 0 to 1 cycle of the count source after setting the TSTART bit to 1 (count starts) while the count is stopped.

During this time, do not access registers associated with timer RA⁽¹⁾ other than the TCSTF bit. Timer RA starts counting at the first valid edge of the count source after The TCSTF bit is set to 1 (during count).

The TCSTF bit remains 1 for 0 to 1 cycle of the count source after setting the TSTART bit to 0 (count stops) while the count is in progress. Timer RA counting is stopped when the TCSTF bit is set to 0.

During this time, do not access registers associated with timer RA⁽¹⁾ other than the TCSTF bit.

NOTE:

1. Registers associated with timer RA: TRACR, TRAIOC, TRAMR, TRAPRE, and TRA.

- When the TRAPRE register is continuously written during count operation (TCSTF bit is set to 1), allow three or more cycles of the count source clock for each write interval.
- When the TRA register is continuously written during count operation (TCSTF bit is set to 1), allow three or more cycles of the prescaler underflow for each write interval.

22.3.2 Notes on Timer RB

- Timer RB stops counting after a reset. Set the values in the timer RB and timer RB prescalers before the count starts.
- Even if the prescaler and timer RB is read out in 16-bit units, these registers are read 1 byte at a time by the MCU. Consequently, the timer value may be updated during the period when these two registers are being read.
- In programmable one-shot generation mode and programmable wait one-shot generation mode, when setting the TSTART bit in the TRBCR register to 0 (count stops) or setting the TOSSP bit in the TRBOCR register to 1 (one-shot stops), the timer reloads the value of reload register and stops. Therefore, in programmable one-shot generation mode and programmable wait one-shot generation mode, read the timer count value before the timer stops.
- The TCSTF bit remains 0 (count stops) for 1 to 2 cycles of the count source after setting the TSTART bit to 1 (count starts) while the count is stopped.

During this time, do not access registers associated with timer RB⁽¹⁾ other than the TCSTF bit.

The TCSTF bit remains 1 for 1 to 2 cycles of the count source after setting the TSTART bit to 0 (count stops) while the count is in progress. Timer RB counting is stopped when the TCSTF bit is set to 0.

During this time, do not access registers associated with timer RB⁽¹⁾ other than the TCSTF bit.

NOTE:

1. Registers associated with timer RB: TRBCR, TRBOCR, TRBIOC, TRBMR, TRBPRES, TRBSC, and TRBPR.

- If the TSTOP bit in the TRBCR register is set to 1 during timer operation, timer RB stops immediately.
- If 1 is written to the TOSST or TOSSP bit in the TRBOCR register, the value of the TOSSTF bit changes after one or two cycles of the count source have elapsed. If the TOSSP bit is written to 1 during the period between when the TOSST bit is written to 1 and when the TOSSTF bit is set to 1, the TOSSTF bit may be set to either 0 or 1 depending on the content state. Likewise, if the TOSST bit is written to 1 during the period between when the TOSSP bit is written to 1 and when the TOSSTF bit is set to 0, the TOSSTF bit may be set to either 0 or 1.

22.3.2.1 Timer mode

The following workaround should be performed in timer mode.

To write to registers TRBPRES and TRBPR during count operation (TCSTF bit is set to 1), note the following points:

- When the TRBPRES register is written continuously, allow three or more cycles of the count source for each write interval.
- When the TRBPR register is written continuously, allow three or more cycles of the prescaler underflow for each write interval.

22.3.2.2 Programmable waveform generation mode

The following three workarounds should be performed in programmable waveform generation mode.

- (1) To write to registers TRBPRE and TRBPR during count operation (TCSTF bit is set to 1), note the following points:
 - When the TRBPRE register is written continuously, allow three or more cycles of the count source for each write interval.
 - When the TRBPR register is written continuously, allow three or more cycles of the prescaler underflow for each write interval.
- (2) To change registers TRBPRE and TRBPR during count operation (TCSTF bit is set to 1), synchronize the TRBO output cycle using a timer RB interrupt, etc. This operation should be performed only once in the same output cycle. Also, make sure that writing to the TRBPR register does not occur during period A shown in Figures 22.2 and 22.3.

The following shows the detailed workaround examples.

- Workaround example (a):
As shown in Figure 22.2, write to registers TRBSC and TRBPR in the timer RB interrupt routine. These write operations must be completed by the beginning of period A.

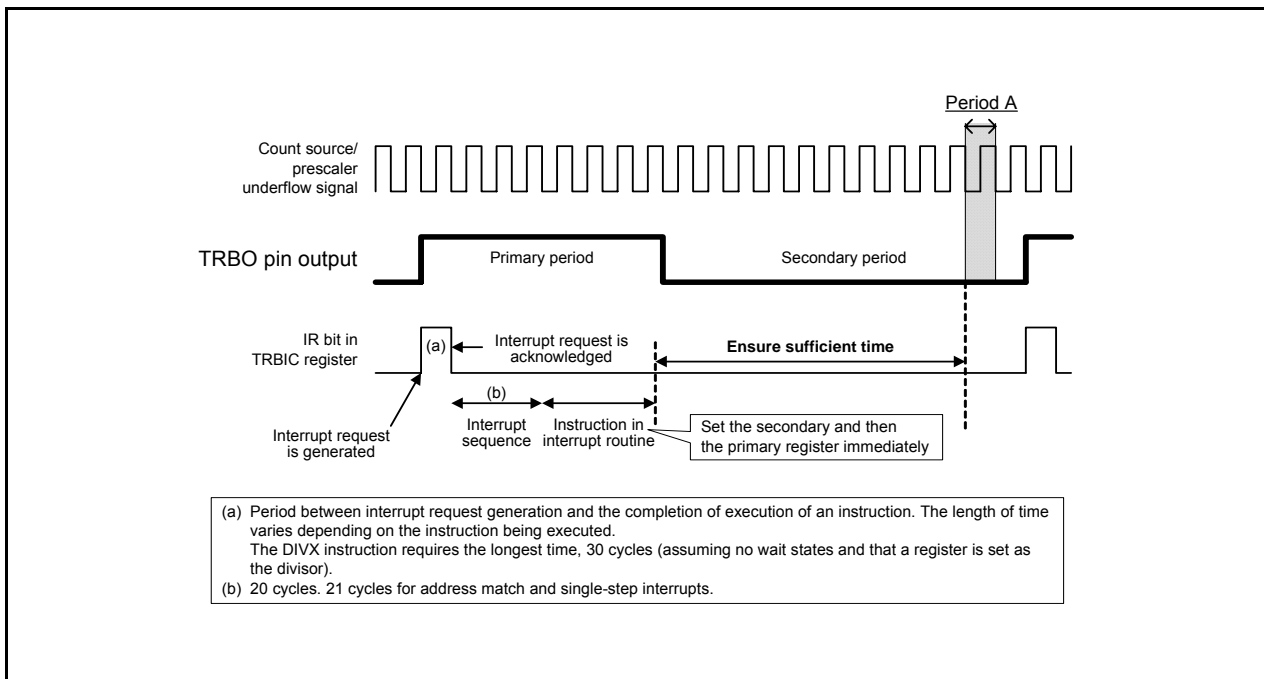


Figure 22.2 Workaround Example (a) When Timer RB interrupt is Used

- Workaround example (b):

As shown in Figure 22.3 detect the start of the primary period by the TRBO pin output level and write to registers TRBSC and TRBPR. These write operations must be completed by the beginning of period A.

If the port register's bit value is read after the port direction register's bit corresponding to the TRBO pin is set to 0 (input mode), the read value indicates the TRBO pin output value.

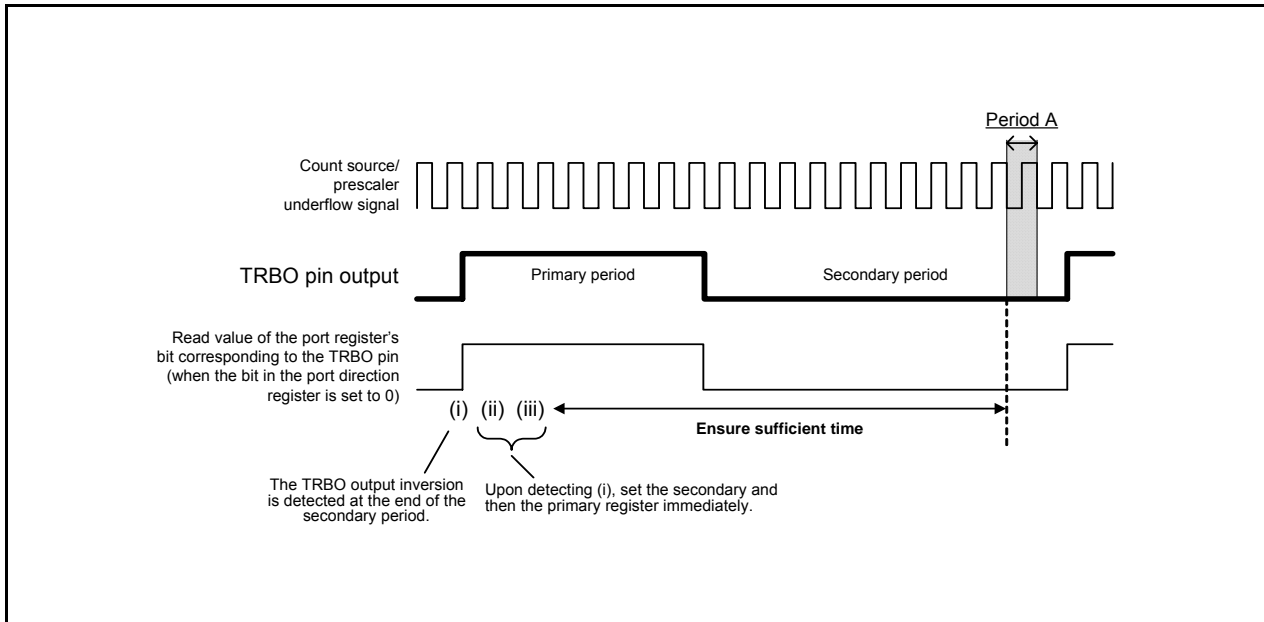


Figure 22.3 Workaround Example (b) When TRBO Pin Output Value is Read

- (3) To stop the timer counting in the primary period, use the TSTOP bit in the TRBCR register. In this case, registers TRBPRES and TRBPR are initialized and their values are set to the values after reset.

22.3.2.3 Programmable one-shot generation mode

The following two workarounds should be performed in programmable one-shot generation mode.

- (1) To write to registers TRBPRES and TRBPR during count operation (TCSTF bit is set to 1), note the following points:
 - When the TRBPRES register is written continuously during count operation (TCSTF bit is set to 1), allow three or more cycles of the count source for each write interval.
 - When the TRBPR register is written continuously during count operation (TCSTF bit is set to 1), allow three or more cycles of the prescaler underflow for each write interval.
- (2) Do not set both the TRBPRES and TRBPR registers to 00h.

22.3.2.4 Programmable wait one-shot generation mode

The following three workarounds should be performed in programmable wait one-shot generation mode.

- (1) To write to registers TRBPRE and TRBPR during count operation (TCSTF bit is set to 1), note the following points:
 - When the TRBPRE register is written continuously, allow three or more cycles of the count source for each write interval.
 - When the TRBPR register is written continuously, allow three or more cycles of the prescaler underflow for each write interval.
- (2) Do not set both the TRBPRE and TRBPR registers to 00h.
- (3) Set registers TRBSC and TRBPR using the following procedure.
 - (a) To use “ $\overline{\text{INT0}}$ pin one-shot trigger enabled” as the count start condition
Set the TRBSC register and then the TRBPR register. At this time, after writing to the TRBPR register, allow an interval of 0.5 or more cycles of the count source before trigger input from the $\overline{\text{INT0}}$ pin.
 - (b) To use “writing 1 to TOSST bit” as the start condition
Set the TRBSC register, the TRBPR register, and then TOSST bit. At this time, after writing to the TRBPR register, allow an interval of 0.5 or more cycles of the count source before writing to the TOSST bit.

22.3.3 Notes on Timer RC

22.3.3.1 TRC Register

- The following note applies when the CCLR bit in the TRCCR1 register is set to 1 (clear TRC register at compare match with TRCGRA register).

When using a program to write a value to the TRC register while the TSTART bit in the TRCMR register is set to 1 (count starts), ensure that the write does not overlap with the timing with which the TRC register is set to 0000h.

If the timing of the write to the TRC register and the setting of the TRC register to 0000h coincide, the write value will not be written to the TRC register and the TRC register will be set to 0000h.

- Reading from the TRC register immediately after writing to it can result in the value previous to the write being read out. To prevent this, execute the JMP.B instruction between the read and the write instructions.

```

Program Example      MOV.W      #XXXXh, TRC          ;Write
                   JMP.B      L1              ;JMP.B instruction
                   L1:        MOV.W      TRC,DATA      ;Read

```

22.3.3.2 TRCSR Register

Reading from the TRCSR register immediately after writing to it can result in the value previous to the write being read out. To prevent this, execute the JMP.B instruction between the read and the write instructions.

```

Program Example      MOV.B      #XXh, TRCSR        ;Write
                   JMP.B      L1              ;JMP.B instruction
                   L1:        MOV.B      TRCSR,DATA    ;Read

```

22.3.3.3 Count Source Switching

- Stop the count before switching the count source.

Switching procedure

- Set the TSTART bit in the TRCMR register to 0 (count stops).
- Change the settings of bits TCK2 to TCK0 in the TRCCR1 register.

- After switching the count source from fOCO40M to another clock, allow a minimum of two cycles of f1 to elapse after changing the clock setting before stopping fOCO40M.

Switching procedure

- Set the TSTART bit in the TRCMR register to 0 (count stops).
- Change the settings of bits TCK2 to TCK0 in the TRCCR1 register.
- Wait for a minimum of two cycles of f1.
- Set the FRA00 bit in the FRA0 register to 0 (high-speed on-chip oscillator off).

22.3.3.4 Input Capture Function

- The pulse width of the input capture signal should be three cycles or more of the timer RC operation clock (refer to **Table 14.12 Timer RC Operation Clock**).
- The value of the TRC register is transferred to the TRCGRj register one or two cycles of the timer RC operation clock after the input capture signal is input to the TRCIOj (j = A, B, C, or D) pin (when the digital filter function is not used).

22.3.3.5 TRCMR Register in PWM2 Mode

When the CSEL bit in the TRCCR2 register is set to 1 (count stops at compare match with the TRCGRA register), do not set the TRCMR register at compare match timing of registers TRC and TRCGRA.

22.3.4 Notes on Timer RD

22.3.4.1 TRDSTR Register

- Set the TRDSTR register using the MOV instruction.
- When the CSELi (i = 0 to 1) is set to 0 (the count stops at compare match of registers TRDi and TRDGRAi), the count does not stop and the TSTARTi bit remains unchanged even if 0 (count stops) is written to the TSTARTi bit.
- Therefore, set the TSTARTi bit to 0 to change other bits without changing the TSTARTi bit when the CSELi bit is set to 0.
- To stop counting by a program, set the TSTARTi bit after setting the CSELi bit to 1. Although the CSELi bit is set to 1 and the TSTARTi bit is set to 0 at the same time (with 1 instruction), the count cannot be stopped.
- Table 22.1 lists the TRDIOji (j = A, B, C, or D) Pin Output Level when Count Stops to use the TRDIOji (j = A, B, C, or D) pin with the timer RD output.

Table 22.1 TRDIOji (j = A, B, C, or D) Pin Output Level when Count Stops

| Count Stop | TRDIOji Pin Output when Count Stops |
|---|--|
| When the CSELi bit is set to 1, set the TSTARTi bit to 0 and the count stops. | Hold the output level immediately before the count stops. |
| When the CSELi bit is set to 0, the count stops at compare match of registers TRDi and TRDGRAi. | Hold the output level after output changes by compare match. |

22.3.4.2 TRDi Register (i = 0 or 1)

- When writing the value to the TRDi register by a program while the TSTARTi bit in the TRDSTR register is set to 1 (count starts), avoid overlapping with the timing for setting the TRDi register to 0000h, and then write. If the timing for setting the TRDi register to 0000h overlaps with the timing for writing the value to the TRDi register, the value is not written and the TRDi register is set to 0000h. These precautions are applicable when selecting the following by bits CCLR2 to CCLR0 in the TRDCRi register.
 - 001b (Clear by the TRDi register at compare match with the TRDGRAi register.)
 - 010b (Clear by the TRDi register at compare match with the TRDGRBi register.)
 - 011b (Synchronous clear)
 - 101b (Clear by the TRDi register at compare match with the TRDGRCi register.)
 - 110b (Clear by the TRDi register at compare match with the TRDGRDi register.)
- When writing the value to the TRDi register and continuously reading the same register, the value before writing may be read. In this case, execute the JMP.B instruction between the writing and reading.

```

Program example      MOV.W      #XXXXh, TRD0      ;Writing
                    JMP.B      L1              ;JMP.B
                    L1:        MOV.W      TRD0,DATA      ;Reading
  
```

22.3.4.3 TRDSRi Register (i = 0 or 1)

When writing the value to the TRDSRi register and continuously reading the same register, the value before writing may be read. In this case, execute the JMP.B instruction between the writing and reading.

```

Program example      MOV.B      #XXh, TRDSR0      ;Writing
                    JMP.B      L1              ;JMP.B
                    L1:        MOV.B      TRDSR0,DATA      ;Reading
  
```

22.3.4.4 Count Source Switch

- Switch the count source after the count stops.

Change procedure

- (1) Set the TSTART_i (i = 0 or 1) bit in the TRDSTR register to 0 (count stops).
- (2) Change bits TCK2 to TCK0 in the TRDCR_i register.

- When changing the count source from fOCO40M to another source and stopping fOCO40M, wait 2 cycles of f₁ or more after setting the clock switch, and then stop fOCO40M.

Change procedure

- (1) Set the TSTART_i (i = 0 or 1) bit in the TRDSTR register to 0 (count stops).
- (2) Change bits TCK2 to TCK0 in the TRDCR_i register.
- (3) Wait 2 or more cycles of f₁.
- (4) Set the FRA00 bit in the FRA0 register to 0 (high-speed on-chip oscillator stops).

22.3.4.5 Input Capture Function

- Set the pulse width of the input capture signal to 3 or more cycles of the timer RD operation clock (refer to **Table 14.26 Timer RD Operation Clocks**).
- The value in the TRD_i register is transferred to the TRDGR_{ji} register 2 to 3 cycles of the timer RD operation clock after the input capture signal is applied to the TRDIO_{ji} pin (i = 0 or 1, j = either A, B, C, or D) (no digital filter).

22.3.4.6 Reset Synchronous PWM Mode

- When reset synchronous PWM mode is used for motor control, make sure OLS0 = OLS1.
- Set to reset synchronous PWM mode by the following procedure:

Change procedure

- (1) Set the TSTART0 bit in the TRDSTR register to 0 (count stops).
- (2) Set bits CMD1 to CMD0 in the TRDFCR register to 00b (timer mode, PWM mode, and PWM3 mode).
- (3) Set bits CMD1 to CMD0 to 01b (reset synchronous PWM mode).
- (4) Set the other registers associated with timer RD again.

22.3.4.7 Complementary PWM Mode

- When complementary PWM mode is used for motor control, make sure OLS0 = OLS1.
- Change bits CMD1 to CMD0 in the TRDFCR register in the following procedure.

Change procedure: When setting to complementary PWM mode (including re-set), or changing the transfer timing from the buffer register to the general register in complementary PWM mode.

- (1) Set both the TSTART0 and TSTART1 bits in the TRDSTR register to 0 (count stops).
- (2) Set bits CMD1 to CMD0 in the TRDFCR register to 00b (timer mode, PWM mode, and PWM3 mode).
- (3) Set bits CMD1 to CMD0 to 10b or 11b (complementary PWM mode).
- (4) Set the registers associated with other timer RD again.

Change procedure: When stopping complementary PWM mode

- (1) Set both the TSTART0 and TSTART1 bits in the TRDSTR register to 0 (count stops).
- (2) Set bits CMD1 to CMD0 to 00b (timer mode, PWM mode, and PWM3 mode).

- Do not write to TRDGRA0, TRDGRB0, TRDGRA1, or TRDGRB1 register during operation. When changing the PWM waveform, transfer the values written to registers TRDGRD0, TRDGRC1, and TRDGRD1 to registers TRDGRB0, TRDGRA1, and TRDGRB1 using the buffer operation. However, to write data to the TRDGRD0, TRDGRC1, or TRDGRD1 register, set bits BFD0, BFC1, and BFD1 to 0 (general register). After this, bits BFD0, BFC1, and BFD1 may be set to 1 (buffer register). The PWM period cannot be changed.

- If the value in the TRDGRA0 register is assumed to be m , the TRD0 register counts $m-1$, m , $m+1$, m , $m-1$, in that order, when changing from increment to decrement operation.

When changing from m to $m+1$, the IMFA bit is set to 1. Also, bits CMD1 to CMD0 in the TRDFCR register are set to 11b (complementary PWM mode, buffer data transferred at compare match between registers TRD0 and TRDGRA0), the content in the buffer registers (TRDGRD0, TRDGRC1, and TRDGRD1) is transferred to the general registers (TRDGRB0, TRDGRA1, and TRDGRB1).

During $m+1$, m , and $m-1$ operation, the IMFA bit remains unchanged and data are not transferred to registers such as the TRDGRA0 register.

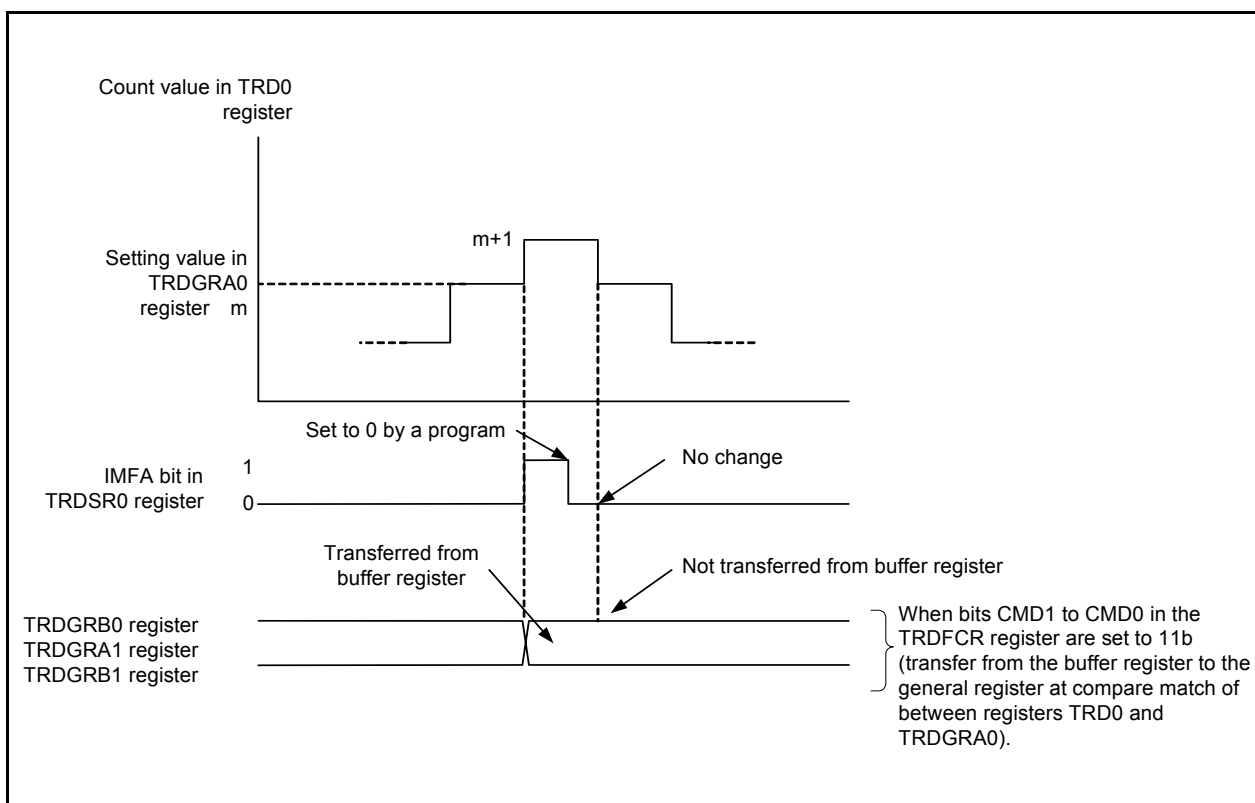


Figure 22.4 Operation at Compare Match between Registers TRD0 and TRDGRA0 in Complementary PWM Mode

- The TRD1 register counts 1, 0, FFFFh, 0, 1, in that order, when changing from decrement to increment operation.

The UDF bit is set to 1 when changing between 1, 0, and FFFFh operation. Also, when bits CMD1 to CMD0 in the TRDFCR register are set to 10b (complementary PWM mode, buffer data transferred at underflow in the TRD1 register), the content in the buffer registers (TRDGRD0, TRDGRC1, and TRDGRD1) is transferred to the general registers (TRDGRB0, TRDGRA1, and TRDGRB1). During FFFFh, 0, 1 operation, data are not transferred to registers such as the TRDGRB0 register. Also, at this time, the OVF bit remains unchanged.

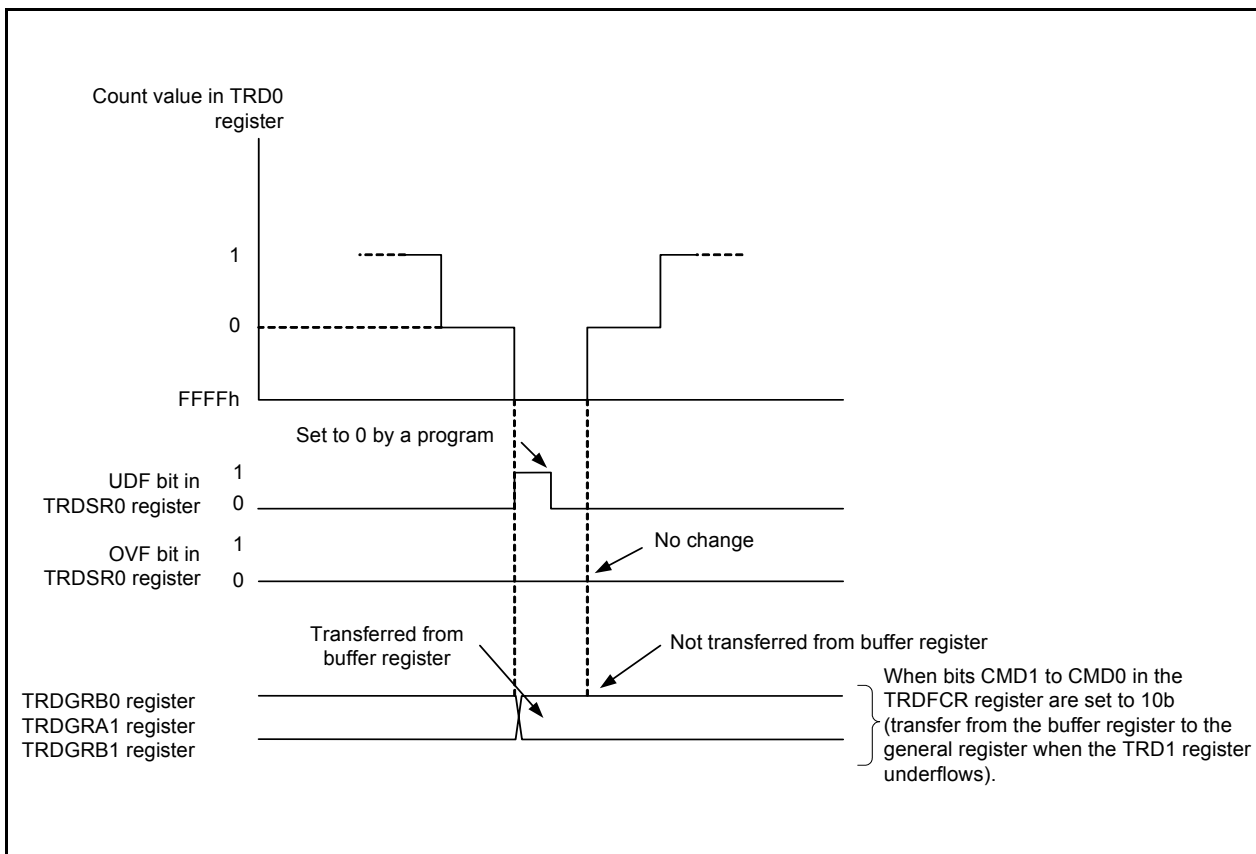


Figure 22.5 Operation when TRD1 Register Underflows in Complementary PWM Mode

- Select with bits CMD1 to CMD0 the timing of data transfer from the buffer register to the general register. However, transfer takes place with the following timing in spite of the value of bits CMD1 to CMD0 in the following cases:

Value in buffer register \geq value in TRDGRA0 register:

Transfer take place at underflow of the TRD1 register.

After this, when the buffer register is set to 0001h or above and a smaller value than the value of the TRDGRA0 register, and the TRD1 register underflows for the first time after setting, the value is transferred to the general register. After that, the value is transferred with the timing selected by bits CMD1 to CMD0.

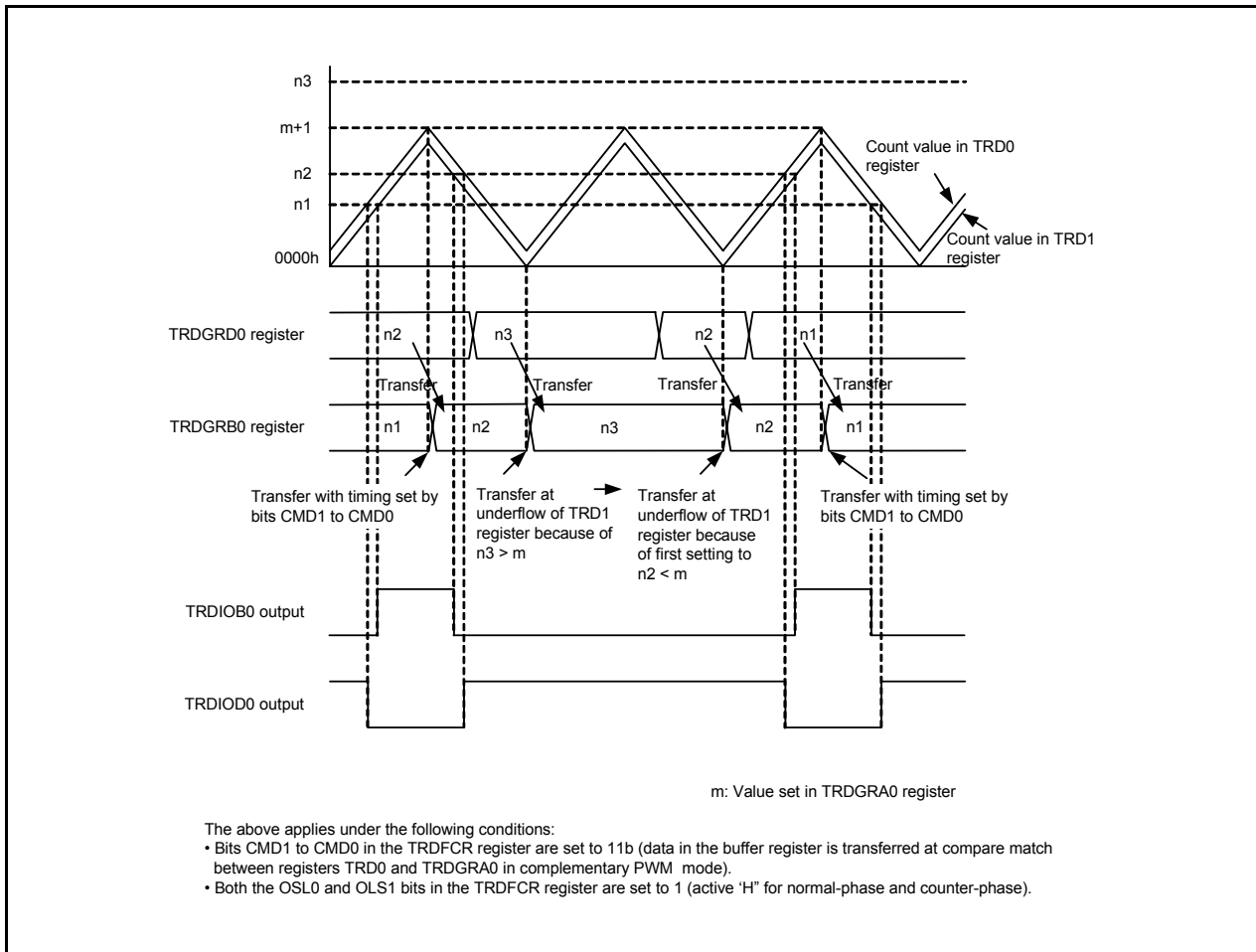


Figure 22.6 Operation when Value in Buffer Register \geq Value in TRDGRA0 Register in Complementary PWM Mode

When the value in the buffer register is set to 0000h:

Transfer takes place at compare match between registers TRD0 and TRDGRA0.

After this, when the buffer register is set to 0001h or above and a smaller value than the value of the TRDGRA0 register, and a compare match occurs between registers TRD0 and TRDGRA0 for the first time after setting, the value is transferred to the general register. After that, the value is transferred with the timing selected by bits CMD1 to CMD0.

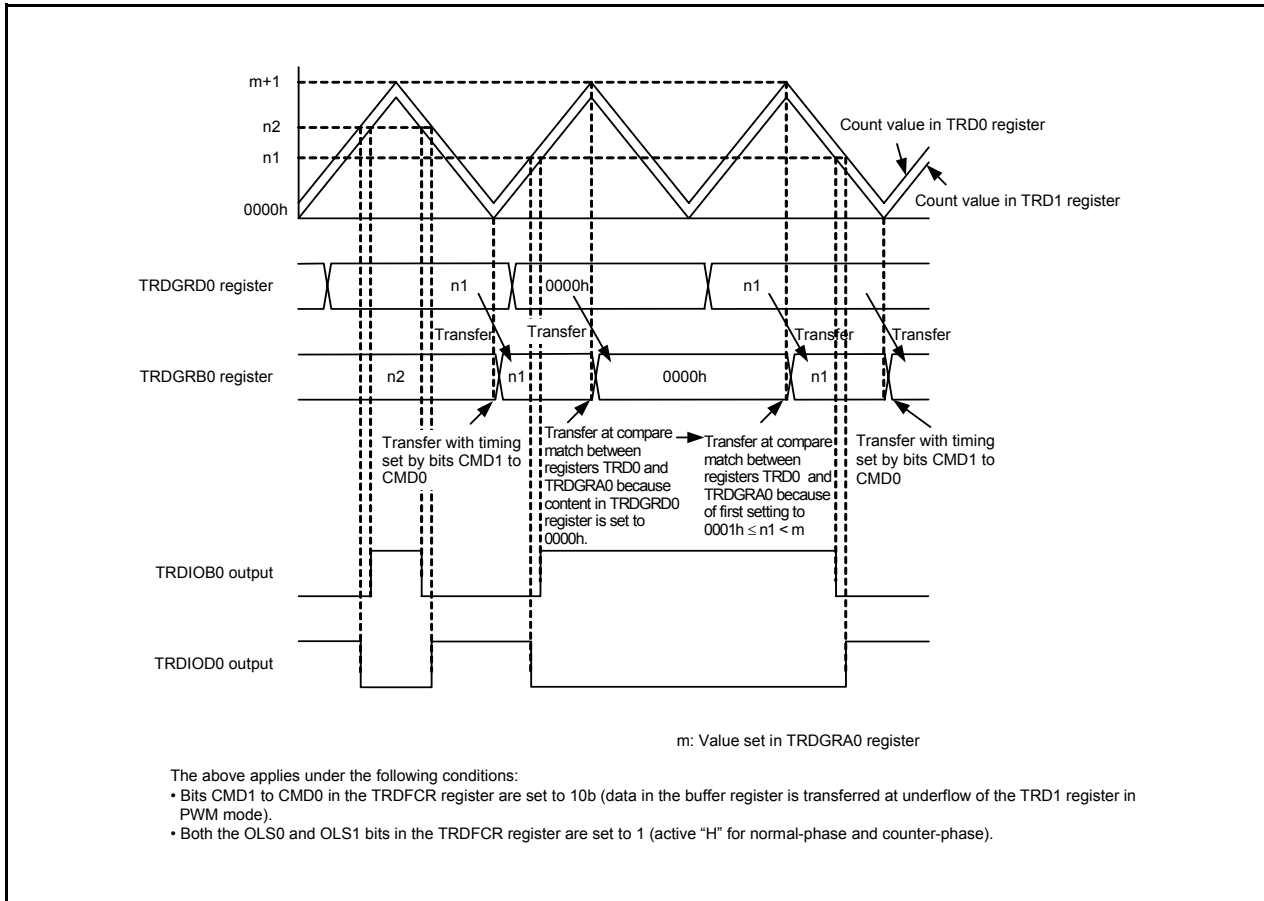


Figure 22.7 Operation when Value in Buffer Register Is Set to 0000h in Complementary PWM Mode

22.3.4.8 Count Source fOCO40M

- The count source fOCO40M can be used with supply voltage $V_{CC} = 3.0$ to 5.5 V. For supply voltage other than that, do not set bits TCK2 to TCK0 in registers TRDCR0 and TRDCR to 110b (select fOCO40M as the count source).

22.3.5 Notes on Timer RE

22.3.5.1 Starting and Stopping Count

Timer RE has the TSTART bit for instructing the count to start or stop, and the TCSTF bit, which indicates count start or stop. Bits TSTART and TCSTF are in the TRECRC1 register.

Timer RE starts counting and the TCSTF bit is set to 1 (count starts) when the TSTART bit is set to 1 (count starts). It takes up to 2 cycles of the count source until the TCSTF bit is set to 1 after setting the TSTART bit to 1. During this time, do not access registers associated with timer RE⁽¹⁾ other than the TCSTF bit.

Also, timer RE stops counting when setting the TSTART bit to 0 (count stops) and the TCSTF bit is set to 0 (count stops). It takes the time for up to 2 cycles of the count source until the TCSTF bit is set to 0 after setting the TSTART bit to 0. During this time, do not access registers associated with timer RE other than the TCSTF bit.

NOTE:

1. Registers associated with timer RE: TRESEC, TREMIN, TREHR, TREWK, TRECRC1, TRECRC2, and TRECSR.

22.3.5.2 Register Setting

Write to the following registers or bits when timer RE is stopped.

- Registers TRESEC, TREMIN, TREHR, TREWK, and TRECRC2
- Bits H12_H24, PM, and INT in TRECRC1 register
- Bits RCS0 to RCS3 in TRECSR register

Timer RE is stopped when bits TSTART and TCSTF in the TRECRC1 register are set to 0 (timer RE stopped).

Also, set all above-mentioned registers and bits (immediately before timer RE count starts) before setting the TRECRC2 register.

Figure 22.8 shows a Setting Example in Real-Time Clock Mode.

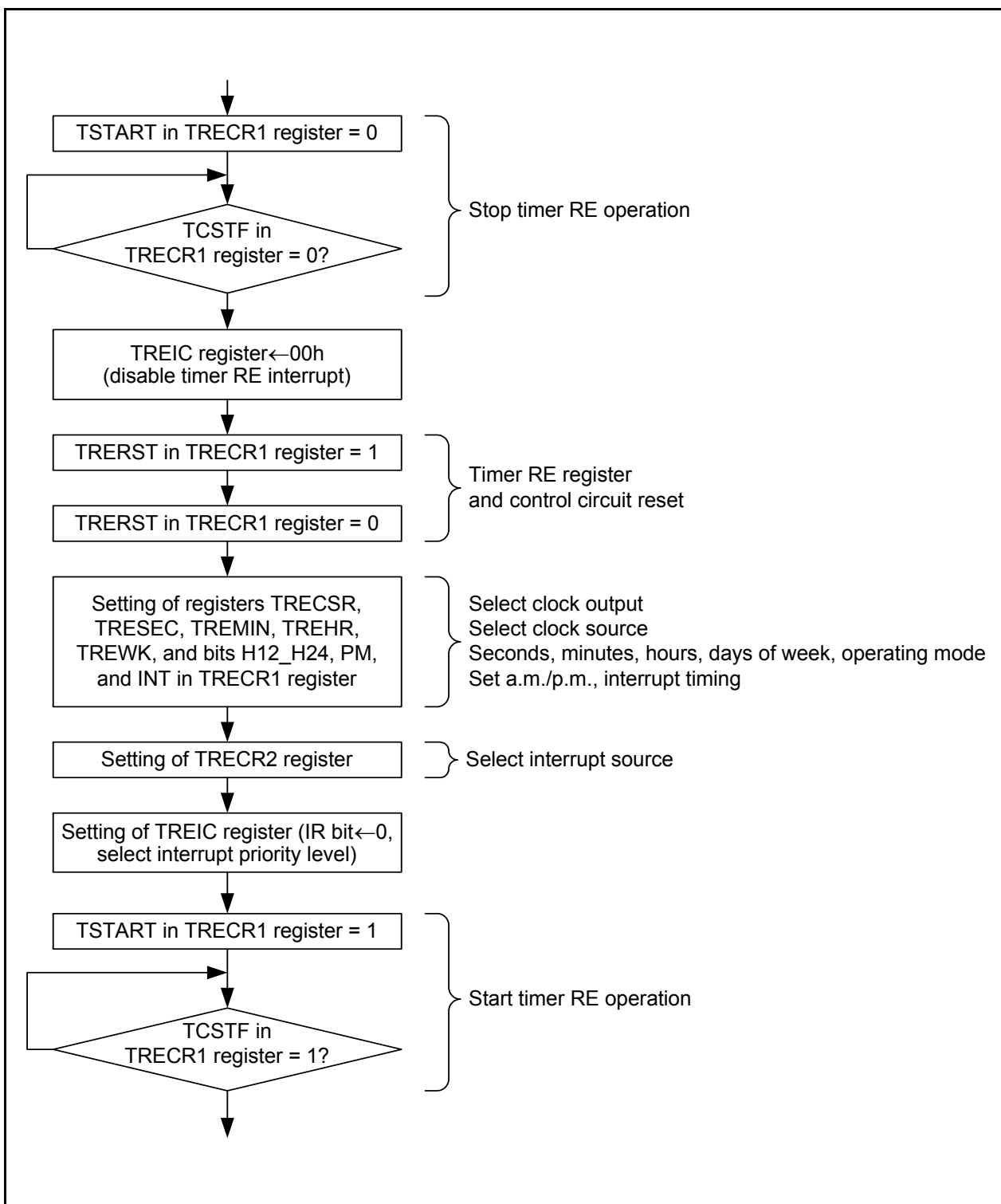


Figure 22.8 Setting Example in Real-Time Clock Mode

22.3.5.3 Time Reading Procedure of Real-Time Clock Mode

In real-time clock mode, read registers TRESEC, TREMIN, TREHR, and TREWK when time data is updated and read the PM bit in the TRECR1 register when the BSY bit is set to 0 (not while data is updated).

Also, when reading several registers, an incorrect time will be read if data is updated before another register is read after reading any register.

In order to prevent this, use the reading procedure shown below.

- Using an interrupt

Read necessary contents of registers TRESEC, TREMIN, TREHR, and TREWK and the PM bit in the TRECR1 register in the timer RE interrupt routine.

- Monitoring with a program 1

Monitor the IR bit in the TREIC register with a program and read necessary contents of registers TRESEC, TREMIN, TREHR, and TREWK and the PM bit in the TRECR1 register after the IR bit in the TREIC register is set to 1 (timer RE interrupt request generated).

- Monitoring with a program 2

- (1) Monitor the BSY bit.

- (2) Monitor until the BSY bit is set to 0 after the BSY bit is set to 1 (approximately 62.5 ms while the BSY bit is set to 1).

- (3) Read necessary contents of registers TRESEC, TREMIN, TREHR, and TREWK and the PM bit in the TRECR1 register after the BSY bit is set to 0.

- Using read results if they are the same value twice

- (1) Read necessary contents of registers TRESEC, TREMIN, TREHR, and TREWK and the PM bit in the TRECR1 register.

- (2) Read the same register as (1) and compare the contents.

- (3) Recognize as the correct value if the contents match. If the contents do not match, repeat until the read contents match with the previous contents.

Also, when reading several registers, read them as continuously as possible.

22.3.6 Notes on Timer RF

- Access registers TRF, TRFM0, and TRFM1 in 16-bit units.

Example of reading timer RF:

```
MOV.W    0290H,R0    ; Read out timer RF
```

- In input capture mode, a capture interrupt request is generated by inputting an edge selected by bits TRFC03 and TRFC04 in the TRFCR0 register even when the TSTART bit in the TRFCR0 register is set to 0 (count stops).

22.4 Notes on Serial Interface

- When reading data from the UiRB (i = 0 to 2) register either in the clock synchronous serial I/O mode or in the clock asynchronous serial I/O mode. Ensure the data is read in 16-bit units. When the high-order byte of the UiRB register is read, bits PER and FER in the UiRB register and the RI bit in the UiC1 register are set to 0. To check receive errors, read the UiRB register and then use the read data.

Example (when reading receive buffer register):

```
MOV.W    00A6H,R0    ; Read the UORB register
```

- When writing data to the UiTB register in the clock asynchronous serial I/O mode with 9-bit transfer data length, write data to the high-order byte first then the low-order byte, in 8-bit units.

Example (when reading transmit buffer register):

```
MOV.B    #XXH,00A3H  ; Write the high-order byte of U0TB register  
MOV.B    #XXH,00A2H  ; Write the low-order byte of U0TB register
```

22.5 Notes on Clock Synchronous Serial Interface

22.5.1 Notes on Clock Synchronous Serial I/O with Chip Select

Set the IICSEL bit in the PMR register to 0 (select clock synchronous serial I/O with chip select function) to use the clock synchronous serial I/O with chip select function.

22.5.2 Notes on I²C bus Interface

Set the IICSEL bit in the PMR register to 1 (select I²C bus interface function) to use the I²C bus interface.

22.5.2.1 Multimaster Operation

The following actions must be performed to use the I²C bus interface in multimaster operation.

- Transfer rate
Set the transfer rate by 1/1.8 or faster than the fastest rate of the other masters. For example, if the fastest transfer rate of the other masters is set to 400 kbps, the I²C-bus transfer rate in this MCU should be set to 223 kbps (= 400/1.18) or more.
- Bits MST and TRS in the ICCR1 register setting
 - (a) Use the MOV instruction to set bits MST and TRS.
 - (b) When arbitration is lost, confirm the contents of bits MST and TRS. If the contents are other than the MST bit set to 0 and the TRS bit set to 0 (slave receive mode), set the MST bit to 0 and the TRS bit to 0 again.

22.5.2.2 Master Receive Mode

Either of the following actions must be performed to use the I²C bus interface in master receive mode.

- (a) In master receive mode while the RDRF bit in the ICSR register is set to 1, read the ICDRR register before the rising edge of the 8th clock.
- (b) In master receive mode, set the RCVD bit in the ICCR1 register to 1 (disables the next receive operation) to perform 1-byte communications.

22.6 Notes on Hardware LIN

For the time-out processing of the header and response fields, use another timer to measure the duration of time with a Synch Break detection interrupt as the starting point.

22.7 Notes on A/D Converter

- Write to each bit (other than ADST bit) in the ADCON0 register, each bit in the ADCON1 register, or the SMP bit in the ADCON2 register when A/D conversion is stopped (before a trigger occurs).
When the VCUT bit in the ADCON1 register is changed from 0 (VREF not connected) to 1 (VREF connected), wait for at least 1 μ s before starting the A/D conversion.
- After changing the A/D operating mode, select an analog input pin again.
- When using the one-shot mode, ensure that A/D conversion is completed before reading the AD0 register. The IR bit in the ADIC register or the ADST bit in the ADCON0 register can be used to determine whether A/D conversion is completed.
- When using the repeat mode 0, select the frequency of the A/D converter operating clock ϕ AD or more for the CPU clock during A/D conversion.
Do not select the fOCO-F for the ϕ AD.
- If the ADST bit in the ADCON0 register is set to 0 (A/D conversion stops) by a program and A/D conversion is forcibly terminated during an A/D conversion operation, the conversion result of the A/D converter will be undefined. If the ADST bit is set to 0 by a program, do not use the value of the AD0 register.
- Connect 0.1 μ F capacitor between the VREF pin and AVSS pin.
- Do not enter stop mode during A/D conversion.
- Do not enter wait mode when the CM02 bit in the CM0 register is set to 1 (peripheral function clock stops in wait mode) during A/D conversion.

22.8 Notes on Flash Memory

22.8.1 CPU Rewrite Mode

22.8.1.1 Operating Speed

Before entering CPU rewrite mode (EW0 mode), select 5 MHz or below for the CPU clock using the CM06 bit in the CM0 register and bits CM16 to CM17 in the CM1 register. This does not apply to EW1 mode.

22.8.1.2 Prohibited Instructions

The following instructions cannot be used in EW0 mode because they reference data in the flash memory: UND, INTO, and BRK.

22.8.1.3 Interrupts

Table 22.2 lists the EW0 Mode Interrupts and Table 22.3 lists the EW1 Mode Interrupts.

Table 22.2 EW0 Mode Interrupts

| Mode | Status | When Maskable Interrupt Request is Acknowledged | When Watchdog Timer, Oscillation Stop Detection, Voltage Monitor 1, or Voltage Monitor 2 Interrupt Request is Acknowledged |
|------|---------------------|---|--|
| EW0 | During auto-erasure | Any interrupt can be used by allocating a vector in RAM | Once an interrupt request is acknowledged, auto-programming or auto-erasure is forcibly stopped immediately and the flash memory is reset. Interrupt handling starts after the fixed period and the flash memory restarts. Since the block during auto-erasure or the address during auto-programming is forcibly stopped, the normal value may not be read. Execute auto-erasure again and ensure it completes normally. Since the watchdog timer does not stop during the command operation, interrupt requests may be generated. Reset the watchdog timer regularly. |
| | Auto-programming | | |

NOTES:

1. Do not use the address match interrupt while a command is being executed because the vector of the address match interrupt is allocated in ROM.
2. Do not use a non-maskable interrupt while block 0 is being automatically erased because the fixed vector is allocated in block 0.

Table 22.3 EW1 Mode Interrupts

| Mode | Status | When Maskable Interrupt Request is Acknowledged | When Watchdog Timer, Oscillation Stop Detection, Voltage Monitor 1, or Voltage Monitor 2 Interrupt Request is Acknowledged |
|------|---|--|--|
| EW1 | During auto-erasure (erase-suspend function enabled) | Auto-erasure is suspended after $t_d(SR-SUS)$ and interrupt handling is executed. Auto-erasure can be restarted by setting the FMR41 bit in the FMR4 register to 0 (erase restart) after interrupt handling completes. | Once an interrupt request is acknowledged, auto-programming or auto-erasure is forcibly stopped immediately and the flash memory is reset. Interrupt handling starts after the fixed period and the flash memory restarts. Since the block during auto-erasure or the address during auto-programming is forcibly stopped, the normal value may not be read. Execute auto-erasure again and ensure it completes normally. Since the watchdog timer does not stop during the command operation, interrupt requests may be generated. Reset the watchdog timer regularly using the erase-suspend function. |
| | During auto-erasure (erase-suspend function disabled) | Auto-erasure has priority and the interrupt request acknowledgement is put on standby. Interrupt handling is executed after auto-erasure completes. | |
| | During auto-programming (program suspend function enabled) | Auto-programming is suspended after $t_d(SR-SUS)$ and interrupt handling is executed. Auto-programming can be restarted by setting the FMR42 bit in the FMR4 register to 0 (program restart) after interrupt handling completes. | |
| | During auto-programming (program suspend function disabled) | Auto-programming has priority and the interrupt request acknowledgement is put on standby. Interrupt handling is executed after auto-programming completes. | |

NOTES:

1. Do not use the address match interrupt while a command is executing because the vector of the address match interrupt is allocated in ROM.
2. Do not use a non-maskable interrupt while block 0 is being automatically erased because the fixed vector is allocated in block 0.

22.8.1.4 How to Access

Write 0 before writing 1 when setting the FMR01, FMR02, or FMR11 bit to 1. Do not generate an interrupt between writing 0 and 1.

22.8.1.5 Rewriting User ROM Area

In EW0 Mode, if the supply voltage drops while rewriting any block in which a rewrite control program is stored, it may not be possible to rewrite the flash memory because the rewrite control program cannot be rewritten correctly. In this case, use standard serial I/O mode.

22.8.1.6 Program

Do not write additions to the already programmed address.

22.8.1.7 Entering Stop Mode or Wait Mode

Do not enter stop mode or wait mode during erase-suspend.

22.8.1.8 Program and Erase Voltage for Flash Memory

To perform programming and erasure, use $V_{CC} = 2.7\text{ V}$ to 5.5 V as the supply voltage. Do not perform programming and erasure at less than 2.7 V .

22.9 Notes on Noise

22.9.1 Inserting a Bypass Capacitor between VCC and VSS Pins as a Countermeasure against Noise and Latch-up

Connect a bypass capacitor (at least 0.1 μF) using the shortest and thickest wire possible.

22.9.2 Countermeasures against Noise Error of Port Control Registers

During rigorous noise testing or the like, external noise (mainly power supply system noise) can exceed the capacity of the MCU's internal noise control circuitry. In such cases the contents of the port related registers may be changed.

As a firmware countermeasure, it is recommended that the port registers, port direction registers, and pull-up control registers be reset periodically. However, examine the control processing fully before introducing the reset routine as conflicts may be created between the reset routine and interrupt routines.

23. Notes on On-Chip Debugger

When using the on-chip debugger to develop and debug programs for the R8C/2A Group and R8C/2B Group take note of the following.

- (1) Do not access the related UART1 registers.
- (2) Some of the user flash memory and RAM areas are used by the on-ship debugger. These areas cannot be accessed by the user.
Refer to the on-chip debugger manual for which areas are used.
- (3) Do not set the address match interrupt (registers AIER, RMAD0, and RMAD1 and fixed vector tables) in a user system.
- (4) Do not use the BRK instruction in a user system.
- (5) Debugging is available under the condition of supply voltage $VCC = 2.7$ to 5.5 V. Debugging with the on-chip debugger under less than 2.7 V is not allowed.

Connecting and using the on-chip debugger has some special restrictions. Refer to the on-chip debugger manual for details.

24. Notes on Emulator Debugger

When using the emulator debugger to develop the R8C/2A Group and R8C/2B Group program and debug, pay the following attention.

- (1) Do not use the following flash memory areas because these areas are used for the emulator debugger. When debugging of these areas, intensive evaluation on the real chip is required.

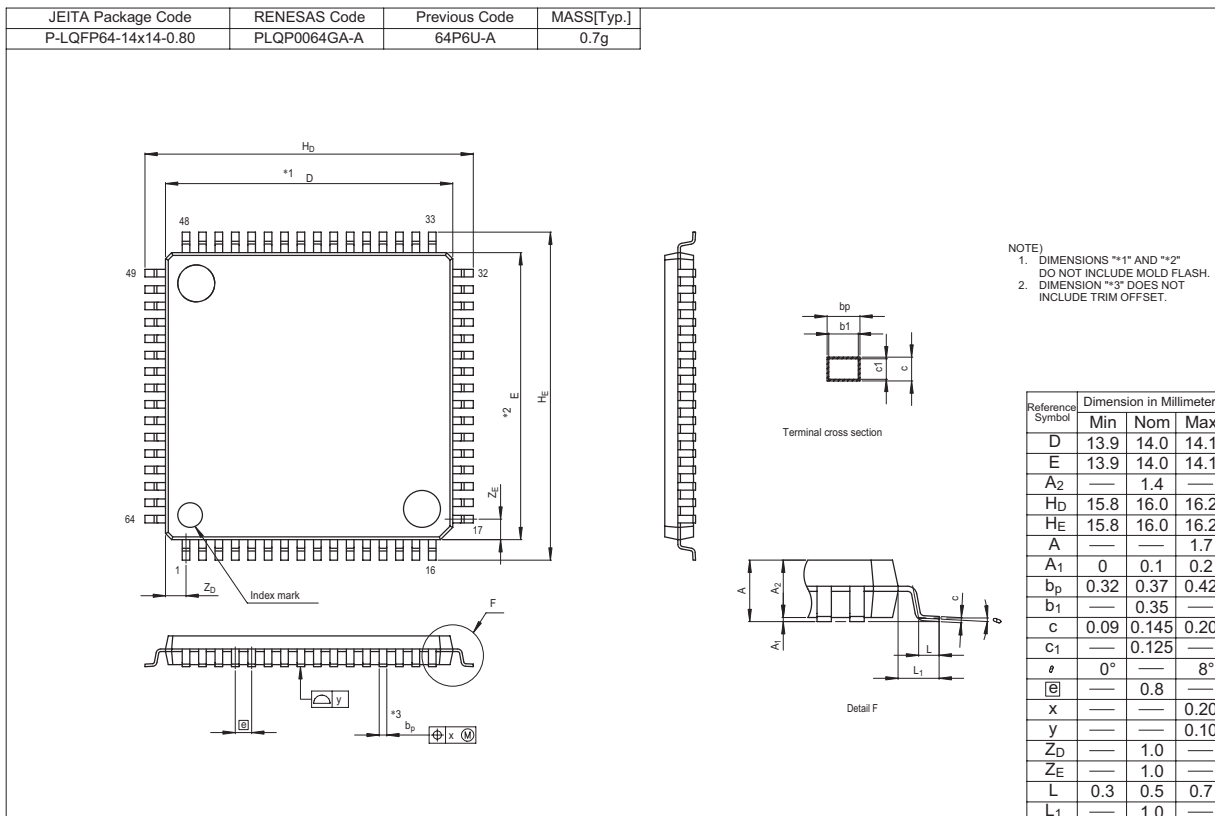
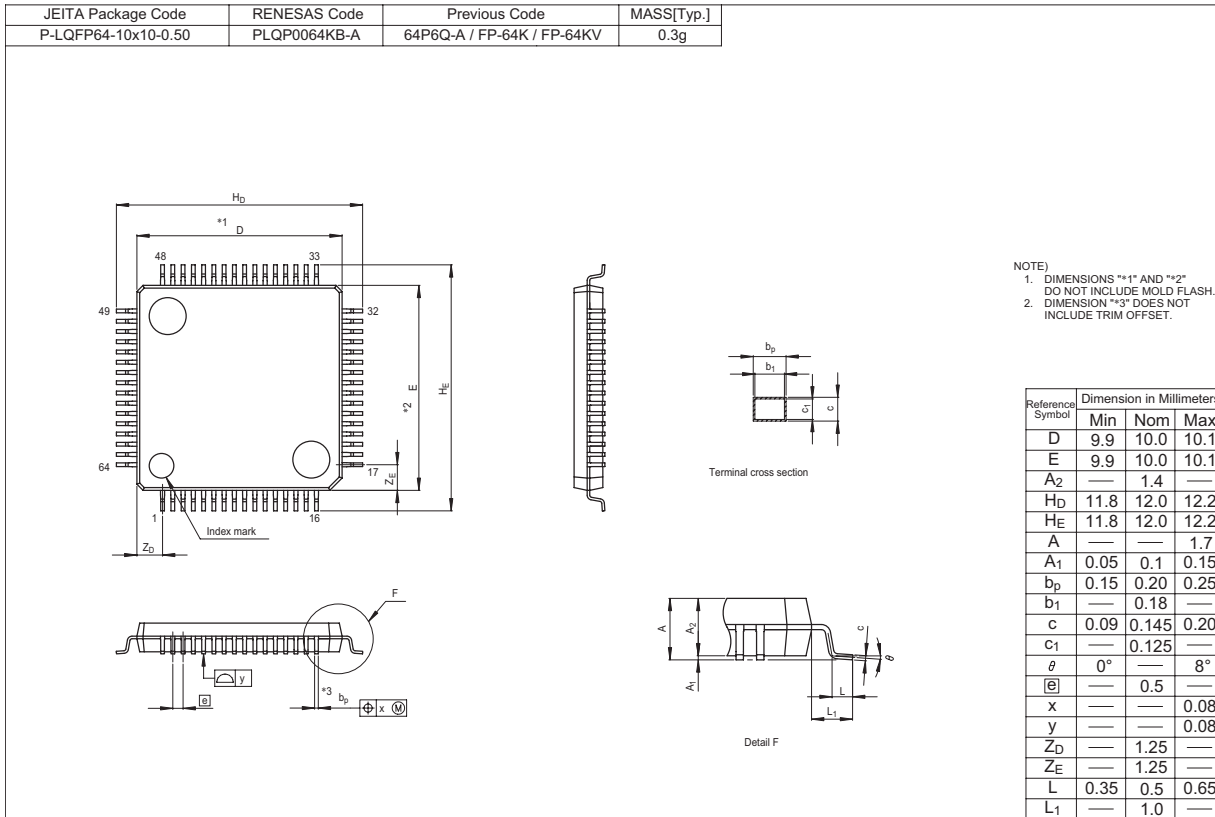
Target product: ROM capacity 128 MB product (Refer to **Table 1.5 Product List for R8C/2A Group** and **Table 1.6 Product List for R8C/2B Group**)

Unusable area: Addresses 20000h to 23FFFh

Connecting and using the emulator debugger has some peculiar restrictions. Refer to each emulator debugger manual for emulator debugger details.

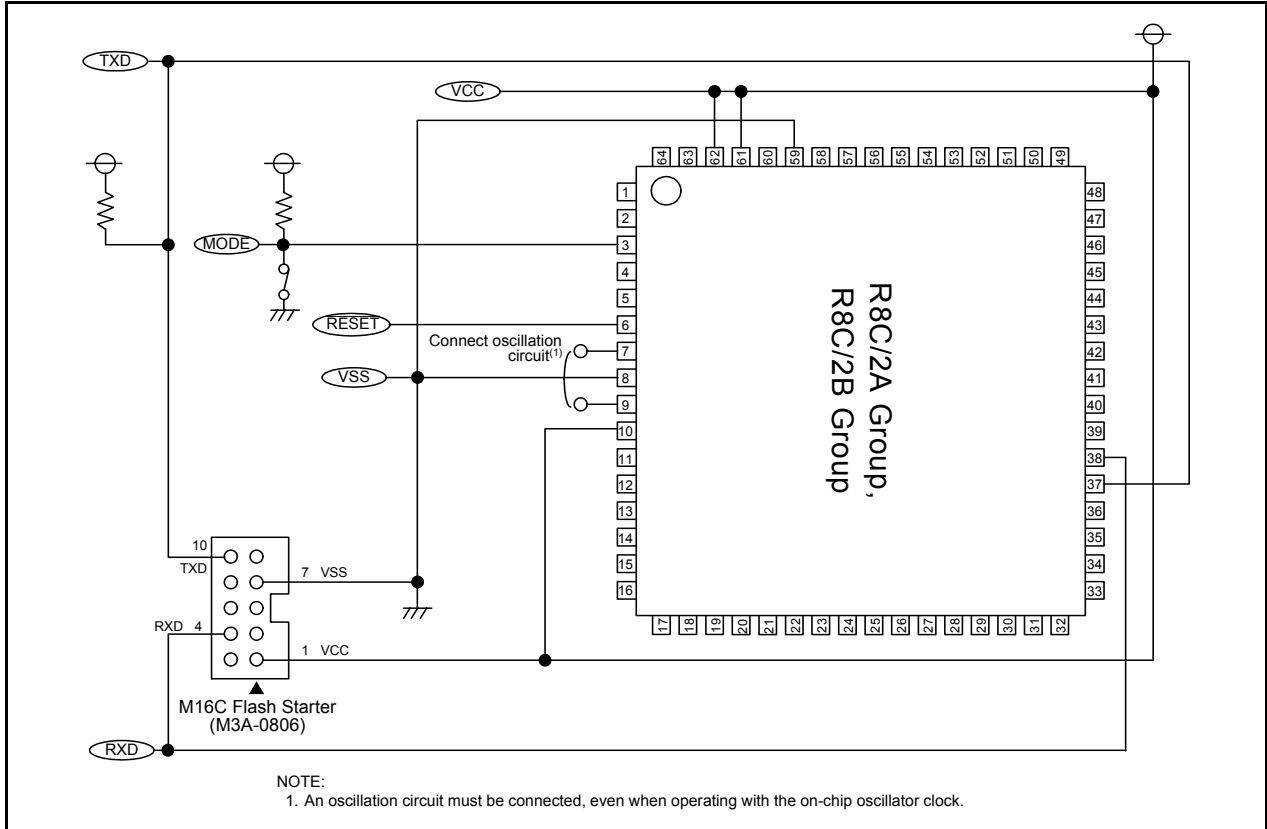
Appendix 1. Package Dimensions

Diagrams showing the latest package dimensions and mounting information are available in the “Packages” section of the Renesas Technology website.

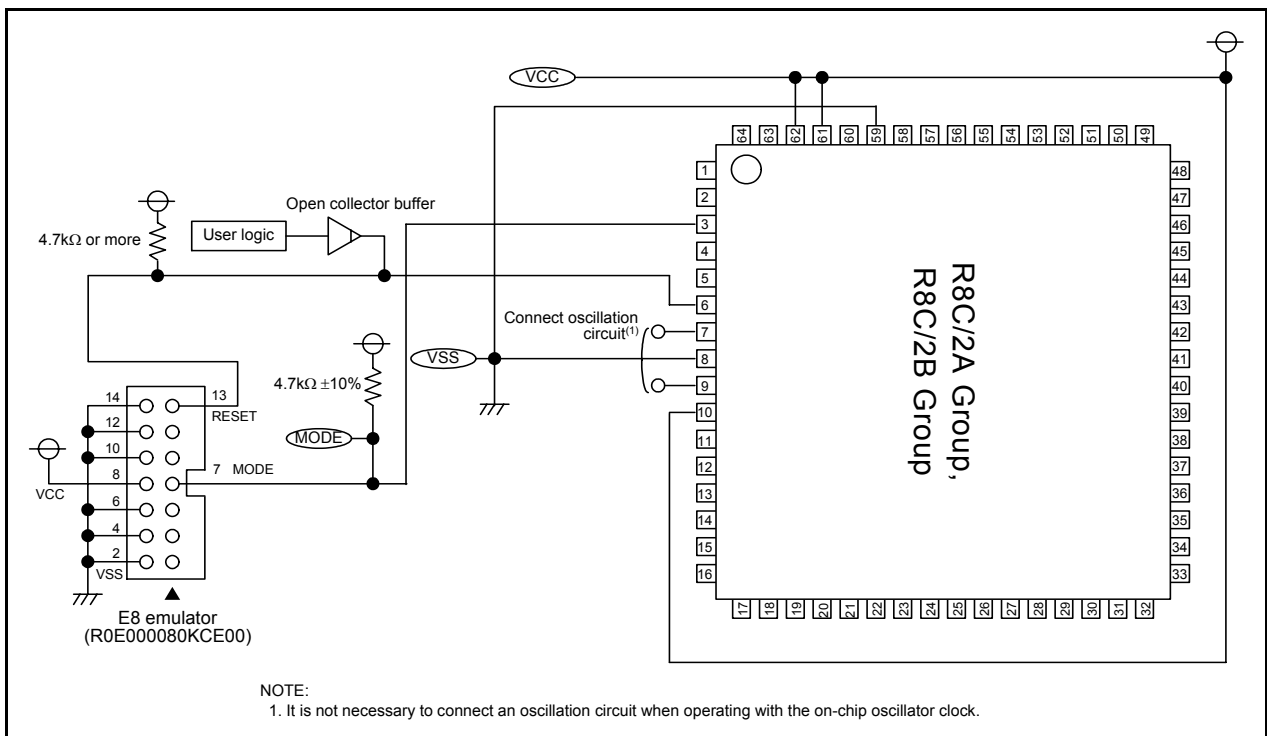


Appendix 2. Connection Examples between Serial Writer and On-Chip Debugging Emulator

Appendix Figure 2.1 shows a Connection Example with M16C Flash Starter (M3A-0806) and Appendix Figure 2.2 shows a Connection Example with E8 Emulator (R0E000080KCE00).



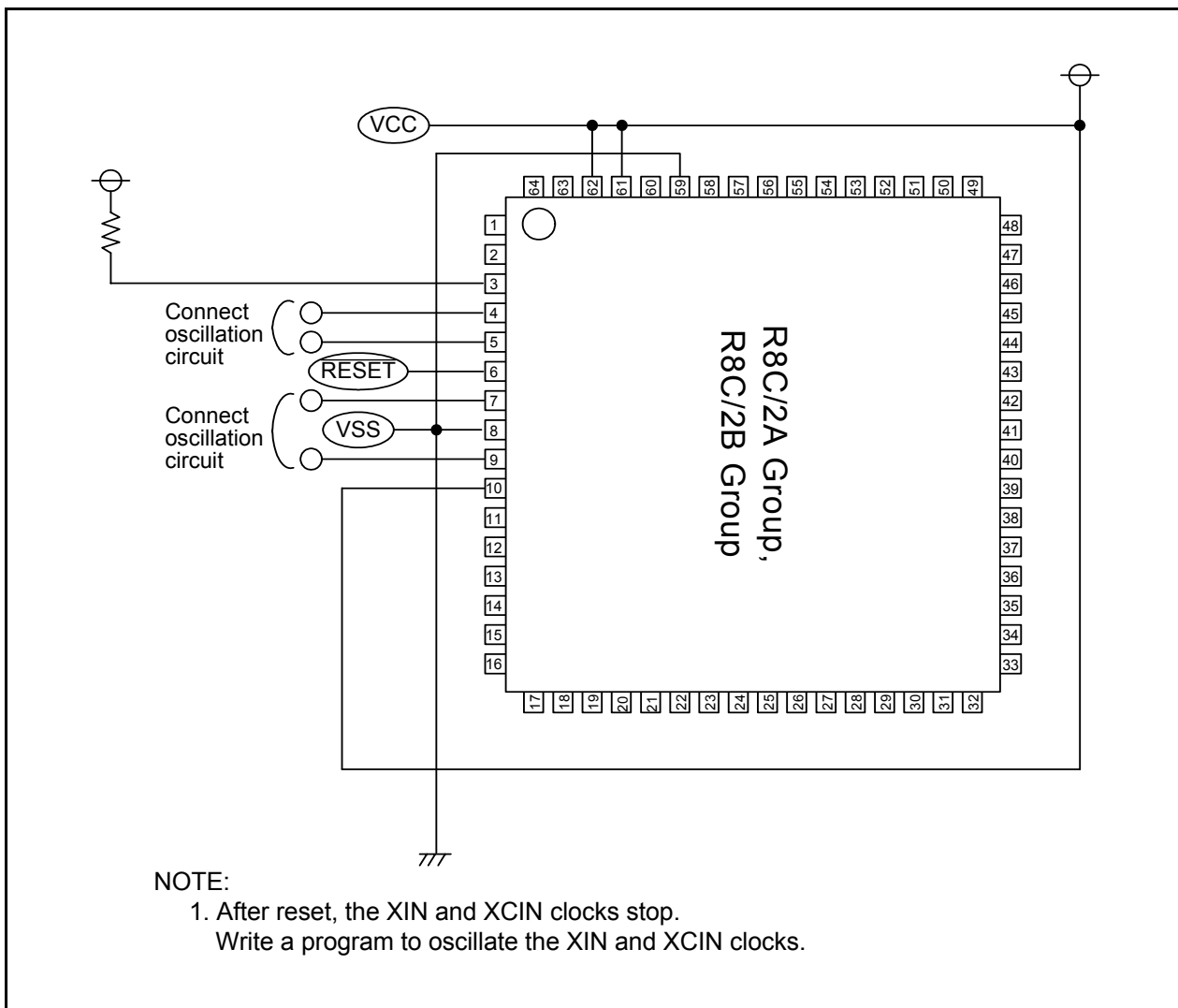
Appendix Figure 2.1 Connection Example with M16C Flash Starter (M3A-0806)



Appendix Figure 2.2 Connection Example with E8 Emulator (R0E000080KCE00)

Appendix 3. Example of Oscillation Evaluation Circuit

Appendix Figure 3.1 shows an Example of Oscillation Evaluation Circuit.



Appendix Figure 3.1 Example of Oscillation Evaluation Circuit

Index

| | | | |
|------------------|---|------------------------------|------------------------------|
| [A] | | [Q] | |
| AD0 | 474 | P2DRR | 70 |
| ADCON0 | 475 | PDi (i = 0 to 6 and 8) | 68 |
| ADCON1 | 476 | PM0 | 90 |
| ADCON2 | 474 | PM1 | 90 |
| ADIC | 127 | PMR | 70, 136, 380, 403, 433 |
| AIER | 143 | PRCR | 121 |
| [C] | | PUR0 | 71 |
| CAPIC | 127 | PUR1 | 71 |
| CM0 | 96 | PUR2 | 71 |
| CM1 | 97 | Pi (i = 0 to 6 and 8) | 69 |
| CMP0IC | 127 | [S] | |
| CMP1IC | 127 | S0RIC | 127 |
| CPSRF | 101 | S0TIC | 127 |
| CSPR | 152 | S1RIC | 127 |
| [E] | | S1TIC | 127 |
| DA0 to DA1 | 488 | S2RIC | 127 |
| DACON | 488 | S2TIC | 127 |
| [G] | | SAR | 432 |
| FMR0 | 498 | RMAD0 | 143 |
| FMR1 | 499 | RMAD1 | 143 |
| FMR4 | 500 | SSCRH | 397 |
| FRA0 | 99 | SSCRL | 398 |
| FRA1 | 99 | SSER | 400 |
| FRA2 | 100 | SSMR | 399 |
| FRA6 | 100 | SSMR2 | 402 |
| FRA7 | 100 | SSRDR | 403 |
| [I] | | SSSR | 401 |
| ICCR1 | 427 | SSTDOR | 403 |
| ICCR2 | 428 | SSUIC | 128 |
| ICDRR | 433 | [U] | |
| ICDRS | 433 | U0C0 to U2C0 | 378 |
| ICDRT | 432 | U0C1 to U2C1 | 379 |
| ICIER | 430 | U0BRG to U2BRG | 377 |
| ICMR | 429 | U0MR to U2MR | 377 |
| ICSR | 431 | U0RB to U2RB | 379 |
| IICIC | 128 | U0TB to U2TB | 378 |
| INT0IC | 129 | U1SR | 380 |
| INT1IC | 129 | TRA | 160 |
| INT2IC | 129 | TRACR | 159 |
| INT3IC | 129 | TRAIC | 127 |
| INTEN | 137 | TRAIOC | 159, 161, 164, 166, 168, 171 |
| INTF | 138 | TRAMR | 160 |
| [K] | | TRAPRE | 160 |
| KIEN | 141 | TRC | 202 |
| KUPIC | 127 | TRBCR | 175 |
| [M] | | TRCCR1 | 199, 222, 226, 231 |
| LINCR | 459 | TRCCR2 | 203 |
| LINCR2 | 459 | TRCDF | 204 |
| LINST | 460 | TRCGRA | 202 |
| MSTCR | 198, 250, 265, 283, 297, 308, 323, 396, 426 | TRCGRB | 202 |
| [O] | | TRCGRC | 202 |
| OCD | 98 | TRCGRD | 202 |
| OFS | 36, 152, 493 | TRBIC | 127 |
| | | TRCIC | 128 |
| | | TRCIER | 200 |
| | | TRBIOC | 176, 178, 182, 184, 189 |
| | | TRCIOR0 | 206, 215, 220 |
| | | TRCIOR1 | 206, 216, 221 |
| | | TRBMR | 176 |
| | | TRCMR | 198 |
| | | TRBOCR | 175 |
| | | TRCOER | 205 |

| | |
|----------------------------|------------------------------|
| TRBPR | 177 |
| TRBPRE | 177 |
| TRBSC | 177 |
| TRCSR | 201 |
| TRD0 | 259, 276, 291, 303, 316, 331 |
| TRD0IC | 128 |
| TRD1 | 259, 276, 291, 316 |
| TRD1IC | 128 |
| TRDCR0 | 255, 271, 288, 301, 313, 329 |
| TRDCR1 | 255, 271, 288, 313 |
| TRECR1 | 348, 355 |
| TRECR2 | 349, 355 |
| TRECSR | 350, 356 |
| TRDDF0 | 254 |
| TRDDF1 | 254 |
| TRDFCR | 253, 268, 286, 299, 311, 326 |
| TRDGRAi (i = 0 to 1) | 260, 276, 292, 304, 316, 332 |
| TRDGRC1 | 316 |
| TRDGRBi (i = 0 to 1) | 260, 276, 292, 304, 316, 332 |
| TRDGRCi (i = 0 to 1) | 260, 276, 292, 304, 332 |
| TRDGRDi (i = 0 to 1) | 260, 276, 292, 304, 316, 332 |
| TREIC | 127 |
| TRDIER0 | 259, 275, 290, 303, 315, 331 |
| TRDIER1 | 259, 275, 290, 303, 315, 331 |
| TRDIORA0 | 256, 272 |
| TRDIORA1 | 256, 272 |
| TRDIORC0 | 257, 273 |
| TRDIORC1 | 257, 273 |
| TREHR | 347 |
| TREMIN | 346, 354 |
| TRDMR | 251, 266, 284, 298, 310, 325 |
| TRDOCR | 270, 288, 328 |
| TRDOER1 | 269, 287, 300, 312, 327 |
| TRDOER2 | 269, 287, 300, 312, 327 |
| TRDPMR | 252, 267, 285 |
| TRDPOCR0 | 291 |
| TRDPOCR1 | 291 |
| TRESEC | 346, 354 |
| TRDSR0 | 258, 274, 289, 302, 314, 330 |
| TRDSR1 | 258, 274, 289, 302, 314, 330 |
| TRDSTR | 251, 266, 284, 298, 309, 324 |
| TREWK | 347 |
| TRF | 363 |
| TRFCR0 | 364 |
| TRFCR1 | 365 |
| TRFIC | 127 |
| TRFM0 | 363 |
| TRFM1 | 363 |
| TRFOUT | 365 |

[W]

| | |
|------------|---------|
| VCA1 | 45 |
| VCA2 | 45, 101 |
| WDC | 151 |
| WDTR | 151 |
| WDTS | 151 |
| VW0C | 46 |
| VW1C | 47 |
| VW2C | 48 |

REVISION HISTORY

R8C/2A Group, R8C/2B Group Hardware Manual

| Rev. | Date | Description | |
|------|--------------|---|---|
| | | Page | Summary |
| 0.01 | Jul 28, 2006 | – | First Edition issued |
| 0.10 | Sep 15, 2006 | 95 110 160 202 254 382 428 430 432 433 461 496 560 | Figure 10.6 FRA1 register NOTE1 revised Figure 10.14 revised Table 14.5 revised Figure 14.39 revised Figure 14.80 revised Figure 15.12 revised Figure 16.33 revised Figure 16.34, Figure 16.35 revised Figure 16.36 revised Figure 16.37 revised Table 17.2 revised 18.6 revised 23 (2) revised, (5) deleted |
| 1.00 | Feb 09, 2007 | All pages 3 5 6 7 17 18 19 23 36 42 72 76 77 88 89 90 91 93 96 | “Preliminary” deleted Table 1.2 revised Table 1.4 revised Table 1.5 and Figure 1.1 revised Table 1.6 and Figure 1.2 revised Figure 3.1 revised Figure 3.2 revised Table 4.1; • 0008h: “Module Standby Control Register” → “Module Operation Enable Register” revised • 000Ah: “00XXX000b” → “00h” revised • 000Fh: “00011111b” → “00X11111b” revised • 002Bh: “High-Speed On-Chip Oscillator Control Register 6” added Table 4.5; 0105h: “LIN Control Register 2” register name revised 5.2 and Figure 5.7 revised Figure 6.5; VCA2 register NOTE6 revised Table 7.17 and Table 7.19 revised Table 7.29 and Table 7.31 revised Table 7.35 revised Table 9.1, Table 9.2 and Table 9.3 revised Table 9.4 added 10 and Table 10.1 NOTE4 revised Figure 10.1 revised Figure 10.3 NOTE4 revised Figure 10.6; FRA0 register NOTE2 revised |

REVISION HISTORY

R8C/2A Group, R8C/2B Group Hardware Manual

| Rev. | Date | Description | |
|------|--|-------------|---|
| | | Page | Summary |
| 1.00 | Feb 09, 2007 | 97 | Figure 10.7; FRA2 register revised, FRA6 register added |
| | | 98 | Figure 10.9 NOTE6 revised |
| | | 99 | Figure 10.10 added |
| | | 101 | 10.2.2 revised |
| | | 106 | 10.5.1.2 and 10.5.1.4 revised |
| | | 108 | Table 10.3 revised |
| | | 110 | 10.5.2.5 and Figure 10.14 revised |
| | | 112 | Figure 10.15 revised |
| | | 117 | 10.7.1 and 10.7.2 revised |
| | | 118 | Figure 11.1 revised |
| | | 121 | 12.1.3.1 revised |
| | | 135 | Figure 12.14 revised and Figure 12.15 added |
| | | 136 | Figure 12.16 NOTE1 revised |
| | | 139 | Table 12.6 revised |
| | | 143 | 12.6.4 deleted |
| | | 148 | Figure 13.2; WDC register revised |
| | | 180 | Table 14.10 NOTE2 added |
| | | 185 | Table 14.11 NOTE2 added |
| | | 192 | Figure 14.25 revised |
| | | 201 | Table 14.15 revised |
| | | 224 | Table 14.23 revised |
| | | 227 | Figure 14.57 revised |
| | | 244 | Figure 14.67 revised |
| | | 258 | Table 14.40 revised |
| | | 259 | Figure 14.82 revised |
| | | 260 | Figure 14.83; TRDSTR register revised |
| | | 271 | Figure 14.95 revised |
| | | 274 | Figure 14.97 revised |
| | | 276 | Table 14.42 revised |
| | | 277 | Figure 14.99 revised |
| | | 278 | Figure 14.100; TRDSTR register revised |
| | | 290 | Table 14.44 revised |
| | | 291 | Figure 14.113 revised |
| 292 | Figure 14.114; TRDSTR register revised | | |
| 302 | Figure 14.124 revised | | |
| 303 | Figure 14.125 revised | | |
| 316 | Table 14.48 revised | | |
| 317 | Figure 14.137 revised | | |
| 318 | Figure 14.138 revised | | |

REVISION HISTORY

R8C/2A Group, R8C/2B Group Hardware Manual

| Rev. | Date | Description | |
|------|---------------------------|-------------|---|
| | | Page | Summary |
| 1.00 | Feb 09, 2007 | 331 | 14.4.12.1 and Table 14.51 revised |
| | | 355 to 367 | 14.6; The following bit name is revised. <ul style="list-style-type: none"> • TRFC00 → TSTART, TRFC01 → TCK0, TRFC02 → TCK1: Bits in TRFCR0 register • TRFC10 → TIPF0, TRFC11 → TIPF1, TRFC12 → CCLR, TRFC13 → TMOD: Bits in TRFCR1 register |
| | | 359 | Figure 14.178 NOTE1 deleted |
| | | 365 | Figure 14.181 revised |
| | | 367 | 14.6.3 revised |
| | | 375 | Table 15.1 NOTE2 revised |
| | | 378 | Figure 15.9 revised |
| | | 381 | Table 15.4 revised |
| | | 384 | Figure 15.12 revised |
| | | 387 | 15.3 revised |
| | | 390 | Figure 16.2 MSTCR register added |
| | | 395 | Figure 16.7 NOTE2 revised |
| | | 420 | Figure 16.25 revised |
| | | 421 | Figure 16.26 NOTE7 added |
| | | 423 | Figure 16.28 NOTE3 revised |
| | | 450 | 16.3.8.2 and 16.3.8.3 added |
| | | 453 | Figure 17.2; LINCR2 register revised |
| | | 456 | Figure 17.5 revised |
| | | 460 | Figure 17.9 revised |
| | | 461 | Figure 17.10 revised |
| | | 462 | 17.4.3 and Figure 17.11 revised |
| | | 463 | 17.4.4 added |
| | | 469 | Figure 18.3 NOTE4 revised |
| | | 471 | Table 18.2 revised |
| | | 472 | Figure 18.5 NOTE4 revised |
| | | 475 | Figure 18.7 NOTE4 revised |
| | | 480 | 18.7 revised |
| | | 483 | Table 20.1 and Table 20.2 revised |
| | | 484 | 20.2 and Figure 20.1 revised |
| | | 485 | Figure 20.2 revised |
| | | 488 | Table 20.3 NOTE1 revised |
| | | 489 | 20.4.1 and 20.4.2; “td(SR-ES)” → “td(SR-SUS)” revised |
| | | 490 | 20.4.2.3 and 20.4.2.4 revised |
| 491 | 20.4.2.15 revised | | |
| 492 | Figure 20.5 revised | | |
| 494 | Figure 20.7 NOTE5 revised | | |

REVISION HISTORY

R8C/2A Group, R8C/2B Group Hardware Manual

| Rev. | Date | Description | |
|------|---|-------------|---|
| | | Page | Summary |
| 1.00 | Feb 09, 2007 | 496 | Figure 20.9 “any area other than the flash memory” → “the RAM” revised |
| | | 497 | Figure 20.11; “any area other than the flash memory” → “the RAM”, “15 μs” → “30 μs”, and NOTES 1 and 3 revised |
| | | 499 | 20.4.3.4 revised |
| | | 500 | Figure 20.13 revised |
| | | 501 | 20.4.3.5 revised |
| | | 502 | Figure 20.15 revised |
| | | 504 | Table 20.6 “FRM00 Register” → “FRM0 Register” revised |
| | | 506 | Table 20.7 revised |
| | | 514 | Table 21.2 revised |
| | | 515 | Table 21.3 and Table 21.4; NOTE1 revised |
| | | 520 | Table 21.11 revised |
| | | 527 | Table 21.17 revised |
| | | 529 | Table 21.21 and Figure 21.11; “i = 0 to 2” revised |
| | | 531 | Table 21.24 revised |
| | | 533 | Table 21.28 revised, Figure 21.16 “i = 0 to 2” revised |
| | | 535 | Table 21.31 revised |
| | | 536 | Table 21.34 revised |
| | | 537 | Table 21.35 and Figure 21.21; “i = 0 to 2” revised |
| | | 538 | 22.1.1 and 22.1.2 revised |
| | | 539 | 22.2.4 deleted |
| | | 545 | 22.3.4.1 and Table 22.1 revised |
| | | 554 | 22.3.6 revised |
| | | 555 | 22.4 revised |
| | | 557 | 22.5.2.2 and 22.5.2.3 added |
| | | 559 | 22.7 revised |
| | | 565 | 24. Notes on Emulator Debugger added |
| 567 | Appendix Figure 2.1 and Appendix Figure 2.2 revised | | |
| 568 | Appendix Figure 3.1 NOTE1 revised | | |
| 2.00 | Nov 26, 2007 | – | “RENESAS TECHNICAL UPDATE” reflected: TN-16C-A164A/E, TN-16C-A167A/E |
| | | All pages | “PTLG0064JA-A (64F0G) package” added |
| | | 2, 4 | Table 1.1, Table 1.3 Clock: “Real-time clock (timer RE)” added |
| | | 3, 5 | Table 1.2 and Table 1.4; • Operating Ambient Temperature: Y version added • Package: 64-pin FLGA added NOTE1 added |
| | | 6, 7 | Table 1.5 and Figure 1.1 revised |
| | | 8, 9 | Table 1.6 and Figure 1.2 revised |
| | | 11 | Figure 1.4 “64-pin LQFP Package” added |

REVISION HISTORY

R8C/2A Group, R8C/2B Group Hardware Manual

| Rev. | Date | Description | |
|------|---|--------------|---|
| | | Page | Summary |
| 2.00 | Nov 26, 2007 | 12 | Figure 1.5 added |
| | | 20, 21 | Figure 3.1 and Figure 3.2 revised |
| | | 22 | Table 4.1 002Ch: High-Speed On-Chip Oscillator Control Register 7 added |
| | | 25 | Table 4.4 00F5h: After reset "00h" → "000000XXb" revised |
| | | 36 | Figure 5.3 revised |
| | | 36, 152, 493 | Figure 5.4, Figure 13.3, Figure 20.4 OFS NOTE1 revised |
| | | 37 | 5.1.1, 5.1.2 "Wait for 1/fOCO-S × 20." → "Wait for 10 μs or more." |
| | | 38 | Figure 5.5, Figure 5.6 revised |
| | | 75 | Table 7.17 Function: RXD0 input NOTE1 added |
| | | 79 | Table 7.29 revised |
| | | 85 | Table 7.54 Function: Input port NOTE1 added Table 7.55 Function: RXD2 input NOTE1 added |
| | | 86 | Table 7.58 Function: RXD1 input NOTE1 added |
| | | 94 | Figure 10.1 "Clock prescaler" added |
| | | 99 | Figure 10.6 FRA1 revised |
| | | 100 | Figure 10.7 FRA2: NOTE2 deleted, FRA7 added |
| | | 104 | 10.2.2 "The frequency correction to the FRA1 before use." added |
| | | 117 | 10.6.1 "To use the high-speed on-chip oscillator clock for the CPU clock and then set bits OCD1 to OCD0 to 11b." revised |
| | | 136 | 12.2.1 "... with the pulse output forced cutoff of timer RD and the INT1 pin is shared with the external trigger input pin of timer RA." → "... with the pulse output forced cutoff of timer RC and timer RD, and the external trigger input of timer RB." |
| | | 147 | Figure 12.22 NOTE2 revised |
| | | 156 | Table 14.1 Timer RE: Count sources "• fC32" deleted |
| | | 157 | Table 14.2 Timer RC: "TRDIOA" → "TRCIOA", "TRDIOB" → "TRCIOB", "TRDIOC" → "TRCIOC", "TRDIOD" → "TRCIOD", Timer RF: Input pin "TCIN" → "TRFI" |
| | | 158 | Figure 14.1 "TSTART" → "TCSTF" |
| | | 162 | Figure 14.5 "Both bits TSTART ... are set to 0 (During count)." → "Both bits TSTART ... are set to 1 (During count)." |
| | | 173 | 14.1.6 "• When the TRAPRE register is for each write interval. • When the TRA register is for each write interval." added |
| | | 174 | 14.2 "The reload register and counter are allocated at the same address" deleted |
| | | 177 | Figure 14.15 "Programmable one-shot mode" → "Programmable one-shot generation mode" |
| 180 | Figure 14.17 "Both bits TSTART are set to 0 (During count)." → "Both bits TSTART are set to 1 (During count)." | | |

| Rev. | Date | Description | |
|------|---|-------------|--|
| | | Page | Summary |
| 2.00 | Nov 26, 2007 | 183 | Table 14.10 Count stop conditions: “• When the TOSSP is set to 0 (one-shot stops).” → “• When the TOSSP is set to 1 (one-shot stops).” |
| | | 191 | 14.2.5 NOTE TRBIOC added |
| | | 191 to 194 | 14.2.5.1, 14.2.5.2, 14.2.5.3, 14.2.5.4 added |
| | | 204 | Figure 14.37 TRCIOR0: b3 revised, NOTE4 added |
| | | 213 | 14.3.4 “The TRCGRA register can input-capture trigger input.” added Table 14.17 revised |
| | | 214 | Figure 14.43 revised |
| | | 215 | Figure 14.44 b3 revised, NOTE3 added |
| | | 218 | Table 14.19 Select functions: “or output level inverted” → “or toggle output” |
| | | 220 | Figure 14.48 b3 revised |
| | | 223 | Figure 14.51 “• The CCLR bit in the TRCCR1 register is set to 0” → “• The CCLR bit in the TRCCR1 register is set to 1” |
| | | 269 | Figure 14.88 TRDOER1 revised |
| | | 274 | Figure 14.93 revised |
| | | 280 | Figure 14.99 revised |
| | | 289 | Figure 14.107 revised |
| | | 298 | Figure 14.116 “TRD” → “TRD0” |
| | | 302 | Figure 14.120 revised |
| | | 304 | Figure 14.123 NOTE1 revised |
| | | 309 | Figure 14.127 b0, b1 revised |
| | | 310 | Figure 14.128 “TRD” → “TRD0” |
| | | 314 | Figure 14.132 revised |
| | | 324 | Figure 14.140 revised |
| | | 330 | Figure 14.146 revised |
| | | 334 | Figure 14.150 “TSTP0 bit in TRDSTR register” → “CSEL0 bit in TRDSTR register” |
| | | 350 | Figure 14.164 b0, b1: “Set to 00 in real-time clock mode.” → “Set to 00b in real-time clock mode.” |
| | | 359 | Figure 14.173 revised |
| | | 365 | Figure 14.179 NOTE4 added |
| | | 377 | Figure 15.4 UARTi Transmit/Receive Mode Register revised |
| | | 380 | Figure 15.7 After Reset: “00h” → “000000XXb”, b1-b0 revised |
| | | 388 | Table 15.5 NOTE2 added |
| | | 397 | Figure 16.3 NOTE4 deleted |
| 398 | Figure 16.4 SOLP: “Cannot write to this.” → “The SOLP bit remains unchanged even if 1 is written to it.”, NOTE4 deleted | | |
| 399 | Figure 16.5 NOTE2 deleted | | |

REVISION HISTORY

R8C/2A Group, R8C/2B Group Hardware Manual

| Rev. | Date | Description | |
|------|---|-------------|---|
| | | Page | Summary |
| 2.00 | Nov 26, 2007 | 400 | Figure 16.6 NOTE1 deleted |
| | | 401 | Figure 16.7 NOTE7 revised |
| | | 402 | Figure 16.8 NOTE5 revised |
| | | 403 | Figure 16.9 SSTDR NOTE1 deleted, SSRDR NOTE2 deleted |
| | | 417 | Figure 16.19 revised |
| | | 423 | 16.2.8.1 deleted |
| | | 427 | Figure 16.26 NOTE6 deleted |
| | | 428 | Figure 16.27 NOTE5 deleted |
| | | 429 | Figure 16.28 NOTE7 deleted |
| | | 430 | Figure 16.29 NOTE3 deleted |
| | | 431 | Figure 16.30 NOTE7 deleted |
| | | 432 | Figure 16.31 SAR, ICDRT NOTE1 deleted |
| | | 433 | Figure 16.31 ICDRR NOTE1 deleted |
| | | 456 | 16.3.8.1 deleted |
| | | 457 | Figure 17.1 revised |
| | | 459 | 17.3 “• LIN Special Function Register (LINCR2)” → “• LIN Control Register 2 (LINCR2)” |
| | | 462, 463 | Figure 17.5, Figure 17.6 revised |
| | | 464 | Figure 17.7 revised |
| | | 466 | Figure 17.9 revised |
| | | 469 | Figure 17.12 revised |
| | | 484 | Figure 18.11 revised |
| | | 494 | Table 20.3 Areas in which a rewrite control program can be executed: EW1 Mode “Executing directly in user ROM area is possible” → “Executing directly in user ROM or RAM area possible” |
| | | 500 | Figure 20.7 NOTE5 revised |
| | | 502 | Figure 20.9 NOTE2 “Write to the FMR01 bit in the RAM.” added |
| | | 503 | Figure 20.11 NOTE4 deleted |
| | | 506, 508 | Figure 20.13, Figure 20.15 revised |
| | | 520 | Table 21.1 Pd: Rated Value “TBD” → “700” revised |
| | | 521 | Table 21.2 NOTE2 revised |
| | | 527 | Table 21.11 revised |
| | | 562 | Figure 22.8 revised |
| 566 | 22.5.1.1, 22.5.2.1 deleted | | |
| 577 | Package Dimensions “PTLG0064JA-A (64F0G) package” added | | |
| 578 | Appendix Figure 2.1, Appendix Figure 2.2 revised | | |
| 579 | Appendix Figure 3.1 revised | | |

R8C/2A Group, R8C/2B Group Hardware Manual

Publication Date: Rev.0.01 Jul 28, 2006
Rev.2.00 Nov 26, 2007

Published by: Sales Strategic Planning Div.
Renesas Technology Corp.

© 2007. Renesas Technology Corp., All rights reserved. Printed in Japan

R8C/2A Group, R8C/2B Group Hardware Manual



Renesas Electronics Corporation

1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0324-0200