

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/14 Group, R8C/15 Group

Hardware Manual

RENESAS 16-BIT SINGLE-CHIP
MICROCOMPUTER
M16C FAMILY / R8C/Tiny 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 Technology Corp. without notice. Please review the latest information published by Renesas Technology Corp. through various means, including the Renesas Technology Corp. website (<http://www.renesas.com>).

Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage. Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of non-flammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein. The information described here may contain technical inaccuracies or typographical errors. Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors. Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (<http://www.renesas.com>).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination. Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.

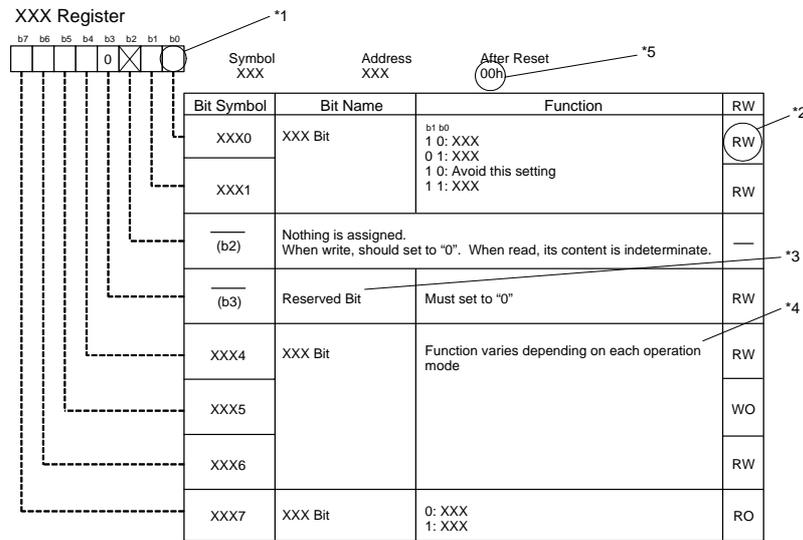
How to Use This Manual

1. Introduction

This hardware manual provides detailed information on the R8C/14 Group, R8C/15 Group of microcomputers.
Users are expected to have basic knowledge of electric circuits, logical circuits and microcomputers.

2. Register Diagram

The symbols, and descriptions, used for bit function in each register are shown 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 concerned. As the bit may be use for future functions, set to "0" when writing to this bit.
•Do not set to this value
The operation is not guaranteed when a value is set.
•Function varies depending on mode of operation
Bit function varies depending on peripheral function mode.
Refer to respective register for each mode.

*5
Follow the text in each manual for binary and hexadecimal notations.

3. M16C Family Documents

The following documents were prepared for the M16C family.⁽¹⁾

| Document | Contents |
|--------------------------|--|
| Short Sheet | Hardware overview |
| Data Sheet | Hardware overview and electrical characteristics |
| Hardware Manual | Hardware specifications (pin assignments, memory maps, peripheral specifications, electrical characteristics, timing charts). *Refer to the application note for how to use peripheral functions. |
| Software Manual | Detailed description of assembly instructions and microcomputer performance of each instruction |
| Application Note | <ul style="list-style-type: none">• Usage and application examples of peripheral functions• Sample programs• Introduction to the basic functions in the M16C family• Programming method with Assembly and C languages |
| RENESAS TECHNICAL UPDATE | Preliminary report about the specification of a product, a document, etc. |

NOTES:

1. Before using this material, please visit the our website to verify that this is the most updated document available.

Table of Contents

| | |
|---|-------|
| SFR Page Reference | B - 1 |
| 1. Overview | 1 |
| 1.1 Applications | 1 |
| 1.2 Performance Overview | 2 |
| 1.3 Block Diagram | 4 |
| 1.4 Product Information | 5 |
| 1.5 Pin Assignments..... | 7 |
| 1.6 Pin Description | 8 |
| 2. Central Processing Unit (CPU) | 10 |
| 2.1 Data Registers (R0, R1, R2 and R3)..... | 11 |
| 2.2 Address Registers (A0 and A1)..... | 11 |
| 2.3 Frame Base Register (FB) | 11 |
| 2.4 Interrupt Table Register (INTB) | 11 |
| 2.5 Program Counter (PC) | 11 |
| 2.6 User Stack Pointer (USP) and Interrupt Stack Pointer (ISP)..... | 11 |
| 2.7 Static Base Register (SB)..... | 11 |
| 2.8 Flag Register (FLG)..... | 11 |
| 2.8.1 Carry Flag (C)..... | 11 |
| 2.8.2 Debug Flag (D) | 11 |
| 2.8.3 Zero Flag (Z)..... | 11 |
| 2.8.4 Sign Flag (S)..... | 11 |
| 2.8.5 Register Bank Select Flag (B) | 11 |
| 2.8.6 Overflow Flag (O) | 11 |
| 2.8.7 Interrupt Enable Flag (I Flag)..... | 12 |
| 2.8.8 Stack Pointer Select Flag (U Flag) | 12 |
| 2.8.9 Processor Interrupt Priority Level (IPL) | 12 |
| 2.8.10 Reserved Bit..... | 12 |
| 3. Memory | 13 |
| 3.1 R8C/14 Group | 13 |
| 3.2 R8C/15 Group | 14 |

| | | |
|-------|---|----|
| 4. | Special Function Register (SFR) | 15 |
| 5. | Reset | 19 |
| 5.1 | Hardware Reset | 21 |
| 5.1.1 | When the power supply is stable..... | 21 |
| 5.1.2 | Power on | 21 |
| 5.2 | Power-On Reset Function | 23 |
| 5.3 | Voltage Monitor 1 Reset | 24 |
| 5.4 | Voltage Monitor 2 Reset..... | 24 |
| 5.5 | Watchdog Timer Reset..... | 24 |
| 5.6 | Software Reset..... | 24 |
| 6. | Voltage Detection Circuit | 25 |
| 6.1 | Monitoring VCC Input Voltage..... | 31 |
| 6.1.1 | Monitoring Vdet1 | 31 |
| 6.1.2 | Monitoring Vdet2 | 31 |
| 6.2 | Voltage Monitor 1 Reset..... | 32 |
| 6.3 | Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset | 33 |
| 7. | Processor Mode | 35 |
| 7.1 | Types of Processor Mode | 35 |
| 8. | Bus | 37 |
| 9. | Clock Generation Circuit | 38 |
| 9.1 | Main Clock..... | 45 |
| 9.2 | On-Chip Oscillator Clock..... | 46 |
| 9.2.1 | Low-Speed On-Chip Oscillator Clock | 46 |
| 9.2.2 | High-Speed On-Chip Oscillator Clock | 46 |
| 9.3 | CPU Clock and Peripheral Function Clock..... | 47 |
| 9.3.1 | System Clock..... | 47 |
| 9.3.2 | CPU Clock | 47 |
| 9.3.3 | Peripheral Function Clock (f1, f2, f4, f8, f32) | 47 |
| 9.3.4 | fRING and fRING128..... | 47 |
| 9.3.5 | fRING-fast..... | 47 |
| 9.3.6 | fRING-S..... | 47 |
| 9.4 | Power Control..... | 48 |

| | | |
|--------|--|-----|
| 9.4.1 | Normal Operating Mode | 48 |
| 9.4.2 | Wait Mode | 49 |
| 9.4.3 | Stop Mode | 51 |
| 9.5 | Oscillation Stop Detection Function | 53 |
| 9.5.1 | How to Use Oscillation Stop Detection Function | 53 |
| 10. | Protection | 55 |
| 11. | Interrupt | 56 |
| 11.1 | Interrupt Overview | 56 |
| 11.1.1 | Types of Interrupts..... | 56 |
| 11.1.2 | Software Interrupts | 57 |
| 11.1.3 | Special Interrupts..... | 58 |
| 11.1.4 | Peripheral Function Interrupt..... | 58 |
| 11.1.5 | Interrupts and Interrupt Vector..... | 59 |
| 11.1.6 | Interrupt Control..... | 61 |
| 11.2 | $\overline{\text{INT}}$ Interrupt | 69 |
| 11.2.1 | $\overline{\text{INT0}}$ Interrupt | 69 |
| 11.2.2 | $\overline{\text{INT0}}$ Input Filter..... | 70 |
| 11.2.3 | $\overline{\text{INT1}}$ Interrupt | 71 |
| 11.2.4 | $\overline{\text{INT3}}$ Interrupt | 72 |
| 11.3 | Key Input Interrupt..... | 74 |
| 11.4 | Address Match Interrupt..... | 76 |
| 12. | Watchdog Timer | 78 |
| 12.1 | When Count Source Protection Mode Disabled..... | 81 |
| 12.2 | When Count Source Protection Mode Enabled..... | 82 |
| 13. | Timers | 83 |
| 13.1 | Timer X..... | 84 |
| 13.1.1 | Timer Mode | 87 |
| 13.1.2 | Pulse Output Mode..... | 88 |
| 13.1.3 | Event Counter Mode..... | 90 |
| 13.1.4 | Pulse Width Measurement Mode | 92 |
| 13.1.5 | Pulse Period Measurement Mode | 95 |
| 13.2 | Timer Z..... | 98 |
| 13.2.1 | Timer Mode | 103 |

| | | |
|------------|---|------------|
| 13.2.2 | Programmable Waveform Generation Mode..... | 105 |
| 13.2.3 | Programmable One-shot Generation Mode | 108 |
| 13.2.4 | Programmable Wait One-shot Generation Mode | 111 |
| 13.3 | Timer C..... | 115 |
| 13.3.1 | Input Capture Mode..... | 121 |
| 13.3.2 | Output Compare Mode..... | 123 |
| 14. | Serial Interface | 125 |
| 14.1 | Clock Synchronous Serial I/O Mode | 130 |
| 14.1.1 | Polarity Select Function..... | 133 |
| 14.1.2 | LSB First/MSB First Select Function | 133 |
| 14.1.3 | Continuous Receive Mode | 134 |
| 14.2 | Clock Asynchronous Serial I/O (UART) Mode | 135 |
| 14.2.1 | CNTR0 Pin Select Function..... | 138 |
| 14.2.2 | Bit Rate..... | 139 |
| 15. | Clock Synchronous Serial I/O with Chip Select (SSU) | 140 |
| 15.1 | Transfer Clock..... | 149 |
| 15.1.1 | Association between Transfer Clock Polarity, Phase and Data | 149 |
| 15.2 | SS Shift Register (SSTRSR)..... | 151 |
| 15.2.1 | Association between Data I/O Pin and SS Shift Register..... | 151 |
| 15.3 | Interrupt Requests..... | 152 |
| 15.4 | Communication Modes and Pin Functions..... | 153 |
| 15.5 | Clock Synchronous Communication Mode | 154 |
| 15.5.1 | Initialization in Clock Synchronous Communication Mode | 154 |
| 15.5.2 | Data Transmit..... | 155 |
| 15.5.3 | Data Receive | 157 |
| 15.5.4 | Data Transmit/Receive | 159 |
| 15.6 | Operation in 4-Wire Bus Communication Mode | 161 |
| 15.6.1 | Initialization in 4-Wire Bus Communication Mode | 161 |
| 15.6.2 | Data Transmit | 163 |
| 15.6.3 | Data Receive | 165 |
| 15.6.4 | $\overline{\text{SCS}}$ Pin Control and Arbitration | 167 |
| 16. | A/D Converter | 168 |
| 16.1 | One-Shot Mode..... | 172 |

| | | |
|------------|--|------------|
| 16.2 | Repeat Mode..... | 174 |
| 16.3 | Sample and Hold..... | 176 |
| 16.4 | A/D Conversion Cycles | 176 |
| 16.5 | Internal Equivalent Circuit of Analog Input | 177 |
| 16.6 | Inflow Current Bypass Circuit | 178 |
| 17. | Programmable I/O Ports | 179 |
| 17.1 | Functions of Programmable I/O Ports..... | 179 |
| 17.2 | Effect on Peripheral Functions | 179 |
| 17.3 | Pins Other than Programmable I/O Ports..... | 179 |
| 17.4 | Port setting | 186 |
| 17.5 | Unassigned Pin Handling | 190 |
| 18. | Flash Memory Version | 191 |
| 18.1 | Overview | 191 |
| 18.2 | Memory Map | 193 |
| 18.3 | Functions To Prevent Flash Memory from Rewriting | 195 |
| 18.3.1 | ID Code Check Function | 195 |
| 18.3.2 | ROM Code Protect Function | 196 |
| 18.4 | CPU Rewrite Mode..... | 197 |
| 18.4.1 | EW0 Mode..... | 198 |
| 18.4.2 | EW1 Mode..... | 198 |
| 18.4.3 | Software Commands | 205 |
| 18.4.4 | Status Register | 209 |
| 18.4.5 | Full Status Check | 210 |
| 18.5 | Standard Serial I/O Mode..... | 212 |
| 18.5.1 | ID Code Check Function | 212 |
| 18.6 | Parallel I/O Mode..... | 216 |
| 18.6.1 | ROM Code Protect Function | 216 |
| 19. | Electrical Characteristics | 217 |
| 20. | Precautions | 235 |
| 20.1 | Stop Mode and Wait Mode..... | 235 |
| 20.1.1 | Stop Mode | 235 |
| 20.1.2 | Wait Mode | 235 |
| 20.2 | Interrupts | 236 |

| | | |
|----------------------|---|-----|
| 20.2.1 | Reading Address 00000h | 236 |
| 20.2.2 | SP Setting..... | 236 |
| 20.2.3 | External Interrupt and Key Input Interrupt | 236 |
| 20.2.4 | Watchdog Timer Interrupt..... | 236 |
| 20.2.5 | Changing Interrupt Factor..... | 237 |
| 20.2.6 | Changing Interrupt Control Register | 238 |
| 20.3 | Clock Generation Circuit | 239 |
| 20.3.1 | Oscillation Stop Detection Function..... | 239 |
| 20.3.2 | Oscillation Circuit Constants..... | 239 |
| 20.4 | Timers | 240 |
| 20.4.1 | Timers X and Z..... | 240 |
| 20.4.2 | Timer X..... | 240 |
| 20.4.3 | Timer Z | 241 |
| 20.4.4 | Timer C..... | 241 |
| 20.5 | Serial Interface | 242 |
| 20.6 | Clock Synchronous Serial I/O (SSU) with Chip Select..... | 243 |
| 20.6.1 | Access Registers Associated with SSU | 243 |
| 20.7 | A/D Converter..... | 244 |
| 20.8 | Flash Memory Version | 245 |
| 20.8.1 | CPU Rewrite Mode..... | 245 |
| 20.9 | Noise | 248 |
| 20.9.1 | Insert a bypass capacitor between VCC and VSS pins as the countermeasures against noise and latch-up..... | 248 |
| 20.9.2 | Countermeasures against Noise Error of Port Control Registers..... | 248 |
| 21. | Precaution for On-chip Debugger | 249 |
| Appendix 1. | Package Dimensions | 250 |
| Appendix 2. | Connecting Example between Serial Writer and On-Chip Debugging Emulator | 251 |
| Appendix 3. | Example of Oscillation Evaluation Circuit | 252 |
| Register Index | | 253 |

SFR Page Reference

| Address | Register | Symbol | Page |
|---------|--|--------|------|
| 0000h | | | |
| 0001h | | | |
| 0002h | | | |
| 0003h | | | |
| 0004h | Processor Mode Register 0 | PM0 | 35 |
| 0005h | Processor Mode Register 1 | PM1 | 36 |
| 0006h | System Clock Control Register 0 | CM0 | 40 |
| 0007h | System Clock Control Register 1 | CM1 | 41 |
| 0008h | | | |
| 0009h | Address Match Interrupt Enable Register | AIER | 77 |
| 000Ah | Protect Register | PRCR | 55 |
| 000Bh | | | |
| 000Ch | Oscillation Stop Detection Register | OCD | 42 |
| 000Dh | Watchdog Timer Reset Register | WDTR | 80 |
| 000Eh | Watchdog Timer Start Register | WDTS | 80 |
| 000Fh | Watchdog Timer Control Register | WDC | 79 |
| 0010h | Address Match Interrupt Register 0 | RMAD0 | 77 |
| 0011h | | | |
| 0012h | | | |
| 0013h | | | |
| 0014h | Address Match Interrupt Register 1 | RMAD1 | 77 |
| 0015h | | | |
| 0016h | | | |
| 0017h | | | |
| 0018h | | | |
| 0019h | | | |
| 001Ah | | | |
| 001Bh | | | |
| 001Ch | Count Source Protection Mode Register | CSPR | 80 |
| 001Dh | | | |
| 001Eh | INT0 Input Filter Select Register | INT0F | 69 |
| 001Fh | | | |
| 0020h | High-Speed On-Chip Oscillator Control Register 0 | HRA0 | 43 |
| 0021h | High-Speed On-Chip Oscillator Control Register 1 | HRA1 | 44 |
| 0022h | High-Speed On-Chip Oscillator Control Register 2 | HRA2 | 44 |
| 0023h | | | |
| 0024h | | | |
| 0025h | | | |
| 0026h | | | |
| 0027h | | | |
| 0028h | | | |
| 0029h | | | |
| 002Ah | | | |
| 002Bh | | | |
| 002Ch | | | |
| 002Dh | | | |
| 002Eh | | | |
| 002Fh | | | |
| 0030h | | | |
| 0031h | Voltage Detection Register 1 | VCA1 | 28 |
| 0032h | Voltage Detection Register 2 | VCA2 | 28 |
| 0033h | | | |
| 0034h | | | |
| 0035h | | | |
| 0036h | Voltage Monitor 1 Circuit Control Register | VW1C | 29 |
| 0037h | Voltage Monitor 2 Circuit Control Register | VW2C | 30 |
| 0038h | | | |
| 0039h | | | |
| 003Ah | | | |
| 003Bh | | | |
| 003Ch | | | |
| 003Dh | | | |
| 003Eh | | | |
| 003Fh | | | |

| Address | Register | Symbol | Page |
|---------|---|--------|------|
| 0040h | | | |
| 0041h | | | |
| 0042h | | | |
| 0043h | | | |
| 0044h | | | |
| 0045h | | | |
| 0046h | | | |
| 0047h | | | |
| 0048h | | | |
| 0049h | | | |
| 004Ah | | | |
| 004Bh | | | |
| 004Ch | | | |
| 004Dh | Key Input Interrupt Control Register | KUPIC | 61 |
| 004Eh | A/D Conversion Interrupt Control Register | ADIC | 61 |
| 004Fh | SSU Interrupt Control Register | SSUAIC | 61 |
| 0050h | Compare 1 Interrupt Control Register | CMP1IC | 61 |
| 0051h | UART0 Transmit Interrupt Control Register | S0TIC | 61 |
| 0052h | UART0 Receive Interrupt Control Register | S0RIC | 61 |
| 0053h | | | |
| 0054h | | | |
| 0055h | | | |
| 0056h | Timer X Interrupt Control Register | TXIC | 61 |
| 0057h | | | |
| 0058h | Timer Z Interrupt Control Register | TZIC | 61 |
| 0059h | INT1 Interrupt Control Register | INT1IC | 61 |
| 005Ah | INT3 Interrupt Control Register | INT3IC | 61 |
| 005Bh | Timer C Interrupt Control Register | TCIC | 61 |
| 005Ch | Compare 0 Interrupt Control Register | CMP0IC | 61 |
| 005Dh | INT0 Interrupt Control Register | INT0IC | 62 |
| 005Eh | | | |
| 005Fh | | | |
| 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 | | | |

NOTES:

- Blank columns are all reserved space. No access is allowed.

| Address | Register | Symbol | Page |
|---------|---|--------|--------|
| 0080h | Timer Z Mode Register | TZMR | 99 |
| 0081h | | | |
| 0082h | | | |
| 0083h | | | |
| 0084h | Timer Z Waveform Output Control Register | PUM | 101 |
| 0085h | Prescaler Z | PREZ | 100 |
| 0086h | Timer Z Secondary | TZSC | 100 |
| 0087h | Timer Z Primary | TZPR | 100 |
| 0088h | | | |
| 0089h | | | |
| 008Ah | Timer Z Output Control Register | TZOC | 101 |
| 008Bh | Timer X Mode Register | TXMR | 85 |
| 008Ch | Prescaler X | PREX | 86 |
| 008Dh | Timer X | TX | 86 |
| 008Eh | Timer Count Source Set Register | TCSS | 86,102 |
| 008Fh | | | |
| 0090h | Timer C | TC | 117 |
| 0091h | | | |
| 0092h | | | |
| 0093h | | | |
| 0094h | | | |
| 0095h | | | |
| 0096h | External Input Enable Register | INTEN | 69 |
| 0097h | | | |
| 0098h | Key Input Enable Register | KIEN | 75 |
| 0099h | | | |
| 009Ah | Timer C Control Register 0 | TCC0 | 118 |
| 009Bh | Timer C Control Register 1 | TCC1 | 119 |
| 009Ch | Capture, Compare 0 Register | TM0 | 117 |
| 009Dh | | | |
| 009Eh | Compare 1 Register | TM1 | 117 |
| 009Fh | | | |
| 00A0h | UART0 Transmit/Receive Mode Register | U0MR | 128 |
| 00A1h | UART0 Bit Rate Register | U0BRG | 127 |
| 00A2h | UART0 Transmit Buffer Register | U0TB | 127 |
| 00A3h | | | |
| 00A4h | UART0 Transmit/Receive Control Register 0 | U0C0 | 128 |
| 00A5h | UART0 Transmit/Receive Control Register 1 | U0C1 | 129 |
| 00A6h | UART0 Receive Buffer Register | U0RB | 127 |
| 00A7h | | | |
| 00A8h | | | |
| 00A9h | | | |
| 00AAh | | | |
| 00ABh | | | |
| 00ACh | | | |
| 00ADh | | | |
| 00AEh | | | |
| 00AFh | | | |
| 00B0h | UART Transmit/Receive Control Register 2 | U0CON | 129 |
| 00B1h | | | |
| 00B2h | | | |
| 00B3h | | | |
| 00B4h | | | |
| 00B5h | | | |
| 00B6h | | | |
| 00B7h | | | |
| 00B8h | SS Control Register H | SSCRH | 142 |
| 00B9h | SS Control Register L | SSCRL | 143 |
| 00BAh | SS Mode Register | SSMR | 144 |
| 00BBh | SS Enable Register | SSER | 145 |
| 00BCh | SS Status Register | SSSR | 146 |
| 00BDh | SS Mode Register 2 | SSMR2 | 147 |
| 00BEh | SS Transmit Data Register | SSTDR | 148 |
| 00BFh | SS Receive Data Register | SSRDR | 148 |

NOTES:

- Blank columns, 0100h to 01AFh and 01C0h to 02FFh are all reserved. No access is allowed.

| Address | Register | Symbol | Page |
|---------|---|--------|--------|
| 00C0h | A/D Register | AD | 171 |
| 00C1h | | | |
| 00C2h | | | |
| 00C3h | | | |
| 00C4h | | | |
| 00C5h | | | |
| 00C6h | | | |
| 00C7h | | | |
| 00C8h | | | |
| 00C9h | | | |
| 00CAh | | | |
| 00CBh | | | |
| 00CCh | | | |
| 00CDh | | | |
| 00CEh | | | |
| 00CFh | | | |
| 00D0h | | | |
| 00D1h | | | |
| 00D2h | | | |
| 00D3h | | | |
| 00D4h | A/D Control Register 2 | ADCON2 | 171 |
| 00D5h | | | |
| 00D6h | A/D Control Register 0 | ADCON0 | 170 |
| 00D7h | A/D Control Register 1 | ADCON1 | 170 |
| 00D8h | | | |
| 00D9h | | | |
| 00DAh | | | |
| 00DBh | | | |
| 00DCh | | | |
| 00DDh | | | |
| 00DEh | | | |
| 00DFh | | | |
| 00E0h | | | |
| 00E1h | Port P1 Register | P1 | 184 |
| 00E2h | | | |
| 00E3h | Port P1 Direction Register | PD1 | 184 |
| 00E4h | | | |
| 00E5h | Port P3 Register | P3 | 184 |
| 00E6h | | | |
| 00E7h | Port P3 Direction Register | PD3 | 184 |
| 00E8h | Port P4 Register | P4 | 184 |
| 00E9h | | | |
| 00EAh | Port P4 Direction Register | PD4 | 184 |
| 00EBh | | | |
| 00ECh | | | |
| 00EDh | | | |
| 00EEh | | | |
| 00EFh | | | |
| 00F0h | | | |
| 00F1h | | | |
| 00F2h | | | |
| 00F3h | | | |
| 00F4h | | | |
| 00F5h | | | |
| 00F6h | | | |
| 00F7h | | | |
| 00F8h | | | |
| 00F9h | | | |
| 00FAh | | | |
| 00FBh | | | |
| 00FCh | Pull-Up Control Register 0 | PUR0 | 185 |
| 00FDh | Pull-Up Control Register 1 | PUR1 | 185 |
| 00FEh | Port P1 Drive Capacity Control Register | DRR | 185 |
| 00FFh | Timer C Output Control Register | TCOUT | 120 |
| | | | |
| 01B3h | Flash Memory Control Register 4 | FMR4 | 201 |
| 01B4h | | | |
| 01B5h | Flash Memory Control Register 1 | FMR1 | 201 |
| 01B6h | | | |
| 01B7h | Flash Memory Control Register 0 | FMR0 | 200 |
| | | | |
| 0FFFh | Optional Function Select Register | OFS | 79,196 |

1. Overview

This MCU is built using the high-performance silicon gate CMOS process using a R8C/Tiny Series CPU core and is packaged in a 20-pin plastic molded LSSOP. This MCU operates using sophisticated instructions featuring a high level of instruction efficiency. With 1 Mbyte of address space, it is capable of executing instructions at high speed.

Furthermore, the data flash ROM (1KB × 2blocks) is embedded in the R8C/15 group.

The difference between R8C/14 and R8C/15 groups is only the existence of the data flash ROM. Their peripheral functions are the same.

1.1 Applications

Electric household appliance, office equipment, housing equipment (sensor, security), general industrial equipment, audio, etc.

1.2 Performance Overview

Table 1.1 lists the Performance Outline of the R8C/14 Group and Table 1.2 lists the Performance Outline of the R8C/15 Group.

Table 1.1 Performance Outline of the R8C/14 Group

| | Item | Performance |
|--|--|---|
| CPU | Number of Basic Instructions | 89 instructions |
| | Minimum Instruction Execution Time | 50ns(f(XIN)=20MHz, VCC=3.0 to 5.5V) 100ns(f(XIN)=10MHz, VCC=2.7 to 5.5V) |
| | Operating Mode | Single-chip |
| | Memory Space | 1 Mbyte |
| | Memory Capacity | See Table 1.3 R8C/14 Group Product Information |
| | Peripheral Function | Port |
| LED Drive Port | | I/O port: 4 pins |
| Timer | | Timer X: 8 bits × 1 channel, Timer Z: 8 bits × 1 channel (Each timer equipped with 8-bit prescaler) Timer C: 16 bits × 1 channel (Circuits of input capture and output compare) |
| Serial Interface | | 1 channel Clock synchronous serial I/O, UART |
| Chip-Select Clock Synchronous Serial I/O (SSU) | | 1 channel |
| A/D Converter | | 10-bit A/D converter: 1 circuit, 4 channels |
| Watchdog Timer | | 15 bits × 1 channel (with prescaler) Reset start selectable, Count source protection mode |
| Interrupt | | Internal: 9 factors, External: 4 factors, Software: 4 factors, Priority level: 7 levels |
| Clock Generation Circuit | | 2 circuits <ul style="list-style-type: none"> • Main clock oscillation circuit (Equipped with a built-in feedback resistor) • On-chip oscillator (high speed, low speed) Equipped with frequency adjustment function on high-speed on-chip oscillator |
| Oscillation Stop Detection Function | | Main clock oscillation stop detection function |
| Voltage Detection Circuit | | Included |
| Power-On Reset Circuit | | Included |
| Electric Characteristics | | Supply Voltage |
| | Power Consumption | Typ. 9mA (VCC=5.0V, f(XIN)=20MHz) Typ. 5mA (VCC=3.0V, f(XIN)=10MHz) Typ. 35μA (VCC=3.0V, wait mode, peripheral clock off) Typ. 0.7μA (VCC=3.0V, stop mode) |
| Flash Memory | Program/Erase Supply Voltage | VCC=2.7 to 5.5V |
| | Program/Erase Endurance | 100 times |
| Operating Ambient Temperature | -20 to 85°C -40 to 85°C (D Version) | |
| Package | 20-pin plastic mold LSSOP | |

Table 1.2 Performance Outline of the R8C/15 Group

| Item | | Performance |
|-------------------------------|--|--|
| CPU | Number of Basic Instructions | 89 instructions |
| | Minimum Instruction Execution Time | 50ns (f(XIN)=20MHz, VCC=3.0 to 5.5V) 100ns (f(XIN)=10MHz, VCC=2.7 to 5.5V) |
| | Operating Mode | Single-chip |
| | Memory Space | 1 Mbyte |
| | Memory Capacity | See Table 1.4 R8C/15 Group Product Information |
| Peripheral Function | Port | I/O : 13 pins (including LED drive port), Input : 2 pins |
| | LED drive port | I/O port: 4 pins |
| | Timer | Timer X: 8 bits × 1 channel, Timer Z: 8 bits × 1 channel (Each timer equipped with 8-bit prescaler) Timer C: 16 bits × 1 channel (Circuits of input capture and output compare) |
| | Serial Interface | 1 channel Clock synchronous serial I/O, UART |
| | Chip-select clock synchronous serial I/O (SSU) | 1 channel |
| | A/D Converter | 10-bit A/D converter: 1 circuit, 4 channels |
| | Watchdog Timer | 15 bits × 1 channel (with prescaler) Reset start selectable, Count source protection mode |
| | Interrupt | Internal: 9 factors, External: 4 factors, Software: 4 factors Priority level: 7 levels |
| | Clock Generation Circuit | 2 circuits • Main clock generation circuit (Equipped with a built-in feedback resistor) • On-chip oscillator (high speed, low speed) Equipped with frequency adjustment function on high-speed on-chip oscillator |
| | Oscillation Stop Detection Function | Main clock oscillation stop detection function |
| | Voltage Detection Circuit | Included |
| | Power on Reset Circuit | Included |
| Electric Characteristics | Supply Voltage | VCC=3.0 to 5.5V (f(XIN)=20MHz) VCC=2.7 to 5.5V (f(XIN)=10MHz) |
| | Power Consumption | Typ. 9mA (VCC=5.0V, f(XIN)=20MHz) Typ. 5mA (VCC=3.0V, f(XIN)=10MHz) Typ. 35μA (VCC=3.0V, wait mode, peripheral clock off) Typ. 0.7μA (VCC=3.0V, stop mode) |
| Flash Memory | Program/Erase Supply Voltage | VCC=2.7 to 5.5V |
| | Program/Erase Endurance | 10,000 times (Data flash) 1,000 times (Program ROM) |
| Operating Ambient Temperature | -20 to 85°C -40 to 85°C (D Version) | |
| Package | 20-pin plastic mold LSSOP | |

1.3 Block Diagram

Figure 1.1 shows a Block Diagram.

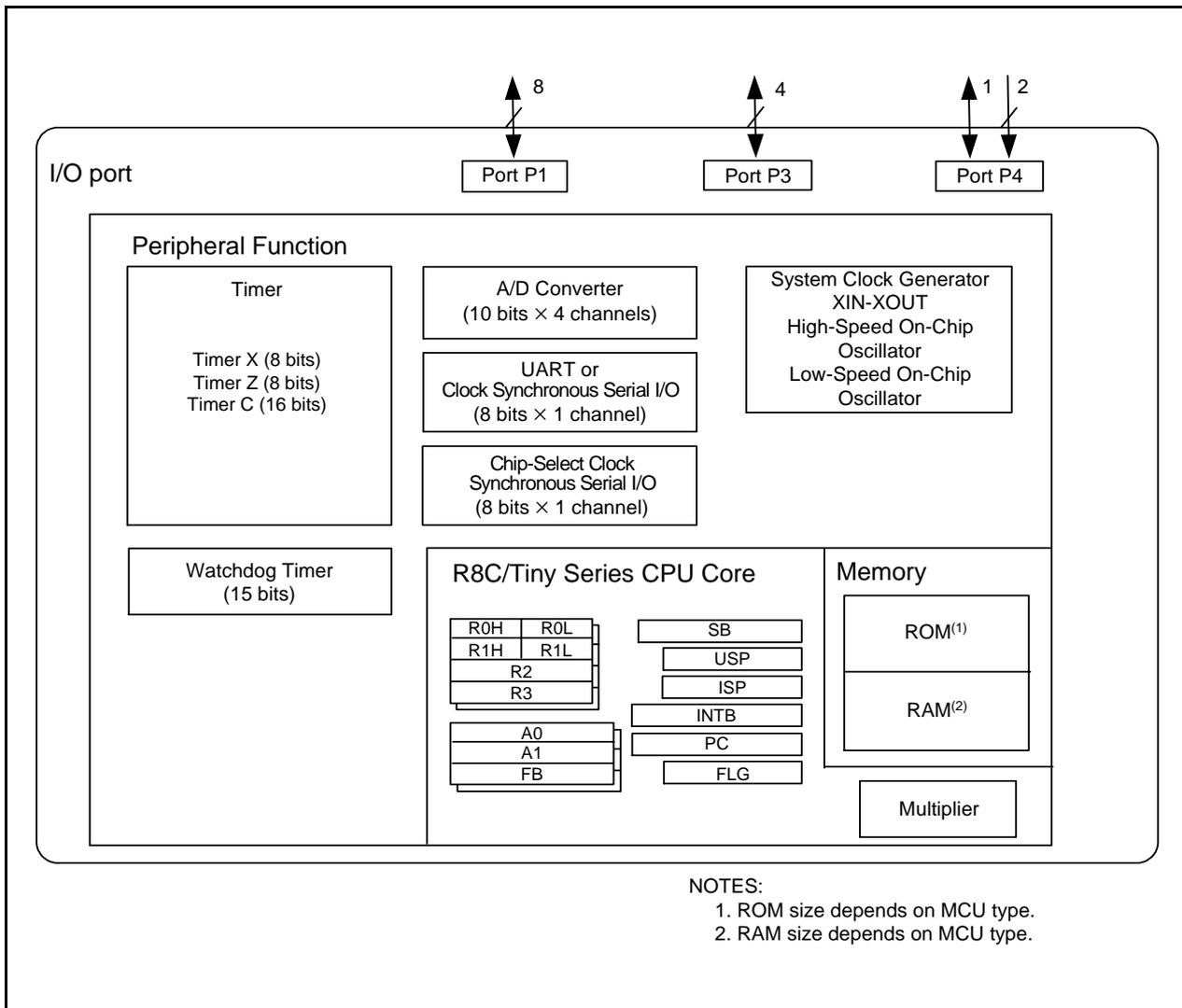


Figure 1.1 Block Diagram

1.4 Product Information

Table 1.3 lists the Product Information of R8C/14 Group and Table 1.4 lists the Product Information of R8C/15 Group.

Table 1.3 Product Information of R8C/14 Group As of Jan 2006

| Type No. | ROM capacity | RAM capacity | Package type | Remarks |
|-------------|--------------|--------------|--------------|----------------------|
| R5F21142SP | 8 Kbytes | 512 bytes | PLSP0020JB-A | Flash memory version |
| R5F21143SP | 12 Kbytes | 768 bytes | PLSP0020JB-A | |
| R5F21144SP | 16 Kbytes | 1 Kbyte | PLSP0020JB-A | |
| R5F21142DSP | 8 Kbytes | 512 bytes | PLSP0020JB-A | D version |
| R5F21143DSP | 12 Kbytes | 768 bytes | PLSP0020JB-A | |
| R5F21144DSP | 16 Kbytes | 1 Kbyte | PLSP0020JB-A | |

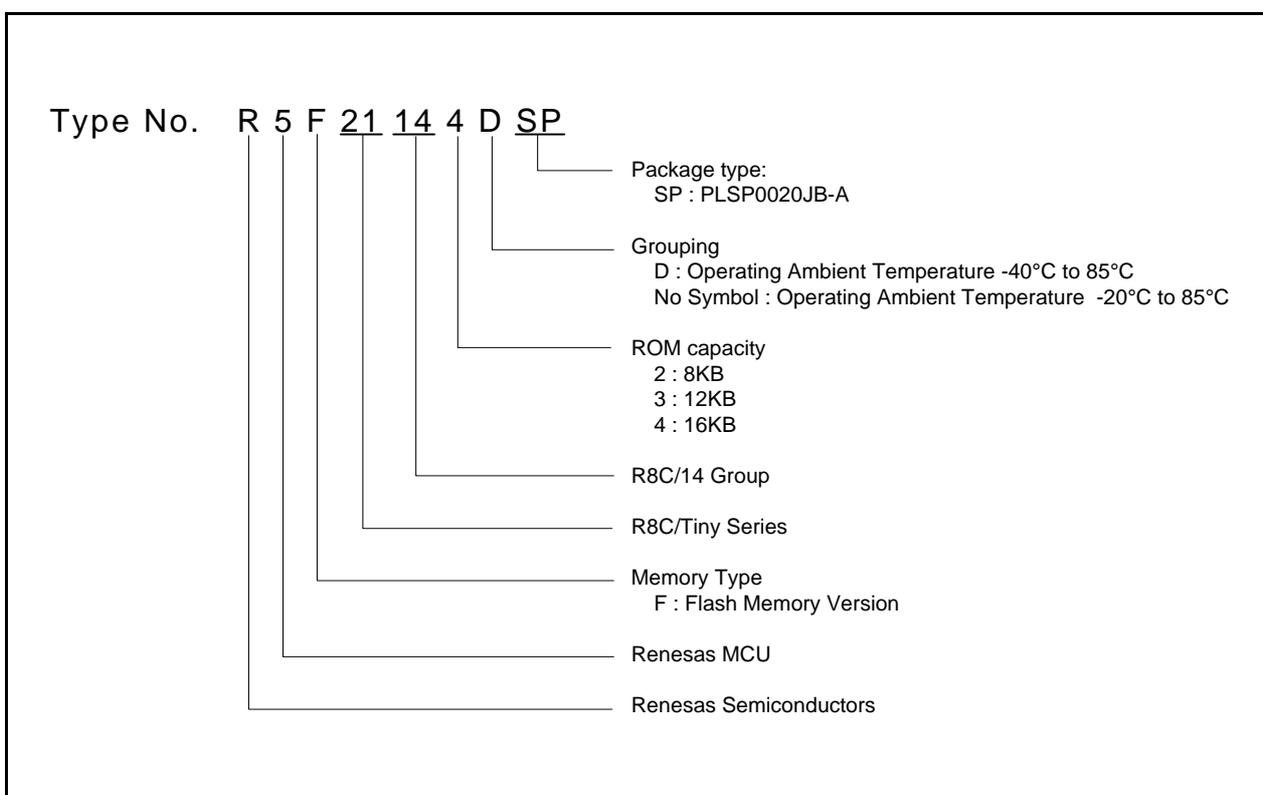


Figure 1.2 Part Number, Memory Size and Package of R8C/14 Group

Table 1.4 Product Information of R8C/15 Group As of Jan 2006

| Type No. | ROM capacity | | RAM capacity | Package type | Remarks |
|-------------|--------------|-------------|--------------|--------------|---------------------------------------|
| | Program ROM | Data flash | | | |
| R5F21152SP | 8 Kbytes | 1 Kbyte x 2 | 512 bytes | PLSP0020JB-A | Flash memory version D version |
| R5F21153SP | 12 Kbytes | 1 Kbyte x 2 | 768 bytes | PLSP0020JB-A | |
| R5F21154SP | 16 Kbytes | 1 Kbyte x 2 | 1 Kbyte | PLSP0020JB-A | |
| R5F21152DSP | 8 Kbytes | 1 Kbyte x 2 | 512 bytes | PLSP0020JB-A | |
| R5F21153DSP | 12 Kbytes | 1 Kbyte x 2 | 768 bytes | PLSP0020JB-A | |
| R5F21154DSP | 16 Kbytes | 1 Kbyte x 2 | 1 Kbyte | PLSP0020JB-A | |

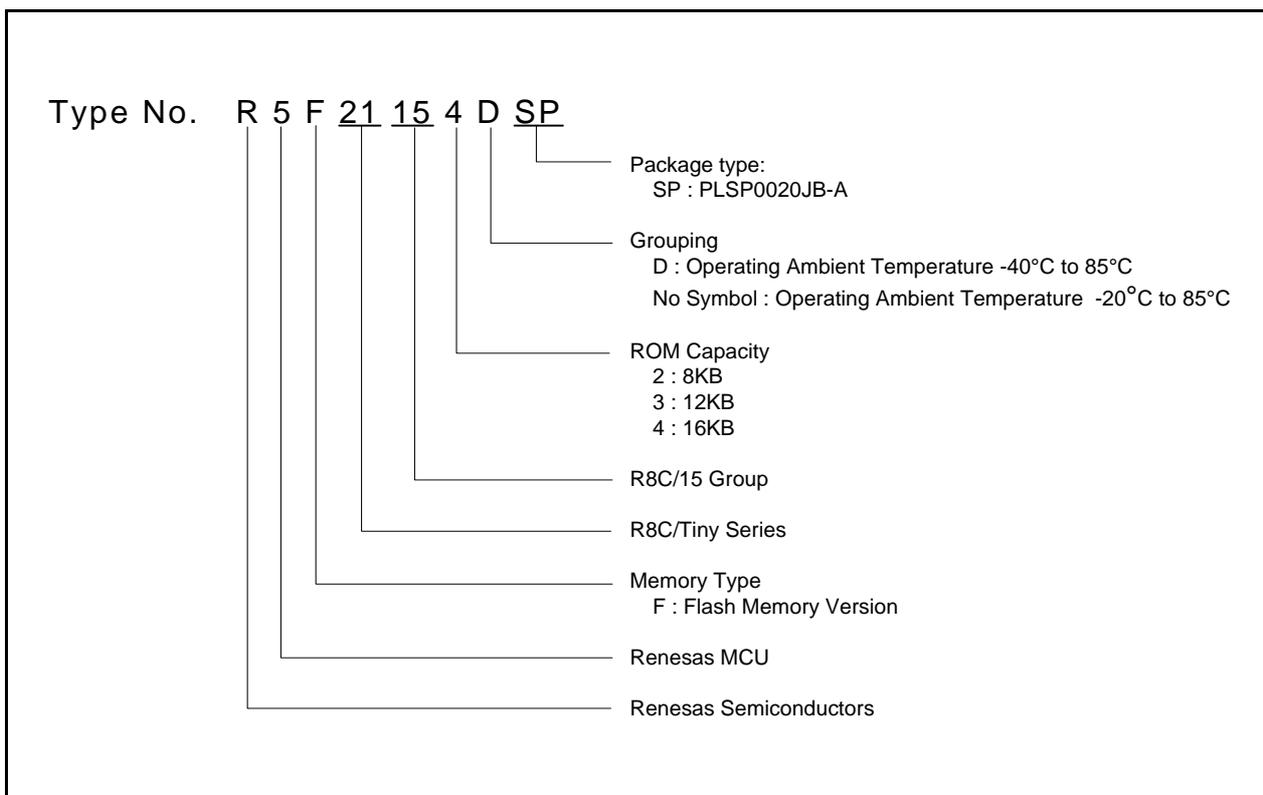


Figure 1.3 Part Number, Memory Size and Package of R8C/15 Group

1.5 Pin Assignments

Figure 1.4 shows the PLSP0020JB-A Package Pin Assignment (top view).

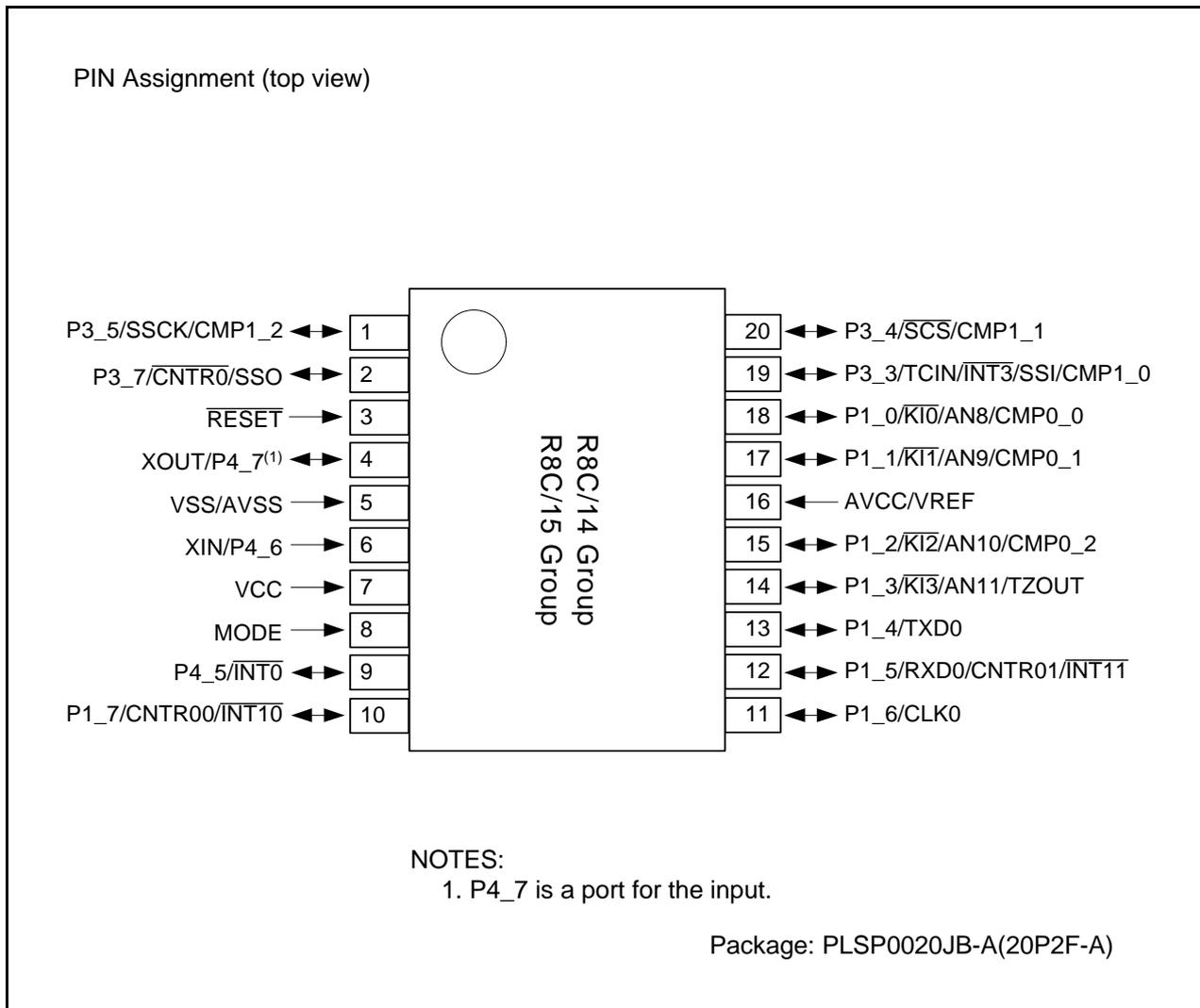


Figure 1.4 PLSP0020JB-A Package Pin Assignment (top view)

1.6 Pin Description

Table 1.5 lists the Pin Description and Table 1.6 lists the Pin Name Information by Pin Number.

Table 1.5 Pin Description

| Function | Pin name | I/O type | Description |
|-----------------------------------|--|----------|---|
| Power Supply Input | VCC VSS | I | Apply 2.7V to 5.5V to the VCC pin. Apply 0V to the VSS pin |
| Analog Power Supply Input | AVCC AVSS | I | Power supply input pins to A/D converter. Connect AVCC to VCC. Apply 0V to AVSS. 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 |
| Main Clock Input | XIN | I | These pins are provided for the main clock generation circuit I/O. Connect a ceramic resonator or a crystal oscillator between the XIN and XOUT pins. To use an externally derived clock, input it to the XIN pin and leave the XOUT pin open. |
| Main Clock Output | XOUT | O | |
| $\overline{\text{INT}}$ Interrupt | $\overline{\text{INT0}}$, $\overline{\text{INT1}}$, $\overline{\text{INT3}}$ | I | INT interrupt input pins |
| Key Input Interrupt | KI0 to KI3 | I | Key input interrupt input pins |
| Timer X | CNTR0 | I/O | Timer X I/O pin |
| | $\overline{\text{CNTR0}}$ | O | Timer X output pin. |
| Timer Z | TZOUT | O | Timer Z output pin |
| Timer C | TCIN | I | Timer C input pin |
| | CMP0_0 to CMP0_2, CMP1_0 to CMP1_2 | O | Timer C output pins. |
| Serial Interface | CLK0 | I/O | Transfer clock I/O pin. |
| | RXD0 | I | Serial data input pin. |
| | TXD0 | O | Serial data output 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 Connect VREF to VCC |
| A/D Converter | AN8 to AN11 | I | Analog input pins to A/D converter |
| I/O Port | P1_0 to P1_7, P3_3 to P3_5, P3_7, P4_5 | I/O | These are CMOS I/O ports. Each port contains an I/O select direction register, allowing each pin in that port to be directed for input or output individually. Any port set to input can select whether to use a pull-up resistor or not by program. P1_0 to P1_3 also function as LED drive ports. |
| Input Port | P4_6, P4_7 | I | Port for input-only |

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

Table 1.6 Pin Name Information by Pin Number

| Pin Number | Control Pin | Port | I/O Pin of Peripheral Function | | | | |
|------------|---------------------------|------|--------------------------------|---------------------------|------------------|---|---------------|
| | | | Interrupt | Timer | Serial Interface | Clock Synchronous Serial I/O with Chip Select | A/D Converter |
| 1 | | P3_5 | | CMP1_2 | | SSCK | |
| 2 | | P3_7 | | $\overline{\text{CNTR0}}$ | | SSO | |
| 3 | $\overline{\text{RESET}}$ | | | | | | |
| 4 | XOUT | P4_7 | | | | | |
| 5 | VSS/AVSS | | | | | | |
| 6 | XIN | P4_6 | | | | | |
| 7 | VCC | | | | | | |
| 8 | MODE | | | | | | |
| 9 | | P4_5 | $\overline{\text{INT0}}$ | | | | |
| 10 | | P1_7 | $\overline{\text{INT10}}$ | CNTR00 | | | |
| 11 | | P1_6 | | | CLK0 | | |
| 12 | | P1_5 | $\overline{\text{INT11}}$ | CNTR01 | RXD0 | | |
| 13 | | P1_4 | | | TXD0 | | |
| 14 | | P1_3 | $\overline{\text{KI3}}$ | TZOUT | | | AN11 |
| 15 | | P1_2 | $\overline{\text{KI2}}$ | CMP0_2 | | | AN10 |
| 16 | AVCC/VREF | | | | | | |
| 17 | | P1_1 | $\overline{\text{KI1}}$ | CMP0_1 | | | AN9 |
| 18 | | P1_0 | $\overline{\text{KI0}}$ | CMP0_0 | | | AN8 |
| 19 | | P3_3 | $\overline{\text{INT3}}$ | TCIN/CMP1_0 | | SSI | |
| 20 | | P3_4 | | CMP1_1 | | $\overline{\text{SCS}}$ | |

2. Central Processing Unit (CPU)

Figure 2.1 shows the CPU Register. The CPU contains 13 registers. Of these, R0, R1, R2, R3, A0, A1 and FB comprise a register bank. Two sets of register banks are provided.

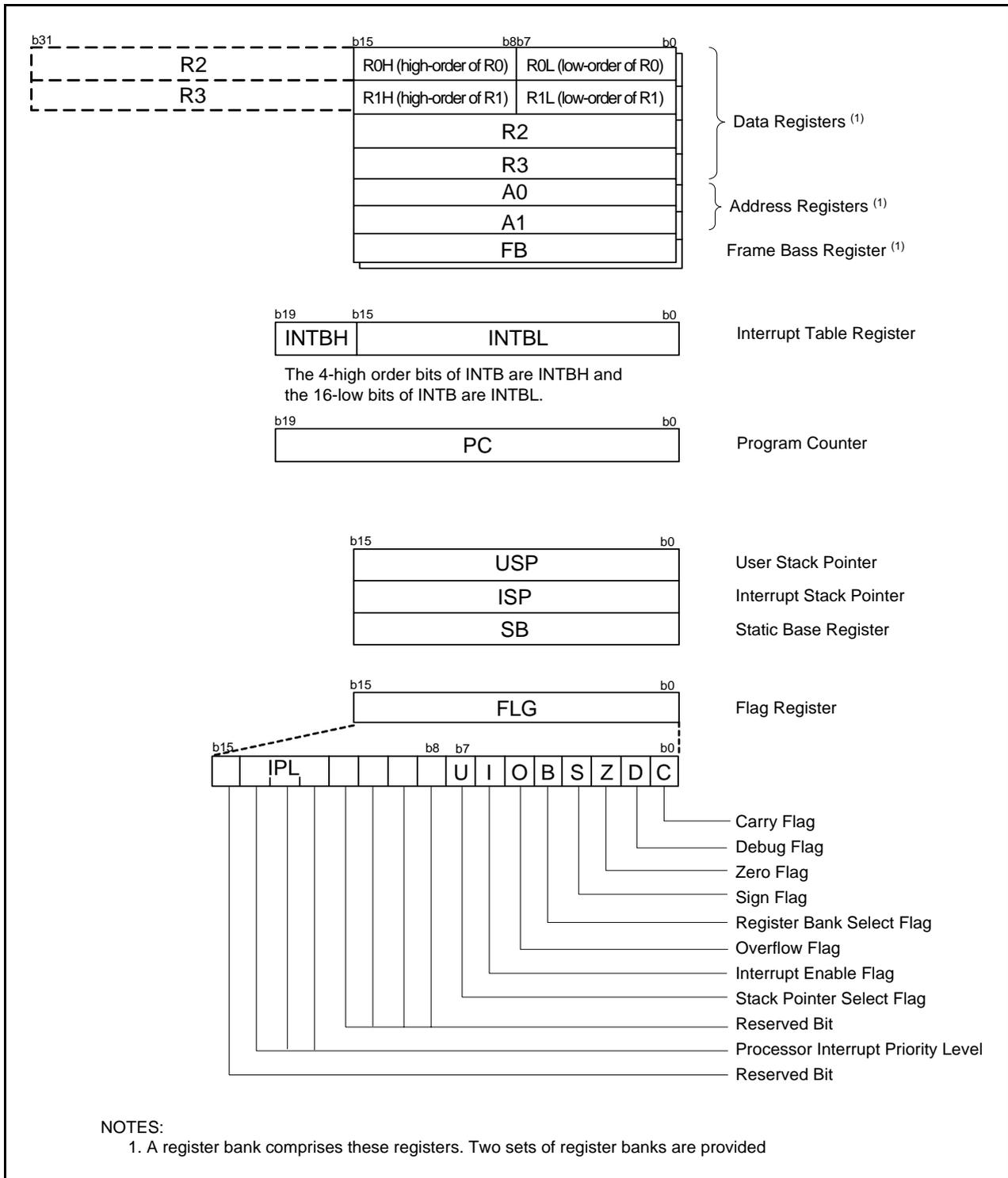


Figure 2.1 CPU Register

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. The R0 can be split into high-order bit (R0H) and low-order bit (R0L) to be used separately as 8-bit data registers. The same applies to R1H and R1L as R0H and R0L. R2 can be combined with R0 to be used as a 32-bit data register (R2R0). The same applies to R3R1 as R2R0.

2.2 Address Registers (A0 and A1)

A0 is a 16-bit register for address register indirect addressing and address register relative addressing. They also are used for transfer, arithmetic and logic operations. The same applies to A1 as A0. A0 can be combined with A0 to be used 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 indicates the start address of an interrupt vector table.

2.5 Program Counter (PC)

PC, 20 bits wide, indicates the address of an instruction to be executed.

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

The stack pointer (SP), USP and ISP, are 16 bits wide each. 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 a 11-bit register indicating the CPU state.

2.8.1 Carry Flag (C)

The C flag retains a carry, borrow, or shift-out bit that has occurred in the arithmetic logic unit.

2.8.2 Debug Flag (D)

The D flag is for debug only. Set to "0".

2.8.3 Zero Flag (Z)

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

2.8.4 Sign Flag (S)

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

2.8.5 Register Bank Select Flag (B)

The register bank 0 is selected when the B flag is "0". The 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 the operation resulted in an overflow; otherwise, "0".

2.8.7 Interrupt Enable Flag (I Flag)

The I flag enables a maskable interrupt.

An interrupt is 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 Flag)

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, 3 bits wide, assigns processor interrupt priority levels from level 0 to level 7.

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

2.8.10 Reserved Bit

When write to this bit, set to "0". When read, its content is indeterminate.

3. Memory

3.1 R8C/14 Group

Figure 3.1 is a Memory Map of R8C/14 Group. The R8C/14 group provides 1-Mbyte address space from addresses 00000h to FFFFFh.

The internal ROM is allocated lower addresses beginning with address 0FFFFh. For example, a 16-Kbyte internal ROM is allocated addresses 0C000h 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 1-Kbyte internal RAM is allocated addresses 00400h to 007FFh. The internal RAM is used not only for storing data but for calling subroutines and stacks when interrupt request is acknowledged.

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

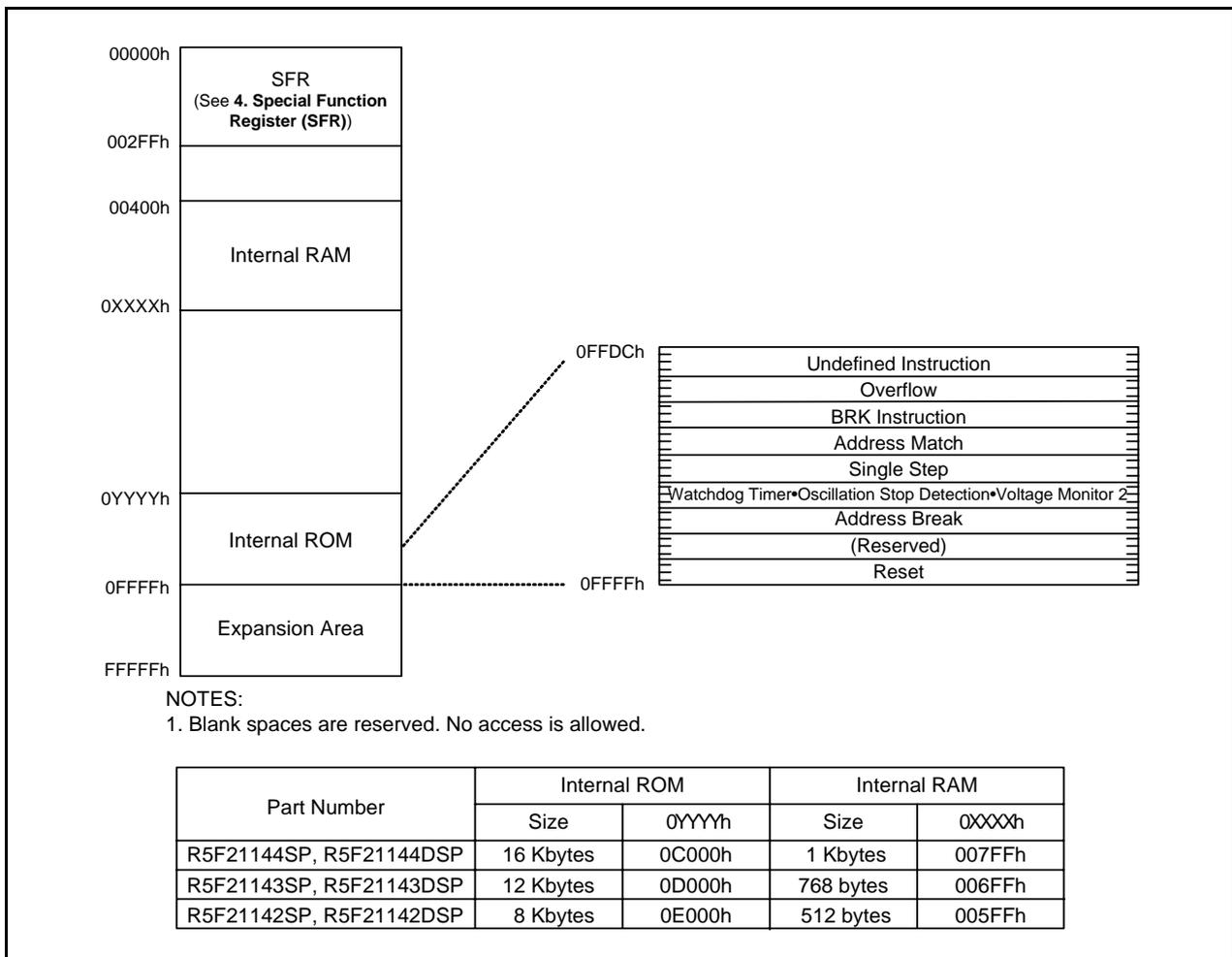


Figure 3.1 Memory Map of R8C/14 Group

3.2 R8C/15 Group

Figure 3.2 is a Memory Map of R8C/15 Group. The R8C/15 group provides 1-Mbyte address space from addresses 00000h to FFFFFh.

The internal ROM (program ROM) is allocated lower addresses beginning with address 0FFFFh. For example, a 16-Kbyte internal ROM is allocated addresses 0C000h 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 is allocated higher addresses beginning with address 00400h. For example, a 1-Kbyte internal RAM is allocated addresses 00400h to 007FFh. The internal RAM is used not only for storing data but for calling subroutines and stacks when interrupt request is acknowledged.

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

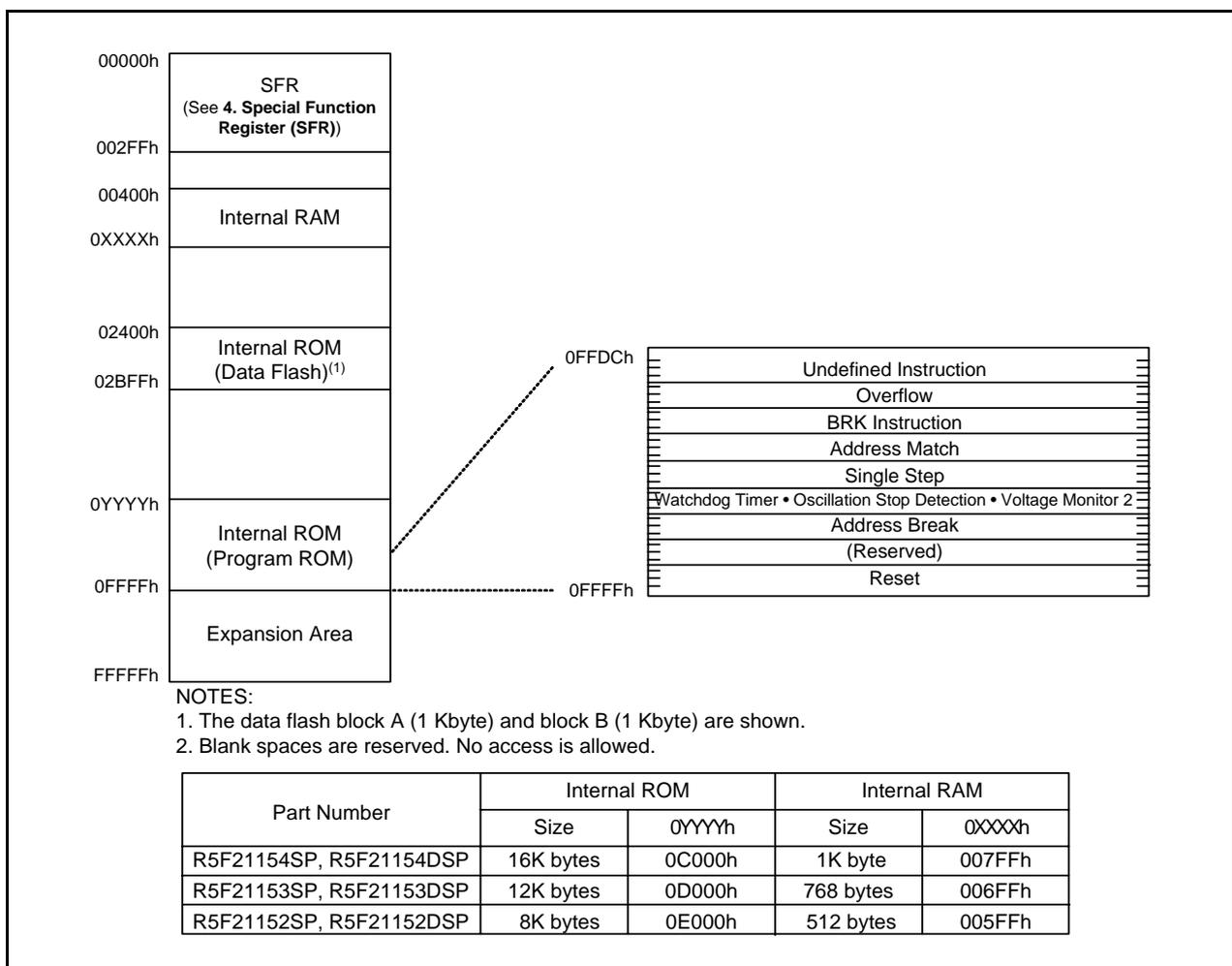


Figure 3.2 Memory Map of R8C/15 Group

4. Special Function Register (SFR)

SFR (Special Function Register) is the control register of peripheral functions. Tables 4.1 to 4.4 list the SFR information.

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 | | | |
| 0009h | Address Match Interrupt Enable Register | AIER | 00h |
| 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 | 00011111b |
| 0010h | Address Match Interrupt Register 0 | RMAD0 | 00h |
| 0011h | | | 00h |
| 0012h | | | X0h |
| 0013h | | | |
| 0014h | Address Match Interrupt Register 1 | RMAD1 | 00h |
| 0015h | | | 00h |
| 0016h | | | X0h |
| 0017h | | | |
| 0018h | | | |
| 0019h | | | |
| 001Ah | | | |
| 001Bh | | | |
| 001Ch | Count Source Protection Mode Register | CSPR | 00h |
| 001Dh | | | |
| 001Eh | INT0 Input Filter Select Register | INT0F | 00h |
| 001Fh | | | |
| 0020h | High-Speed On-Chip Oscillator Control Register 0 | HRA0 | 00h |
| 0021h | High-Speed On-Chip Oscillator Control Register 1 | HRA1 | When shipping |
| 0022h | High-Speed On-Chip Oscillator Control Register 2 | HRA2 | 00h |
| 0023h | | | |
| 0024h | | | |
| 0025h | | | |
| 0026h | | | |
| 0027h | | | |
| 0028h | | | |
| 0029h | | | |
| 002Ah | | | |
| 002Bh | | | |
| 002Ch | | | |
| 002Dh | | | |
| 002Eh | | | |
| 002Fh | | | |
| 0030h | | | |
| 0031h | Voltage Detection Register 1 ⁽²⁾ | VCA1 | 00001000b |
| 0032h | Voltage Detection Register 2 ⁽²⁾ | VCA2 | 00h ⁽³⁾ 01000000b ⁽⁴⁾ |
| 0033h | | | |
| 0034h | | | |
| 0035h | | | |
| 0036h | Voltage Monitor 1 Circuit Control Register ⁽²⁾ | VW1C | 0000X000b ⁽³⁾ 0100X001b ⁽⁴⁾ |
| 0037h | Voltage Monitor 2 Circuit Control Register ⁽⁵⁾ | VW2C | 00h |
| 0038h | | | |
| 0039h | | | |
| 003Ah | | | |
| 003Bh | | | |
| 003Ch | | | |
| 003Dh | | | |
| 003Eh | | | |
| 003Fh | | | |

X: Undefined

NOTES:

- Blank spaces are reserved. No access is allowed.
- Software reset, the watchdog timer reset or the voltage monitor 2 reset does not affect this register.
- Owing to Hardware reset.
- Owing to Power-on reset or the voltage monitor 1 reset.
- Software reset, the watchdog timer reset or the voltage monitor 2 reset does not affect the b2 and b3.

Table 4.2 SFR Information(2)(1)

| Address | Register | Symbol | After reset |
|---------|---|--------|-------------|
| 0040h | | | |
| 0041h | | | |
| 0042h | | | |
| 0043h | | | |
| 0044h | | | |
| 0045h | | | |
| 0046h | | | |
| 0047h | | | |
| 0048h | | | |
| 0049h | | | |
| 004Ah | | | |
| 004Bh | | | |
| 004Ch | | | |
| 004Dh | Key Input Interrupt Control Register | KUPIC | XXXXX000b |
| 004Eh | A/D Conversion Interrupt Control Register | ADIC | XXXXX000b |
| 004Fh | SSU Interrupt Control Register | SSUAIC | 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 | | | |
| 0054h | | | |
| 0055h | | | |
| 0056h | Timer X Interrupt Control Register | TXIC | XXXXX000b |
| 0057h | | | |
| 0058h | Timer Z Interrupt Control Register | TZIC | XXXXX000b |
| 0059h | INT1 Interrupt Control Register | INT1IC | XXXXX000b |
| 005Ah | INT3 Interrupt Control Register | INT3IC | XXXXX000b |
| 005Bh | Timer C Interrupt Control Register | TCIC | XXXXX000b |
| 005Ch | Compare 0 Interrupt Control Register | CMP0IC | XXXXX000b |
| 005Dh | INT0 Interrupt Control Register | INT0IC | XX00X000b |
| 005Eh | | | |
| 005Fh | | | |
| 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:

- Blank spaces are reserved. No access is allowed.

Table 4.3 SFR Information(3)(1)

| Address | Register | Symbol | After reset |
|---------|---|--------|--------------------|
| 0080h | Timer Z Mode Register | TZMR | 00h |
| 0081h | | | |
| 0082h | | | |
| 0083h | | | |
| 0084h | Timer Z Waveform Output Control Register | PUM | 00h |
| 0085h | Prescaler Z Register | PREZ | FFh |
| 0086h | Timer Z Secondary Register | TZSC | FFh |
| 0087h | Timer Z Primary Register | TZPR | FFh |
| 0088h | | | |
| 0089h | | | |
| 008Ah | Timer Z Output Control Register | TZOC | 00h |
| 008Bh | Timer X Mode Register | TXMR | 00h |
| 008Ch | Prescaler X Register | PREX | FFh |
| 008Dh | Timer X Register | TX | FFh |
| 008Eh | Timer Count Source Setting Register | TCSS | 00h |
| 008Fh | | | |
| 0090h | Timer C Register | TC | 00h |
| 0091h | | | 00h |
| 0092h | | | |
| 0093h | | | |
| 0094h | | | |
| 0095h | | | |
| 0096h | External Input Enable Register | INTEN | 00h |
| 0097h | | | |
| 0098h | Key Input Enable Register | KIEN | 00h |
| 0099h | | | |
| 009Ah | Timer C Control Register 0 | TCC0 | 00h |
| 009Bh | Timer C Control Register 1 | TCC1 | 00h |
| 009Ch | Capture, Compare 0 Register | TM0 | 00h |
| 009Dh | | | 00h ⁽²⁾ |
| 009Eh | Compare 1 Register | TM1 | FFh |
| 009Fh | | | FFh |
| 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 | | | |
| 00A9h | | | |
| 00AAh | | | |
| 00ABh | | | |
| 00ACh | | | |
| 00ADh | | | |
| 00AEh | | | |
| 00AFh | | | |
| 00B0h | UART Transmit/Receive Control Register 2 | U0CON | 00h |
| 00B1h | | | |
| 00B2h | | | |
| 00B3h | | | |
| 00B4h | | | |
| 00B5h | | | |
| 00B6h | | | |
| 00B7h | | | |
| 00B8h | SS Control Register H | SSCRH | 00h |
| 00B9h | SS Control Register L | SSCRL | 7Dh |
| 00BAh | SS Mode Register | SSMR | 18h |
| 00BBh | SS Enable Register | SSER | 00h |
| 00BCh | SS Status Register | SSSR | 00h |
| 00BDh | SS Mode Register 2 | SSMR2 | 00h |
| 00BEh | SS Transmit Data Register | SSTDR | FFh |
| 00BFh | SS Receive Data Register | SSRDR | FFh |

X: Undefined

NOTES:

- Blank spaces are reserved. No access is allowed.
- When output compare mode (the TCC13 bit in the TCC1 register = 1) is selected, the value after reset is "FFFFh".

Table 4.4 SFR Information(4)(1)

| Address | Register | Symbol | After reset |
|---------|---|--------|-------------|
| 00C0h | A/D Register | AD | XXh |
| 00C1h | | | XXh |
| 00C2h | | | |
| 00C3h | | | |
| 00C4h | | | |
| 00C5h | | | |
| 00C6h | | | |
| 00C7h | | | |
| 00C8h | | | |
| 00C9h | | | |
| 00CAh | | | |
| 00CBh | | | |
| 00CCh | | | |
| 00CDh | | | |
| 00CEh | | | |
| 00CFh | | | |
| 00D0h | | | |
| 00D1h | | | |
| 00D2h | | | |
| 00D3h | | | |
| 00D4h | A/D Control Register 2 | ADCON2 | 00h |
| 00D5h | | | |
| 00D6h | A/D Control Register 0 | ADCON0 | 00000XXb |
| 00D7h | A/D Control Register 1 | ADCON1 | 00h |
| 00D8h | | | |
| 00D9h | | | |
| 00DAh | | | |
| 00DBh | | | |
| 00DCh | | | |
| 00DDh | | | |
| 00DEh | | | |
| 00DFh | | | |
| 00E0h | | | |
| 00E1h | Port P1 Register | P1 | XXh |
| 00E2h | | | |
| 00E3h | Port P1 Direction Register | PD1 | 00h |
| 00E4h | | | |
| 00E5h | Port P3 Register | P3 | XXh |
| 00E6h | | | |
| 00E7h | Port P3 Direction Register | PD3 | 00h |
| 00E8h | Port P4 Register | P4 | XXh |
| 00E9h | | | |
| 00EAh | Port P4 Direction Register | PD4 | 00h |
| 00EBh | | | |
| 00ECh | | | |
| 00EDh | | | |
| 00EEh | | | |
| 00EFh | | | |
| 00F0h | | | |
| 00F1h | | | |
| 00F2h | | | |
| 00F3h | | | |
| 00F4h | | | |
| 00F5h | | | |
| 00F6h | | | |
| 00F7h | | | |
| 00F8h | | | |
| 00F9h | | | |
| 00FAh | | | |
| 00FBh | | | |
| 00FCh | Pull-Up Control Register 0 | PUR0 | 00XX0000b |
| 00FDh | Pull-Up Control Register 1 | PUR1 | XXXXXX0Xb |
| 00FEh | Port P1 Drive Capacity Control Register | DRR | 00h |
| 00FFh | Timer C Output Control Register | TCOUT | 00h |
| 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 |
| 0FFFh | Optional Function Select Register | OFS | (2) |

X: Undefined

NOTES:

- Blank columns, 0100h to 01B2h and 01B8h to 02FFh are all reserved. No access is allowed.
- The OFS register cannot be changed by program. Use a flash programmer to write to it.

5. Reset

There are resets: hardware reset, power-on reset, voltage monitor 1 reset, voltage monitor 2 reset, watchdog timer reset and software reset. Table 5.1 lists the Reset Name and Factor.

Table 5.1 Reset Name and Factor

| Reset Name | Factor |
|-------------------------|--|
| Hardware Reset | Input voltage of $\overline{\text{RESET}}$ pin is held "L" |
| Power-On Reset | VCC rises |
| 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 |

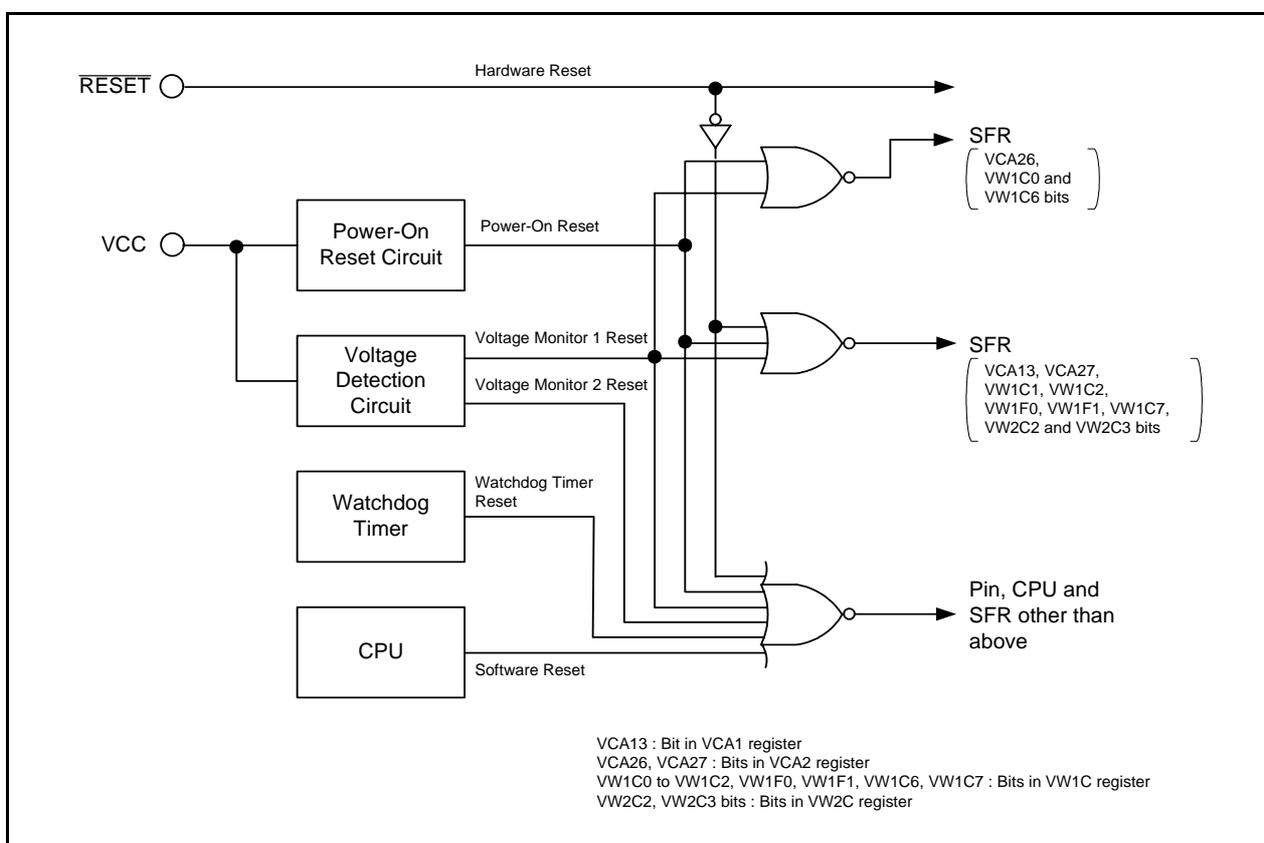


Figure 5.1 Block Diagram of Reset Circuit

Table 5.2 shows the Pin Status after Reset, Figure 5.2 shows CPU Register Status after Reset and Figure 5.3 shows Reset Sequence.

Table 5.2 Pin Status after Reset

| Pin Name | Pin Status |
|--------------------|------------|
| P1 | Input Port |
| P3_3 to P3_5, P3_7 | Input Port |
| P4_5 to P4_7 | Input Port |

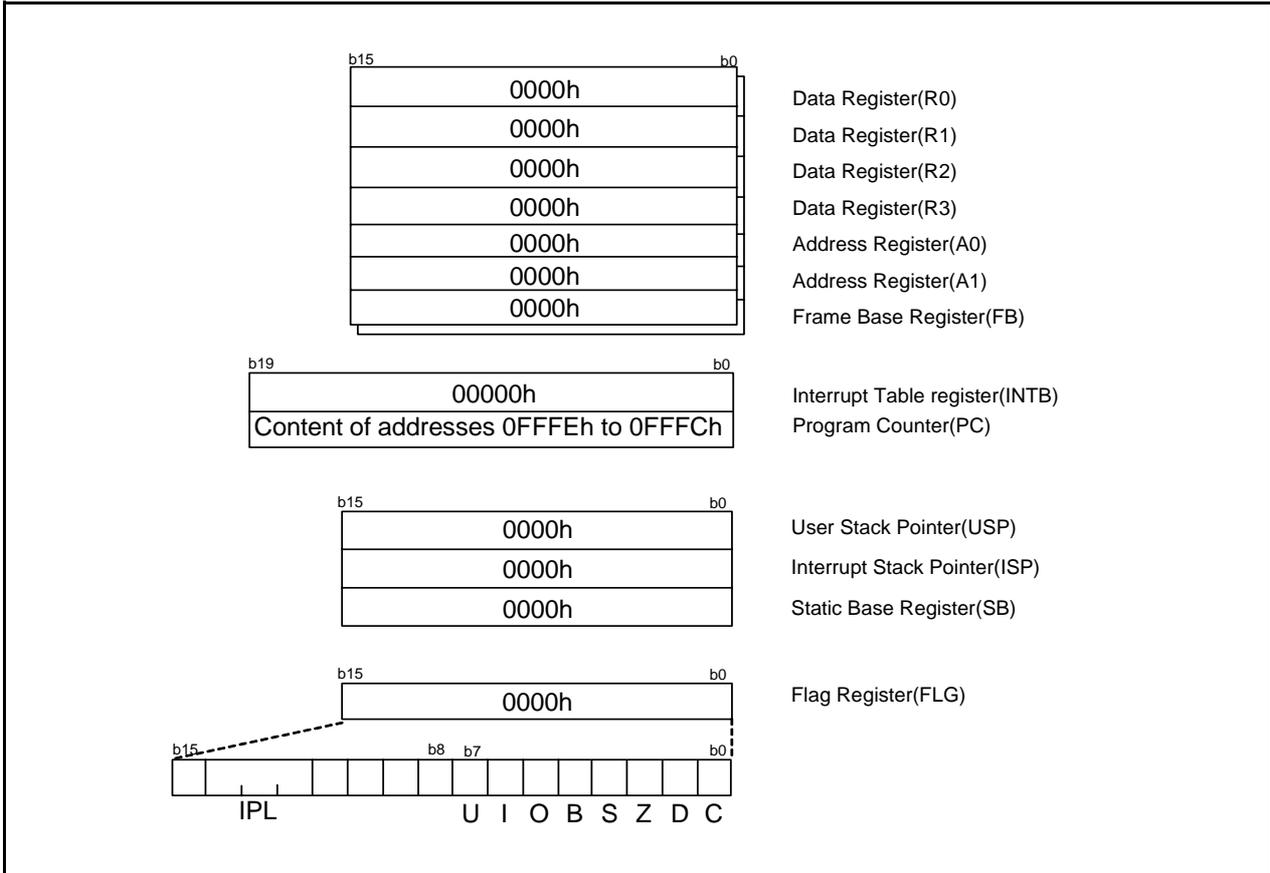


Figure 5.2 CPU Register Status after Reset

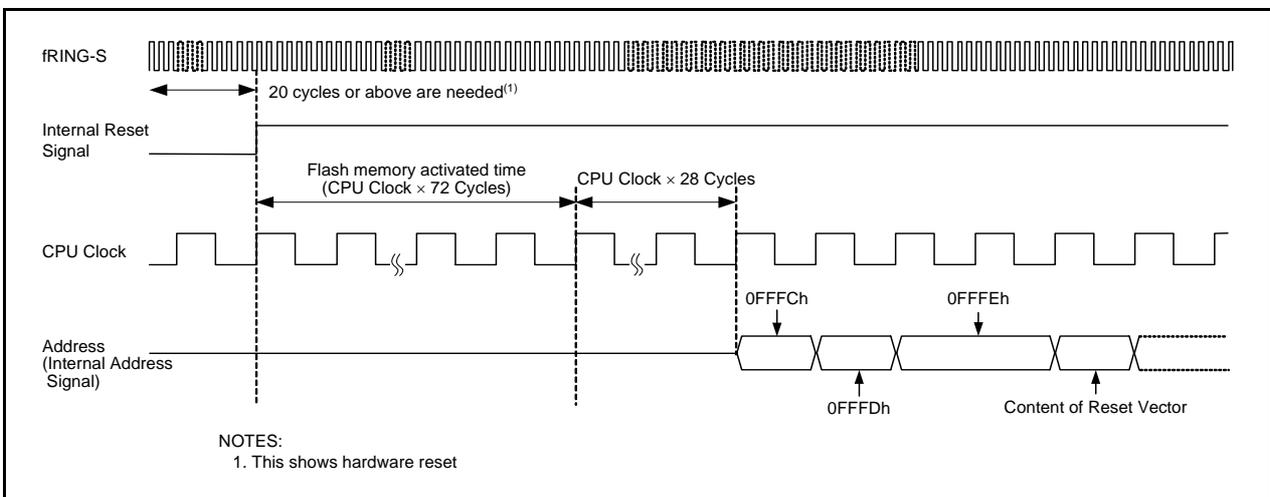


Figure 5.3 Reset Sequence

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 power supply voltage meets the recommended performance condition, the pins, CPU and SFR are reset (refer to **Table 5.2 Pin Status after Reset**). When the input level applied to the $\overline{\text{RESET}}$ pin changes "L" to "H", the 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 for the CPU clock.

Refer to **4. Special Function Register (SFR)** for the status of the SFR after reset.

The internal RAM is not reset. If the $\overline{\text{RESET}}$ pin is pulled "L" during writing to the internal RAM, the internal RAM will be in indeterminate state.

Figure 5.4 shows the Example of Hardware Reset Circuit and Operation and Figure 5.5 shows the Example of Hardware Reset Circuit (Use Example of External Power Supply Voltage Detection Circuit) and Operation.

5.1.1 When the power supply is stable

- (1) Apply an "L" signal to the $\overline{\text{RESET}}$ pin.
- (2) Wait for $500\mu\text{s}$ ($1/\text{fRING-S}\times 20$).
- (3) Apply an "H" signal to the $\overline{\text{RESET}}$ pin.

5.1.2 Power on

- (1) Apply an "L" signal to the $\overline{\text{RESET}}$ pin.
- (2) Let the power supply voltage increase until it meets the recommended performance condition.
- (3) Wait for $t_d(\text{P-R})$ or more until the internal power supply stabilizes (Refer to **19. Electrical Characteristics**).
- (4) Wait for $500\mu\text{s}$ ($1/\text{fRING-S}\times 20$).
- (5) Apply an "H" signal to the $\overline{\text{RESET}}$ pin.

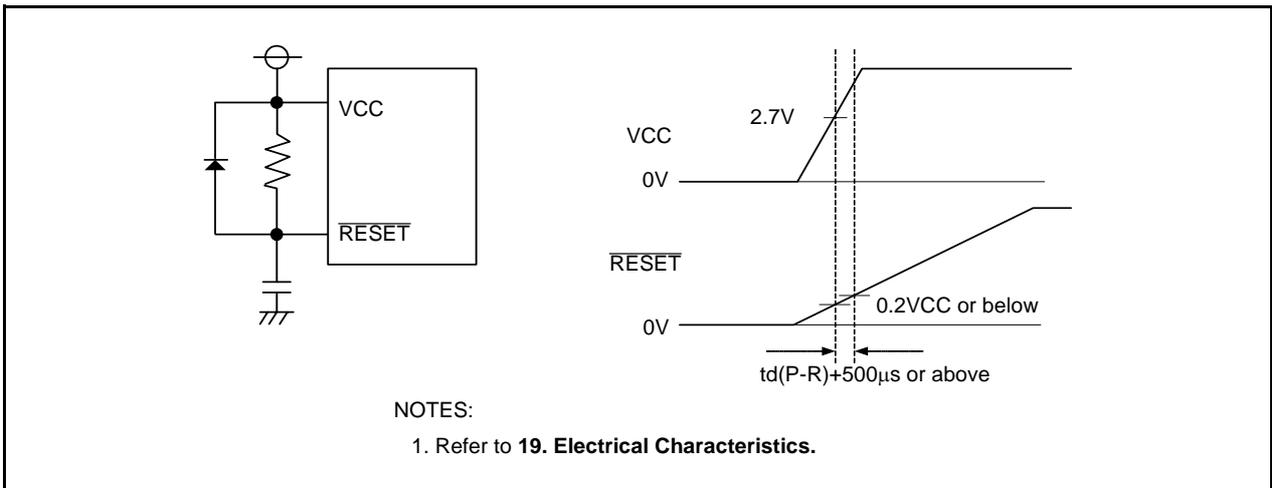


Figure 5.4 Example of Hardware Reset Circuit and Operation

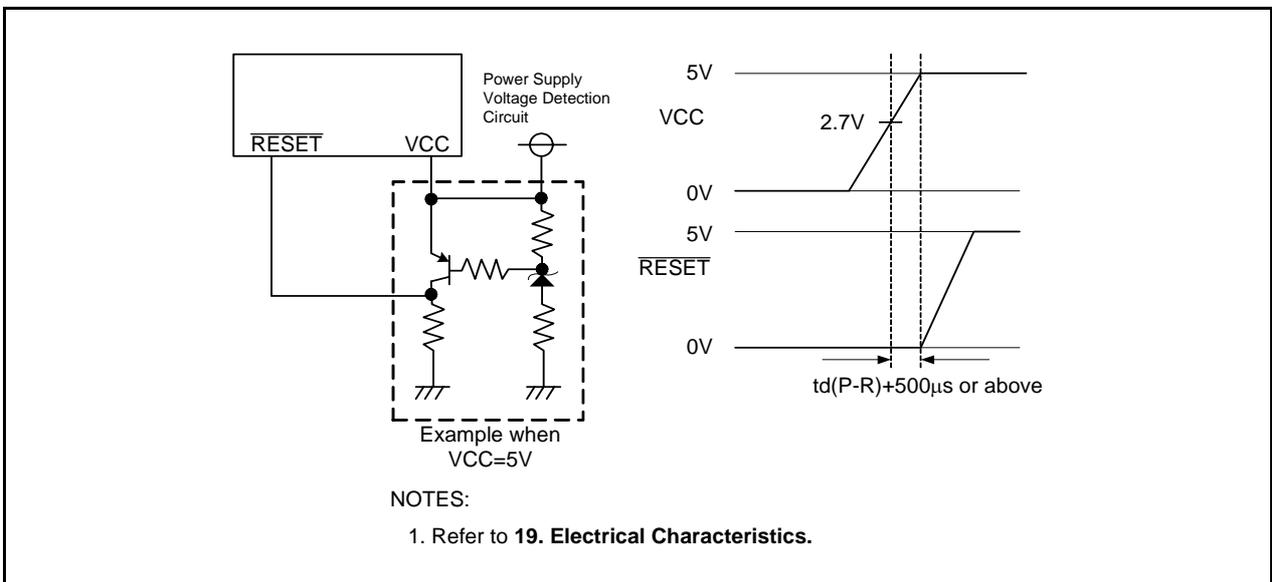


Figure 5.5 Example of Hardware Reset Circuit (Use Example of External Power Supply Voltage Detection Circuit) and Operation

5.2 Power-On Reset Function

When the $\overline{\text{RESET}}$ pin is connected to the VCC pin via about $5\text{k}\Omega$ pull-up resistor and the VCC pin rises, the function is enabled and the microcomputer resets its pins, CPU, and SFR. When a capacitor is connected to the $\overline{\text{RESET}}$ pin, always keep the voltage to the $\overline{\text{RESET}}$ pin 0.8VCC or more.

When the input voltage to the VCC pin reaches to V_{det1} level or above, count operation of the low-speed on-chip oscillator clock starts. When the operation counts the low-speed on-chip oscillator clock for 32 times, the internal reset signal is held "H" and the microcomputer enters the reset sequence (See Figure 5.3). The low-speed on-chip oscillator clock divide-by-8 is automatically selected for the CPU after reset.

Refer to 4. Special Function Register (SFR) for the status of the SFR after power-on reset.

The voltage monitor 1 reset is enabled after power-on reset.

Figure 5.6 shows the Example of Power-On Reset Circuit and Operation.

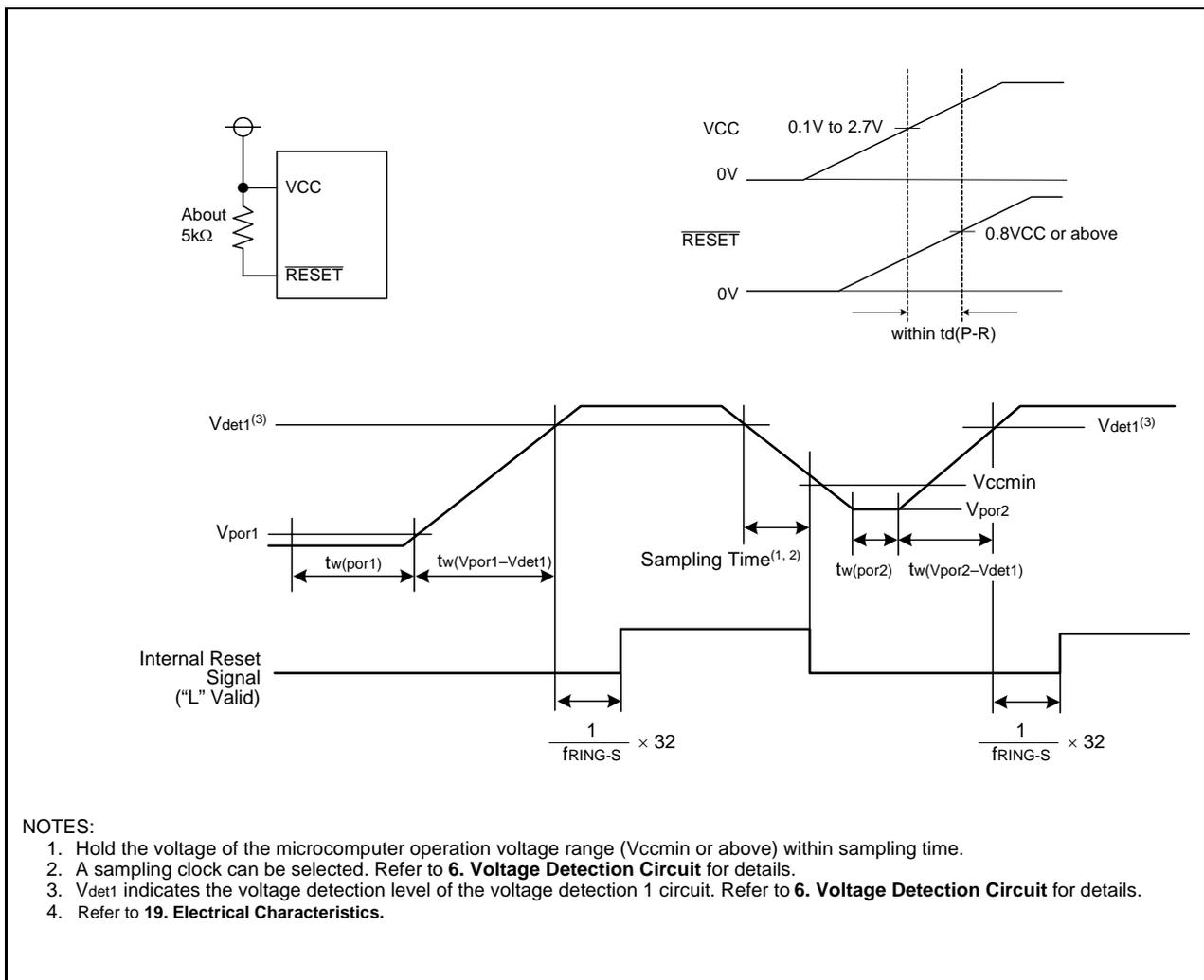


Figure 5.6 Example of Power-On Reset Circuit and Operation

5.3 Voltage Monitor 1 Reset

A reset is applied using the built-in 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 to the Vdet1 level or below, the pins, CPU and SFR are reset.

And when the input voltage to the VCC pin reaches to the Vdet1 level or above, count operation of the low-speed on-chip oscillator clock starts. When the operation counts the low-speed on-chip oscillator clock for 32 times, the internal reset signal is held "H" and the microcomputer enters the reset sequence (See Figure 5.3). The low-speed on-chip oscillator clock divide-by-8 is automatically selected for the CPU after reset.

Refer to **4. Special Function Register (SFR)** for the status of the SFR after voltage monitor 1 reset.

The internal RAM is not reset. When the input voltage to the VCC pin reaches to the Vdet1 level or below during writing to the internal RAM, the internal RAM is in indeterminate state.

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

5.4 Voltage Monitor 2 Reset

A reset is applied using the built-in 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 drops to the Vdet2 level or below, the pins, CPU and SFR are reset and the program is executed beginning with the address indicated by the reset vector. After reset, the low-speed on-chip oscillator clock divide-by-8 is automatically selected for the CPU clock.

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

The internal RAM is not reset. When the input voltage to the VCC pin reaches to the Vdet2 level or below during writing to the internal RAM, the internal RAM is in indeterminate state.

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

5.5 Watchdog Timer Reset

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

After reset, the low-speed on-chip oscillator clock divide-by-8 is automatically selected for the CPU clock.

The watchdog timer reset does not reset some SFRs. Refer to **4. Special Function Register (SFR)** for details.

The internal RAM is not reset. When the watchdog timer underflows, the internal RAM is in indeterminate state.

Refer to **12. Watchdog Timer** for watchdog timer.

5.6 Software Reset

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

The software reset does not reset some SFRs. Refer to **4. Special Function Register (SFR)** for details.

The internal RAM is not reset.

6. Voltage Detection Circuit

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

Table 6.1 lists the Specification of Voltage Detection Circuit and Figures 6.1 to 6.3 show the Block Diagrams. Figures 6.4 to 6.6 show the Associated Registers.

Table 6.1 Specification of Voltage Detection Circuit

| Item | | Voltage Detection 1 | Voltage Detection 2 |
|----------------------------------|---------------------------|---|---|
| VCC Monitor | Voltage to Monitor | Vdet1 | Vdet2 |
| | Detection Target | Whether passing through Vdet1 by rising or falling | Whether passing through Vdet2 by rising or falling |
| | Monitor | None | VCA13 bit in VCA1 register Whether VCC is higher or lower than Vdet2 |
| Process When Voltage Is Detected | Reset | Voltage Monitor 1 Reset | Voltage Monitor 2 Reset |
| | | Reset at Vdet1 > VCC; Restart CPU operation at VCC > Vdet1 | Reset at Vdet2 > VCC Restart CPU operation after a specified time |
| Interrupt | Interrupt | None | 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 |
| | Sampling Time | (Divide-by-n of fRING-S) x 4 n : 1, 2, 4 and 8 | (Divide-by-n of fRING-S) x 4 n : 1, 2, 4 and 8 |

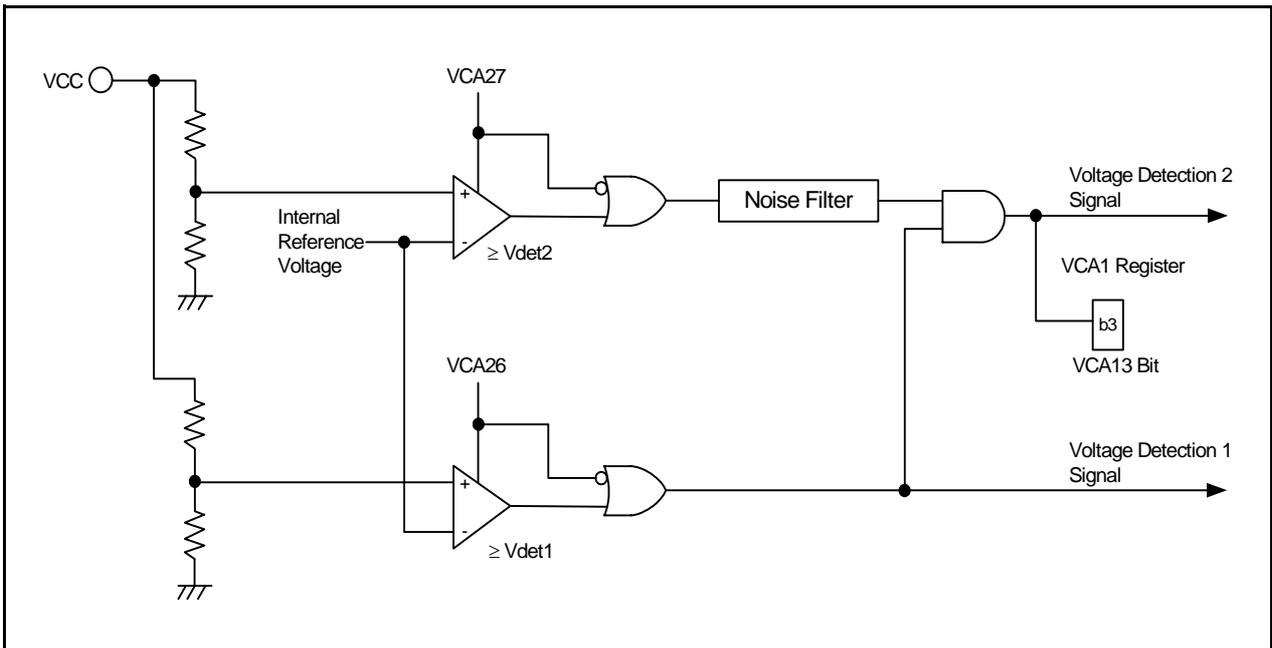


Figure 6.1 Block Diagram of Voltage Detection Circuit

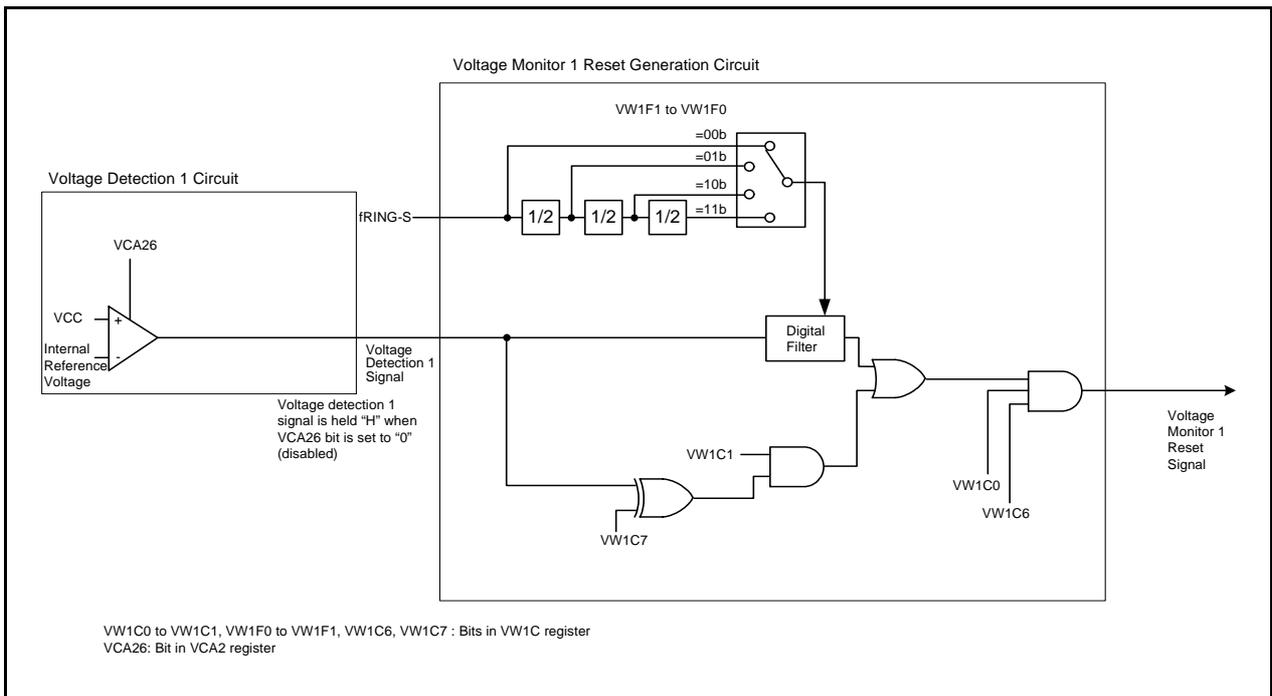


Figure 6.2 Block Diagram of Voltage Monitor 1 Reset Generation Circuit

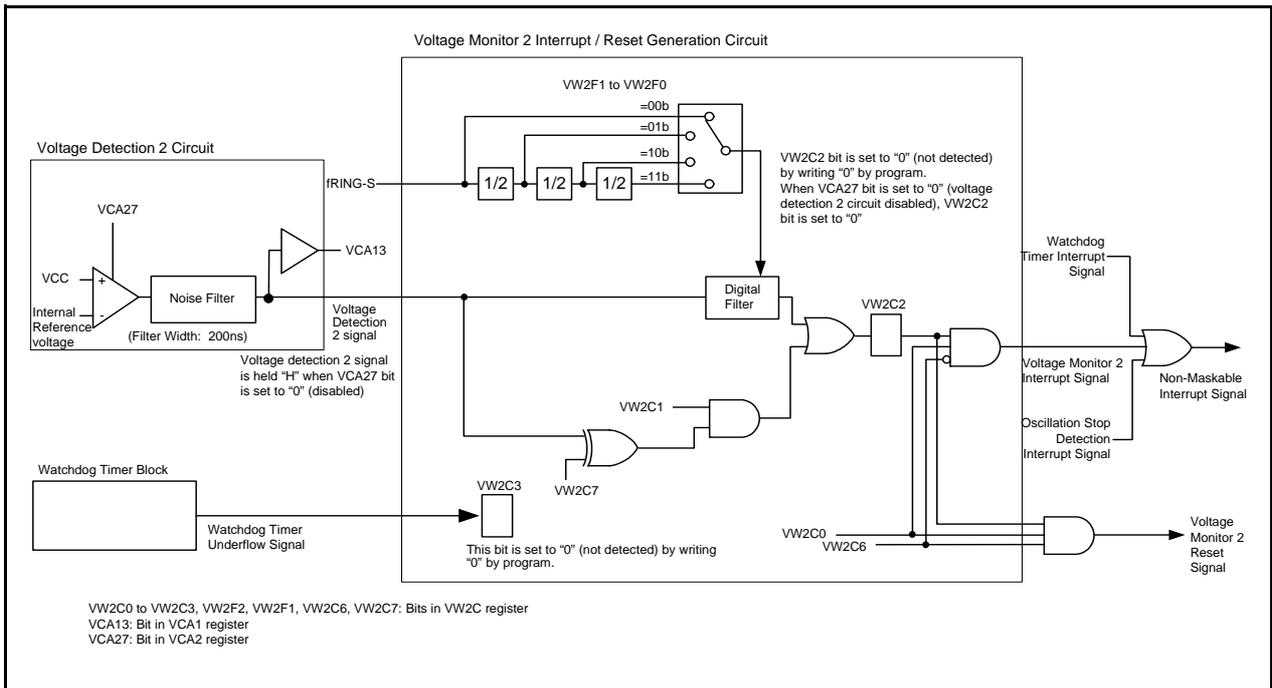


Figure 6.3 Block Diagram of Voltage Monitor 2 Interrupt / Reset Generation Circuit

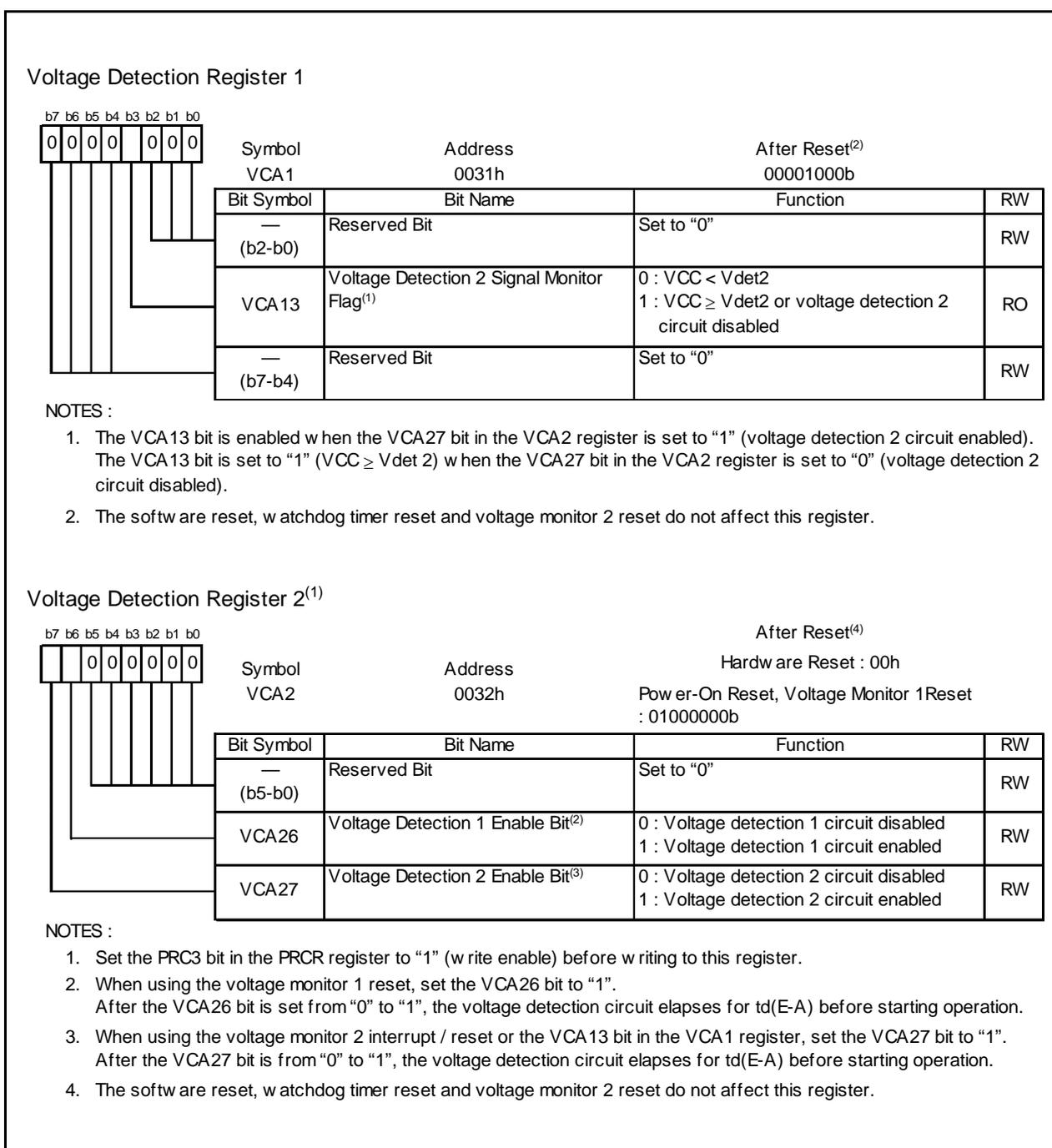


Figure 6.4 VCA1 and VCA2 Registers

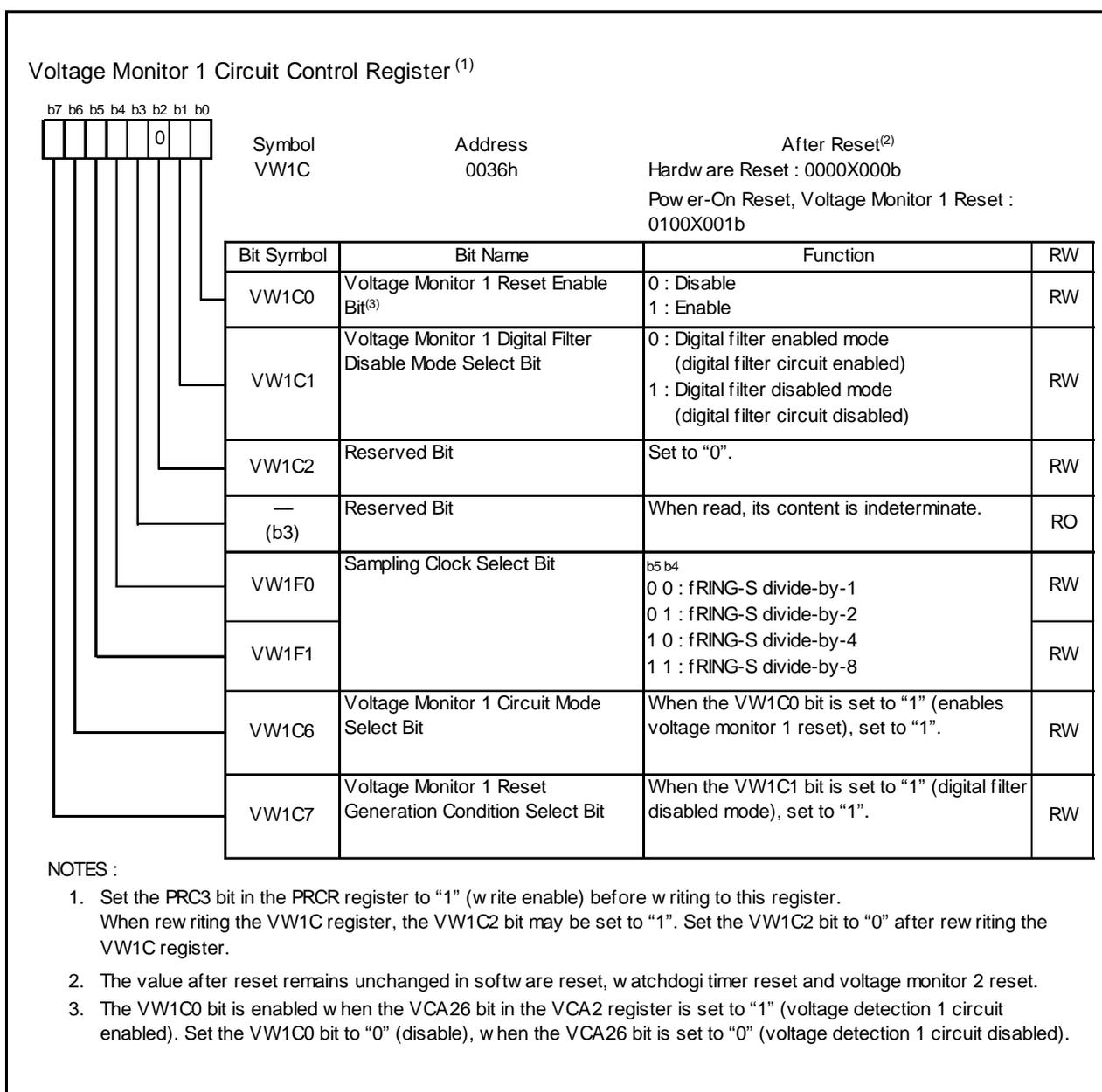


Figure 6.5 VW1C Register

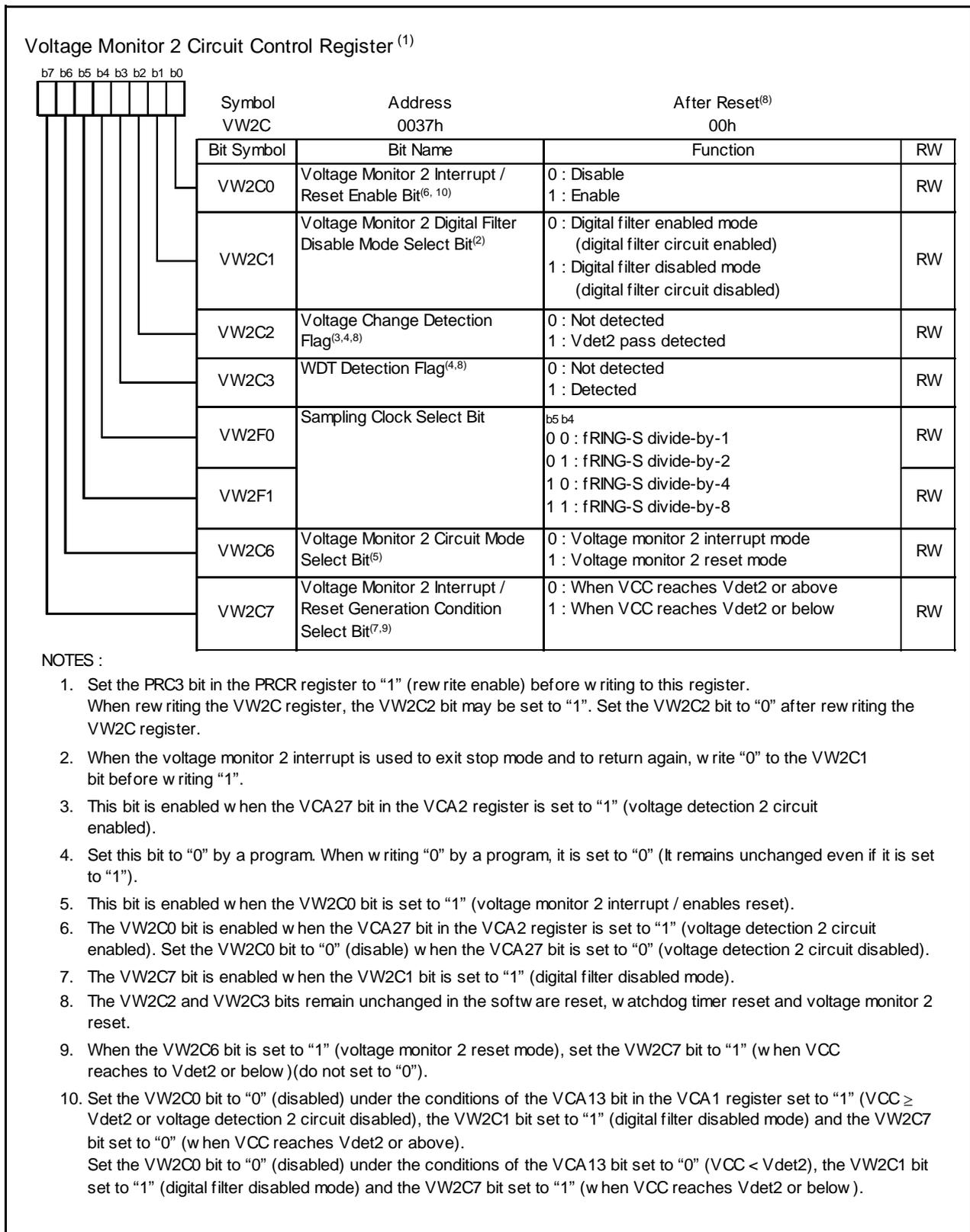


Figure 6.6 VW2C Register

6.1 Monitoring VCC Input Voltage

6.1.1 Monitoring Vdet1

Vdet1 cannot be monitored.

6.1.2 Monitoring Vdet2

Set the VCA27 bit in the VCA2 register to "1" (voltage detection 2 circuit enabled). After $t_d(E-A)$ (refer to **19. Electrical Characteristics**) elapse, Vdet2 can be monitored by the VCA13 bit in the VCA1 register.

6.2 Voltage Monitor 1 Reset

Table 6.2 lists the Setting Procedure of Voltage Monitor 1 Reset Associated Bit and Figure 6.7 shows the Operating Example of Voltage Monitor 1 Reset. When using the voltage monitor 1 reset to exit stop mode, set the VW1C1 bit in the VW1C register to "1" (digital filter disabled).

Table 6.2 Setting Procedure of Voltage Monitor 1 Reset Associated Bit

| Procedure | When Using Digital Filter | When Not Using Digital Filter |
|-----------|---|---|
| 1 | Set the VCA26 bit in the VCA2 register to "1" (voltage detection 1 circuit enabled) | |
| 2 | Wait for $t_d(E-A)$ | |
| 3(1) | Select the sampling clock of the digital filter by the VW1F0 to VW1F1 bits in the VW1C register | Set the VW1C7 bit in the VW1C register to "1" |
| 4(1) | 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(1) | 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" | |
| 7 | Set the CM14 bit in the CM1 register to "0" (low-speed on-chip oscillator on) | – |
| 8 | Wait for the sampling clock of the digital filter x 4 cycles | – (no wait time) |
| 9 | Set the VW1C0 bit in the VW1C register to "1" (enables voltage monitor 1 reset) | |

NOTES:

- When the VW1C0 bit is set to "0" (disabled), procedures 3, 4 and 5 can be executed simultaneously (with 1 instruction).

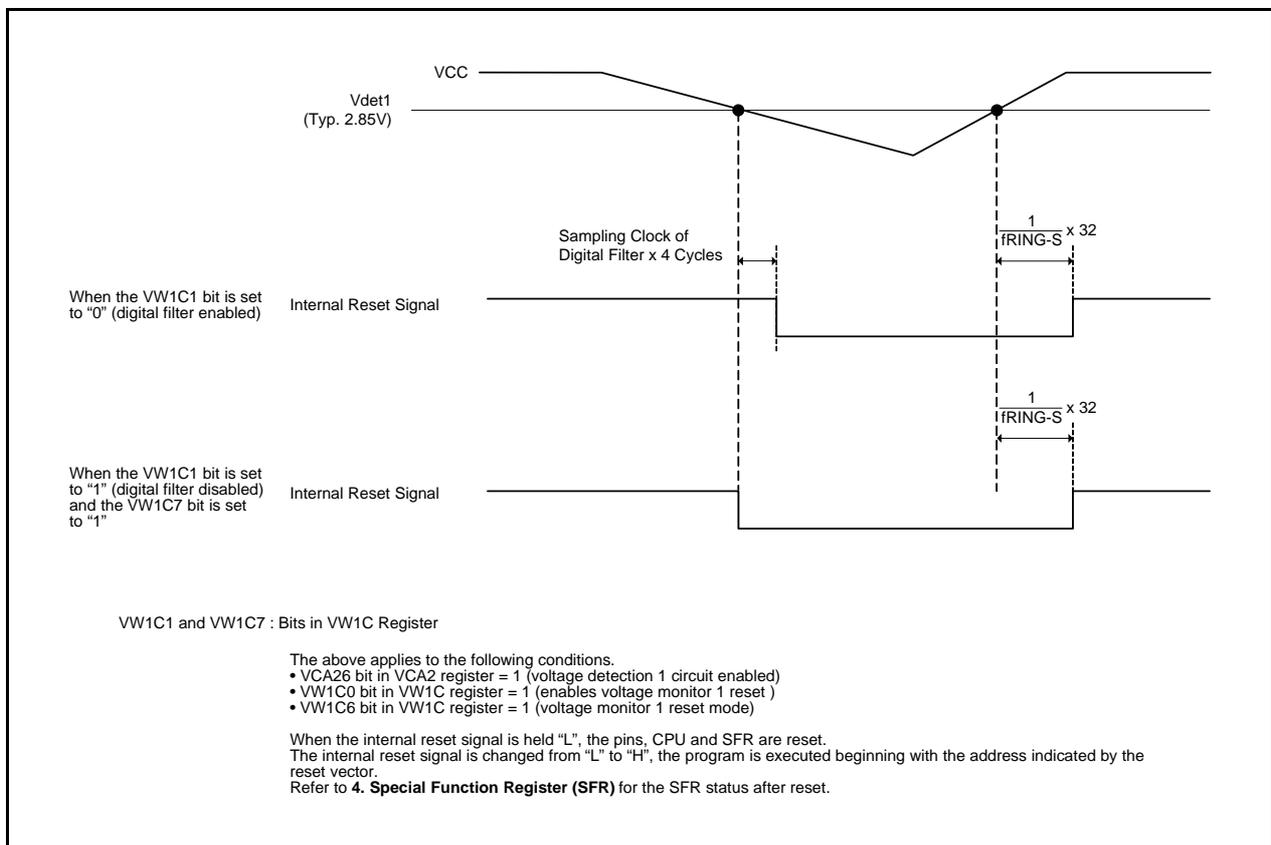


Figure 6.7 Operating Example of Voltage Monitor 1 Reset

6.3 Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset

Table 6.3 lists the Setting Procedure of Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset Associated Bit. Figure 6.8 shows the Operating Example of Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset. When using 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 6.3 Setting Procedure of Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset Associated Bit

| Procedure | 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 ⁽²⁾ | 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 ⁽²⁾ | 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 the sampling clock of the digital filter x 4 cycles | | - (no wait time) | |
| 9 | Set the VW2C0 bit in the VW2C register to "1" (enables voltage monitor 2 interrupt / reset) | | | |

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" (disabled), procedures 3, 4 and 5 can be executed simultaneously (with 1 instruction).

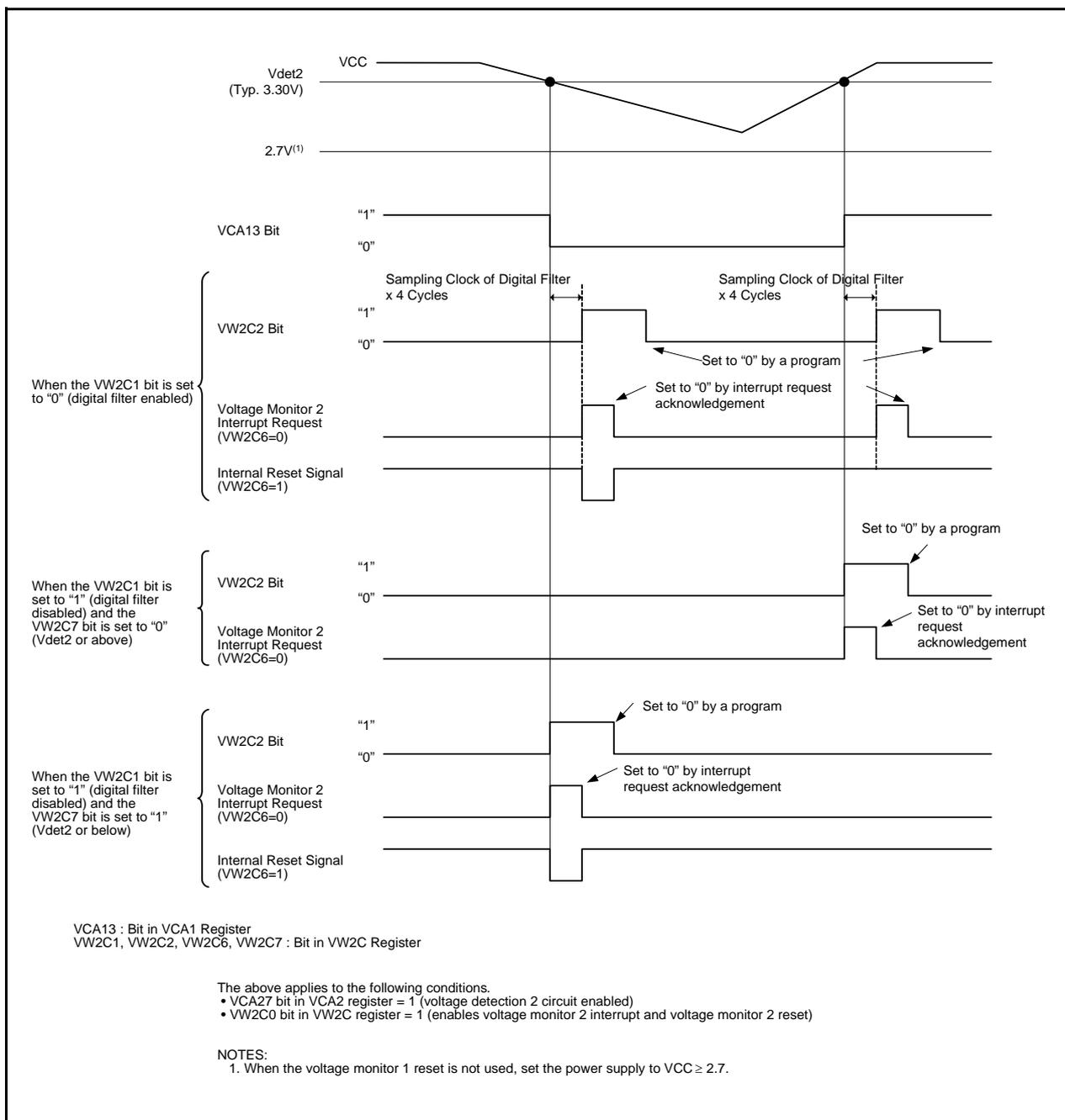


Figure 6.8 Operating Example of Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset

7. Processor Mode

7.1 Types of Processor Mode

Single-chip mode can be selected as processor mode. Table 7.1 lists Features of Processor Mode. Figure 7.1 shows the PM0 Register and Figure 7.2 shows the PM1 Register.

Table 7.1 Features of Processor Mode

| Processor Mode | Access Area | Pins to which I/O ports are assigned |
|------------------|---------------------------------|--|
| Single-Chip Mode | SFR, Internal RAM, Internal ROM | All pins are I/O ports or peripheral function I/O pins |

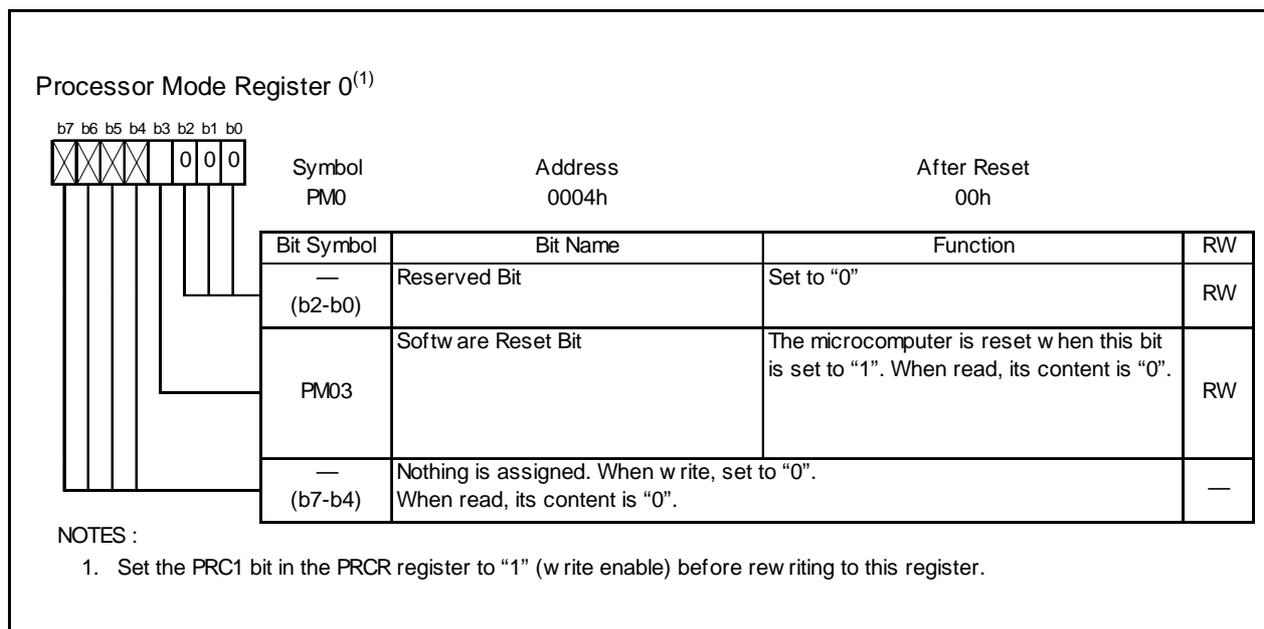


Figure 7.1 PM0 Register

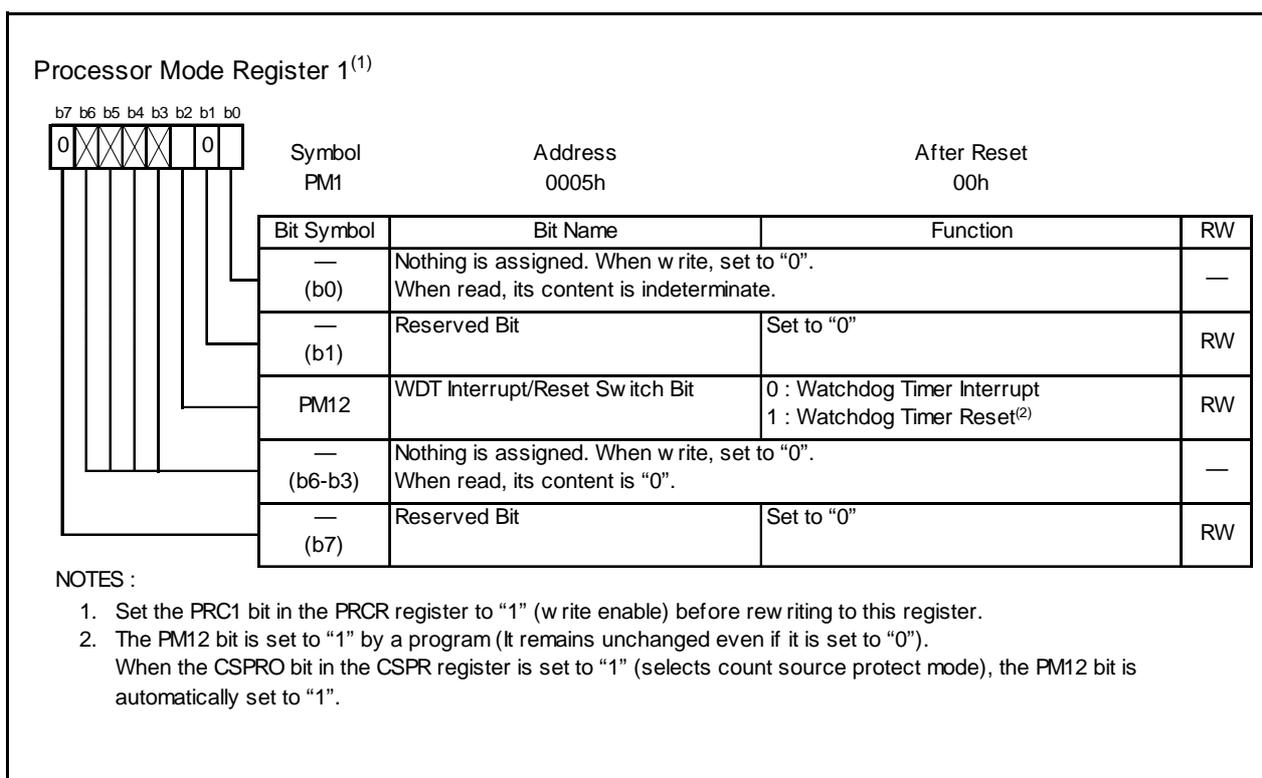


Figure 7.2 PM1 Register

8. Bus

During access, the ROM/RAM and SFR vary from bus cycles. Table 8.1 lists Bus Cycles for Access Space of the R8C/14 Group and Table 8.2 lists Bus Cycles for Access Space of the R8C/15 Group.

The ROM/RAM and SFR are connected to the CPU through an 8-bit bus. When accessing in word-(16 bits) unit, these area are accessed twice in 8-bit unit. Table 8.3 lists Access Unit and Bus Operation.

Table 8.1 Bus Cycles for Access Space of the R8C/14 Group

| Access Area | Bus Cycle |
|-------------|-----------------------|
| SFR | 2 cycles of CPU clock |
| ROM/RAM | 1 cycle of CPU clock |

Table 8.2 Bus Cycles for Access Space of the R8C/15 Group

| Access Area | Bus Cycle |
|-----------------|-----------------------|
| SFR/Data flash | 2 cycles of CPU clock |
| Program ROM/RAM | 1 cycle of CPU clock |

Table 8.3 Access Unit and Bus Operation

| Area | SFR, Data flash | ROM (Program ROM), RAM |
|--------------------------|-----------------|------------------------|
| Even Address Byte Access | | |
| Odd Address Byte Access | | |
| Even Address Word Access | | |
| Odd Address Word Access | | |

9. Clock Generation Circuit

The MCU has two on-chip clock generation circuits:

- Main clock oscillation circuit
- On-chip oscillator (oscillation stop detection function)

Table 9.1 lists Specification of Clock Generation Circuit. Figure 9.1 shows a Clock Generation Circuit. Figures 9.2 to 9.5 show clock-associated registers.

Table 9.1 Specification of Clock Generation Circuit

| Item | Main Clock Oscillation Circuit | On-Chip Oscillator | |
|------------------------------------|--|---|---|
| | | High-Speed On-Chip Oscillator | Low-Speed On-Chip Oscillator |
| Use of Clock | <ul style="list-style-type: none"> • CPU clock source • Peripheral function clock source | <ul style="list-style-type: none"> • CPU clock source • Peripheral function clock source • CPU and peripheral function clock sources when main clock stops oscillating | <ul style="list-style-type: none"> • CPU clock source • Peripheral function clock source • CPU and peripheral function clock sources when main clock stops oscillating |
| Clock Frequency | 0 to 20MHz | Approx. 8MHz | Approx. 125kHz |
| Connectable Oscillator | <ul style="list-style-type: none"> • Ceramic resonator • Crystal oscillator | – | – |
| Oscillator Connect Pins | XIN, XOUT ⁽¹⁾ | (Note 1) | (Note 1) |
| Oscillation Stop, Restart Function | Usable | Usable | Usable |
| Oscillator Status After Reset | Stop | Stop | Oscillate |
| Others | Externally generated clock can be input | – | – |

NOTES:

1. This pin can be used as P4_6 and P4_7 when using the on-chip oscillator clock for a CPU clock while the main clock oscillation circuit is not used.

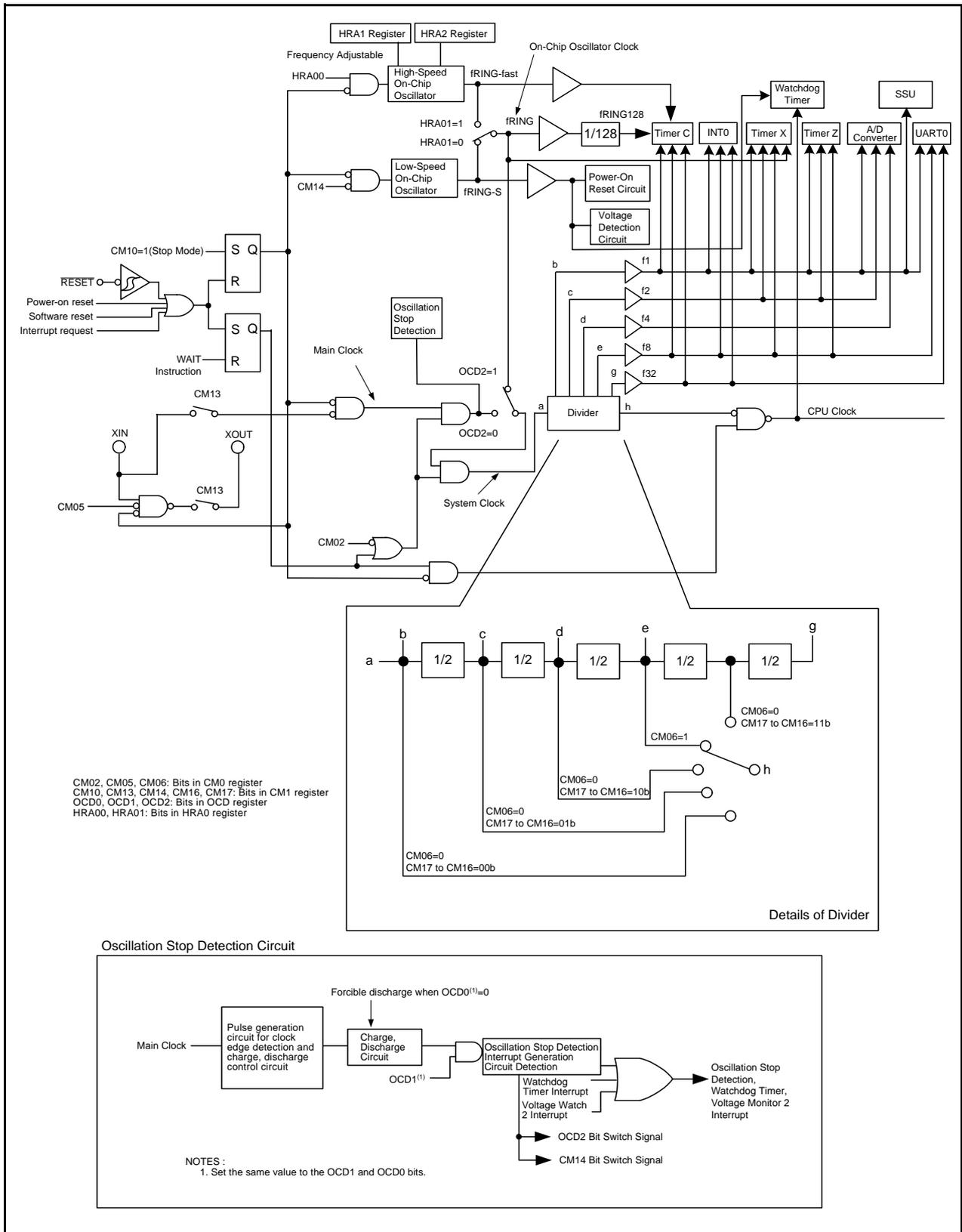


Figure 9.1 Clock Generation Circuit

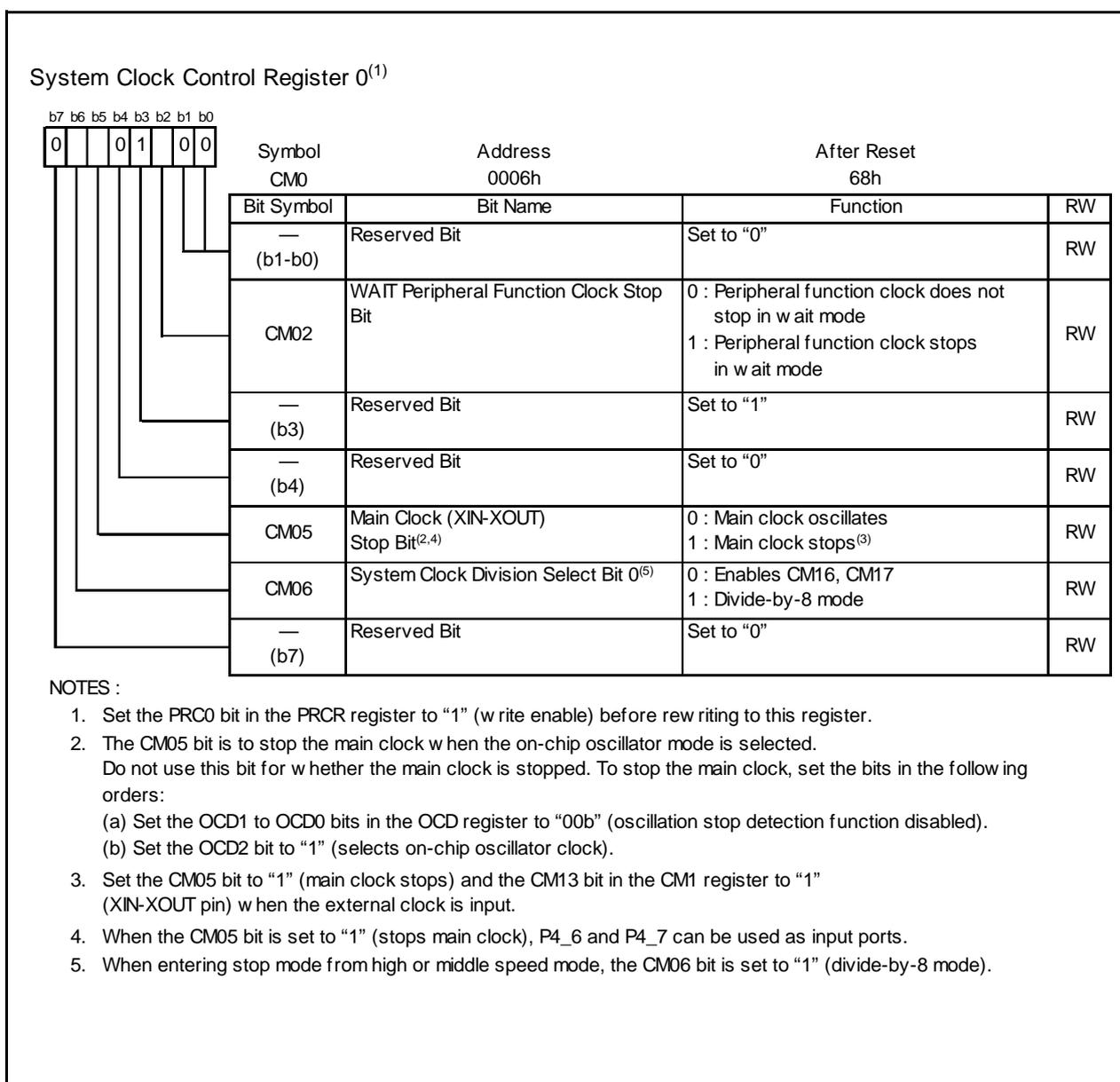


Figure 9.2 CM0 Register

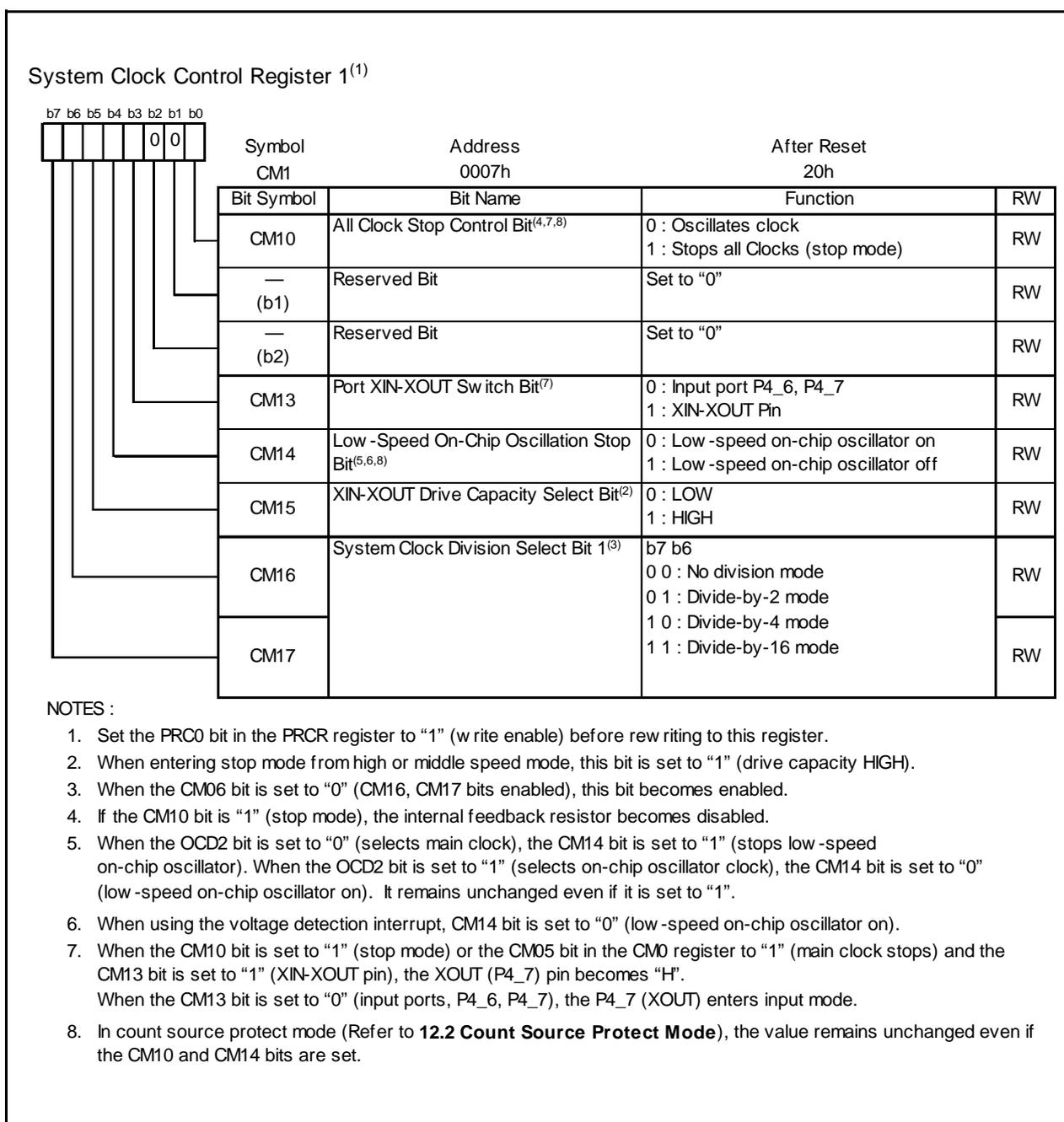


Figure 9.3 CM1 Register

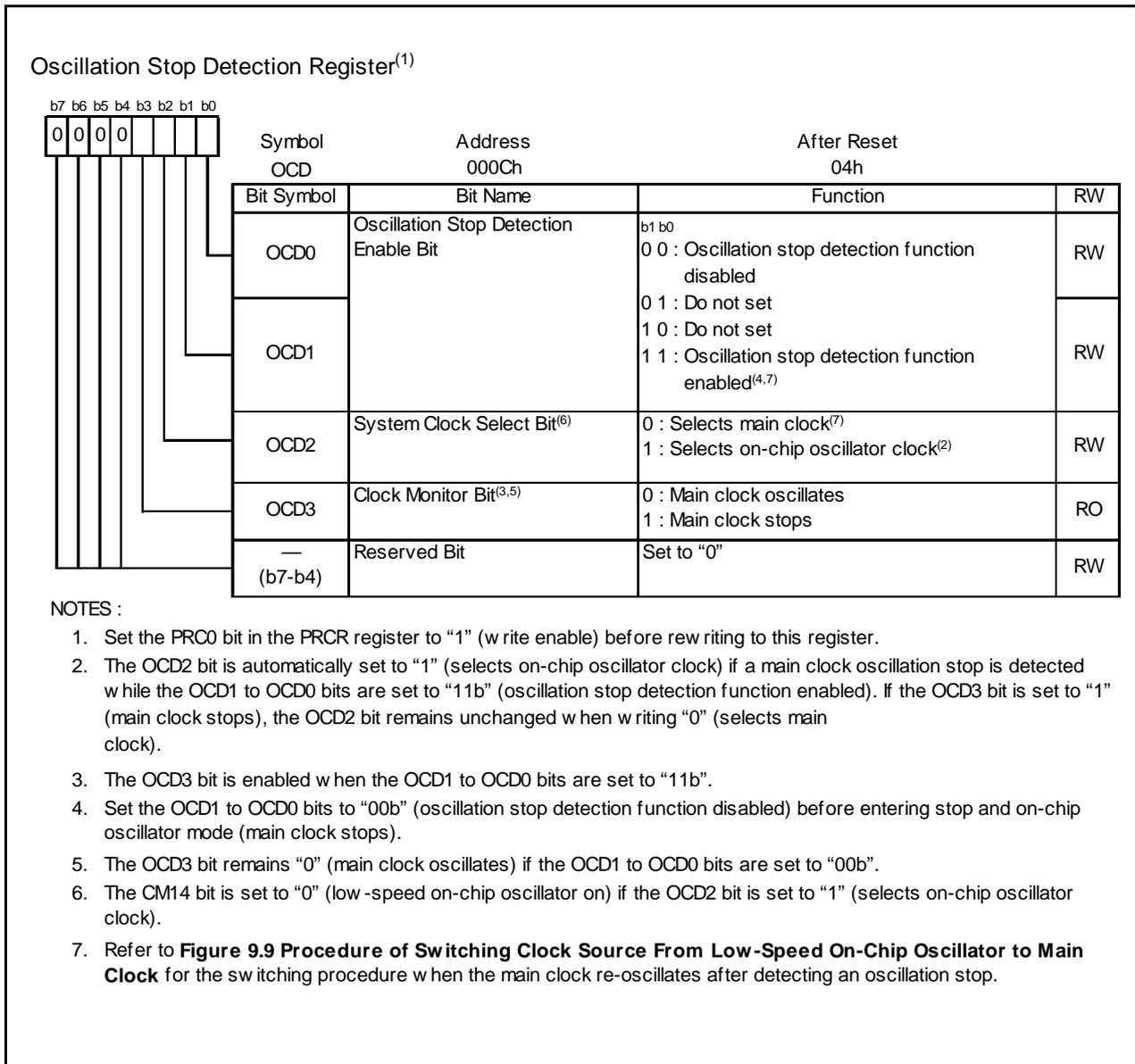


Figure 9.4 OCD Register

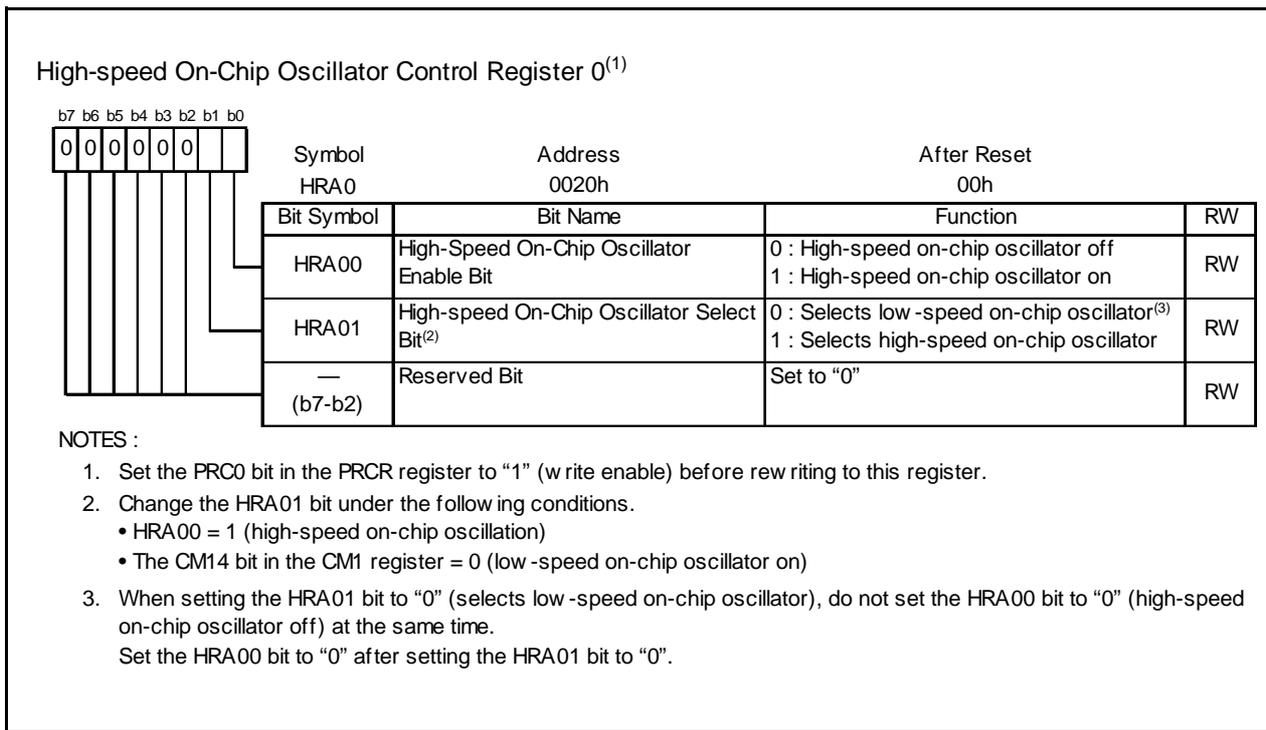


Figure 9.5 HRA0 Register

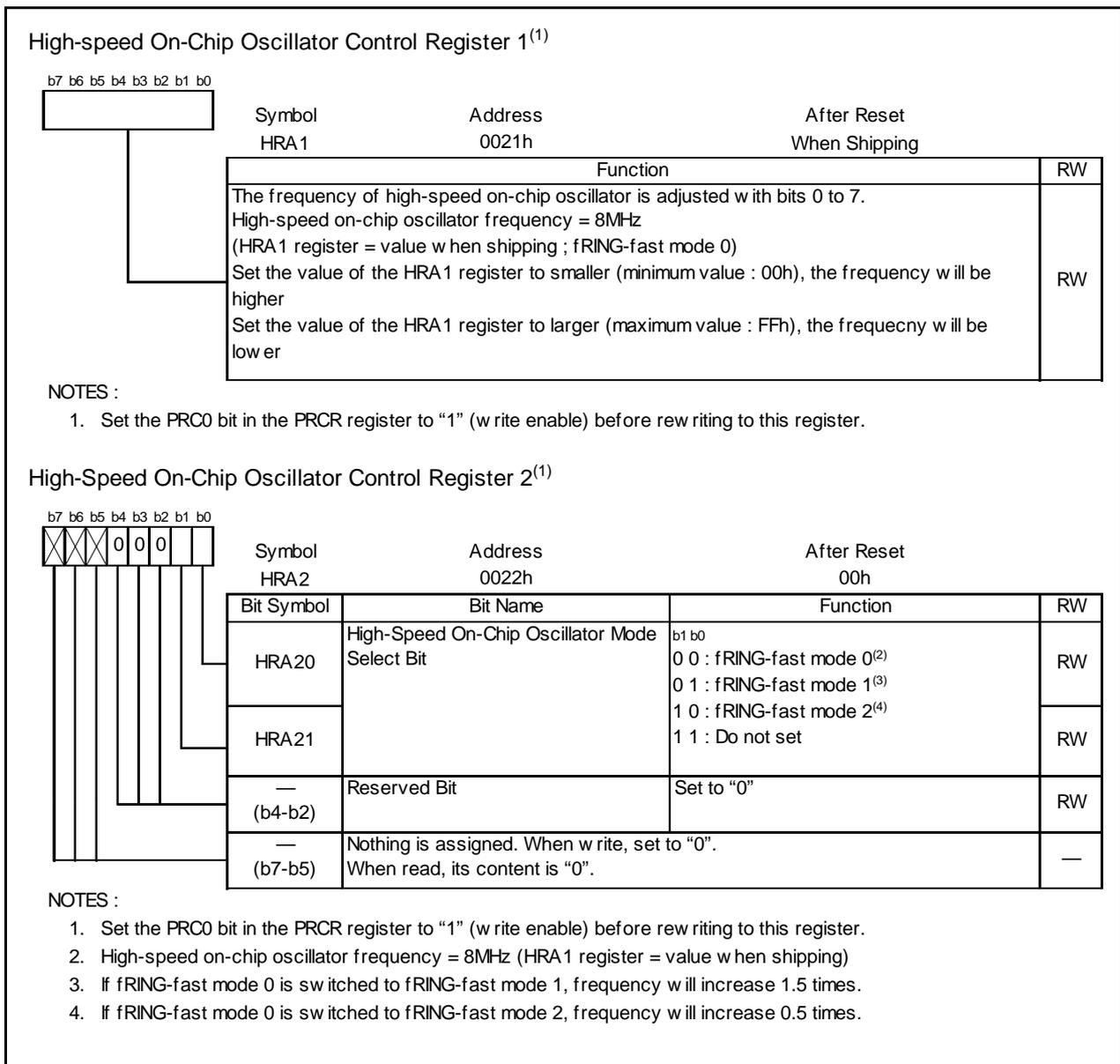


Figure 9.6 HRA1 and HRA2 Registers

The following describes the clocks generated by the clock generation circuit.

9.1 Main Clock

This clock is supplied by a main clock oscillation circuit. This clock is used as the clock source for the CPU and peripheral function clocks. The main clock oscillation circuit is configured by connecting a resonator between the XIN and XOUT pins. The main clock oscillation circuit contains a 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 main clock oscillation circuit may also be configured by feeding an externally generated clock to the XIN pin. Figure 9.7 shows the Examples of Main Clock Connection Circuit.

During reset and after reset, the main clock stops.

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

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

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

When the clocks externally generated to the XIN pin are input, a main clock does not stop if setting the CM05 bit to "1". If necessary, use an external circuit to stop the clock.

In stop mode, all clocks including the main clock stop. Refer to **9.4 Power Control** for details.

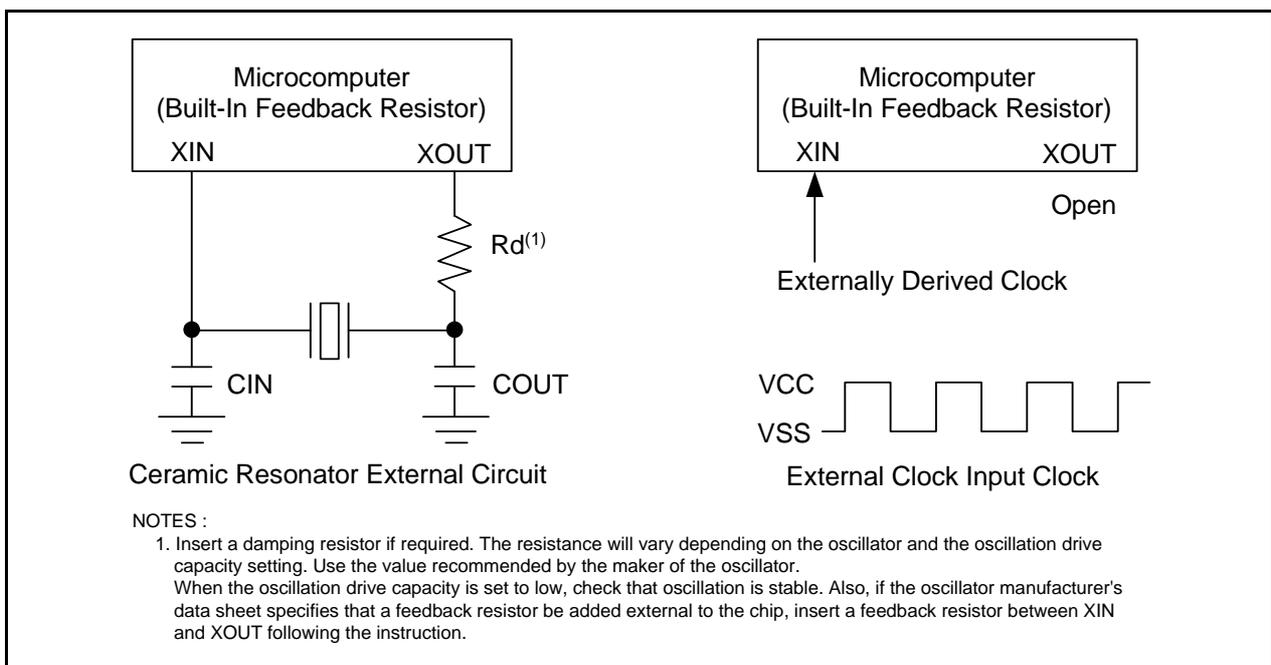


Figure 9.7 Examples of Main Clock Connection Circuit

9.2 On-Chip Oscillator Clock

This clock is supplied by an on-chip oscillator. The on-chip oscillator contains a high-speed on-chip oscillator and a low-speed on-chip oscillator. Either an on-chip oscillator clock is selected by the HRA01 bit in the HRA0 register.

9.2.1 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, fRING, fRING128 and fRING-S.

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

If the main clock stops oscillating when the OCD1 to OCD0 bits in the OCD register are set to "11b" (oscillation stop detection function enabled), the low-speed on-chip oscillator automatically starts operating, supplying the necessary clock for the microcomputer.

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

9.2.2 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, fRING, fRING128, and fRING1-fast.

After reset, the on-chip oscillator clock generated by the high-speed on-chip oscillator stops. The oscillation starts by setting the HRA00 bit in the HRA0 register to "1" (high-speed on-chip oscillator on). The frequency can be adjusted by the HRA1 and HRA2 registers.

Since the difference in delay between the bits, adjust by changing each bit.

9.3 CPU Clock and Peripheral Function Clock

There are two type clocks: a CPU clock to operate the CPU and a peripheral function clock to operate the peripheral functions. Refer to Figure 9.1 Clock Generation Circuit.

9.3.1 System Clock

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

9.3.2 CPU Clock

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

The system clock can be the divide-by-1 (no division), 2, 4, 8 or 16 to produce the CPU clock. Use the CM06 bit in the CM0 register and the CM16 to CM17 bits in the CM1 register to select the value of the division.

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

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

9.3.3 Peripheral Function Clock (f1, f2, f4, f8, f32)

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

The clock f_i ($i=1, 2, 4, 8, 32$) is generated by the system clock divided-by- i . The clock f_i is used for timers X, Y, Z, C, serial interface and A/D converter.

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 stops.

9.3.4 fRING and fRING128

fRING and fRING128 are operating clocks for the peripheral functions.

The fRING runs at the same frequency as the on-chip oscillator clock and can be used as the source for the timer X. The fRING128 is generated by the fRING by dividing it by 128 and can be used for the timer C.

When the WAIT instruction is executed, the clocks fRING and fRING128 do not stop.

9.3.5 fRING-fast

fRING-fast is used as the count source for the timer C. The fRING-fast is generated by the high-speed on-chip oscillator and provided by setting the HRA00 bit to "1".

When the WAIT instruction is executed, the clock fRING-fast does not stop.

9.3.6 fRING-S

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

9.4 Power Control

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

9.4.1 Normal Operating Mode

Normal operating mode is further separated into four modes.

In normal 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 after switching needs to be stabilized and oscillated. If the new clock source is the main clock, allow sufficient wait time in a program until an oscillation is stabilized before exiting.

Table 9.2 Setting and Mode of Clock Associated Bit

| Modes | | OCD Register | CM1 Register | | CM0 Register | |
|--|--------------|--------------|--------------|------|--------------|------|
| | | OCD2 | CM17, CM16 | CM13 | CM06 | CM05 |
| High-Speed Mode | | 0 | 00b | 1 | 0 | 0 |
| Medium-Speed Mode | divide-by-2 | 0 | 01b | 1 | 0 | 0 |
| | divide-by-4 | 0 | 10b | 1 | 0 | 0 |
| | divide-by-8 | 0 | – | 1 | 1 | 0 |
| | divide-by-16 | 0 | 11b | 1 | 0 | 0 |
| High-Speed, Low-Speed On-Chip Oscillator Mode ⁽¹⁾ | no division | 1 | 00b | – | 0 | – |
| | divide-by-2 | 1 | 01b | – | 0 | – |
| | divide-by-4 | 1 | 10b | – | 0 | – |
| | divide-by-8 | 1 | – | – | 1 | – |
| | divide-by-16 | 1 | 11b | – | 0 | – |

NOTES:

1. The low-speed on-chip oscillator is used as the on-chip oscillator clock when the CM14 bit in the CM1 register is set to “0” (low-speed on-chip oscillator on) and the HRA01 bit in the HRA0 register is set to “0”. The high-speed on-chip oscillator is used as the on-chip oscillator clock when the HRA00 bit in the HRA0 register is set to “1” (high-speed on-chip oscillator A on) and the HRA01 bit in the HRA0 register is set to “1”.

9.4.1.1 High-Speed Mode

The main clock divided-by-1 (no division) provides the CPU clock. If the CM14 bit is set to "0" (low-speed on-chip oscillator on) or the HRA00 bit in the HRA0 register is set to "1" (high-speed on-chip oscillator on), the fRING and fRING128 can be used for timers X and C. When the HRA00 bit is set to "1", fRING-fast can be used for timer C. When the CM14 bit is set to "0" (low-speed on-chip oscillator on), fRING-S can be used for the watchdog timer and voltage detection circuit.

9.4.1.2 Medium-Speed Mode

The main clock divided-by-2, -4, -8 or -16 provides the CPU clock. If the CM14 bit is set to "0" (low-speed on-chip oscillator on) or the HRA00 bit in the HRA0 register is set to "1" (high-speed on-chip oscillator on), the fRING and fRING128 can be used for timers X and C. When the HRA00 bit is set to "1", fRING-fast can be used for timer C. When the CM14 bit is set to "0" (low-speed on-chip oscillator on), fRING-S can be used for the watchdog timer and voltage detection circuit.

9.4.1.3 High-Speed, Low-Speed On-Chip Oscillator Mode

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. When the HRA00 bit is set to "1", fRING-fast can be used for timer C. When the CM14 bit is set to "0" (low-speed on-chip oscillator on), fRING-S can be used for the watchdog timer and voltage detection circuit.

9.4.2 Wait Mode

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

9.4.2.1 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. The power consumption can be reduced.

9.4.2.2 Entering Wait Mode

The microcomputer enters wait mode by executing the WAIT instruction.

9.4.2.3 Pin Status in Wait Mode

The status before entering wait mode is maintained.

9.4.2.4 Exiting Wait Mode

The microcomputer exits wait mode by a hardware reset or peripheral function interrupt. When using a hardware reset to exit wait mode, set the ILVL2 to ILVL0 bits for the peripheral function interrupts to "000b" (interrupts disabled) before executing the WAIT instruction.

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 can be used to exit wait mode.

Table 9.3 lists Interrupts to Exit Wait Mode and Usage Conditions.

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 to the ILVL2 to ILVL0 bits in the interrupt control register of the peripheral function interrupts to use for exiting wait mode. Set the ILVL2 to ILVL0 bits of the peripheral function interrupts not to use for exiting wait mode to "000b" (disables interrupt).
- (2) Set the I flag to "1".
- (3) Operate the peripheral function to use for exiting wait mode.

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

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.

Table 9.3 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 |
| SSU 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 X Interrupt | Usable in all modes | Usable in event counter mode |
| Timer Z Interrupt | Usable in all modes | –(Do not use) |
| Timer C Interrupt | Usable in all modes | –(Do not use) |
| $\overline{\text{INT}}$ Interrupt | Usable | Usable ($\overline{\text{INT0}}$ and $\overline{\text{INT3}}$ can be used if there is no filter. |
| Voltage Monitor 2 Interrupt | Usable | Usable |
| Oscillation Stop Detection Interrupt | Usable | –(Do not use) |
| Watchdog Timer Interrupt | Usable in count source protect mode | Usable in count source protect mode |

9.4.3 Stop Mode

Since the oscillator circuits stop in wait mode, the CPU clock and peripheral function clock stop and the CPU and peripheral functions clocked by these clocks stop operating. The least power required to operate the microcomputer is in stop mode. If the voltage applied to the VCC pin is VRAM or more, the internal RAM is maintained.

The peripheral functions clocked by external signals maintain operating. Table 9.4 lists Interrupts to Exit Stop Mode and Usage Conditions.

Table 9.4 Interrupts to Exit Stop Mode and Usage Conditions

| Interrupt | Usage Conditions |
|---|---|
| Key Input Interrupt | – |
| $\overline{\text{INT0}}$ to $\overline{\text{INT1}}$ Interrupts | $\overline{\text{INT0}}$ can be used if there is no filter |
| $\overline{\text{INT3}}$ Interrupt | No filter. Interrupt request is generated at $\overline{\text{INT3}}$ input. (TCC06 bit in TCC0 register is set to “1”) |
| Timer X Interrupt | When external pulse is counted in event counter mode |
| Serial Interface Interrupt | When external clock is selected |
| Voltage Monitor 2 Interrupt | Usable in digital filter disabled mode (VW2C1 bit in VW2C register is set to “1”) |

9.4.3.1 Entering Stop Mode

The microcomputer enters stop mode by setting the CM10 bit in the CM1 register 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 CM10 register is set to “1” (drive capability HIGH of main clock oscillator circuit).

When using stop mode, set the OCD1 to OCD0 bits to “00b” (oscillation stop detection function disabled) before entering stop mode.

9.4.3.2 Pin Status in Stop Mode

The status before entering wait mode 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 port P4_6 and P4_7), the P4_7(XOUT) is held in input status.

9.4.3.3 Exiting Stop Mode

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

When using a hardware reset to exit stop mode, set the ILVL2 to ILVL0 bits for the peripheral function interrupts to “000b” (disables interrupts) before setting the CM10 bit to “1”.

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 to the ILVL2 to ILVL0 bits of the peripheral function interrupts to use for exiting stop mode. Set the ILVL2 to ILVL0 bits of the peripheral function interrupts not to use for exiting stop mode to “000b” (disables interrupt).
- (2) Set the I flag to “1”.
- (3) Operates the peripheral function to use for exiting stop mode.

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

The CPU clock, when exiting stop mode by a peripheral function interrupt, is the divide-by-8 of the clock which is used before entering stop mode.

Figure 9.8 shows the State Transition of Power Control.

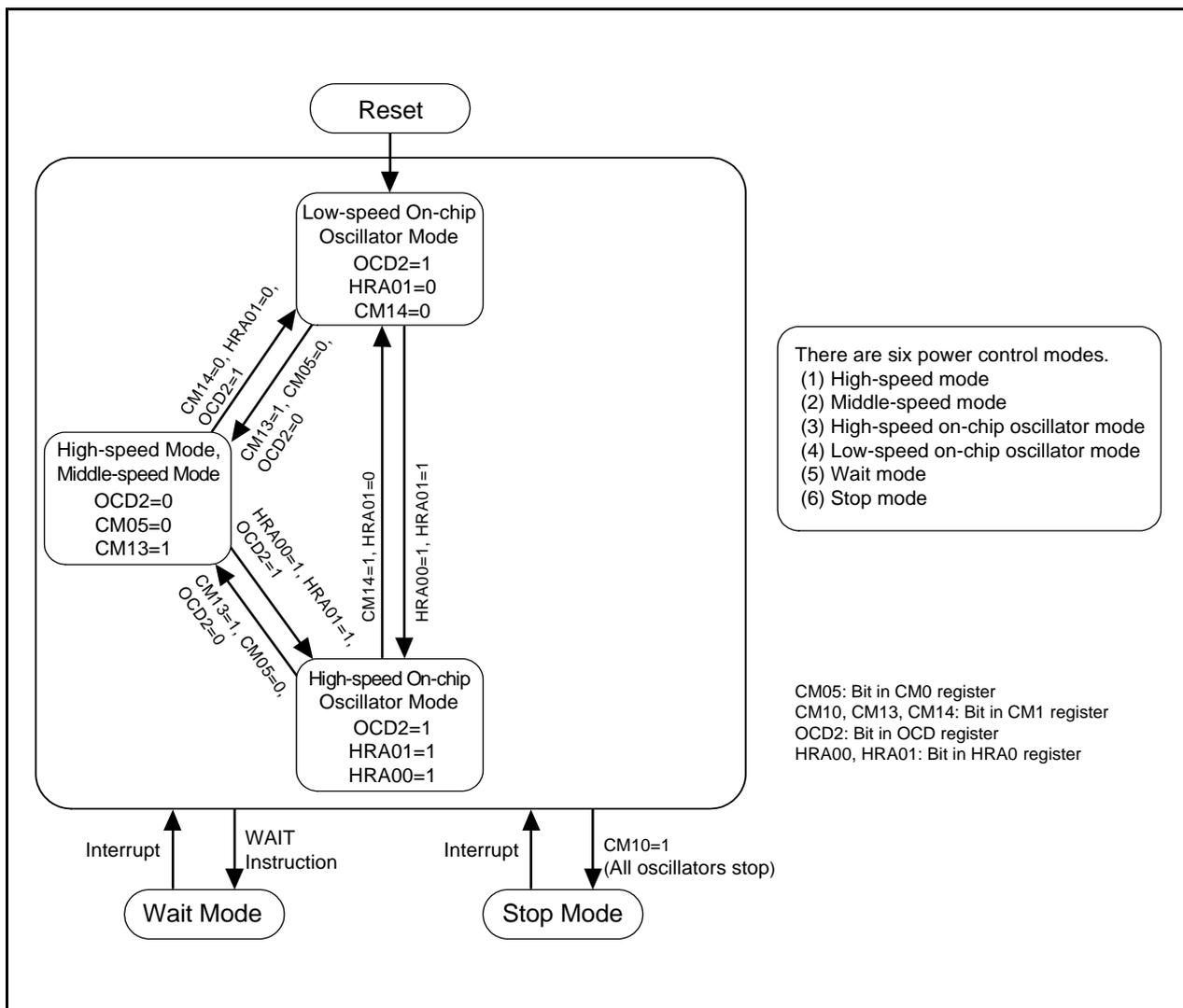


Figure 9.8 State Transition of Power Control

9.5 Oscillation Stop Detection Function

The oscillation stop detection function is a function to detect the stop of the main clock oscillating circuit. The oscillation stop detection function can be enabled and disabled by the OCD1 to OCD0 bits in the OCD register.

Table 9.5 lists the Specification of Oscillation Stop Detection Function.

When the main clock is the CPU clock source and the OCD1 to OCD0 bits are set to "11b" (oscillation stop detection function enabled), the system is placed in the following state if the main clock stops.

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

Table 9.5 Specification of Oscillation Stop Detection Function

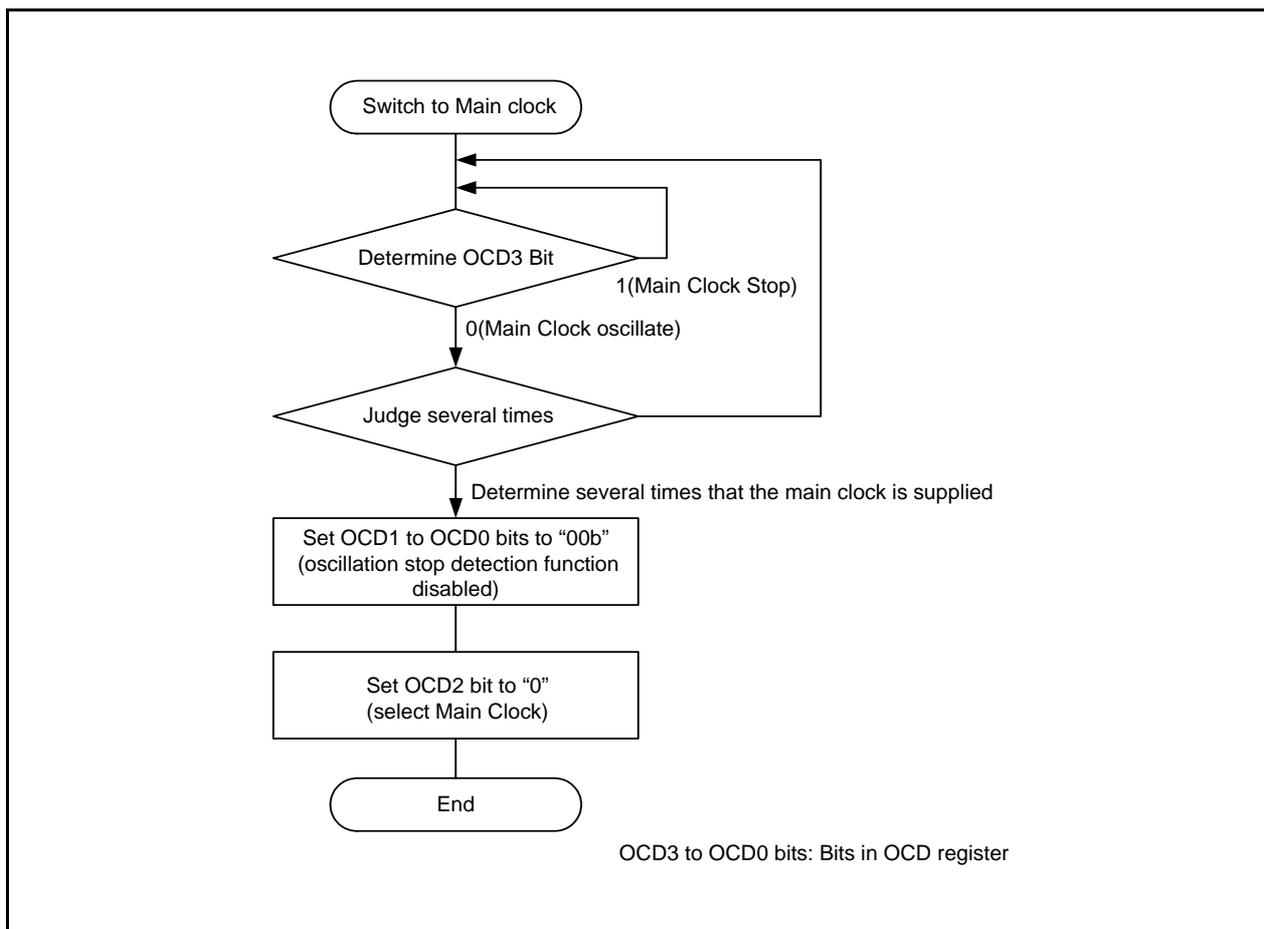
| Item | Specification |
|---|--|
| Oscillation Stop Detection Enable Clock and Frequency Bandwidth | $f(XIN) \geq 2 \text{ MHz}$ |
| Enabled Condition for Oscillation Stop Detection Function | Set OCD1 to OCD0 bits to "11b" (oscillation stop detection function enabled) |
| Operation at Oscillation Stop Detection | Oscillation stop detection interrupt is generated |

9.5.1 How to Use Oscillation Stop Detection Function

- The oscillation stop detection interrupt shares the vector with the voltage monitor 2 interrupt and the watchdog timer interrupt. When using the oscillation stop detection interrupt and watchdog timer interrupt, the interrupt cause needs to be determined. Table 9.6 lists the Determine Interrupt Factor of Oscillation Stop Detection, Watchdog Timer and Voltage Monitor 2 Interrupts.
- When the main clock is re-oscillated after oscillation stop, switch the main clock to the clock source of the CPU clock and peripheral functions by a program.
- Figure 9.9 shows the Procedure of Switching Clock Source From Low-Speed On-Chip Oscillator to Main 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 preparing to stop the main clock by the external cause, set the OCD1 to OCD0 bits to "00b" (oscillation stop detection function disabled) when the main clock stops or oscillates in the program, that is stop mode is selected or the CM05 bit is changed.
- This function cannot be used when the main clock frequency is below 2 MHz. Set the OCD1 to OCD0 bits to "00b" (oscillation stop detection function disabled).
- When using the low-speed on-chip oscillator clock for the CPU clock and clock sources of peripheral functions after detecting the oscillation stop, set the HRA01 bit in the HRA0 register to "0" (low-speed on-chip oscillator selected) and the OCD1 to OCD0 bits to "11b" (oscillation stop detection function enabled).
When using the high-speed on-chip oscillator clock for the CPU clock and clock sources of peripheral functions after detecting the oscillation stop, set the HRA01 bit to "1" (high-speed on-chip oscillator selected) and the OCD1 to OCD0 bits to "11b" (oscillation stop detection function enabled).

Table 9.6 Determine Interrupt Factor of Oscillation Stop Detection, Watchdog Timer and Voltage Monitor 2 Interrupts

| Generated Interrupt Cause | 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 the OCD2 bit = 1 |
| Watchdog Timer | VW2C3 bit in VW2C register = 1 |
| Voltage Monitor 2 | VW2C2 bit in VW2C register = 1 |

**Figure 9.9 Procedure of Switching Clock Source From Low-Speed On-Chip Oscillator to Main Clock**

10. Protection

Protection function protects important registers from being easily overwritten when a program runs out of control. Figure 10.1 shows the PRCR Register. The following lists the registers protected by the PRCR register.

- Registers protected by PRC0 bit : CM0, CM1, and OCD, HRA0, HRA1, HRA2 registers
- Registers protected by PRC1 bit : PM0 and PM1 registers
- Registers protected by PRC3 bit : VCA2, VW1C and VW2C registers

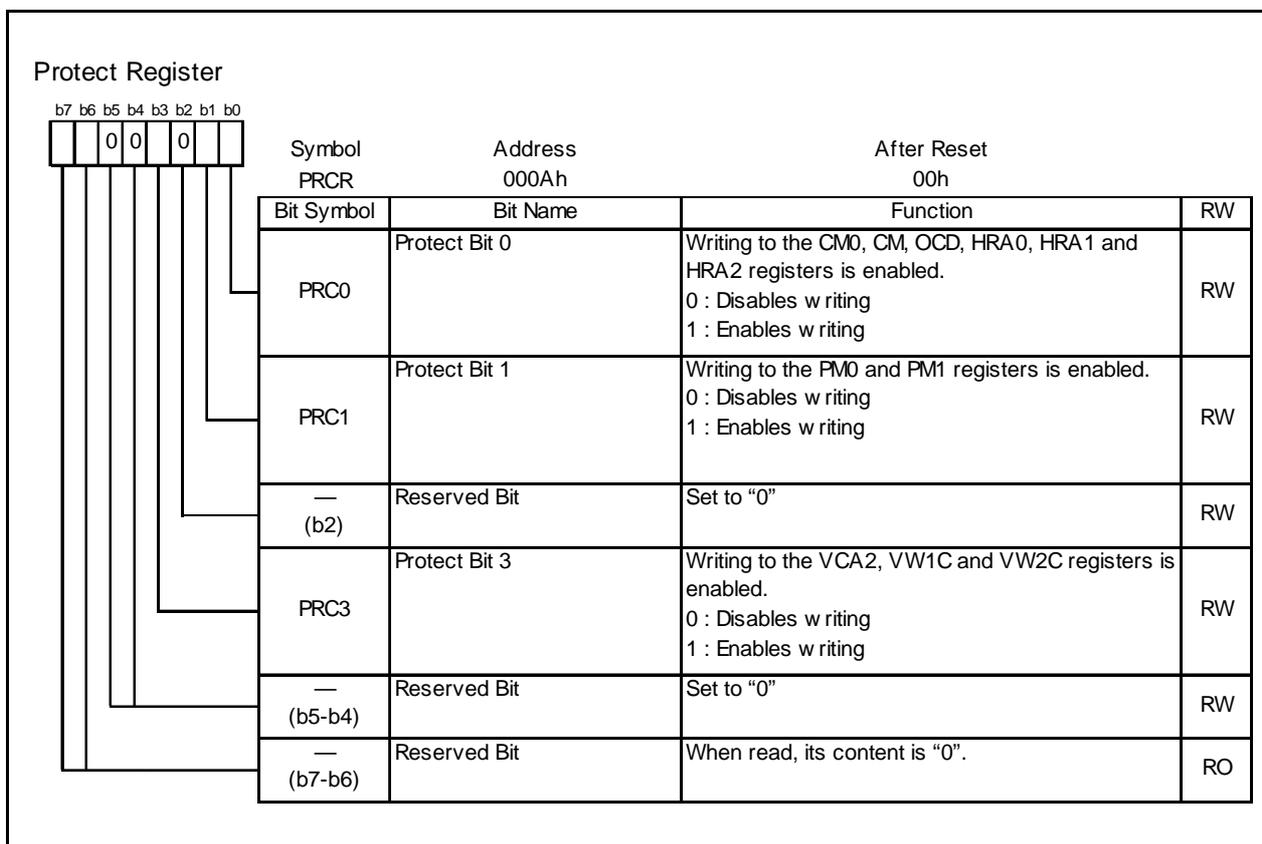


Figure 10.1 PRCR Register

11. Interrupt

11.1 Interrupt Overview

11.1.1 Types of Interrupts

Figure 11.1 shows types of Interrupts.

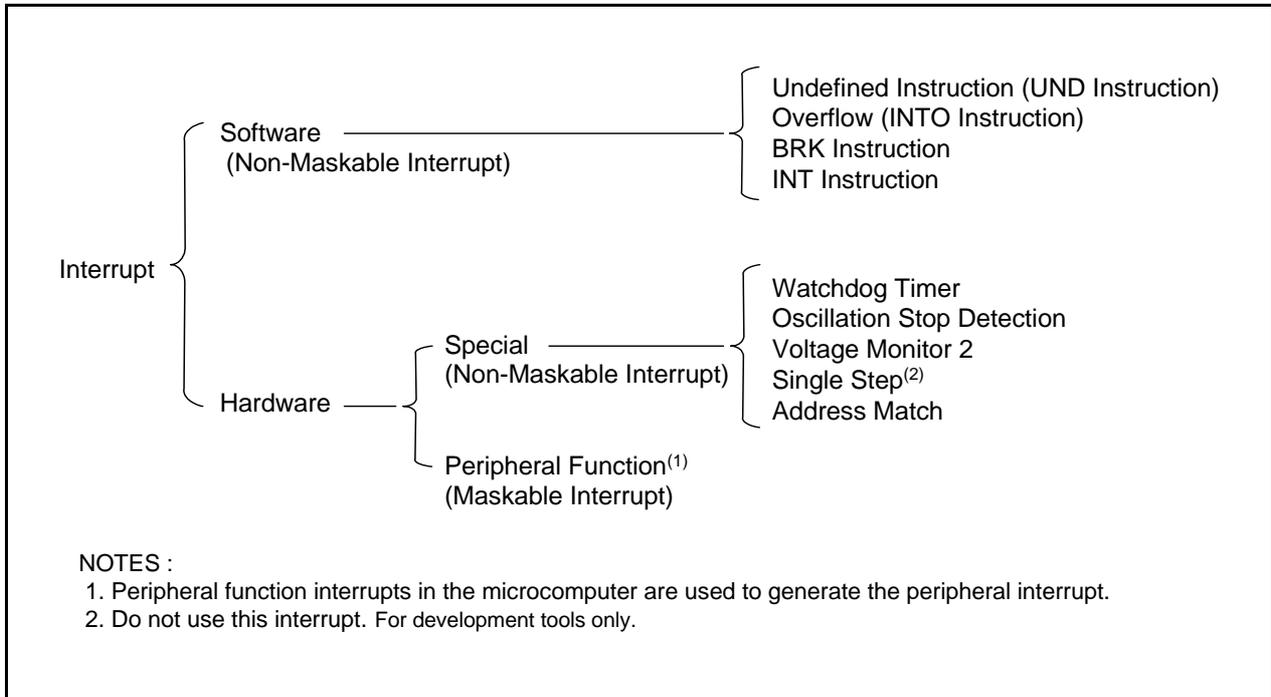


Figure 11.1 Interrupts

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

11.1.2 Software Interrupts

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

11.1.2.1 Undefined Instruction Interrupt

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

11.1.2.2 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 to set the O flag are:

ABS, ADC, ADCF, ADD, CMP, DIV, DIVU, DIVX, NEG, RMPA, SBB, SHA, SUB

11.1.2.3 BRK Interrupt

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

11.1.2.4 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 4 to 31 are assigned to the peripheral function interrupt. Therefore, the microcomputer executes the same interrupt routine when the INT instruction is executed as when a peripheral function interrupt is generated. In software interrupt numbers 0 to 31, the U flag is saved to the stack during instruction execution and set the U flag to "0" (ISP selected) before executing an interrupt sequence. The U flag is restored from the stack when returning from the interrupt routine. In software interrupt numbers 32 to 63, the U flag does not change state during instruction execution, and the selected SP is used.

11.1.3 Special Interrupts

Special interrupts are non-maskable interrupts.

11.1.3.1 Watchdog Timer Interrupt

The watchdog timer interrupt is generated by the watchdog timer. Reset the watchdog timer after the watchdog timer interrupt is generated. For details, refer to **12. Watchdog Timer**.

11.1.3.2 Oscillation Stop Detection Interrupt

Oscillation Stop Detection Interrupt is generated by the oscillation stop detection function. For details of the oscillation stop detection function, refer to **9. Clock Generation Circuit**.

11.1.3.3 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**.

11.1.3.4 Single-Step Interrupt, Address Break Interrupt

Do not use the single-step interrupt. For development tools only.

11.1.3.5 Address Match Interrupt

The address match interrupt is generated immediately before executing an instruction that is stored into an address indicated by the RMAD0 to RMAD1 registers when the AIER0 or AIER1 bit in the AIER register which is set to "1" (address match interrupt enable). For details of the address match interrupt, refer to **11.4 Address Match Interrupt**.

11.1.4 Peripheral Function Interrupt

The peripheral function interrupt is generated by the internal peripheral function of the microcomputer and a maskable interrupt. Refer to **Table 11.2 Relocatable Vector Tables** for the interrupt factor of the peripheral function interrupt. For details of the peripheral function, refer to the description of each peripheral function.

11.1.5 Interrupts and Interrupt Vector

There are 4 bytes in one vector. Set the starting address of interrupt routine in each vector table. When an interrupt request is acknowledged, the CPU branches to the address set in the corresponding interrupt vector. Figure 11.2 shows the Interrupt Vector.

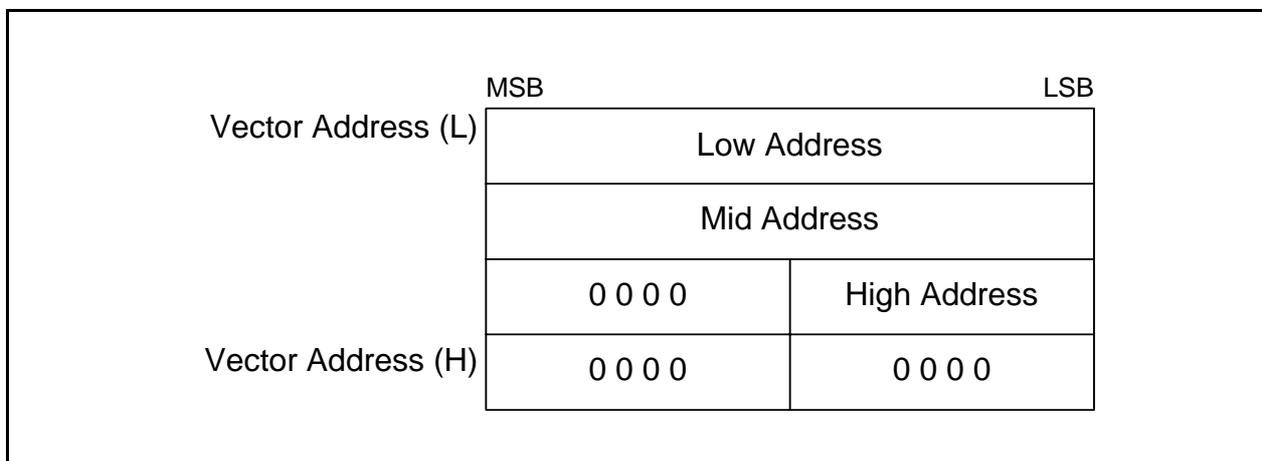


Figure 11.2 Interrupt Vector

11.1.5.1 Fixed Vector Tables

The fixed vector tables are allocated addresses 0FFDCh to 0FFFFh. Table 11.1 lists the Fixed Vector Tables. The vector addresses (H) of fixed vectors are used by the ID code check function. For details, refer to **18.3 Functions To Prevent Flash Memory from Rewriting**.

Table 11.1 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 | | 11.4 Address Match Interrupt |
| Single Step ⁽¹⁾ | 0FFEC h to 0FFEFh | | |
| <ul style="list-style-type: none"> • Watchdog Timer • Oscillation Stop Detection • Voltage Monitor 2 | 0FFF0h to 0FFF3h | | <ul style="list-style-type: none"> • 12. Watchdog Timer • 9. Clock Generation Circuit • 6. Voltage Detection Circuit |
| Address Break ⁽¹⁾ | 0FFF4h to 0FFF7h | | |
| (Reserved) | 0FFF8h to 0FFFBh | | |
| Reset | 0FFFC h to 0FFFFh | | 5. Reset |

1. Do not use the single-step interrupt. For development tools only.

11.1.5.2 Relocatable Vector Tables

The relocatable vector tables occupy 256 bytes from the starting address set in the INTB register. Table 11.2 lists the Relocatable Vector Tables.

Table 11.2 Relocatable Vector Tables

| Interrupt Factor | Vector Address ⁽¹⁾ Address (L) to Address (H) | Software Interrupt Number | Reference |
|-----------------------------------|---|------------------------------|---|
| BRK Instruction ⁽²⁾ | +0 to +3(0000h to 0003h) | 0 | R8C/Tiny Series software manual |
| –(Reserved) | | 1 to 12 | |
| Key Input | +52 to +55(0034h to 0037h) | 13 | 11.3 Key Input Interrupt |
| A/D Converter | +56 to +59(0038h to 003Bh) | 14 | 16. A/D Converter |
| SSU | +60 to +63(003Ch to 003Fh) | 15 | 15. Clock Synchronous Serial I/O with Chip Select (SSU) |
| Compare 1 | +64 to +67(0040h to 0043h) | 16 | 13.3 Timer C |
| UART0 Transmit | +68 to +71(0044h to 0047h) | 17 | 14. Serial Interface |
| UART0 Receive | +72 to +75(0048h to 004Bh) | 18 | |
| –(Reserved) | | 19 | |
| –(Reserved) | | 20 | |
| –(Reserved) | | 21 | |
| Timer X | +88 to +91(0058h to 005Bh) | 22 | 13.1 Timer X |
| –(Reserved) | | 23 | |
| Timer Z | +96 to +99(0060h to 0063h) | 24 | 13.2 Timer Z |
| $\overline{\text{INT}}1$ | +100 to +103(0064h to 0067h) | 25 | 11.2 $\overline{\text{INT}}$ interrupt |
| $\overline{\text{INT}}3$ | +104 to +107(0068h to 006Bh) | 26 | |
| Timer C | +108 to +111(006Ch to 006Fh) | 27 | 13.3 Timer C |
| Compare 0 | +112 to +115(0070h to 0073h) | 28 | |
| $\overline{\text{INT}}0$ | +116 to +119(0074h to 0077h) | 29 | 11.2 $\overline{\text{INT}}$ interrupt |
| –(Reserved) | | 30 | |
| –(Reserved) | | 31 | |
| Software Interrupt ⁽²⁾ | +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 I flag does not disable these interrupts.

11.1.6 Interrupt Control

The following describes enable/disable the maskable interrupts and set the priority order to acknowledge. The contents explained does not apply to the nonmaskable interrupts.

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

Figure 11.3 shows the Interrupt Control Register and Figure 11.4 shows the INT0IC Register

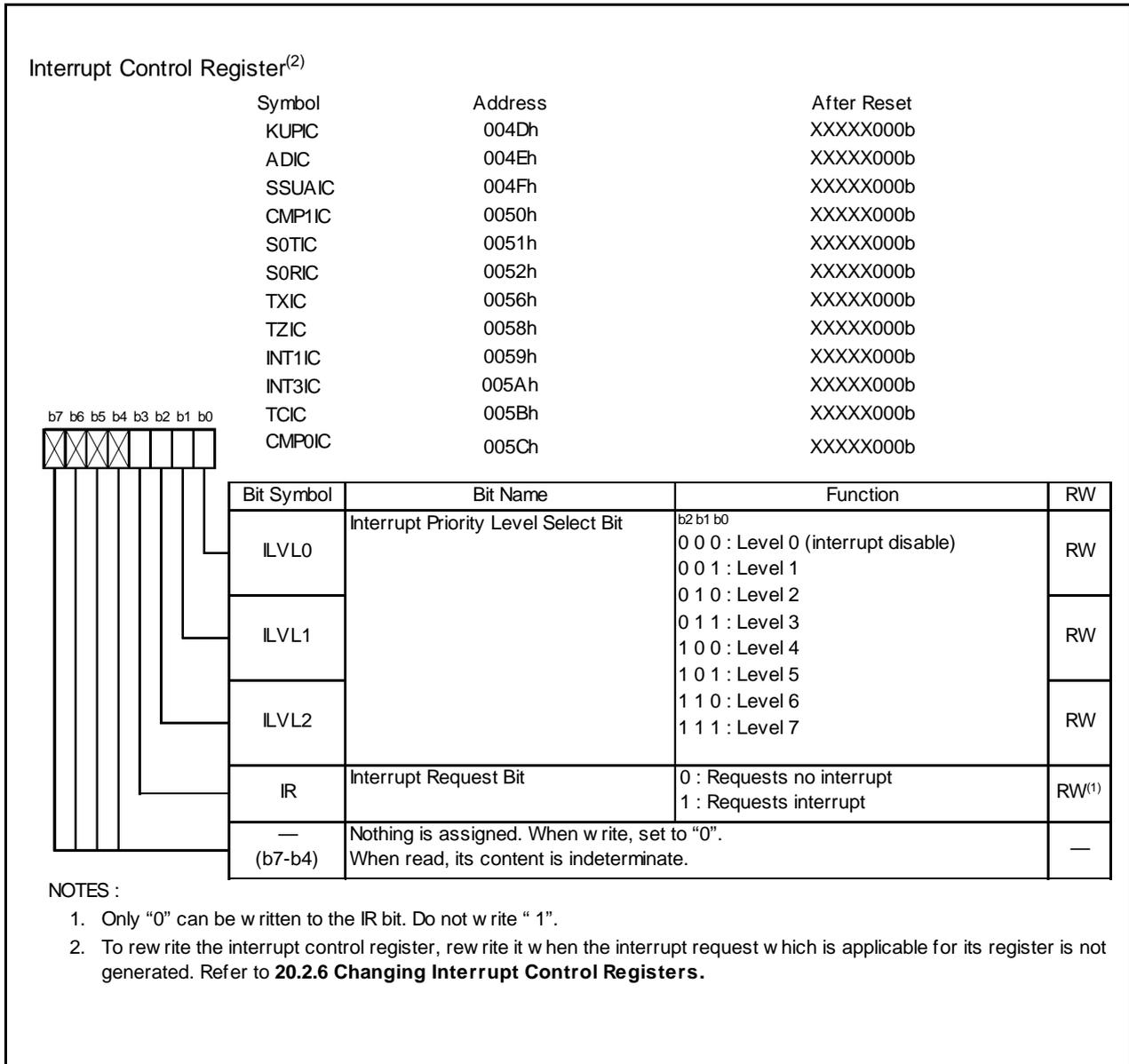


Figure 11.3 Interrupt Control Register

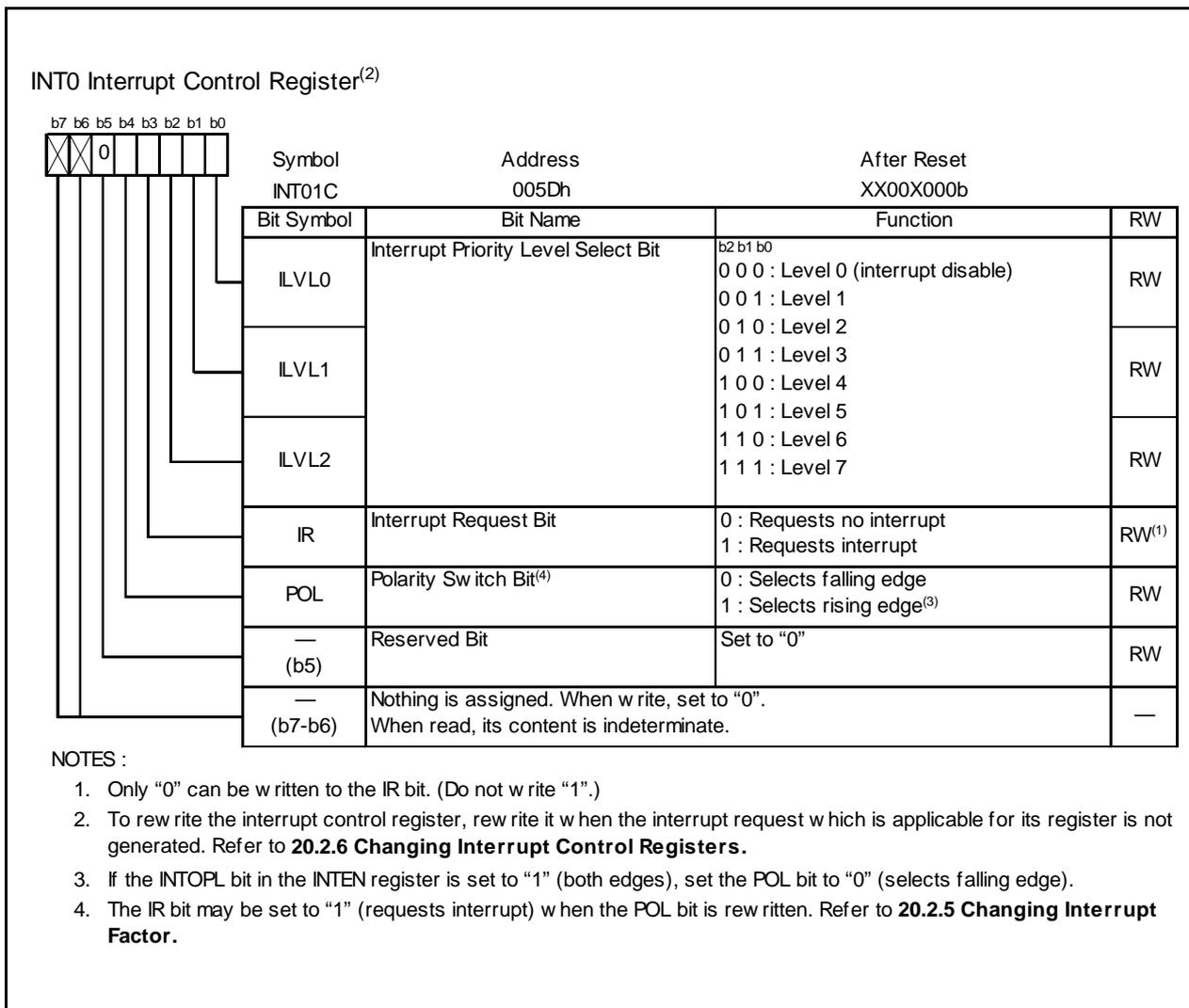


Figure 11.4 INT0IC Register

11.1.6.1 I Flag

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

11.1.6.2 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.

11.1.6.3 ILVL2 to ILVL0 Bits and IPL

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

Table 11.3 lists the Settings of Interrupt Priority Levels and Table 11.4 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, ILVL2 to ILVL0 bits and IPL are independent of each other. They do not affect one another.

Table 11.3 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 11.4 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 |

11.1.6.4 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 executed, the CPU determines its interrupt priority level after the instruction is completed. The CPU starts the interrupt sequence from the following cycle. However, in regards to the SMOVB, SMOVF, SSTR or RMPA instruction, if an interrupt request is generated while executing the instruction, the microcomputer suspends the instruction to start the interrupt sequence. The interrupt sequence is performed as follows. Figure 11.5 shows the Time Required for Executing Interrupt Sequence.

- (1) The CPU gets interrupt information (interrupt number and interrupt request level) by reading the address 00000h. The IR bit for the corresponding interrupt is set to "0" (interrupt not requested).
- (2) The FLG register immediately before entering the interrupt sequence is saved to the CPU internal temporary register⁽¹⁾.
- (3) The I, D and U flags in the FLG register are set as follows:
The I flag is set to "0" (disables interrupts).
The D flag is set to "0" (disables single-step interrupt).
The U flag is set to "0" (ISP selected).
However, the U flag does not change state if an INT instruction for software interrupt numbers 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, the instructions are executed from the starting address of the interrupt routine.

NOTES:

1. This register cannot be used by user.

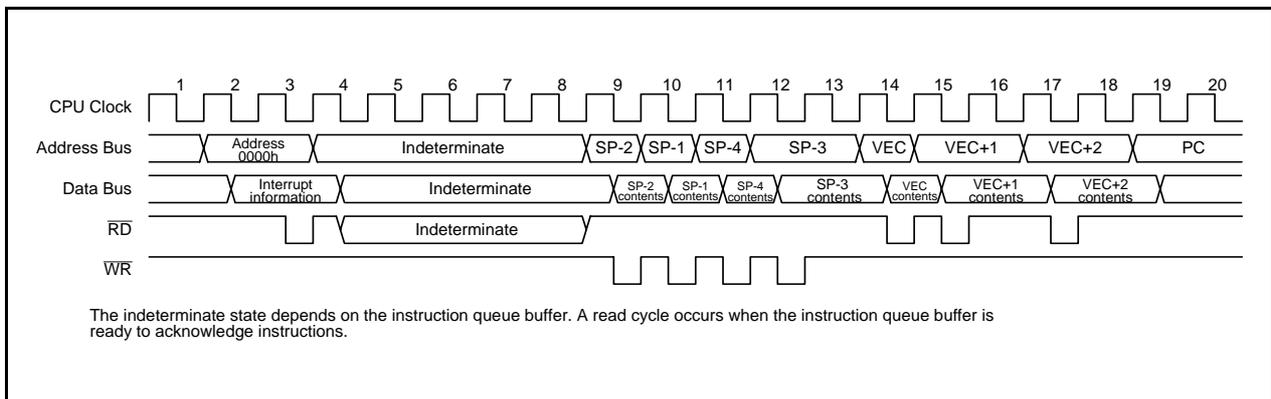


Figure 11.5 Time Required for Executing Interrupt Sequence

11.1.6.5 Interrupt Response Time

Figure 11.6 shows an Interrupt Response Time. The interrupt response time is the period between an interrupt request generation and the execution of the first instruction in an interrupt routine. An interrupt response time includes the period between an interrupt request generation and the completed execution of an instruction (see #a in Figure 11.6) and the period required to perform an interrupt sequence (20 cycles, see #b in Figure 11.6).

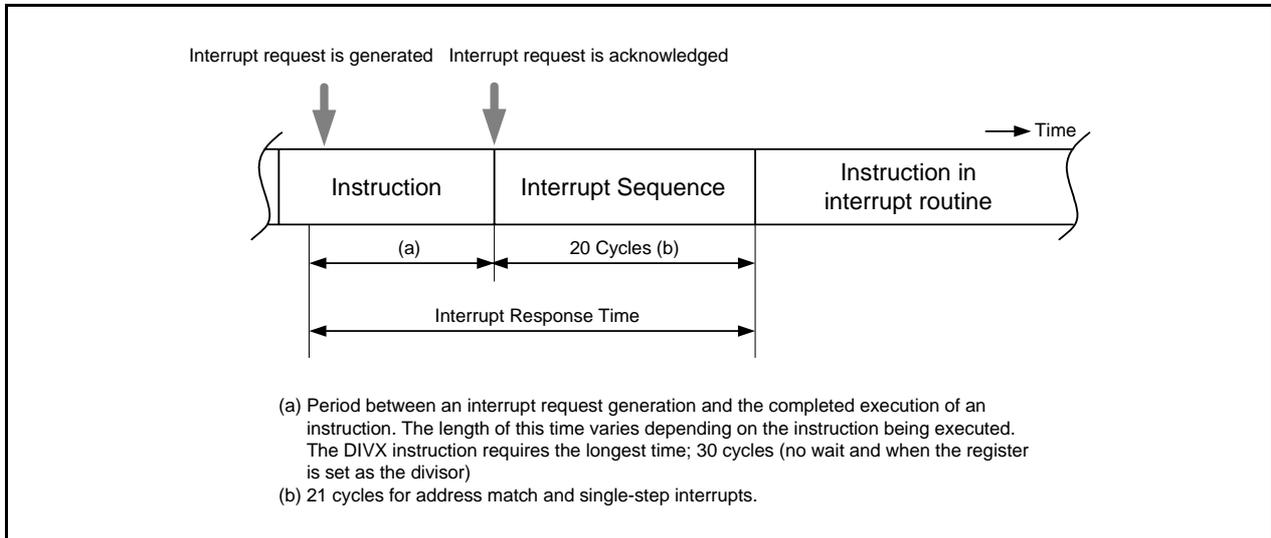


Figure 11.6 Interrupt Response Time

11.1.6.6 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 and special interrupt request are acknowledged, the level listed in Table 11.5 is set to the IPL. Table 11.5 lists the IPL Value When Software or Special Interrupts Is Acknowledged.

Table 11.5 IPL Value When Software or Special Interrupts Is Acknowledged

| Interrupt Factor | Value Set to IPL |
|---|------------------|
| Watchdog Timer, Oscillation Stop Detection, Voltage Monitor 2 | 7 |
| Software, Address Match, Single-Step | Not changed |

11.1.6.7 Saving a Register

In the interrupt sequence, the FLG register and PC are saved to the stack. After 4 high-order bits in the PC and 4 high-order (IPL) and 8 low-order bits in the FLG register, extended to 16 bits, are saved to the stack, the 16 low-order bits in the PC are saved. Figure 11.7 shows the Stack State Before and After Acknowledgement of Interrupt Request. The other necessary registers are saved by a program at the beginning of the interrupt routine. The PUSHM instruction can save several registers in the register bank being currently used⁽¹⁾ with 1 instruction.

NOTES:

1. Selectable from the R0, R1, R2, R3, A0, A1, SB and FB registers.

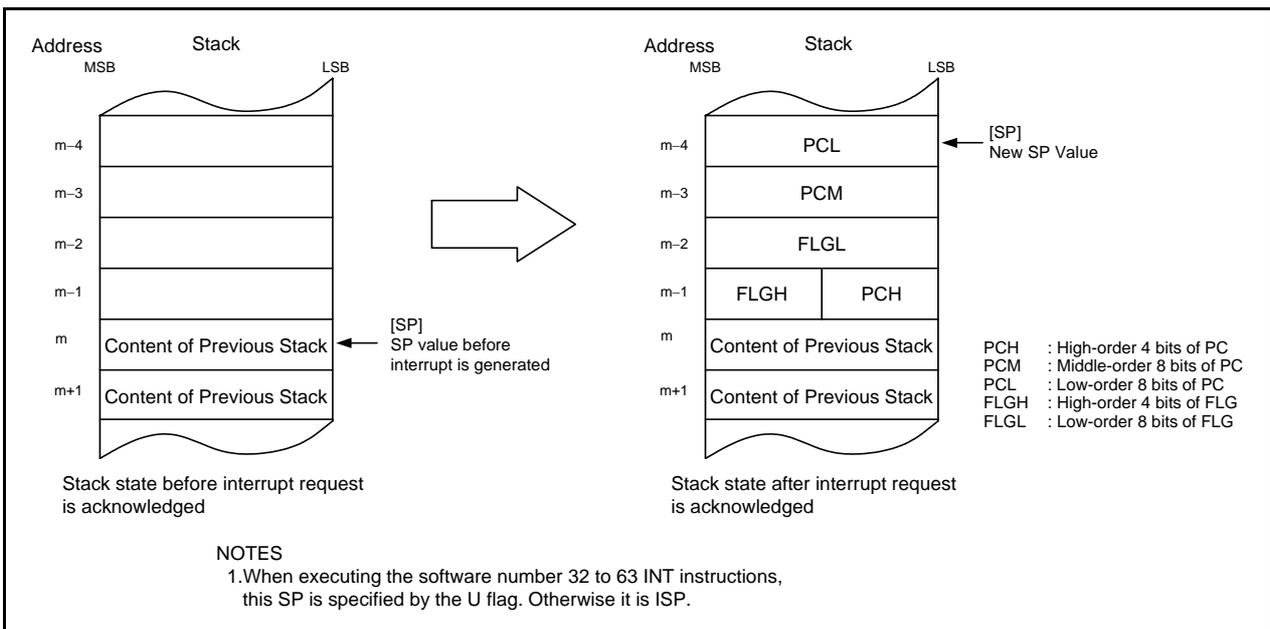


Figure 11.7 Stack State Before and After Acknowledgement of Interrupt Request

The register saving operation which is performed in the interrupt sequence is saved in 8 bits every 4 steps. Figure 11.8 shows the Operation of Saving Register.

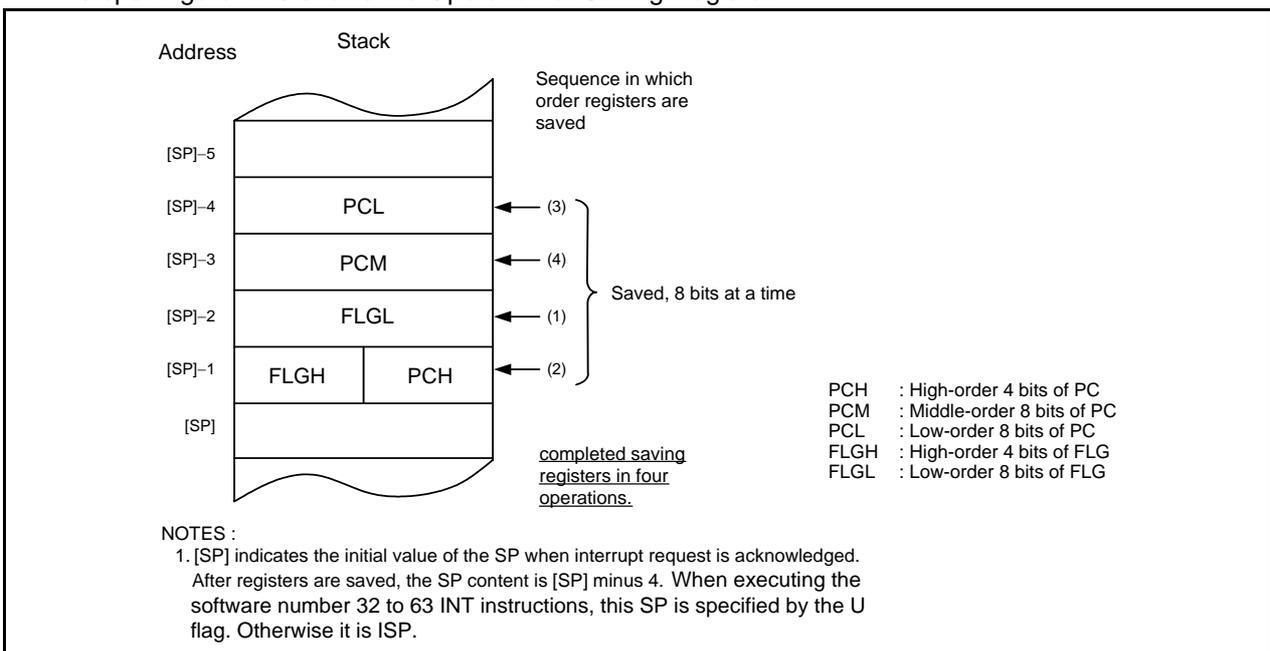


Figure 11.8 Operation of Saving Register

11.1.6.8 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 returned. The program, executed before the interrupt request has been acknowledged, starts running again.

Return the register saved by a program in an interrupt routine using the POPM instruction or others before the REIT instruction.

11.1.6.9 Interrupt Priority

If two or more interrupt requests are generated while executing one instruction, the interrupt with the higher priority is acknowledged.

Set the ILVL2 to ILVL0 bits 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, with the higher priority interrupt acknowledged in hardware.

The priority levels of special interrupts such as reset (reset has the highest priority) and watchdog timer are set by hardware. Figure 11.9 shows the Interrupt Priority Levels of Hardware Interrupt.

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

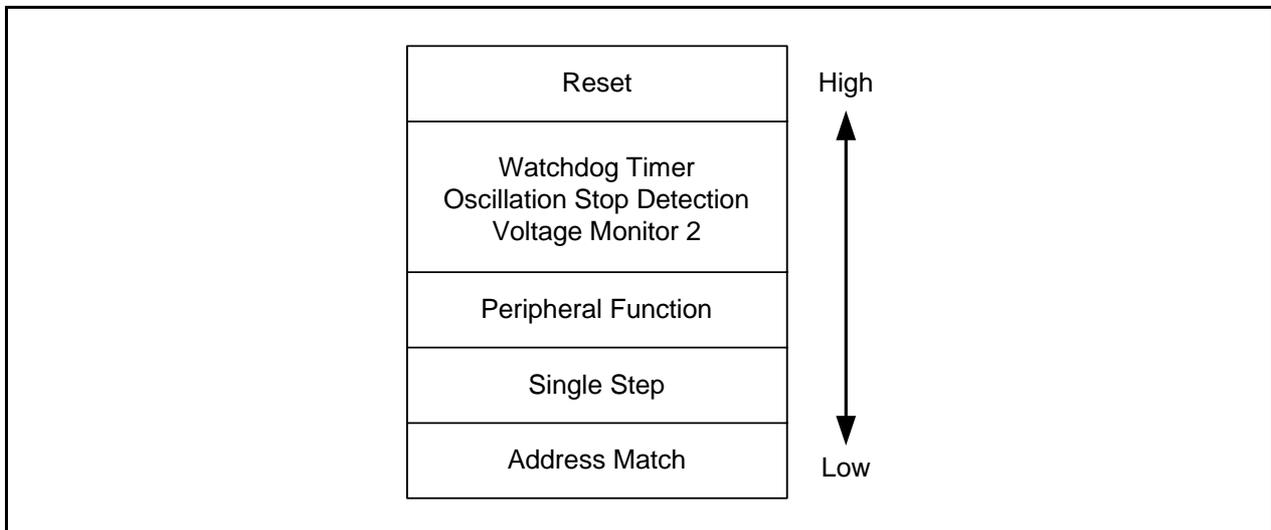


Figure 11.9 Interrupt Priority Levels of Hardware Interrupt

11.1.6.10 Interrupt Priority Judgement Circuit

The interrupt priority judgement circuit selects the highest priority interrupt. Figure 11.10 shows the Judgement Circuit of Interrupts Priority Level.

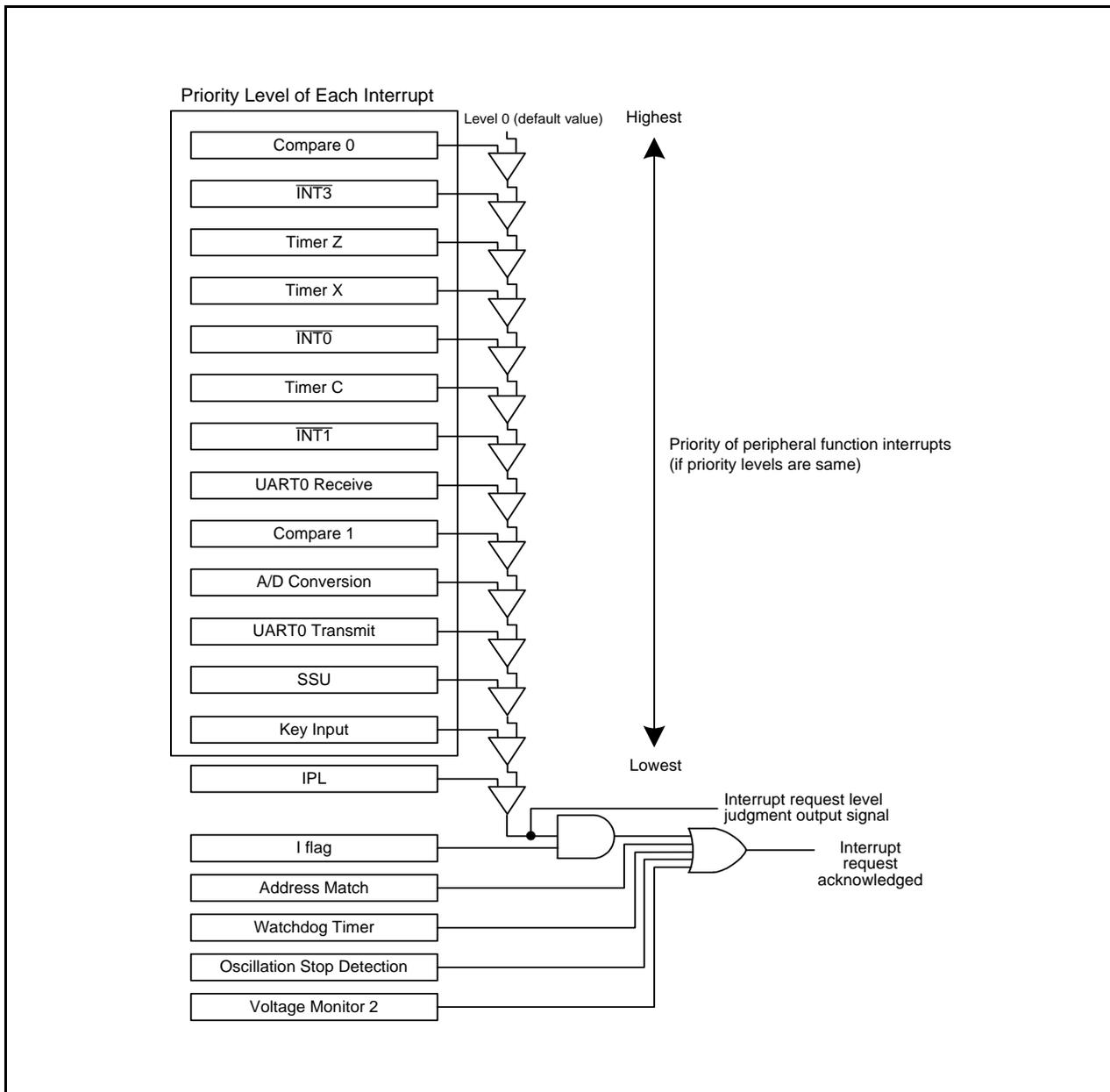


Figure 11.10 Judgement Circuit of Interrupts Priority Level

11.2 $\overline{\text{INT}}$ Interrupt

11.2.1 $\overline{\text{INT0}}$ Interrupt

The $\overline{\text{INT0}}$ interrupt is generated by an $\overline{\text{INT0}}$ input. When using the $\overline{\text{INT0}}$ interrupt, the INT0EN bit in the INTEN register is set to "1" (enable). The edge polarity is selected using the INT0PL bit in the INTEN register and the POL bit in the INT0IC register.

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

The $\overline{\text{INT0}}$ pin is shared with the external trigger input pin of timer Z.

Figure 11.11 shows the INTEN and INT0F Registers.

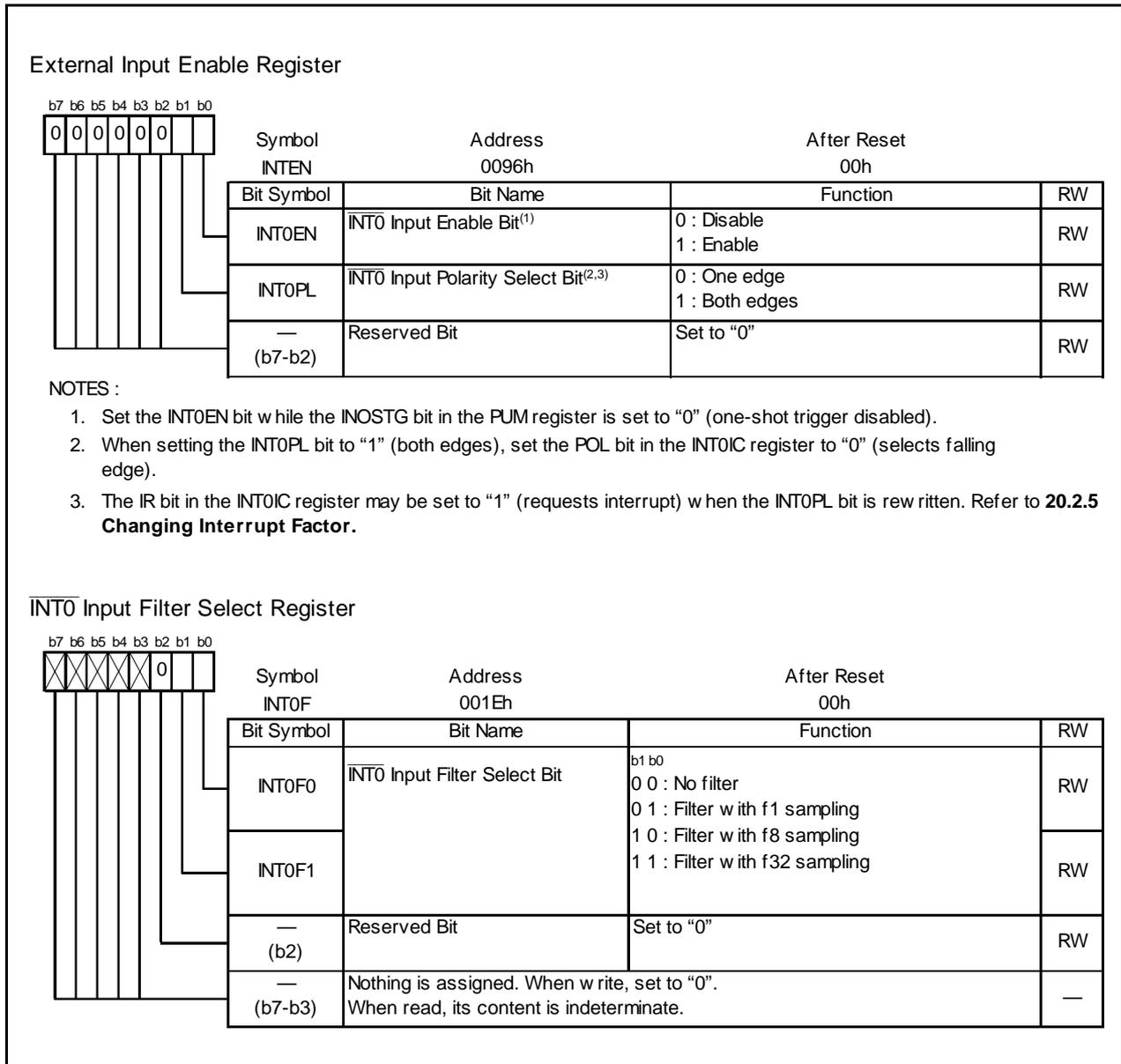


Figure 11.11 INTEN and INT0F Registers

11.2.2 $\overline{\text{INT0}}$ Input Filter

The $\overline{\text{INT0}}$ input contains a digital filter. The sampling clock is selected by the INT0F1 to INT0F0 bits in the INT0F register. The IR bit in the INTOIC register is set to "1" (interrupt requested) when the $\overline{\text{INT0}}$ level is sampled for every sampling clock and the sampled input level matches three times.

Figure 11.12 shows the Configuration of INT0 Input Filter. Figure 11.13 shows the Operating Example of INT0 Input Filter.

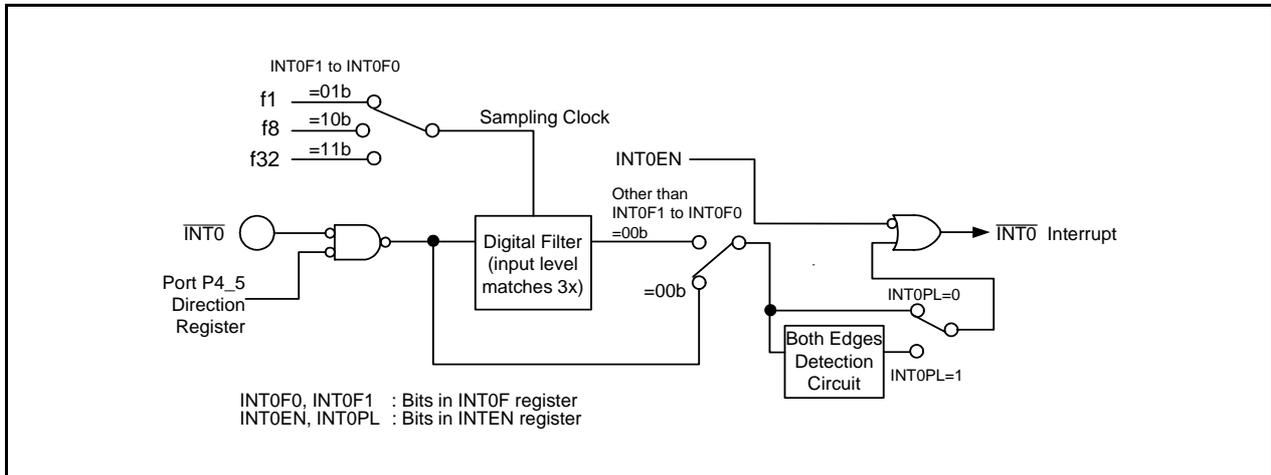


Figure 11.12 Configuration of $\overline{\text{INT0}}$ Input Filter

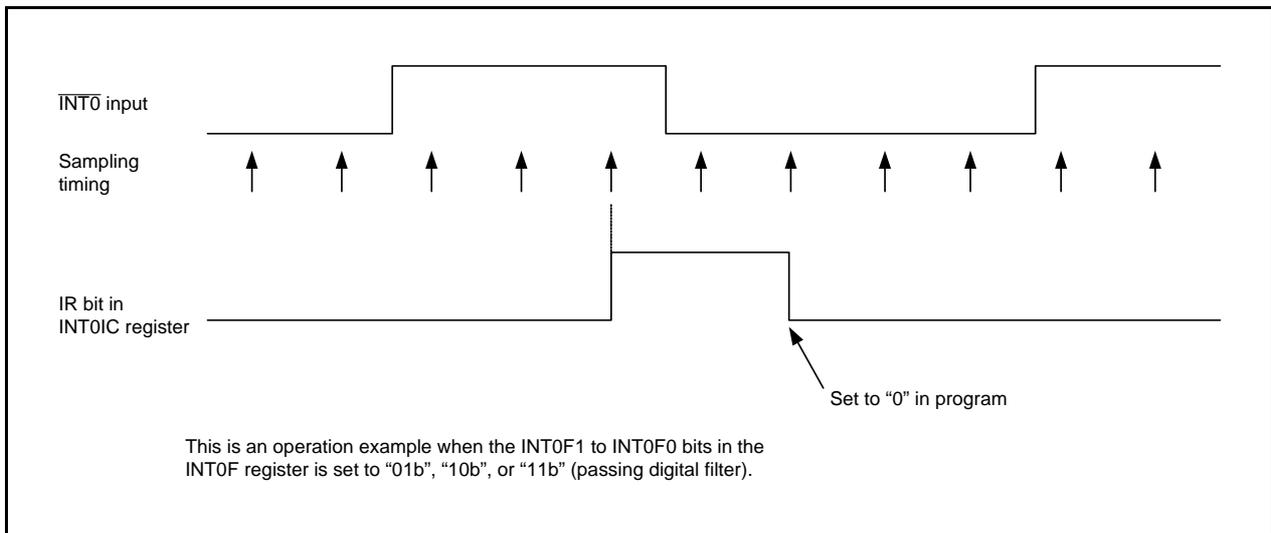


Figure 11.13 Operating Example of $\overline{\text{INT0}}$ Input Filter

11.2.3 $\overline{\text{INT1}}$ Interrupt

The $\overline{\text{INT1}}$ interrupt is generated by $\overline{\text{INT1}}$ inputs. The edge polarity is selected by the R0EDG bit in the TXMR register.

When the CNTRSEL bit in the UCON register is set to "0", the $\overline{\text{INT10}}$ pin becomes the $\overline{\text{INT1}}$ input pin. When the CNTRSEL bit is set to "1", the INT11 pin becomes the INT1 input pin.

The INT10 pin is shared with the CNTR00 pin and the INT11 pin is shared with the CNTR01 pin.

Figure 11.14 shows the TXMR Register when $\overline{\text{INT1}}$ Interrupt is Used.

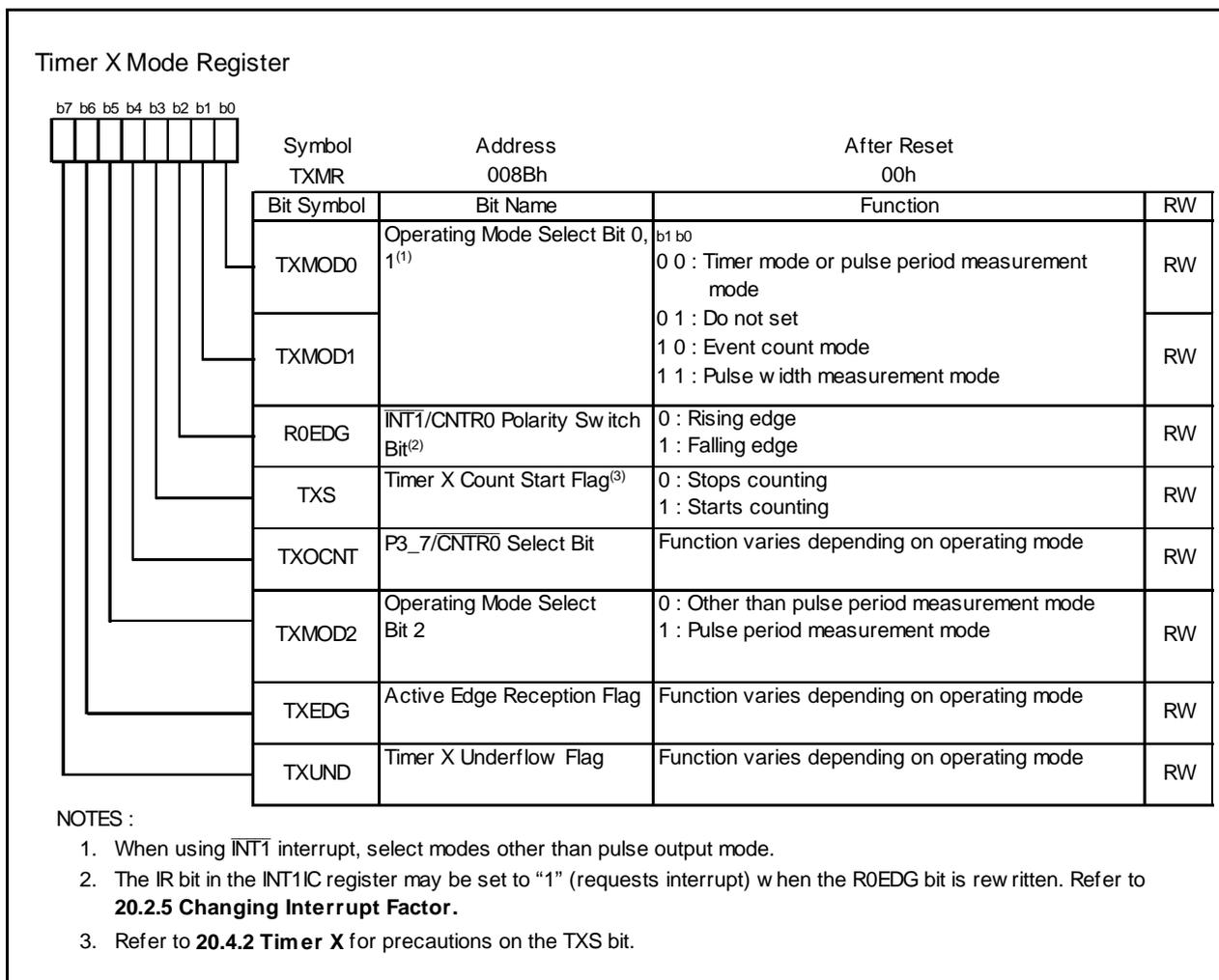


Figure 11.14 TXMR Register when $\overline{\text{INT1}}$ Interrupt is Used

11.2.4 $\overline{\text{INT3}}$ Interrupt

The $\overline{\text{INT3}}$ interrupt is generated by the $\overline{\text{INT3}}$ input. Set the TCC07 bit in the TCC0 register to “0” ($\overline{\text{INT3}}$).

When the TCC06 bit in the TCC0 register is set to “0”, the $\overline{\text{INT3}}$ interrupt request is generated synchronizing with the count source of timer C. When the TCC06 bit is set to “1”, the $\overline{\text{INT3}}$ interrupt request is generated when the $\overline{\text{INT3}}$ is input.

The $\overline{\text{INT3}}$ input contains a digital filter. The IR bit in the INT3IC register is set to “1” (interrupt requested) when the $\overline{\text{INT3}}$ level is sampled for every sampling clock and the sampled input level matches three times. The sampling clock is selected by the TCC11 to TCC10 bits in the TCC1 register. When selecting “Filter”, the interrupt request is generated synchronizing with the sampling clock even if the TCC06 bit is set to “1”. The P3_3 bit in the P3 register indicates the previous value before filtering regardless of the contents set in the TCC11 to TCC10 bits.

The $\overline{\text{INT3}}$ pin is used with the TCIN pin.

When setting the TCC07 bit to “1” (fRING128), the $\overline{\text{INT3}}$ interrupt is generated by the fRING128 clock. The IR bit in the INT3IC register is set to “1” (interrupt requested) every fRING128 clock cycle or every half fRING128 clock cycle.

Figure 11.15 shows the TCC0 Register and Figure 11.16 shows the TCC1 Register.

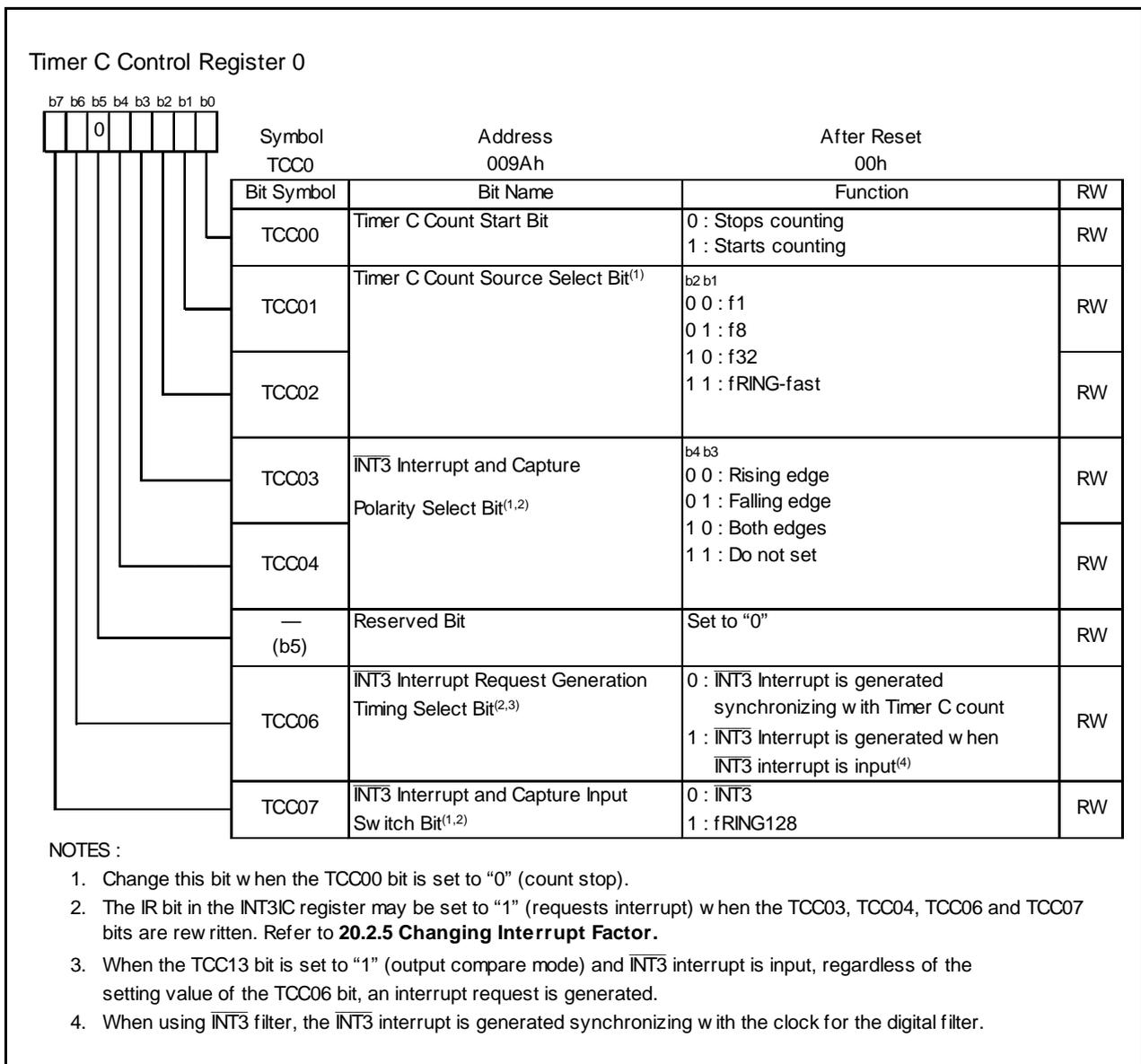


Figure 11.15 TCC0 Register

Timer C Control Register 1

| Symbol | Address | After Reset | |
|------------|--|---|----|
| TCC1 | 009Bh | 00h | |
| Bit Symbol | Bit Name | Function | RW |
| TCC10 | INT3 Filter Select Bit ⁽¹⁾ | b1b0 0 0 : No filter 0 1 : Filter with f1 sampling 1 0 : Filter with f8 sampling 1 1 : Filter with f32 sampling | RW |
| TCC11 | | | RW |
| TCC12 | Timer C Counter Reload Select Bit ^(2,3) | 0 0 : No reload 1 1 : Set TC register to "0000h" w hen compare 1 is matched | RW |
| TCC13 | Compare 0 / Capture Select Bit | 0 0 : Capture Select (input capture mode) ⁽²⁾ 1 1 : Compare 0 Output Select (output compare mode) | RW |
| TCC14 | Compare 0 Output Mode Select Bit ⁽³⁾ | b5 b4 0 0 : CMP output remains unchanged even w hen compare 0 is matched 0 1 : CMP output is reversed w hen compare 0 signal is matched 1 0 : CMP output is set to "L" w hen compare 0 signal is matched 1 1 : CMP output is set to "H" w hen compare 0 signal is matched | RW |
| TCC15 | | | RW |
| TCC16 | Compare 1 Output Mode Select Bit ⁽³⁾ | b7 b6 0 0 : CMP output remains unchanged even w hen compare 1 is matched 0 1 : CMP output is reversed w hen compare 1 signal is matched 1 0 : CMP output is set to "L" w hen compare 1 signal is matched 1 1 : CMP output is set to "H" w hen compare 1 signal is matched | RW |
| TCC17 | | | RW |

NOTES :

- When the same value from the $\overline{\text{INT3}}$ pin is sampled three times continuously, the input is determined.
- When the TCC00 bit in the TCC0 register is set to "0" (count stop), rewrite the TCC13 bit.
- When the TCC13 bit is set to "0" (input capture mode), set the TCC12, TCC14 to TCC17 bits to "0".

Figure 11.16 TCC1 Register

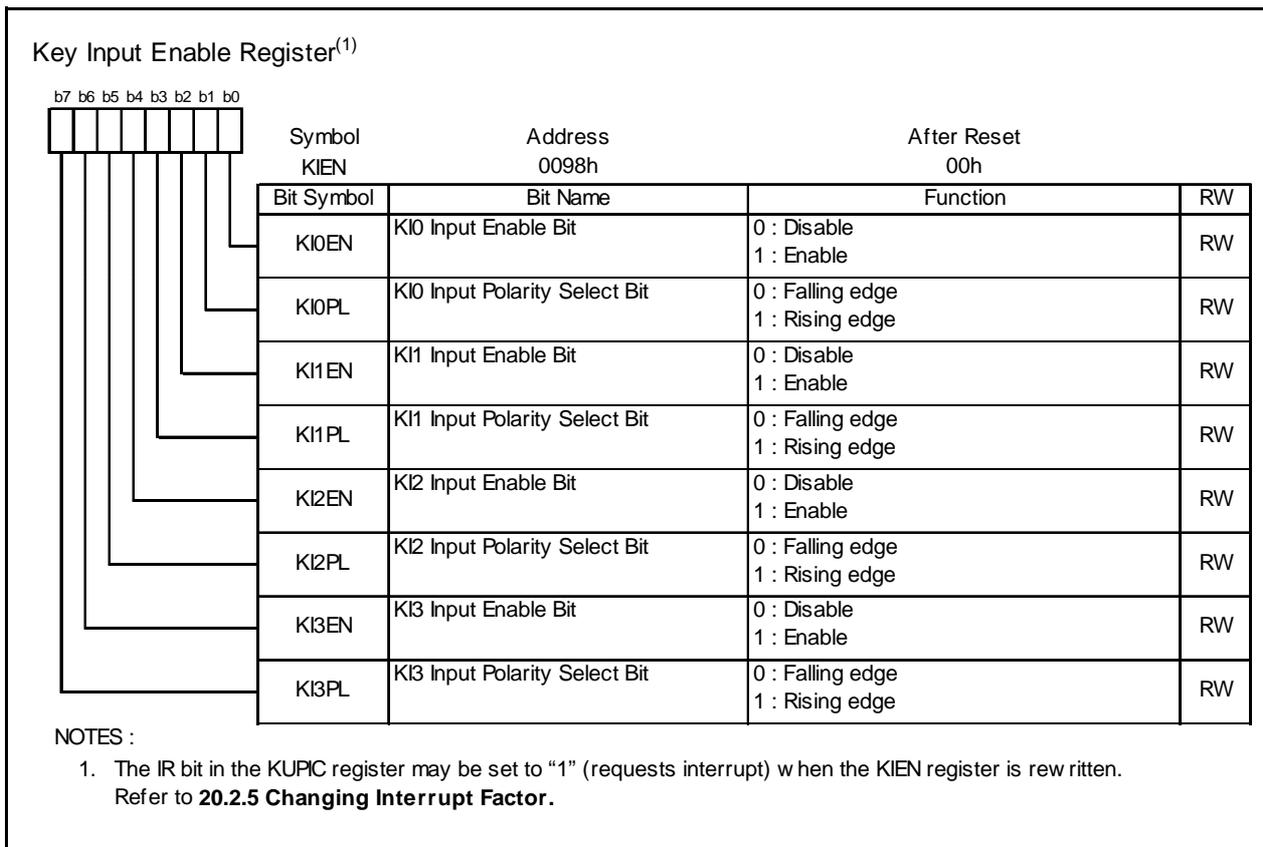


Figure 11.18 KIEN Register

11.4 Address Match Interrupt

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

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

The value of the PC (Refer to **11.1.6.7 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 pushed on the stack). When returning from the address match interrupt, return by one of the following:

- 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 an interrupt request was acknowledged. And then use a jump instruction.

Table 11.6 lists the Value of PC Saved to Stack when Address Match Interrupt is Acknowledged.

Figure 11.19 shows the AIER, RMAD0 to RMAD1 Registers.

Table 11.6 Value of PC Saved to Stack when Address Match Interrupt is Acknowledged

| Address Indicated by RMADi Register (i=0,1) | PC Value Saved ⁽¹⁾ |
|--|---|
| <ul style="list-style-type: none"> • 16-bit operation code instruction • Instruction shown below among 8-bit operation code instructions 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.B:S #IMM8,dest STNZ.B:S #IMM8,dest STZX.B:S #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 **11.1.6.7 Saving a Register** for the PC value saved.

Table 11.7 Between Address Match Interrupt Sources and Associated Registers

| Address Match Interrupt Factor | Address Match Interrupt Enable Bit | Address Match Interrupt Register |
|--------------------------------|------------------------------------|----------------------------------|
| Address Match Interrupt 0 | AIER0 | RMAD0 |
| Address Match Interrupt 1 | AIER1 | RMAD1 |

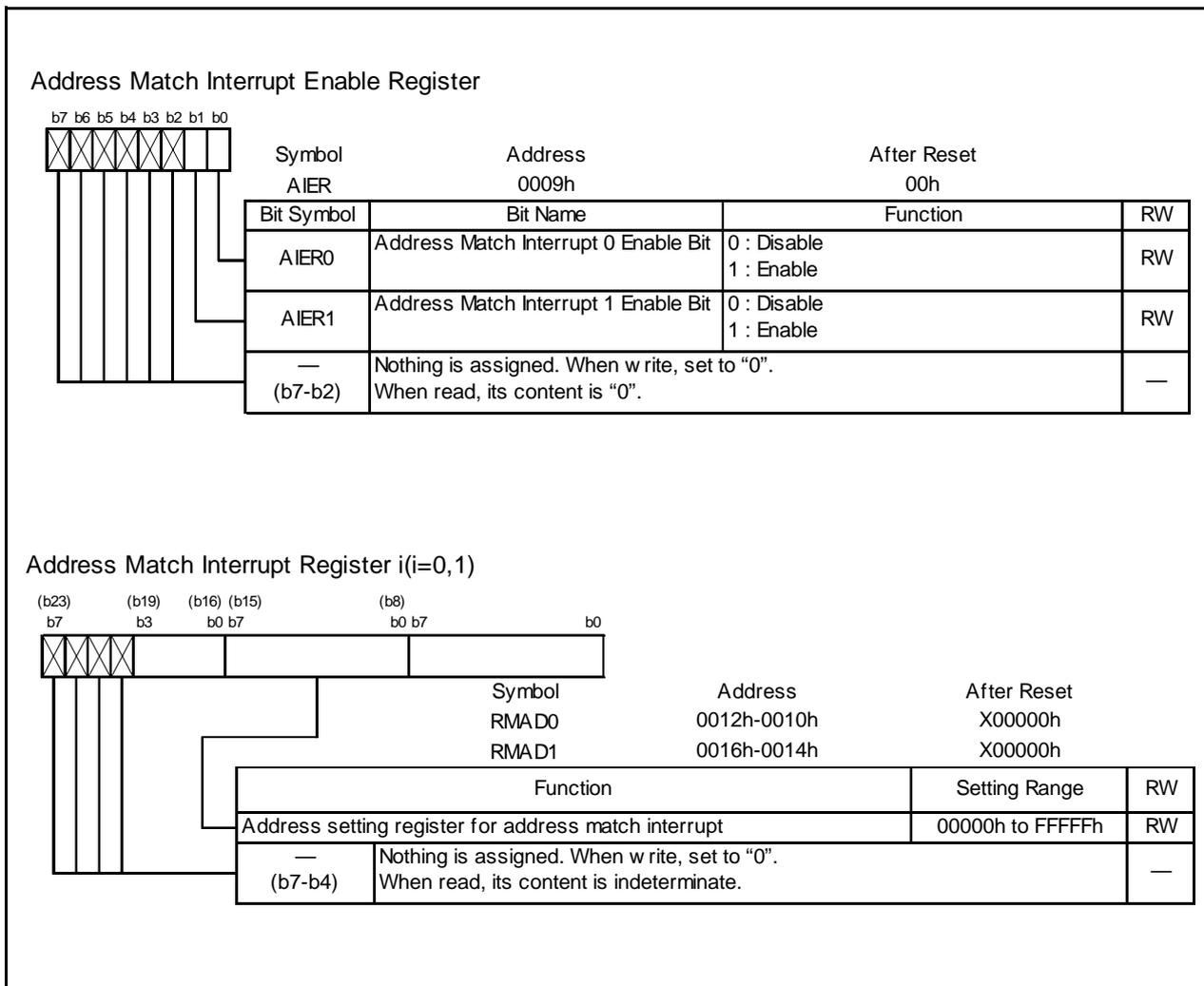


Figure 11.19 AIER, RMAD0 to RMAD1 Registers

12. Watchdog Timer

The watchdog timer is a function to detect when the program is out of control. To use the watchdog timer is recommend for improving reliability of a system. The watchdog timer contains a 15-bit counter and can select count source protection mode is enabled or disabled. Table 12.1 lists the Count Source Protection Mode is Enabled / Disabled

Refer to 5.5 Watchdog Timer Reset for details of the watchdog timer reset.

Figure 12.1 shows the Block Diagram of Watchdog Timer and Figures 12.2 to 12.3 show the OFS, WDC, WDTR, WDTS and CSPR Registers.

Table 12.1 Count Source Protection Mode is Enabled / Disabled

| Item | When Count Source Protection Mode is Disabled | When Count Source Protection Mode is Enabled |
|------------------------------------|---|--|
| Count Source | CPU clock | Low-speed on-chip oscillator clock |
| Count Operation | Decrement | |
| 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 | Either of 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 |
| Operation at the time of Underflow | Watchdog timer interrupt or watchdog timer reset | Watchdog timer reset |

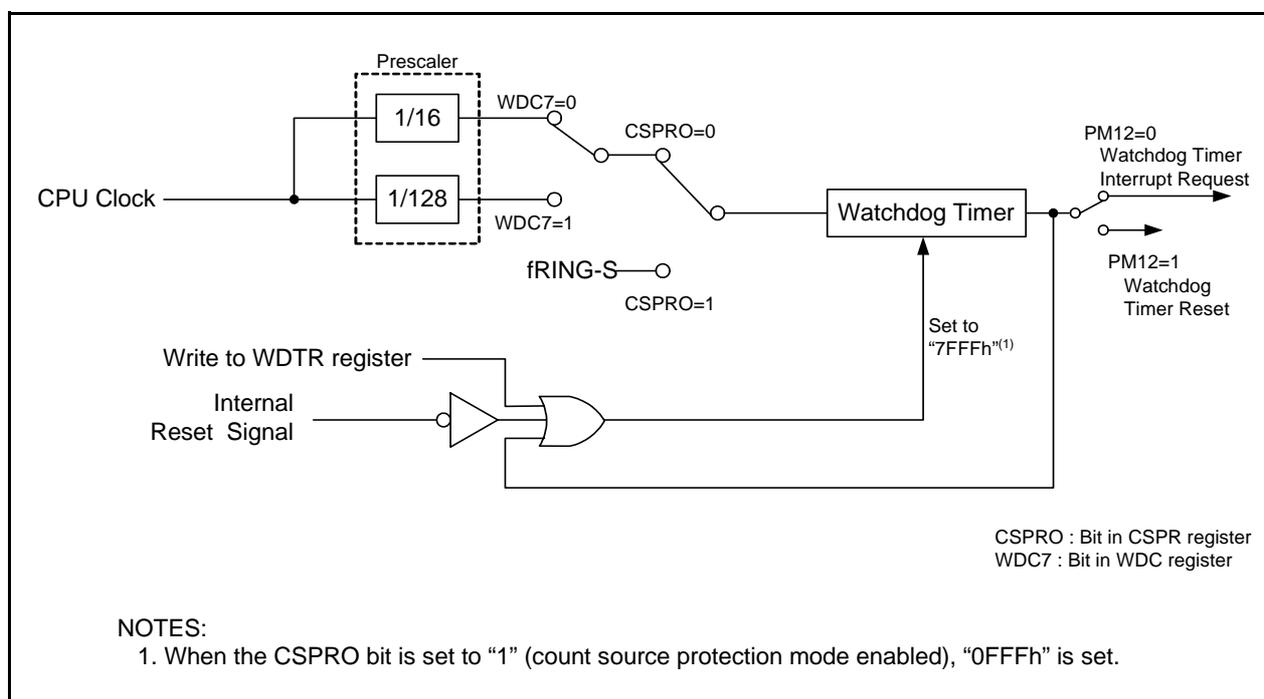


Figure 12.1 Block Diagram of Watchdog Timer

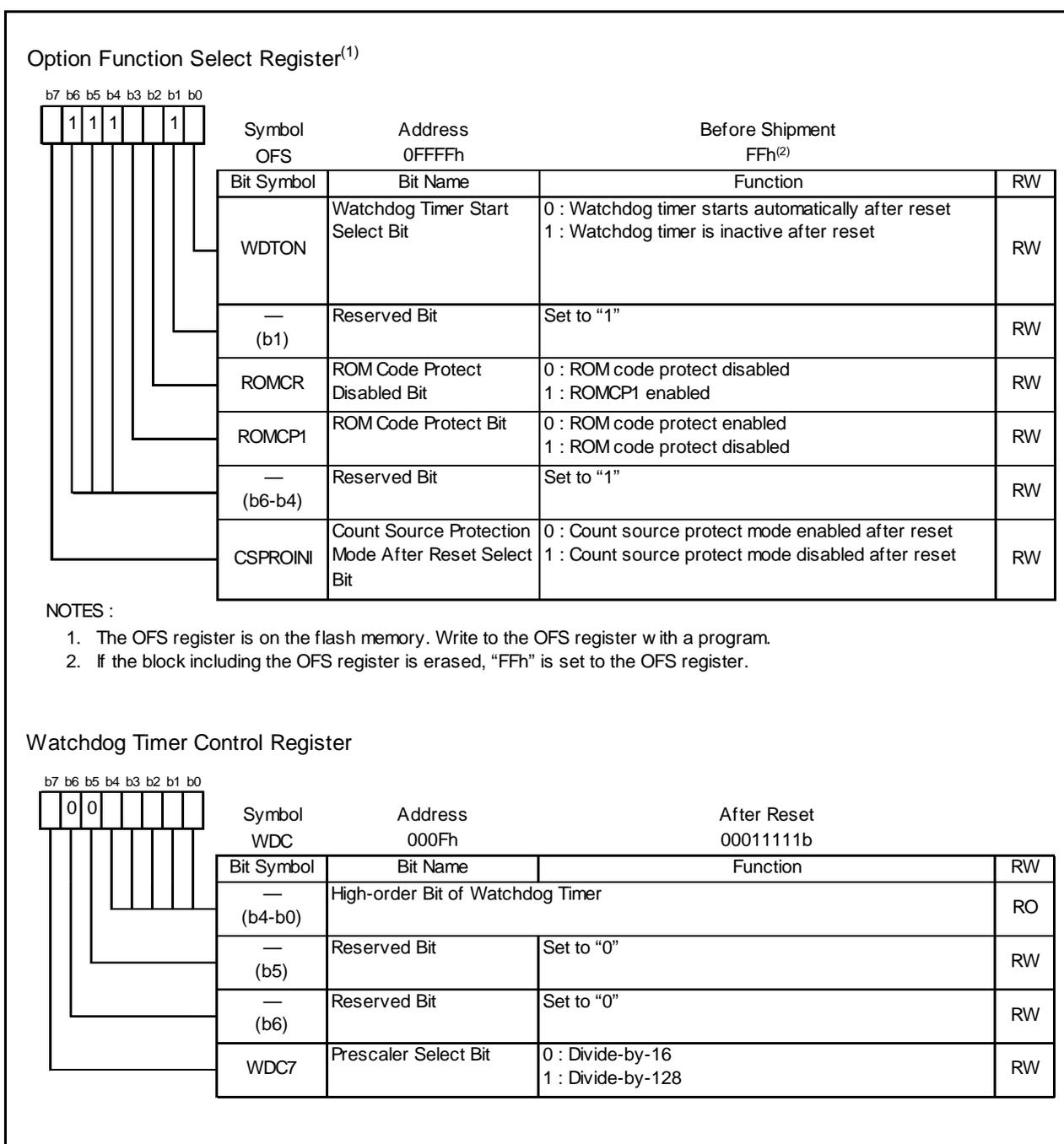


Figure 12.2 OFS and WDC Registers

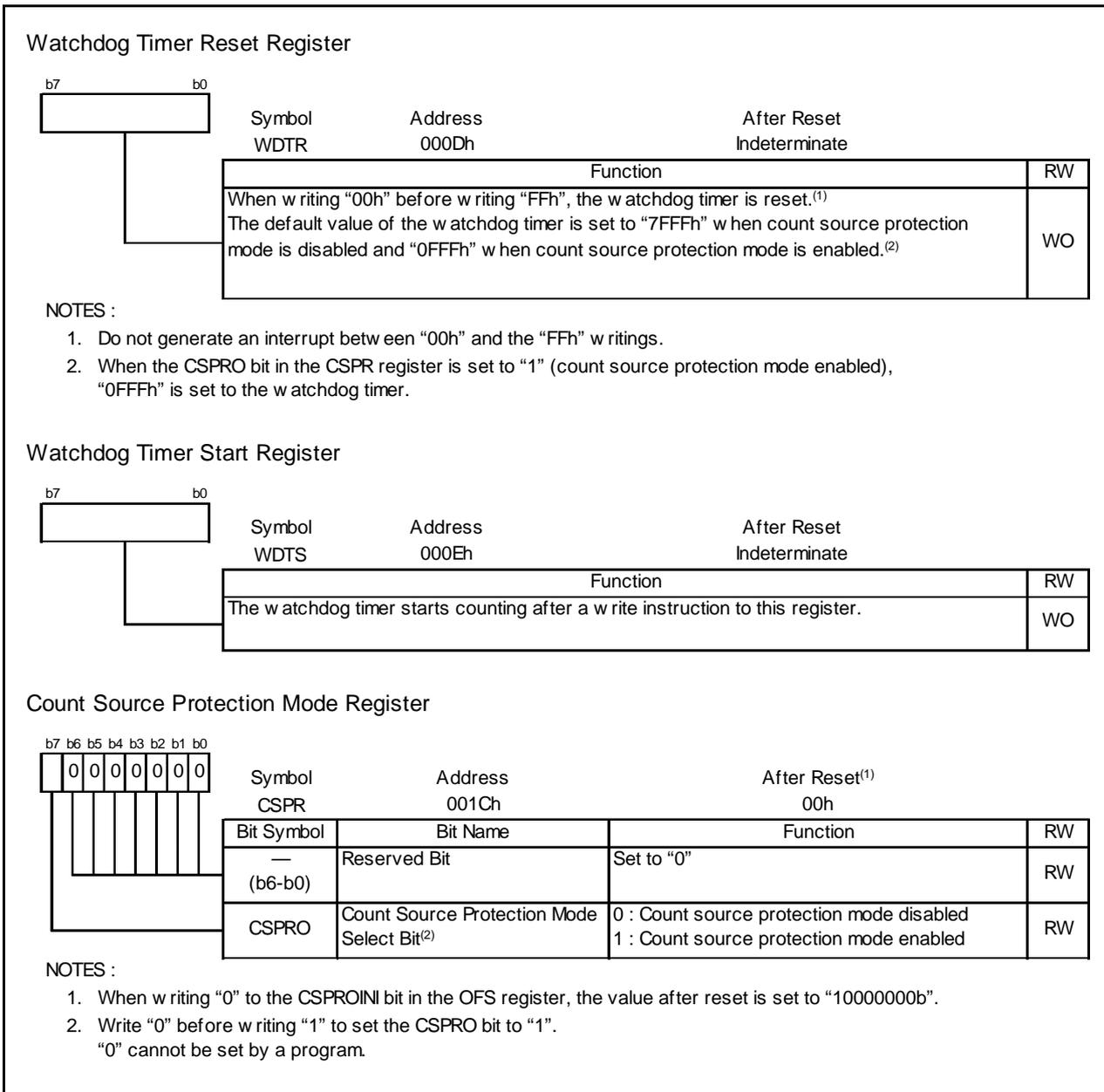


Figure 12.3 WDTR, WDTS and CSPR Registers

12.1 When Count Source Protection Mode Disabled

The count source of the watchdog timer is the CPU clock when count source protection mode is disabled. Table 12.2 lists the Specification of Watchdog Timer (When Count Source Protection Mode is Disabled).

Table 12.2 Specification of Watchdog Timer (When Count Source Protection Mode is Disabled)

| Item | Specification |
|------------------------------------|--|
| Count Source | CPU clock |
| Count Operation | Decrement |
| Period | $\frac{\text{Division ratio of prescaler}(n) \times \text{count value of watchdog timer}(32768)^{(1)}}{\text{CPU clock}}$ n : 16 or 128 (selected by WDC7 bit in WDC register) e.g. When the CPU clock is 16MHz and prescaler is divided by 16, the period is approximately 32.8ms |
| Count Start Condition | The WDTON bit ⁽²⁾ in the OFS register (0FFFFh) selects the operation of watchdog timer after 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 reset and the count starts by writing to the WDTS register • When the WDTON bit is set to "0" (watchdog timer starts automatically after exiting) The watchdog timer and prescaler start counting automatically after reset |
| Reset Condition of Watchdog Timer | <ul style="list-style-type: none"> • Reset • Write "00h" to the WDTR register before writing "FFh" • Underflow |
| Count Stop Condition | Stop and wait modes (inherit the count from the held value after exiting modes) |
| Operation at the 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 5.5 Watchdog Timer Reset) |

NOTES:

1. The watchdog timer is reset when writing "00h" to the WDTR register before writing "FFh". The prescaler is reset after the microcomputer is reset. Some errors occur by the prescaler for the period of the watchdog timer.
2. The WDTON bit cannot be changed by a program. When setting the WDTON bit, write "0" to the bit 0 of the address 0FFFFh by a flash writer.

12.2 When 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 the program is out of control, the clock can be supplied to the watchdog timer. Table 12.3 lists the Specification of Watchdog Timer (When Count Source Protection Mode is Enabled).

Table 12.3 Specification of Watchdog Timer (When Count Source Protection Mode is 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 e.g. Period is approximately 32.8ms when the low-speed on-chip oscillator clock is 125 kHz |
| Count Start Condition | The WDTON bit ⁽¹⁾ in the OFS register (0FFFFh) selects the operation of the watchdog timer after 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 reset and the count starts by writing to the WDTS register When the WDTON bit is set to "0" (watchdog timer starts automatically after reset) The watchdog timer and prescaler start counting automatically after reset |
| Reset Condition of Watchdog Timer | <ul style="list-style-type: none"> Reset Write "00h" to the WDTR register before writing "FFh" Underflow |
| Count Stop Condition | None (the count does not stop in wait mode after the count starts. The microcomputer does not enter stop mode) |
| Operation at the time of Underflow | Watchdog timer reset (refer to 5.5 Watchdog Timer Reset) |
| Register, Bit | <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 states are held in count source protection mode <ul style="list-style-type: none"> Writing to the CM10 bit in the CM1 register disables (It remains unchanged even if it is set to "1". The microcomputer does not enter stop mode) Writing to the CM14 bit in the CM1 register disables (It remains unchanged even if it is set to "1". The low-speed on-chip oscillator does not stop) |

NOTES:

- The WDTON bit cannot be changed by a program. When setting the WDTON bit, write "0" to the bit 0 of the address 0FFFFh by a flash writer.
- Even if writing "0" to the CSPROINI bit in the OFS register, the CSPPRO bit is set to "1". The CSPROINI bit cannot be changed by a program. When setting the CSPROINI bit, write "0" to the bit 7 of the address 0FFFFh by a flash writer.

13. Timers

The microcomputer contains two 8-bit timers with 8-bit prescaler and a 16-bit timer. The two 8-bit timers with the 8-bit prescaler contain Timer X and Timer Z. These timers contain a reload register to memorize the default value of the counter. The 16-bit timer is Timer C which contains the input capture and output compare. All these timers operate independently. The count source for each timer is the operating clock that regulates the timing of timer operations such as counting and reloading.

Table 13.1 lists Functional Comparison of Timers.

Table 13.1 Functional Comparison of Timers

| Item | | Timer X | Timer Z | Timer C |
|-------------------|--|---|---|---|
| Configuration | | 8-bit timer with 8-bit prescaler (with reload register) | 8-bit timer with 8-bit prescaler (with reload register) | 16-bit free-run timer (with input capture and output compare) |
| Count | | Decrement | Decrement | Increment |
| Count source | | <ul style="list-style-type: none"> • f1 • f2 • f8 • fRING | <ul style="list-style-type: none"> • f1 • f2 • f8 • Timer X underflow | <ul style="list-style-type: none"> • f1 • f8 • f32 • fRING-fast |
| Function | Timer Mode | provided | provided | not provided |
| | Pulse Output Mode | provided | not provided | not provided |
| | Event Counter Mode | provided | not provided | not provided |
| | Pulse Width Measurement Mode | provided | not provided | not provided |
| | Pulse Period Measurement Mode | provided | not provided | not provided |
| | Programmable Waveform Generation Mode | not provided | provided | not provided |
| | Programmable one-shot generation mode | not provided | provided | not provided |
| | Programmable Wait One-Shot Generation Mode | not provided | provided | not provided |
| | Input Capture Mode | not provided | not provided | provided |
| | Output Compare Mode | not provided | not provided | provided |
| Input Pin | | CNTR0 | $\overline{\text{INT0}}$ | TCIN |
| Output Pin | | $\overline{\text{CNTR0}}$ CNTR0 | TZOUT | CMP0_0 to CMP0_2 CMP1_0 to CMP1_2 |
| Related Interrupt | | Timer X interrupt $\overline{\text{INT1}}$ interrupt | Timer Y interrupt $\overline{\text{INT0}}$ interrupt | Timer C interrupt $\overline{\text{INT3}}$ interrupt Compare 0 interrupt Compare 1 interrupt |
| Timer Stop | | provided | provided | provided |

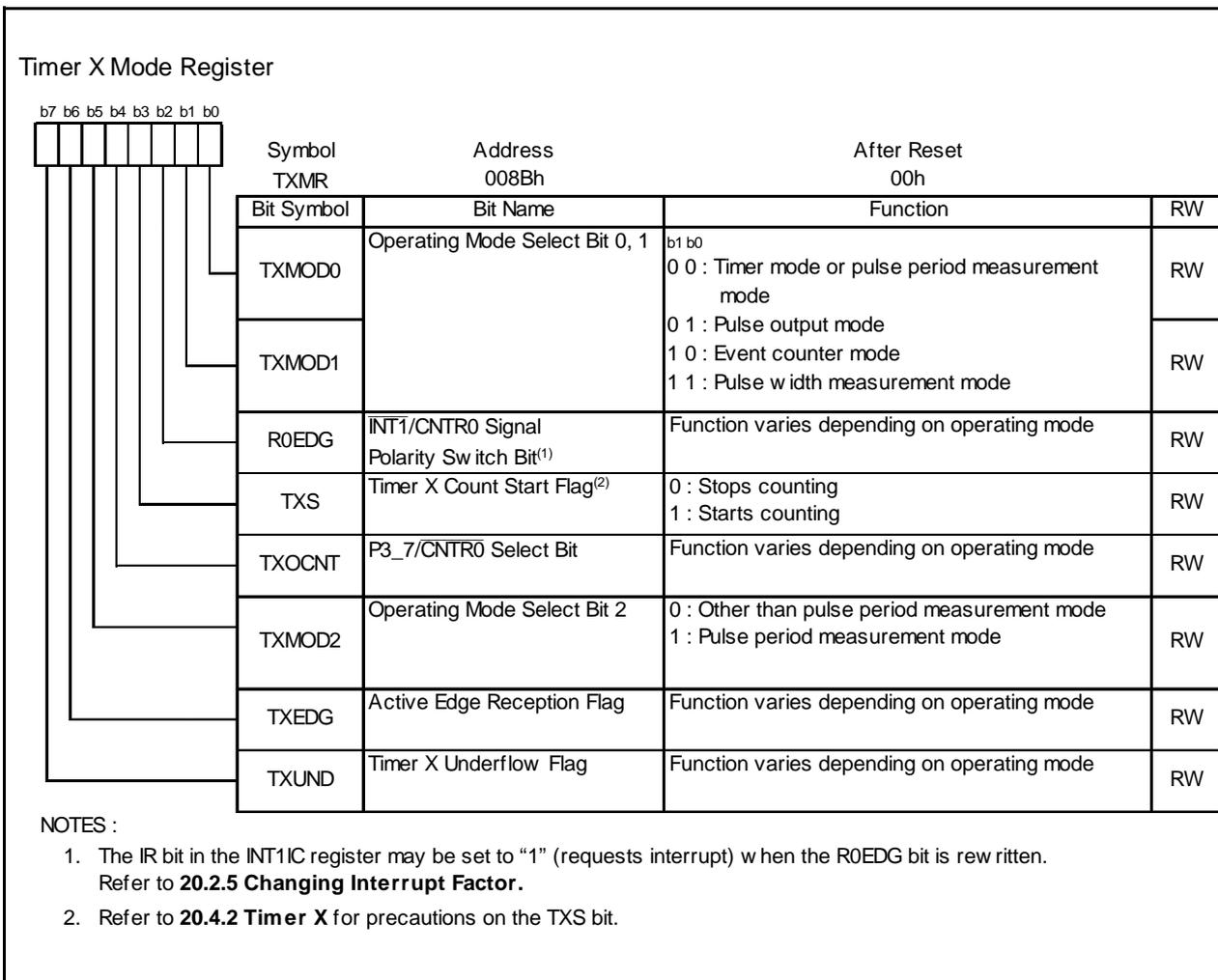


Figure 13.2 TXMR Register

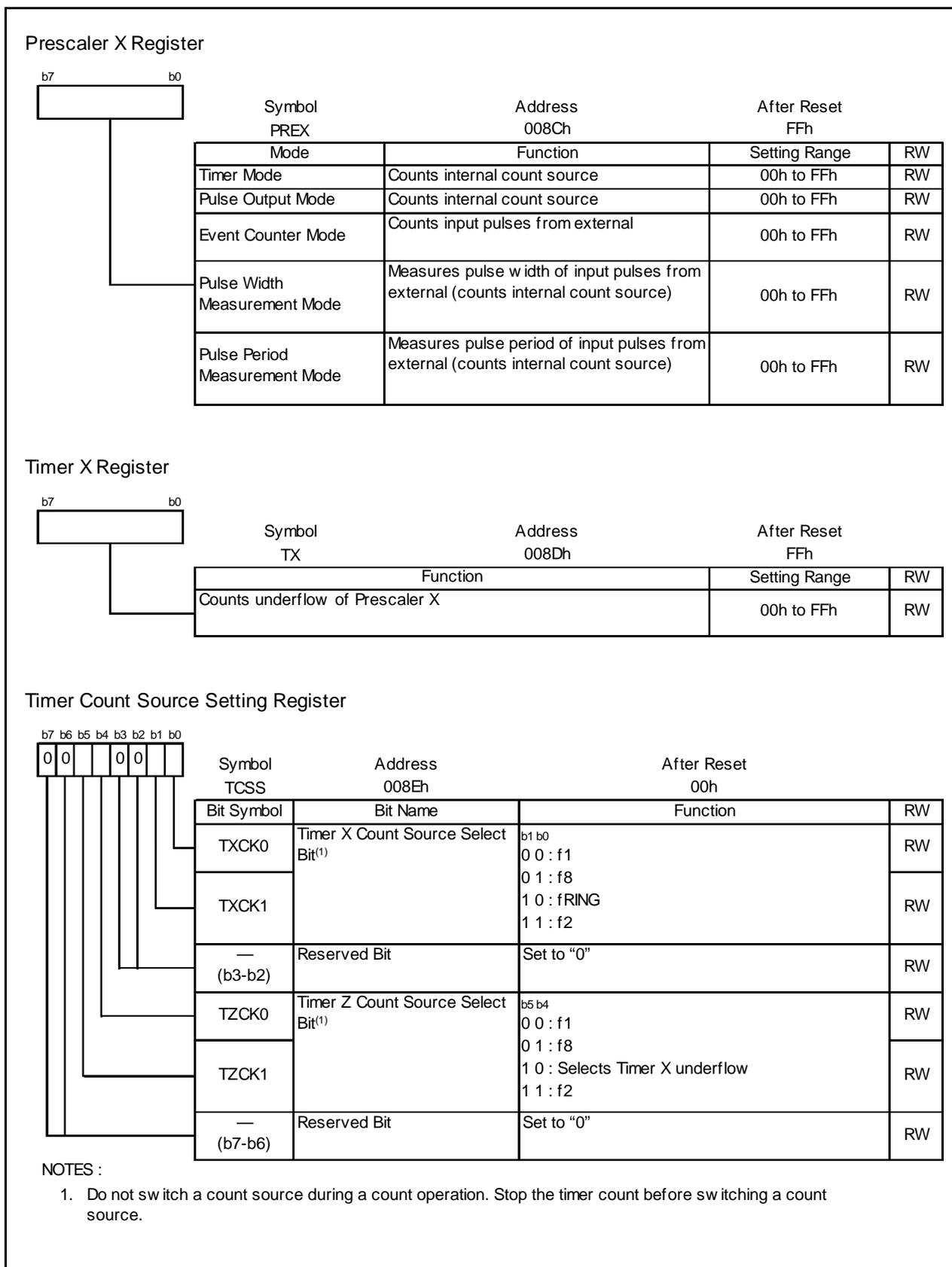


Figure 13.3 PREX, TX and TCSS Registers

13.1.1 Timer Mode

Timer mode is mode to count the count source which is internally generated (See **Table 13.2 Specification of Timer Mode**). Figure 13.4 shows the TXMR Register in Timer Mode.

Table 13.2 Specification of Timer Mode

| Item | Specification |
|---|--|
| Count Source | f1, f2, f8, fRING |
| Count Operation | <ul style="list-style-type: none"> Decrement When the timer underflows, the contents in the reload register is reloaded and the count is inherited |
| Divide Ratio | 1/(n+1)(m+1) n: setting value of PREX register, m: setting value of TX register |
| Count Start Condition | Write "1" (count starts) to the TXS bit in the TXMR register |
| Count Stop Condition | Write "0" (count stops) to the TXS bit in the TXMR register |
| Interrupt Request Generation Timing | When Timer X underflows [Timer X interrupt] |
| INT10/CNTR00, INT11/CNTR01 Pin Function | Programmable I/O port, or $\overline{\text{INT1}}$ interrupt input |
| CNTR0 Pin Function | Programmable I/O port |
| Read from Timer | The count value can be read by reading the TX and PREX registers |
| Write to Timer | <ul style="list-style-type: none"> When writing to the TX and PREX registers while the count stops, the value is written to both the reload register and counter. When writing to the TX and PREX registers during the count, the value is written to each reload register of the TX and PREX registers at the following count source input and the data is transferred to the counter at the second count source input and the count re-starts at the third count source input. |

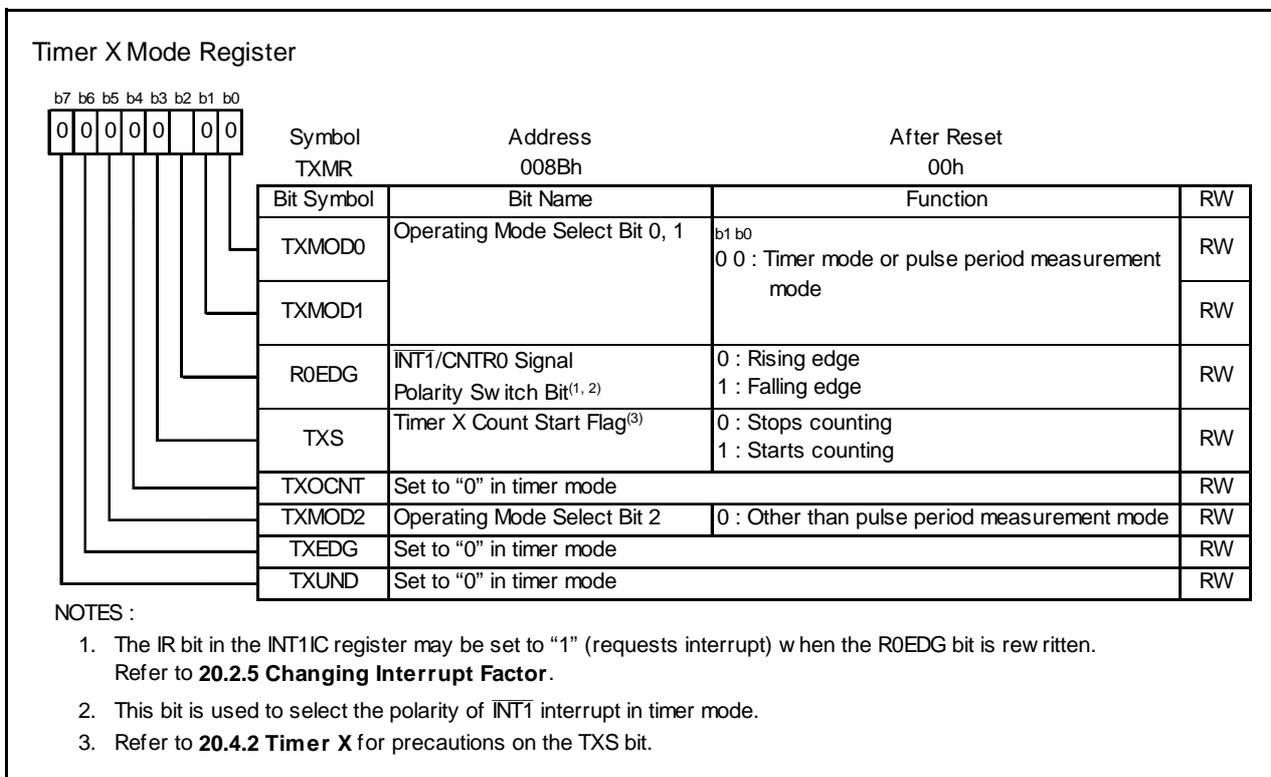


Figure 13.4 TXMR Register in Timer Mode

13.1.2 Pulse Output Mode

Pulse output mode is mode to count the count source internally generated and outputs the pulse which inverts the polarity from the CNTR0 pin each time the timer underflows (See **Table 13.3 Specification of Pulse Output Mode**). Figure 13.5 shows TXMR Register in Pulse Output Mode.

Table 13.3 Specification of Pulse Output Mode

| Item | Specification |
|-------------------------------------|--|
| Count Source | f1, f2, f8, fRING |
| Count Operation | <ul style="list-style-type: none"> • Decrement • When the timer underflows, the contents in the reload register is reloaded and the count is inherited |
| Divide Ratio | 1/(n+1)(m+1) n: setting value of PREX register, m: setting value of TX register |
| Count Start Condition | Write "1" (count starts) to the TXS bit in the TXMR register |
| Count Stop Condition | Write "0" (count stops) to the TXS bit in the TXMR register |
| Interrupt Request Generation Timing | When Timer X underflows [Timer X interrupt] |
| INT10/CNTR0 Pin Function | Pulse output |
| CNTR0 Pin Function | Programmable I/O port or inverted output of CNTR0 |
| Read from timer | The count value can be read by reading the TX and PREX registers. |
| Write to Timer | <ul style="list-style-type: none"> • When writing to the TX and PREX registers while the count stops, the value is written to both the reload register and counter. • When writing to the TX and PREX registers during the count, the value is written to each reload register of the TX and PREX registers at the following count source input and the data is transferred to the counter at the second count source input and the count re-starts at the third count source input. |
| Select Function | <ul style="list-style-type: none"> • $\overline{\text{INT1}}/\text{CNTR0}$ signal polarity switch function The R0EDG bit can select the polarity level when the pulse output starts⁽¹⁾ • Inverted pulse output function The pulse which inverts the polarity of the CNTR0 output can be output from the $\overline{\text{CNTR0}}$ pin (selected by TXOCNT bit) |

NOTES:

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

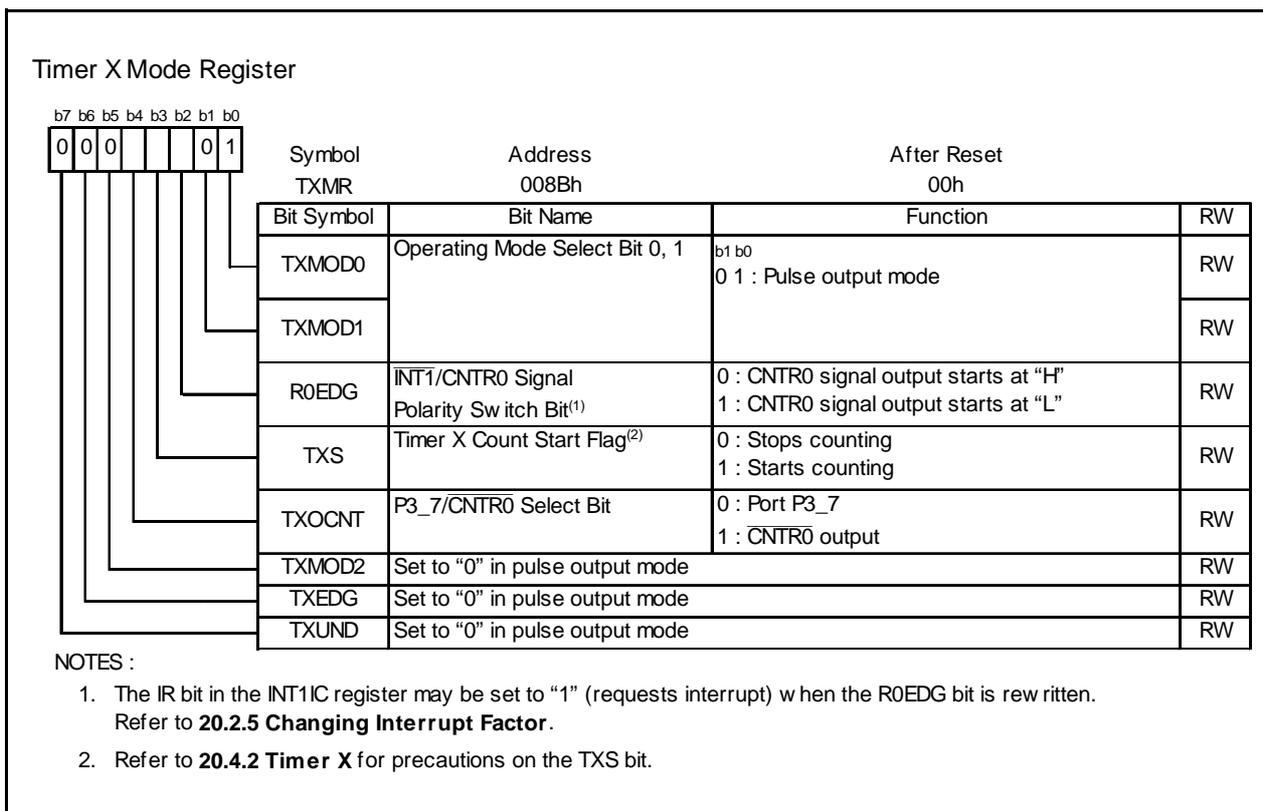


Figure 13.5 TXMR Register in Pulse Output Mode

13.1.3 Event Counter Mode

Event counter mode is mode to count an external signal which inputs from the $\overline{\text{INT1}}/\text{CNTR0}$ pin (See **Table 13.4 Specification of Event Counter Mode**). Figure 13.6 shows TXMR Register in Event Counter Mode.

Table 13.4 Specification of Event Counter Mode

| Item | Specification |
|---|--|
| Count Source | External signal which is input to CNTR0 pin (Active edge is selectable by software) |
| Count Operation | <ul style="list-style-type: none"> • Decrement • When the timer underflows, the contents in the reload register is reloaded and the count is inherited |
| Divide Ratio | $1/(n+1)(m+1)$ n: setting value of PREX register, m: setting value of TX register |
| Count Start Condition | Write "1" (count starts) to the TXS bit in the TXMR register |
| Count Stop Condition | Write "0" (count stops) to the TXS bit in the TXMR register |
| Interrupt Request Generation Timing | <ul style="list-style-type: none"> • When Timer X underflows [Timer X interrupt] |
| $\overline{\text{INT10}}/\text{CNTR00}$, $\overline{\text{INT11}}/\text{CNTR01}$ Pin Function | Count source input ($\overline{\text{INT1}}$ interrupt input) |
| CNTR0 Pin Function | Programmable I/O port |
| Read from Timer | The count value can be read by reading the TX and PREX registers. |
| Write to Timer | <ul style="list-style-type: none"> • When writing to the TX and PREX registers while the count stops, the value is written to both the reload register and counter. • When writing to the TX and PREX registers during the count, the value is written to each reload register of the TX and PREX registers at the following count source input and the data is transferred to the counter at the second count source input and the count re-starts at the third count source input. |
| Select Function | <ul style="list-style-type: none"> • $\overline{\text{INT1}}/\text{CNTR0}$ signal polarity switch function The ROEDG bit can select the active edge of the count source. • Count source input pin select function The CNTRSEL bit in the UCON register can select the CNTR00 or CNTR01 pin |

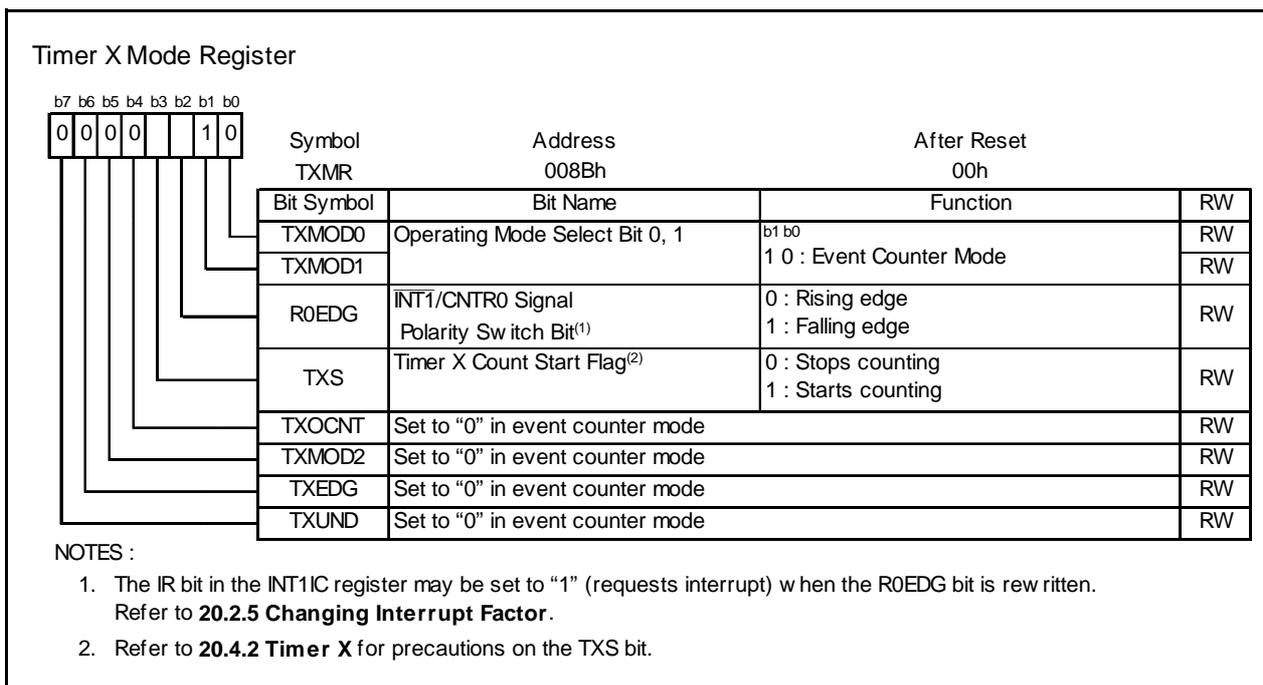


Figure 13.6 TXMR Register in Event Counter Mode

13.1.4 Pulse Width Measurement Mode

Pulse width measurement mode is mode to measure the pulse width of an external signal which inputs from the $\overline{\text{INT1}}$ /CNTR0 pin (See **Table 13.5 Specification of Pulse Width Measurement Mode**). Figure 13.7 shows the TXMR Register in Pulse Width Measurement Mode. Figure 13.8 shows an Operating Example in Pulse Width Measurement Mode.

Table 13.5 Specification of Pulse Width Measurement Mode

| Item | Specification |
|---|--|
| Count Source | f1, f2, f8, fRING |
| Count Operation | <ul style="list-style-type: none"> • Decrement • Continuously counts the selected signal only when the measurement pulse is "H" level, or conversely only "L" level. • When the timer underflows, the contents in the reload register is reloaded and the count is inherited |
| Count Start Condition | Write "1" (count starts) to the TXS bit in the TXMR register |
| Count Stop Condition | Write "0" (count stops) to the TXS bit in the TXMR register |
| Interrupt Request Generation Timing | <ul style="list-style-type: none"> • When Timer X underflows [Timer X interrupt] • Rising or falling of the CNTR0 input (end of measurement period) [$\overline{\text{INT1}}$ interrupt] |
| $\overline{\text{INT10}}$ /CNTR00, $\overline{\text{INT11}}$ /CNTR01 Pin Function | Measurement pulse input ($\overline{\text{INT1}}$ interrupt input) |
| CNTR0 Pin Function | Programmable I/O port |
| Read from Timer | The count value can be read by reading the TX and PREX registers. |
| Write to Timer | <ul style="list-style-type: none"> • When writing to the TX and PREX registers while the count stops, the value is written to both the reload register and counter. • When writing to the TX and PREX registers during the count, the value is written to each reload register of the TX and PREX registers at the following count source input and the data is transferred to the counter at the second count source input and the count re-starts at the third count source input. |
| Select Function | <ul style="list-style-type: none"> • $\overline{\text{INT1}}$/CNTR0 signal polarity switch function The R0EDG bit can select during "H" or "L" level as the input pulse measurement • Measurement pulse input pin select function The CNTRSEL bit in the UCON register can select the CNTR00 or CNTR01 pin |

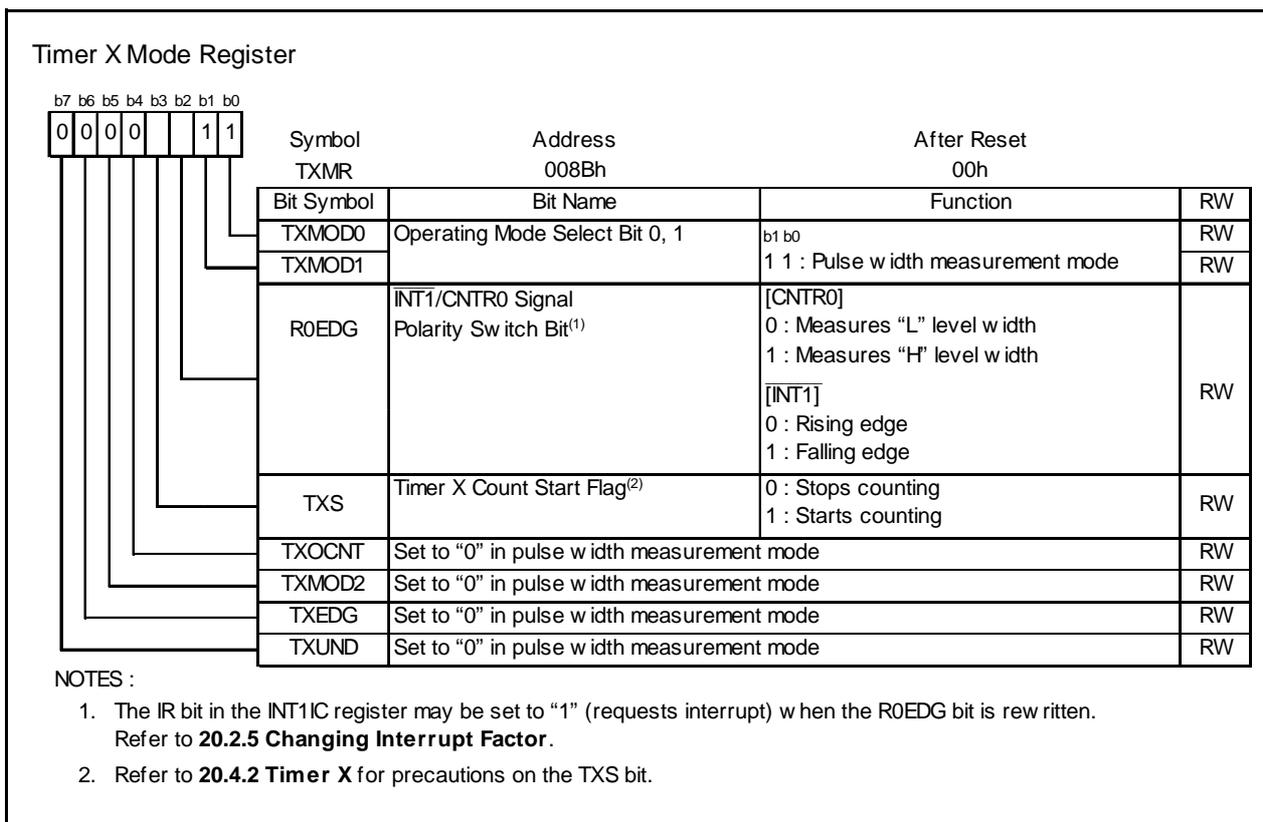


Figure 13.7 TXMR Register in Pulse Width Measurement Mode

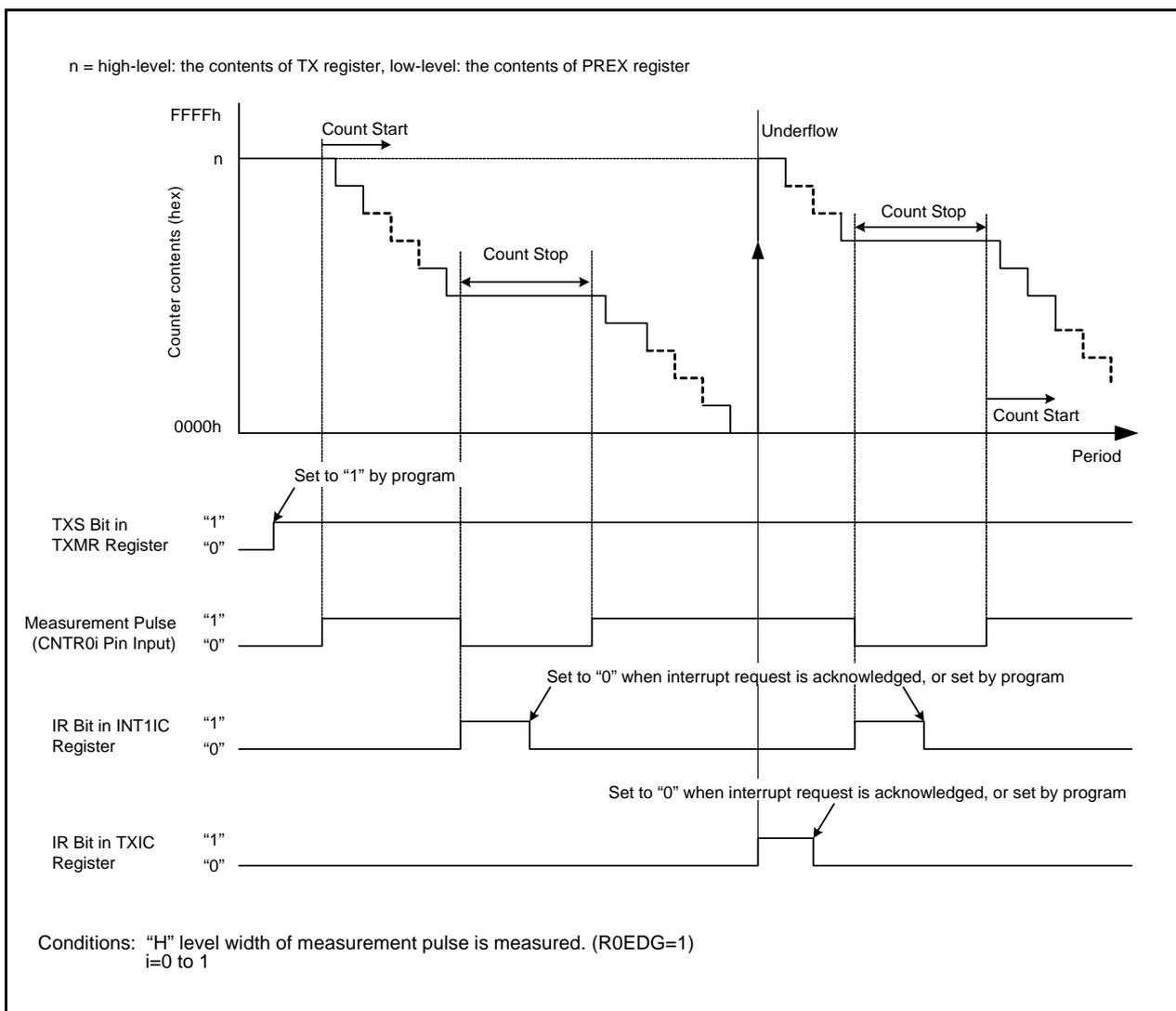


Figure 13.8 Operating Example in Pulse Width Measurement Mode

13.1.5 Pulse Period Measurement Mode

Pulse period measurement mode is mode to measure the pulse period of an external signal which inputs from the $\overline{\text{INT1}}/\text{CNTR0}$ pin (See **Table 13.6 Specification of Pulse Period Measurement Mode**). Figure 13.9 shows the TXMR Register in Pulse Period Measurement Mode. Figure 13.10 shows an Operating Example in Pulse Period Measurement Mode.

Table 13.6 Specification of Pulse Period Measurement Mode

| Item | Specification |
|---|--|
| Count Source | f1, f2, f8, fRING |
| Count Operation | <ul style="list-style-type: none"> Decrement After an active edge of measurement pulse is input, contents for the read-out buffer are retained at the first underflow of prescaler X. Then timer X reloads contents in the reload register at the second underflow of prescaler X and continues counting. |
| Count Start Condition | Write "1" (count start) to the TXS bit in the TXMR register |
| Count Stop Condition | Write "0" (count stop) to TXS bit in TXMR register |
| Interrupt Request Generation Timing | <ul style="list-style-type: none"> When timer X underflows or reloads [timer X interrupt] Rising or falling of $\overline{\text{CNTR0}}$ input (end of measurement period) [$\overline{\text{INT1}}$ interrupt] |
| $\overline{\text{INT10}}/\text{CNTR00}$, $\overline{\text{INT11}}/\text{CNTR01}$ Pin Function | Measurement pulse input ⁽¹⁾ ($\overline{\text{INT1}}$ interrupt input) |
| $\overline{\text{CNTR0}}$ Pin Function | Programmable I/O port |
| Read from Timer | Contents in the read-out buffer can be read by reading the TX register. The value retained in the read-out buffer is released by reading TX register. |
| Write to Timer | <ul style="list-style-type: none"> When writing to the TX and PREX registers while the count stops, the value is written to both the reload register and counter. When writing to the TX and PREX registers during the count, the value is written to each reload register of the TX and PREX registers at the following count source input and the data is transferred to the counter at the second count source input and the count re-starts at the third count source input. |
| Select Function | <ul style="list-style-type: none"> $\overline{\text{INT1}}/\text{CNTR0}$ polarity switch function The R0EDG bit can select the measurement period of input pulse. Measurement pulse input pin select function The CNTRSEL bit in the UCON register can select the CNTR00 or CNTR01 pin. |

NOTES:

- Input the pulse whose period is longer than twice of the prescaler X period. Input the longer pulse for "H" width and "L" width than the prescaler X period. If the shorter pulse than the period is input to the $\overline{\text{CNTR0}}$ pin, the input may be disabled.

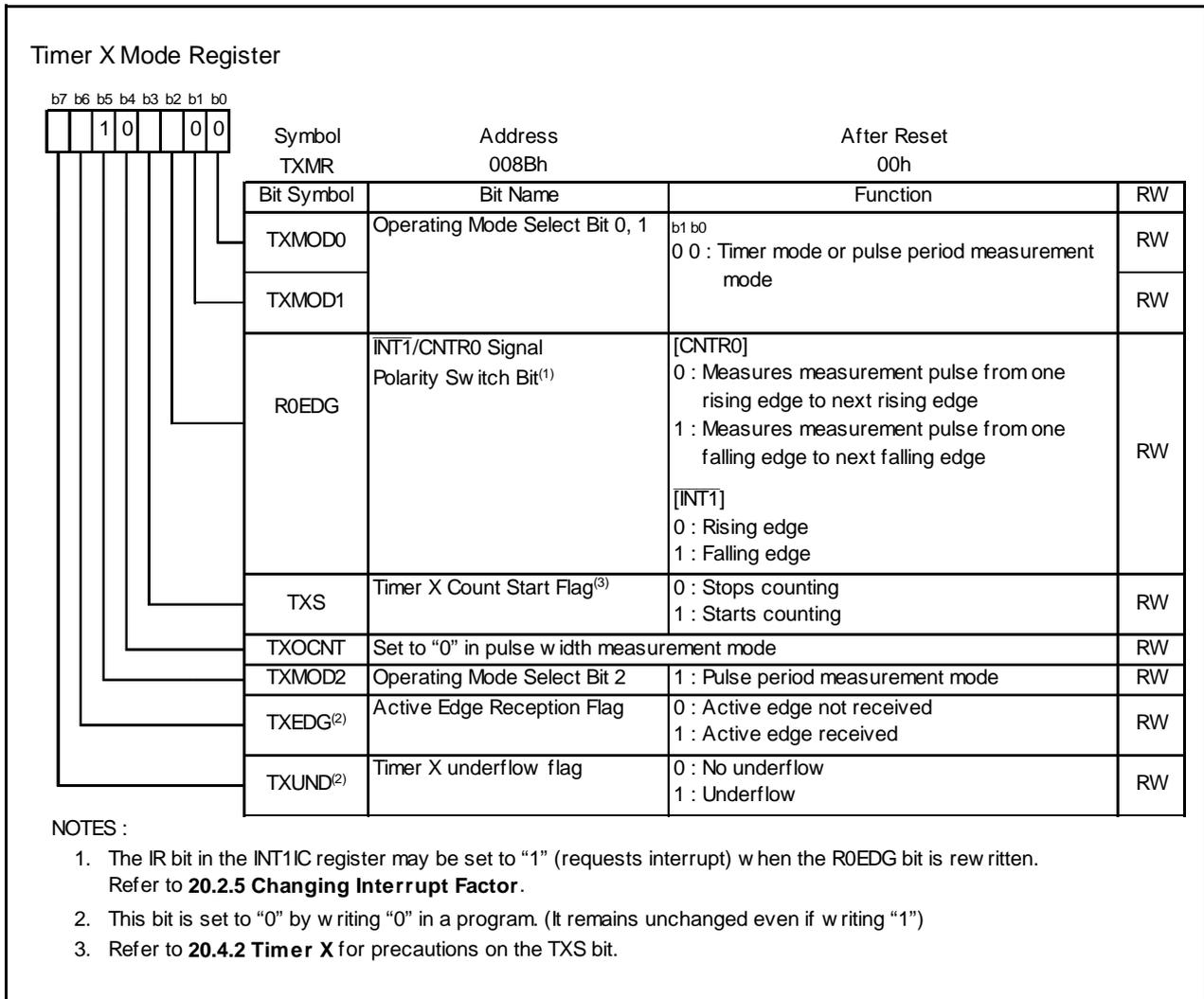


Figure 13.9 TXMR Register in Pulse Period Measurement Mode

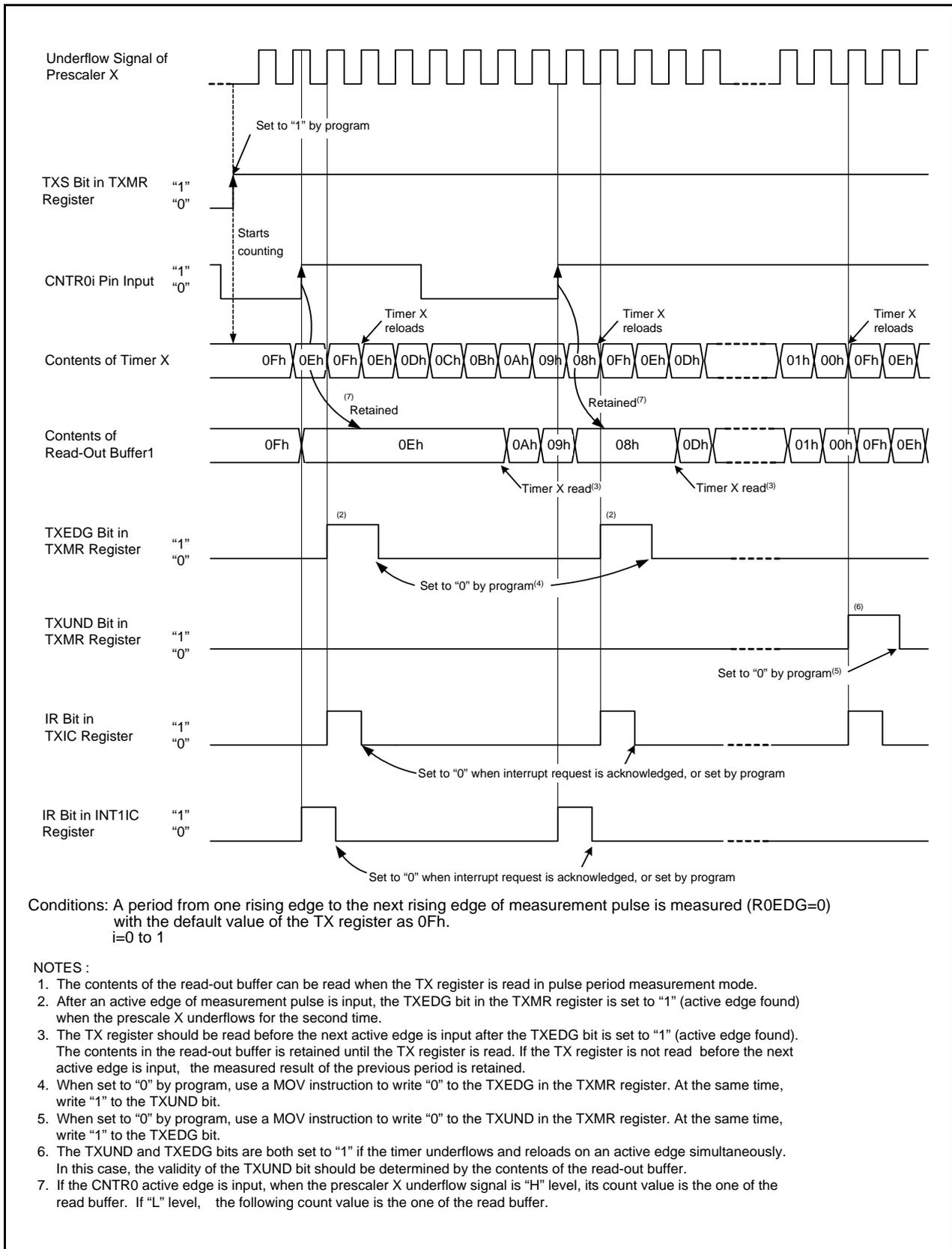


Figure 13.10 Operating Example in Pulse Period Measurement Mode

13.2 Timer Z

Timer Z is an 8-bit timer with an 8-bit prescaler. The prescaler and timer consist of the reload register and counter. The reload register and counter are allocated at the same address. Refer to the **Tables 13.7 to 13.12 for the Specification of Each Modes**. Timer Z contains the timer Z primary and timer Z secondary as the reload register.

Figure 13.11 shows the Block Diagram of Timer Z. Figures 13.12 to 13.15 show the TZMR, PREZ, TZSC, TZPR, TZOC, PUM, and TCSS registers.

Timer Z contains the following four operating modes.

- Timer mode: The timer counts an internal count source or Timer X underflow.
- Programmable waveform generation mode: The timer outputs pulses of a given width successively
- Programmable one-shot generation mode: The timer outputs one-shot pulse.
- Programmable wait one-shot generation mode: The timer outputs delayed one-shot pulse.

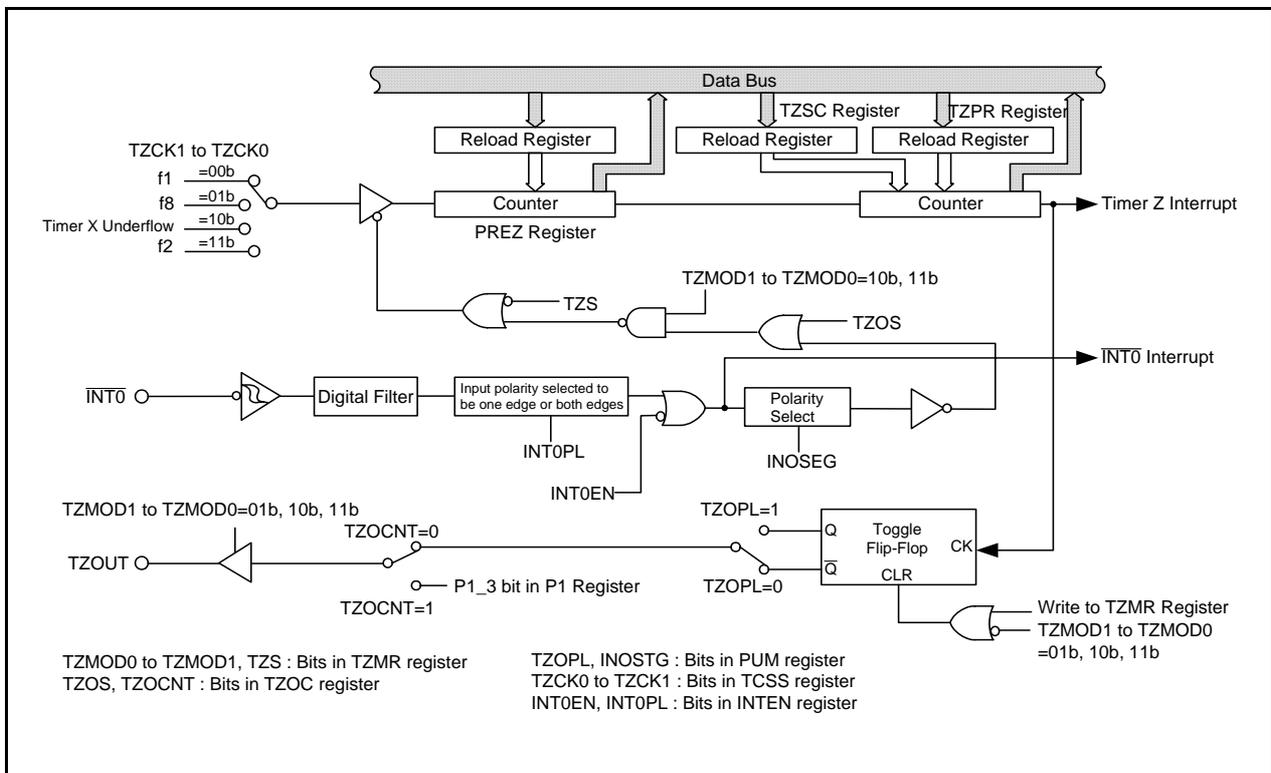


Figure 13.11 Block Diagram of Timer Z

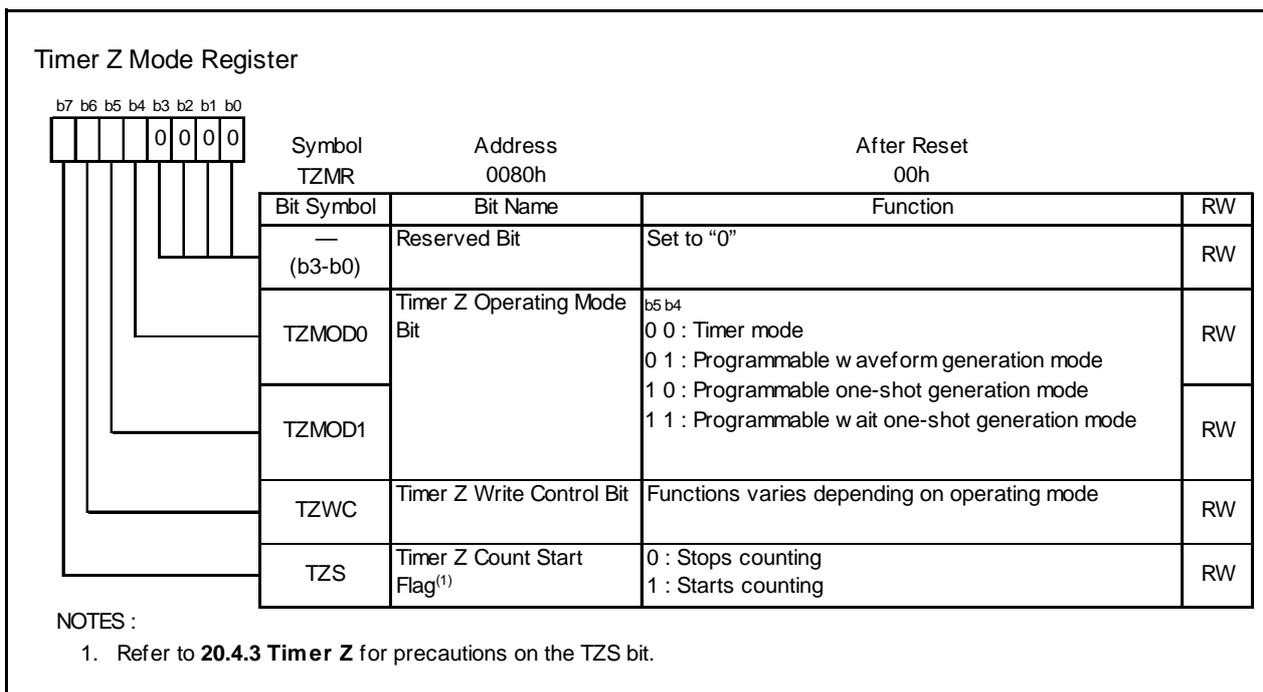


Figure 13.12 TZMR Register

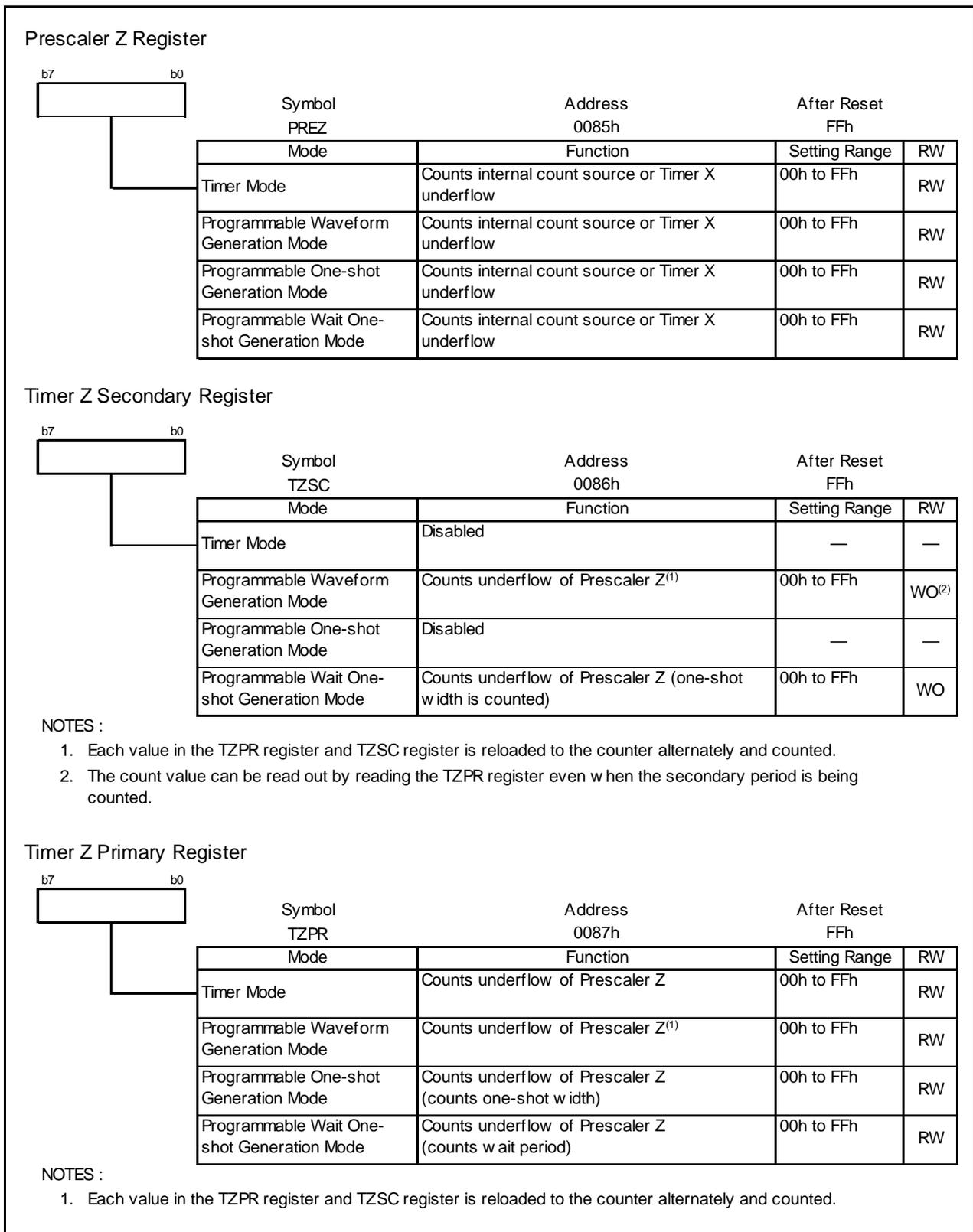


Figure 13.13 PREZ, TZSC and TZPR Registers

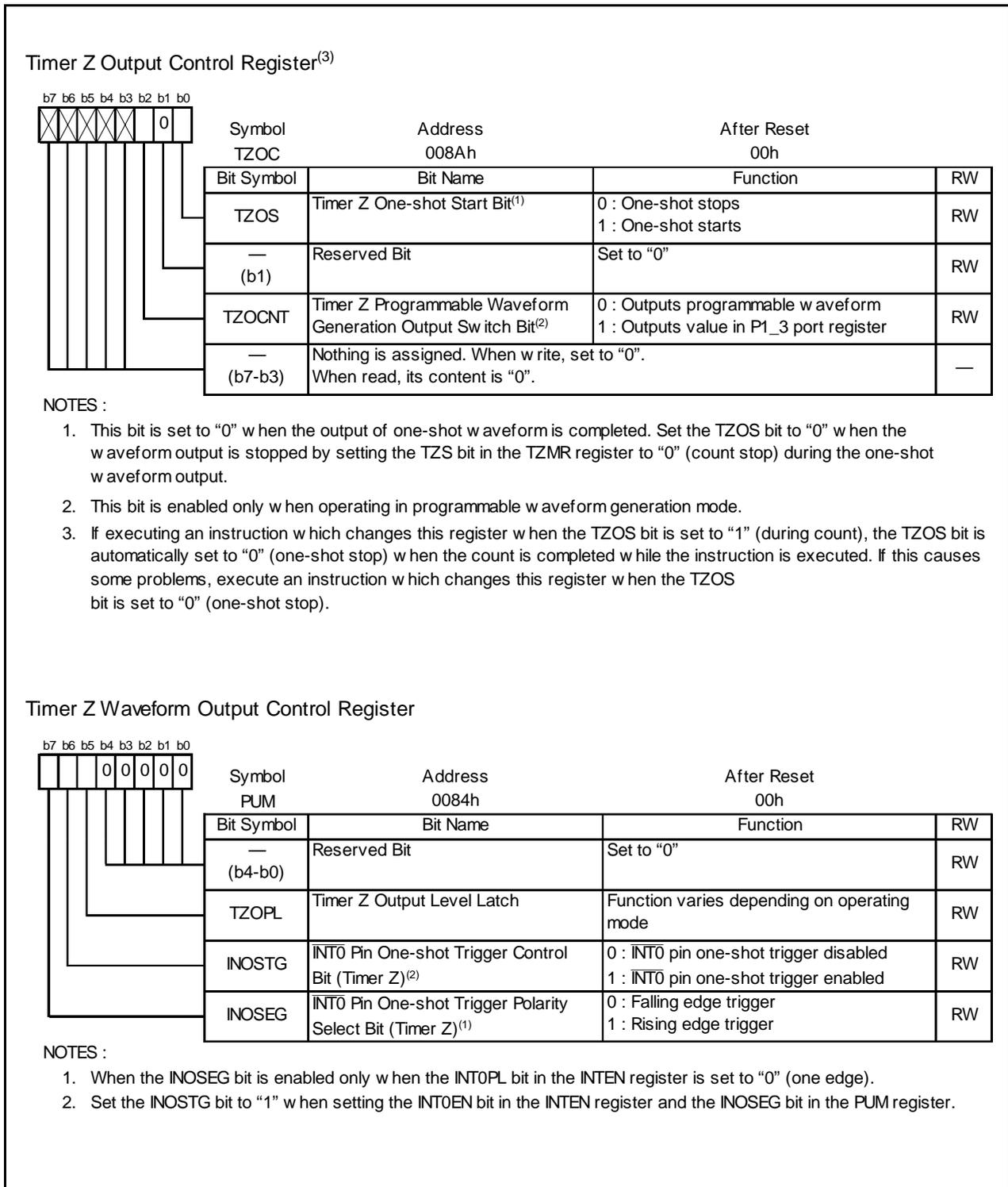


Figure 13.14 TZOC and PUM Registers

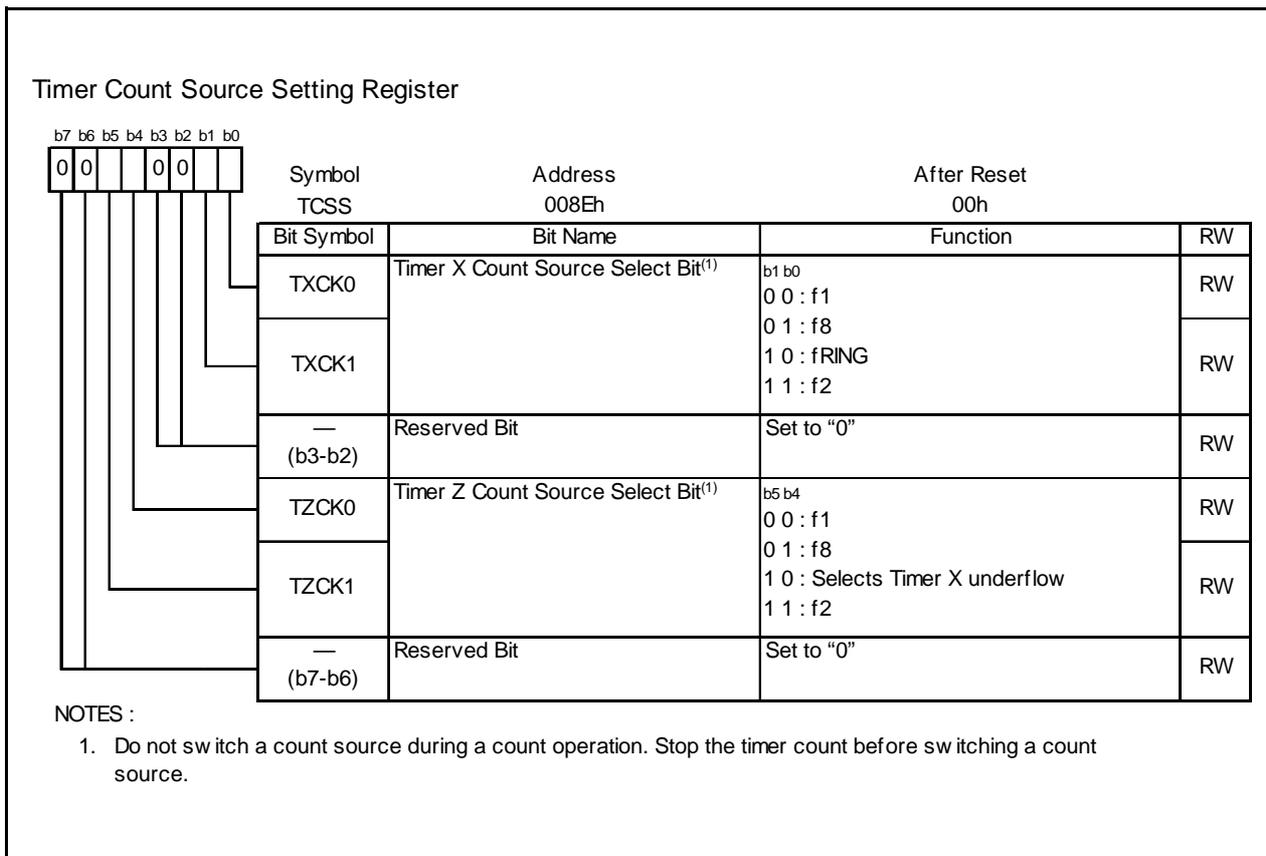


Figure 13.15 TCSS Register

13.2.1 Timer Mode

Timer mode is mode to count a count source which is internally generated or Timer X underflow (see **Table 13.7 Specification of Timer Mode**). The TZSC register is unused in timer mode. Figure 13.16 shows the TZMR and PUM Registers in Timer Mode.

Table 13.7 Specification of Timer Mode

| Item | Specification |
|-------------------------------------|--|
| Count Source | f1, f2, f8, Timer X underflow |
| Count Operation | <ul style="list-style-type: none"> Decrement When the timer underflows, it reloads the reload register contents before the count continues (When Timer Z underflows, the contents of Timer Z primary reload register is reloaded) |
| Divide Ratio | $1/(n+1)(m+1)$ fi: Count source frequency n: setting value in PREZ register, m: setting value in TZPR register |
| Count Start Condition | Write "1" (count starts) to the TZS bit in the TZMR register |
| Count Stop Condition | Write "0" (count stops) to the TZS bit in the TZMR register |
| Interrupt Request Generation Timing | <ul style="list-style-type: none"> When Timer Z underflows [Timer Z interrupt] |
| TZOUT Pin Function | Programmable I/O port |
| INT0 Pin Function | Programmable I/O port, or $\overline{\text{INT0}}$ interrupt input |
| Read from Timer | The count value can be read out by reading the TZPR and PREZ registers |
| Write to Timer ⁽¹⁾ | <ul style="list-style-type: none"> When writing to the TZPR and PREZ registers while the count stops, the value is written to both the reload register and counter. When writing to the TZPR and PREZ registers during the count while the TZWC bit is set to "0" (writing to the reload register and counter simultaneously), the value is written to each reload register of the TZPR and PREZ registers at the following count source input and the data is transferred to the counter at the second count source input and the count re-starts at the third count source input. When the TZWC bit is set to "1" (writing to only the reload register), the value is written to each reload register of the TZPR and PREZ registers (the data is transferred to the counter at the following reload). |

NOTES:

- The IR bit in the TZIC register is set to "1" (interrupt requested) when writing to the TZPR or PREZ register while both of the following conditions are met.

<Conditions>

- TZWC bit in TZMR register is set to "0" (write to reload register and counter simultaneously)
- TZS bit in TZMR register is set to "1" (count starts)

When writing to the TZPR or PREZ register in the above state, disable an interrupt before writing.

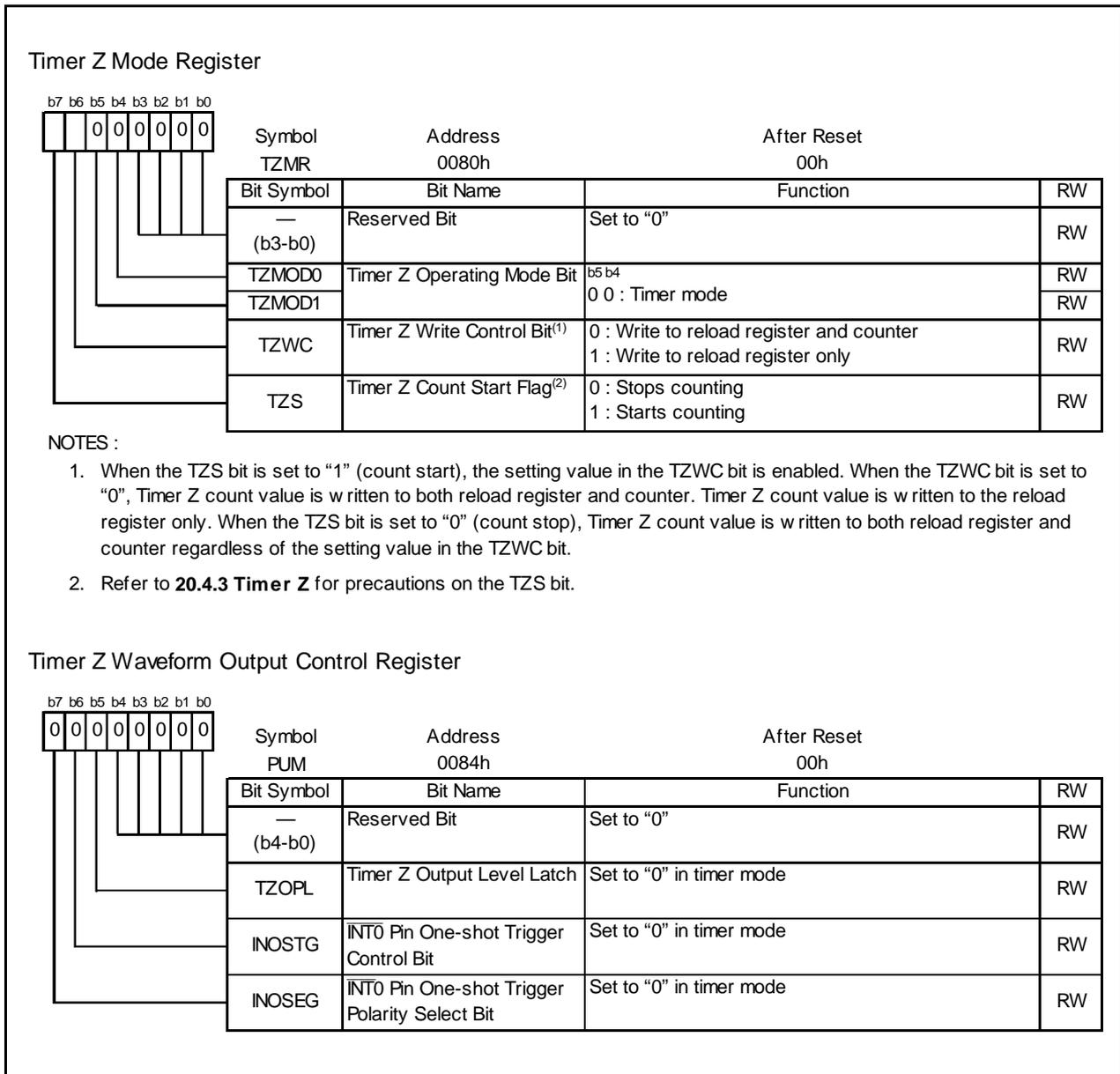


Figure 13.16 TZMR and PUM Registers in Timer Mode

13.2.2 Programmable Waveform Generation Mode

Programmable waveform generation mode is mode to invert the signal output from the TZOUT pin each time the counter underflows, while the values in the TZPR and TZSC registers are counted alternately (See **Table 13.8 Specification of Programmable Waveform Generation Mode**). A counting starts by counting the value set in the TZPR register. Figure 13.17 shows TZMR and PUM Registers in Programmable Waveform Generation Mode. Figure 13.18 shows Operating Example of Timer Z in Programmable Waveform Generation Mode.

Table 13.8 Specification of Programmable Waveform Generation Mode

| Item | Specification |
|-------------------------------------|---|
| Count Source | f1, f2, f8, Timer X underflow |
| Count Operation | <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 : Setting value in PREZ register, m : setting value in TZPR register, p : setting value in TZSC register |
| Count Start Condition | Write "1" (count start) to the TZS bit in the TZMR register |
| Count Stop Condition | Write "0" (count stop) to the TZS bit in the TZMR register |
| Interrupt Request Generation Timing | In half of count source, after Timer Z underflows during secondary period (at the same time as the TZout output change) [Timer Z interrupt] |
| TZOUT Pin Function | Pulse output (When using this function as a programmable I/O port, set to timer mode.) |
| INT0 Pin Function | Programmable I/O port, or $\overline{\text{INT0}}$ interrupt input |
| Read from timer | The count value can be read out by reading the TZPR and PREZ registers ⁽¹⁾ . |
| Write to timer | The value written to the TZSC, PREZ and TZPR registers is written to the reload register only ⁽²⁾ |
| Select function | <ul style="list-style-type: none"> • Output level latch select function The TZOPL bit can select the output level during primary and secondary periods. • Programmable waveform generation output switch function When the TZOCNT bit in the TZOC register is set to "0", the output from the TZOUT pin is inverted synchronously when Timer Z underflows. And when setting to "1", output the value in the P1_3 bit from the TZOUT pin⁽³⁾ |

NOTES:

1. Even when counting the secondary period, read out the TZPR register.
2. The setting value in the TZPR register and TZSC register are made effective by writing a value to the TZPR register. The set values are reflected to the waveform output beginning with the following primary period after writing to the TZPR register.
3. The TZOCNT bit is enabled by the followings.
 - When count starts.
 - When the timer Z interrupt request is generated. The contents after the TZOCNT bit is changed are reflected from the output of the following primary period.

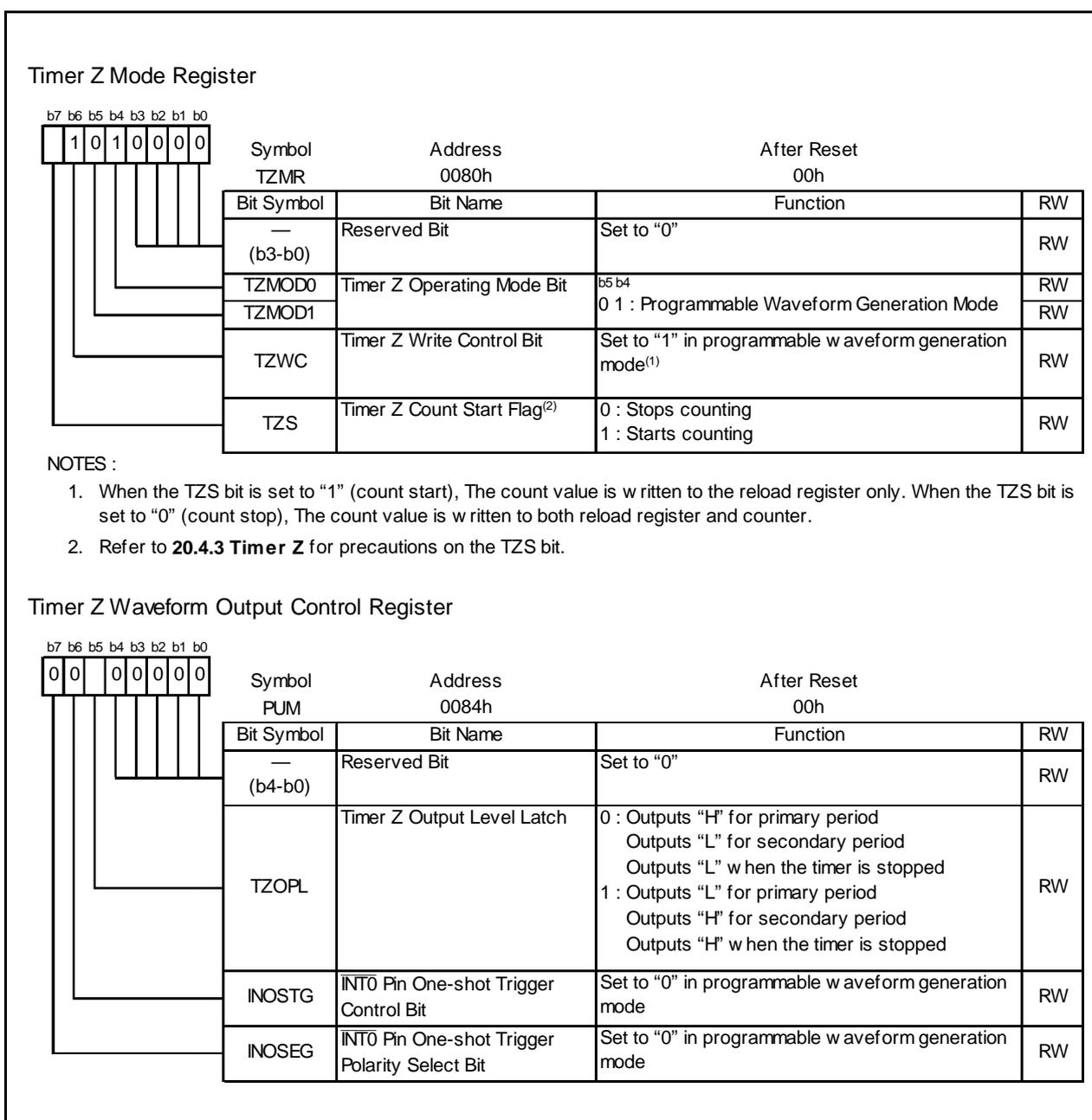


Figure 13.17 TZMR and PUM Registers in Programmable Waveform Generation Mode

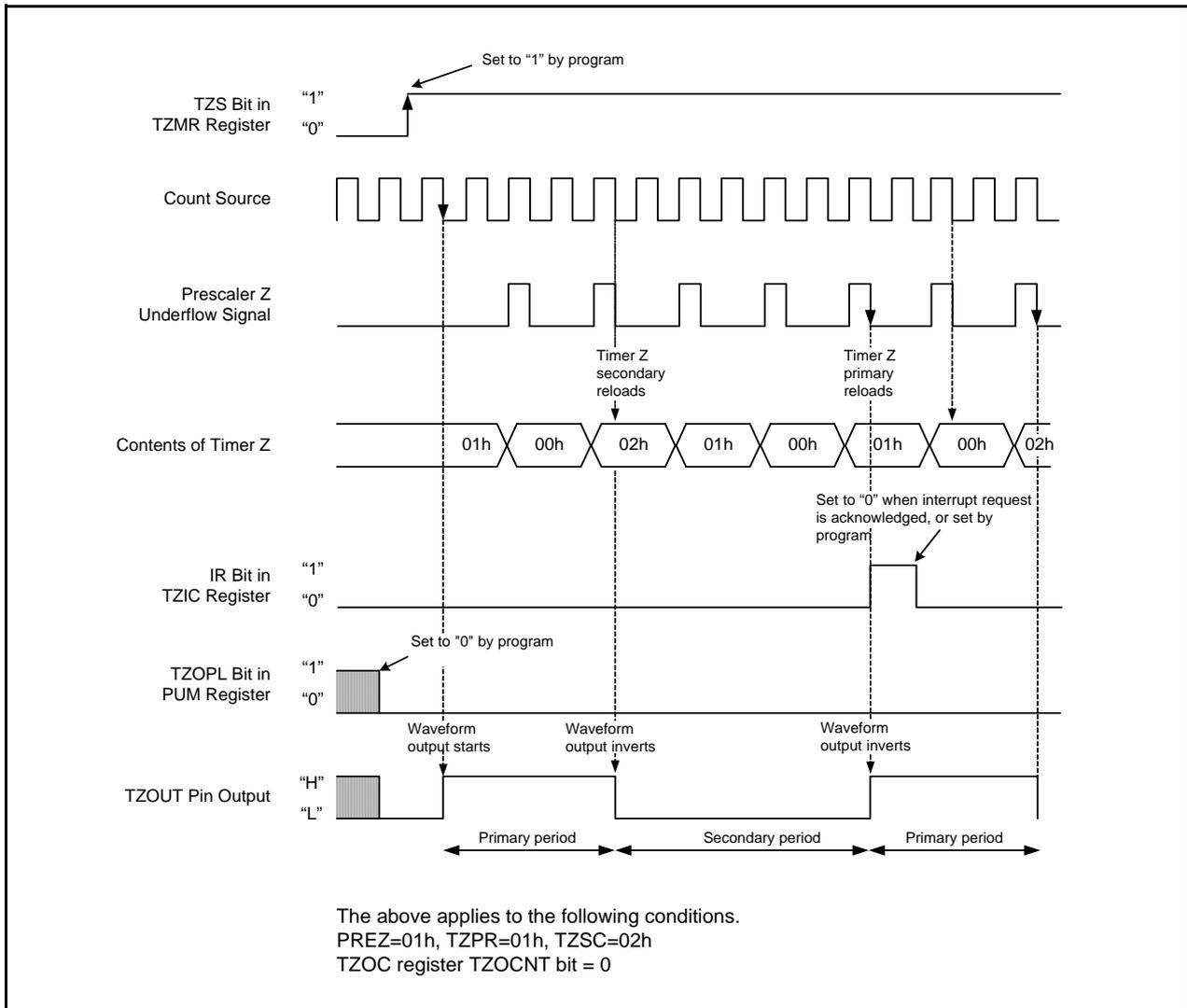


Figure 13.18 Operating Example of Timer Z in Programmable Waveform Generation Mode

13.2.3 Programmable One-shot Generation Mode

Programmable one-shot generation mode is mode to output the one-shot pulse from the TZOUT pin by a program or an external trigger input (input to the $\overline{\text{INT0}}$ pin). (see **Table 13.9 Specification of Programmable One-Shot Generation Mode**). 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 TZPR register. The TZSC register is unused in this mode. Figure 13.19 shows the TZMR and PUM Registers in Programmable One-Shot Generation Mode. Figure 13.20 shows an Operating Example in Programmable One-Shot Generation Mode.

Table 13.9 Specification of Programmable One-Shot Generation Mode

| Item | Specification |
|---------------------------------------|---|
| Count Source | f1, f2, f8, Timer X underflow |
| Count Operation | <ul style="list-style-type: none"> Decrement the setting value in the TZPR register When the timer underflows, it reloads the contents of the reload register before the count completes and the TZOS bit is set to "0" (one-shot stops). When a 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 PREZ register, m: setting value in TZPR register |
| Count Start Condition | <ul style="list-style-type: none"> Set the TZOS bit in the TZOC register to "1" (one-shot starts) ⁽¹⁾ Input active trigger to the $\overline{\text{INT0}}$ pin⁽²⁾ |
| Count Stop Condition | <ul style="list-style-type: none"> When reloading completes after the count value is set to "00h" When the TZS bit in the TZMR register is set to "0" (count stops) When the TZOS bit in the TZOC register is set to "0" (one-shot stops) |
| Interrupt Request Generation Timing | In half cycles of count source, after the timer underflows (at the same time as the TZOUT output ends) [Timer Z interrupt] |
| TZOUT Pin Function | Pulse output (When using this function as a programmable I/O port, set to timer mode.) |
| $\overline{\text{INT0}}$ Pin Function | <ul style="list-style-type: none"> When the INOSTG bit in the PUM register is set to "0" ($\overline{\text{INT0}}$ one-shot trigger disabled) programmable I/O port or $\overline{\text{INT0}}$ interrupt input When the INOSTG bit in the PUM register is set to "1" ($\overline{\text{INT0}}$ one-shot trigger enabled) external trigger ($\overline{\text{INT0}}$ interrupt input) |
| Read from Timer | The count value can be read out by reading the TZPR and PREZ registers. |
| Write to Timer | The value written to the TZPR and PREZ registers is written to the reload register only ⁽³⁾ . |
| Select Function | <ul style="list-style-type: none"> Output level latch select function The TZOPL bit can select the output level of the one-shot pulse waveform. $\overline{\text{INT0}}$ pin one-shot trigger control and polarity select functions The INOSTG bit can select the trigger input from the $\overline{\text{INT0}}$ pin is active or inactive. Also, the INOSEG bit can select the active trigger polarity. |

NOTES:

- Set the TZS bit in the TZMR register to "1" (count starts).
- Set the TZS bit to "1" (count starts), the $\overline{\text{INT0}}$ EN bit in the INTEN register to "1" (enables $\overline{\text{INT0}}$ input), and the INOSTG bit in the PUM register to "1" ($\overline{\text{INT0}}$ one-shot trigger enabled). A trigger which is input during the count cannot be acknowledged, however the $\overline{\text{INT0}}$ interrupt request is generated.
- The set value is reflected at the following one-shot pulse after writing to the TZPR register.

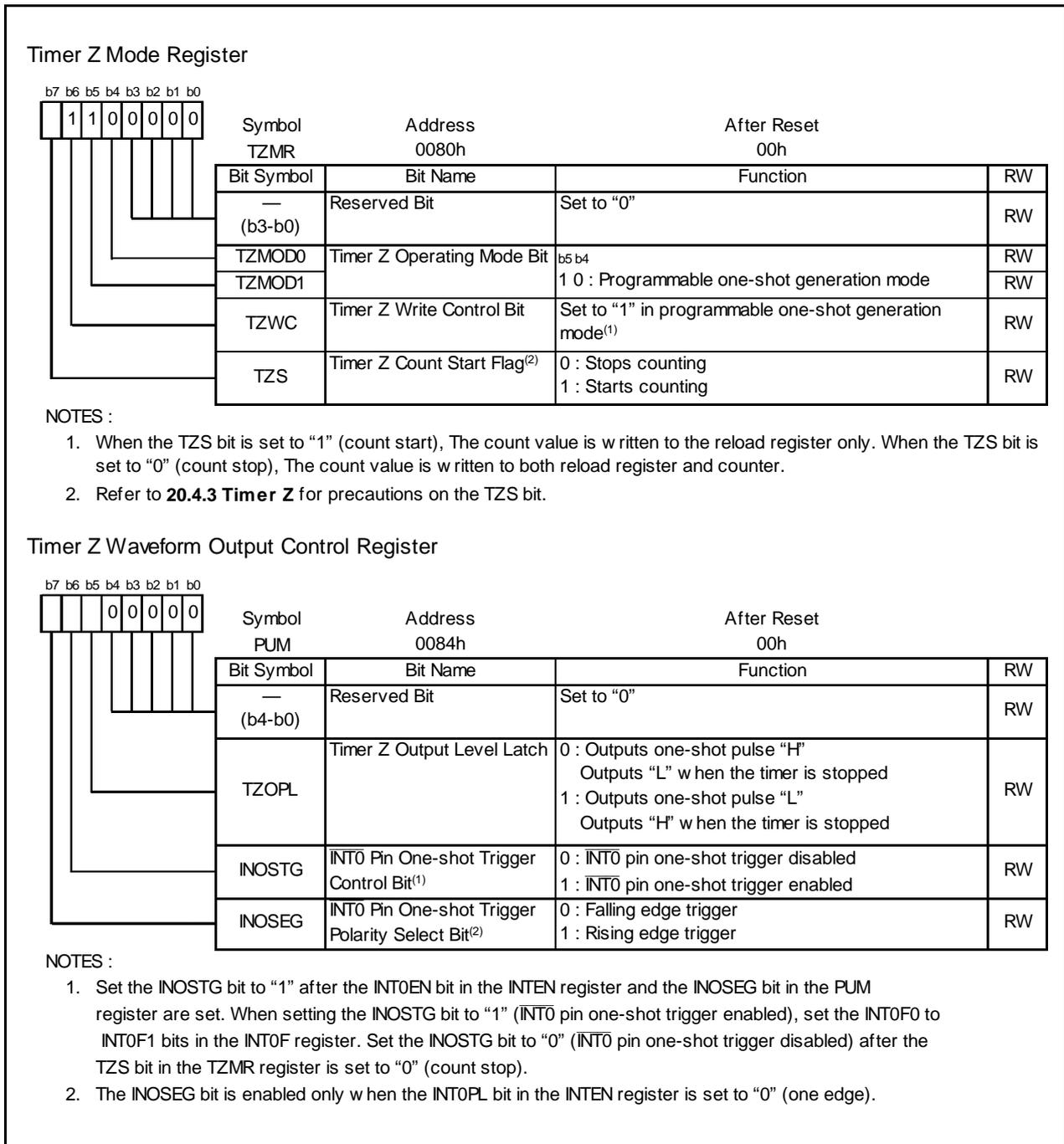


Figure 13.19 TZMR and PUM Registers in Programmable One-Shot Generation Mode

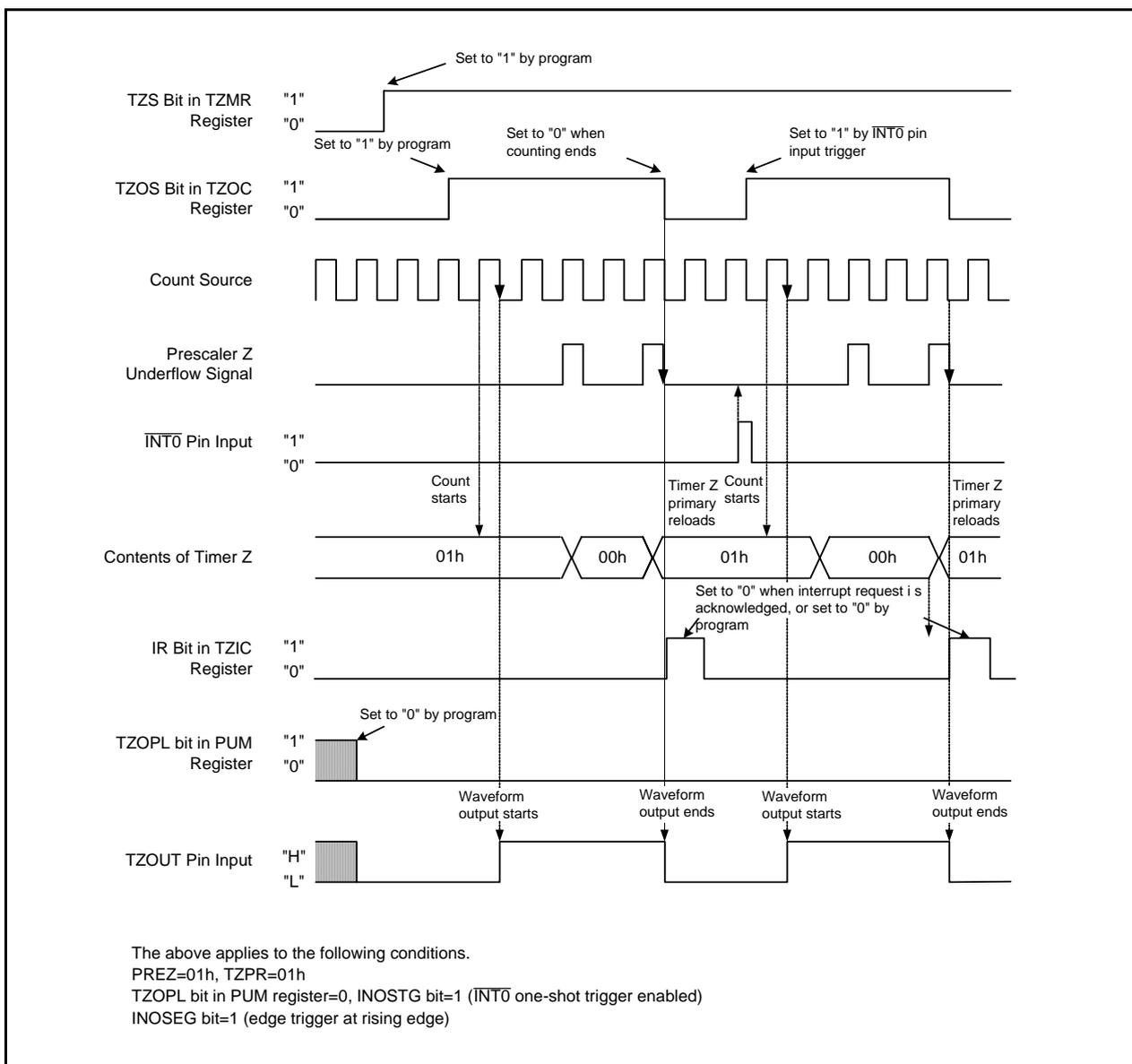


Figure 13.20 Operating Example in Programmable One-Shot Generation Mode

13.2.4 Programmable Wait One-shot Generation Mode

Programmable wait one-shot generation mode is mode to output the one-shot pulse from the TZOUT pin by the external trigger input (input to the INT0 pin) (see **Table 13.10 Specification of Programmable Wait One-Shot Generation Mode Specifications**). When a trigger is generated from this point, the timer starts outputting pulses only once for a given length of time equal to the setting value in the TZSC register after waiting for a given length of time equal to the setting value in the TZPR register. Figure 13.21 shows the TZMR and PUM Registers in Programmable Wait One-Shot Generation Mode. Figure 13.22 shows an Operating Example in Programmable Wait One-Shot Generation Mode.

Table 13.10 Specification of Programmable Wait One-Shot Generation Mode Specifications

| Item | Specification |
|-------------------------------------|---|
| Count Source | f1, f2, f8, Timer X underflow |
| Count Operation | <ul style="list-style-type: none"> • Decrement the setting value in Timer Z primary • When a count of TZPR register underflows, the timer reloads the contents of the TZSC register before the count continues. • When a count of the TZSC register underflows, the timer reloads the contents of the TZPR register before the count completes and the TZOS bit is set to "0". • When a 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 : setting value in PREZ register, m : setting value in TZPR register |
| One-Shot Pulse Output Time | $(n+1)(p+1)/f_i$ f_i : Count source frequency n : setting value in PREZ register, p : setting value in TZSC register |
| Count Start Condition | <ul style="list-style-type: none"> • Set the TZOS bit in the TZOC register to "1" (one-shot starts)⁽¹⁾ • Input active trigger to the $\overline{INT0}$ pin⁽²⁾ |
| Count Stop Condition | <ul style="list-style-type: none"> • When reloading completes after Timer Z underflows during secondary period • When the TZS bit in the TZMR register is set to "0" (count stops) • When the TZOS bit in the TZOC register is set to "0" (one-shot stops) |
| Interrupt Request Generation Timing | In half cycles of the count source after Timer Z underflows during secondary period (complete at the same time as waveform output from the TZOUT pin) [timer Z interrupt] |
| TZOUT Pin Function | Pulse output (When using this function as a programmable I/O port, set to timer mode.) |
| $\overline{INT0}$ Pin Function | <ul style="list-style-type: none"> • When the INOSTG bit in the PUM register is set to "0" ($\overline{INT0}$ one-shot trigger disabled), programmable I/O port or $\overline{INT0}$ interrupt input • When the INOSTG bit in the PUM register is set to "1" ($\overline{INT0}$ one-shot trigger enabled), external trigger ($\overline{INT0}$ interrupt input) |
| Read from Timer | The count value can be read out by reading the TZPR and PREZ registers. |
| Write to Timer | The value written to the TZPR register, PREZ and TZSC register is written to reload register only ⁽³⁾ . |
| Select Function | <ul style="list-style-type: none"> • Output level latch select function The output level for the one-shot pulse waveform is selected by the TZOPL bit. • $\overline{INT0}$ pin one-shot trigger control function and polarity select function Trigger input from the $\overline{INT0}$ pin can be set to active or inactive by the INOSTG bit. Also, an active trigger's polarity can be selected by the INOSEG bit. |

NOTES:

1. Set the TZS bit in the TZMR register to "1" (count starts).
2. Set the TZS bit to "1" (count starts), the $\overline{INT0EN}$ bit in the INTEN register to "1" (enables $\overline{INT0}$ input), and the INOSTG bit in the PUM register to "1" (enabling $\overline{INT0}$ one-shot trigger). A trigger which is input during the count cannot be acknowledged, however the $\overline{INT0}$ interrupt request is generated.
3. The setting values are reflected beginning with the following one-shot pulse after writing to the TZPR register.

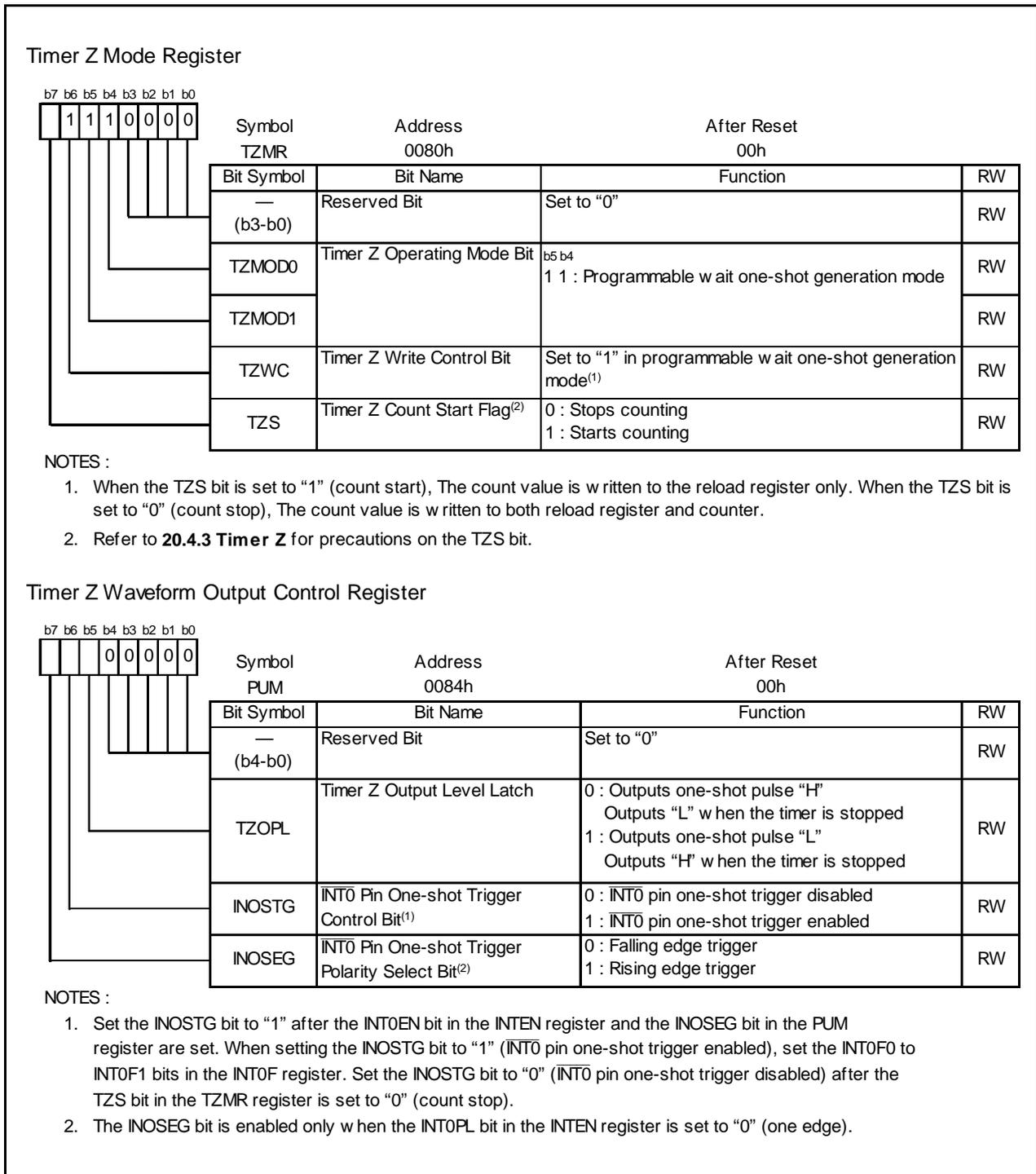


Figure 13.21 TZMR and PUM Registers in Programmable Wait One-Shot Generation Mode

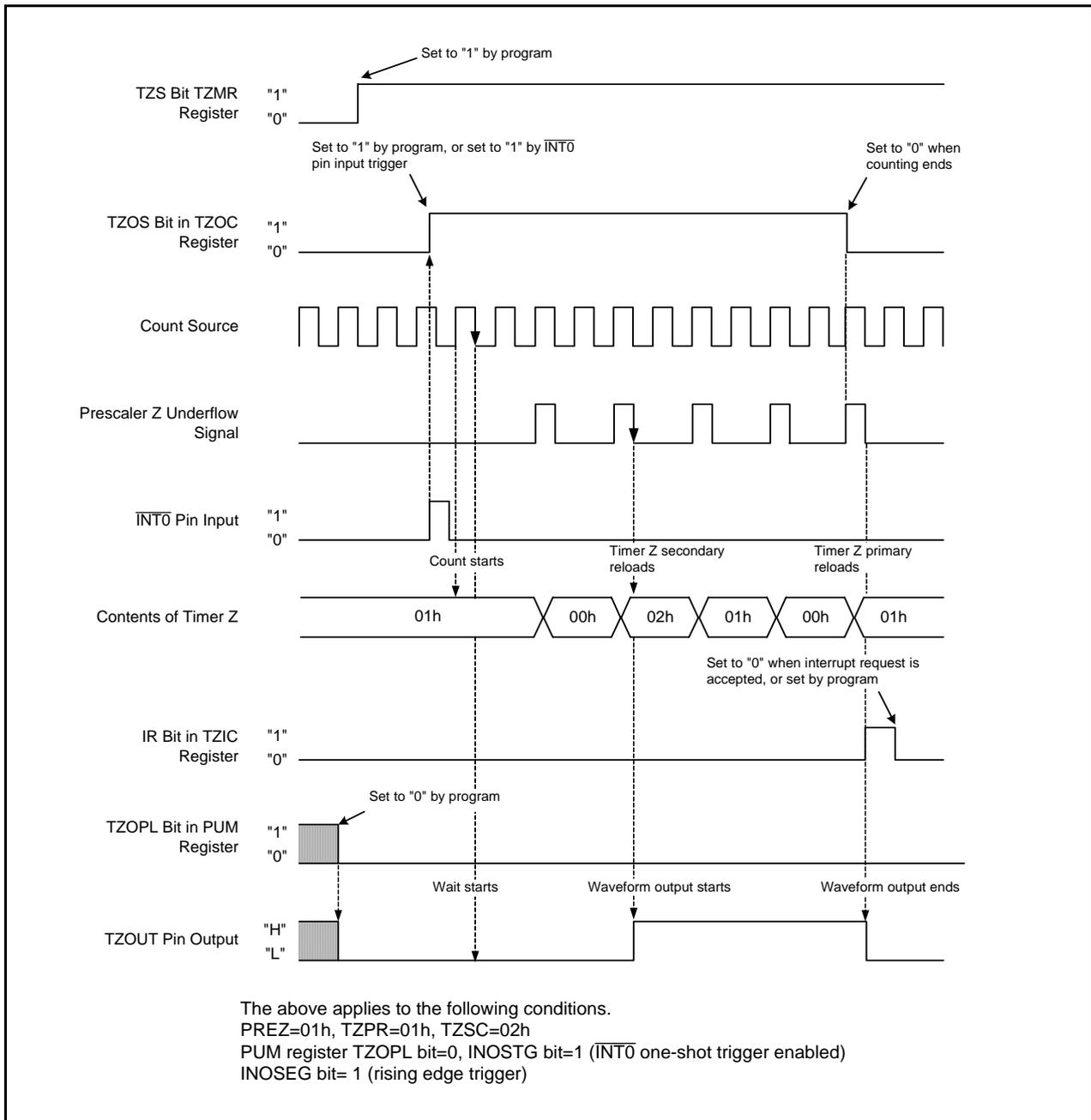


Figure 13.22 Operating Example in Programmable Wait One-Shot Generation Mode

13.3 Timer C

Timer C is a 16-bit timer. Figure 13.23 shows the Block Diagram of Timer C. Figure 13.24 shows the Block Diagram of CMP Waveform Generation Unit. Figure 13.25 shows the Block Diagram of CMP Waveform Output Unit.

Timer C has two modes: input capture mode and output compare mode. Figure 13.26 to 13.29 show the Timer C-associated registers.

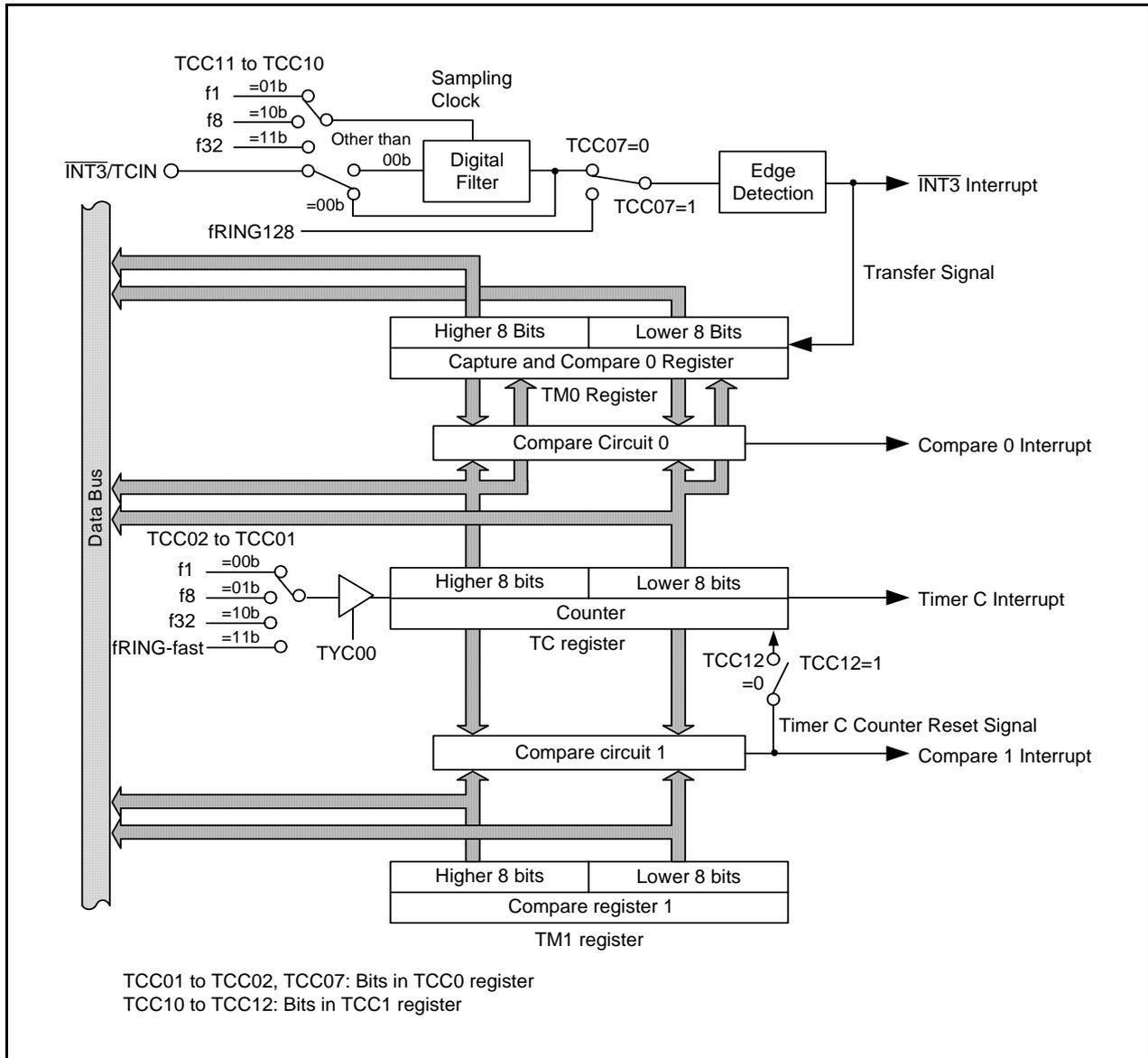


Figure 13.23 Block Diagram of Timer C

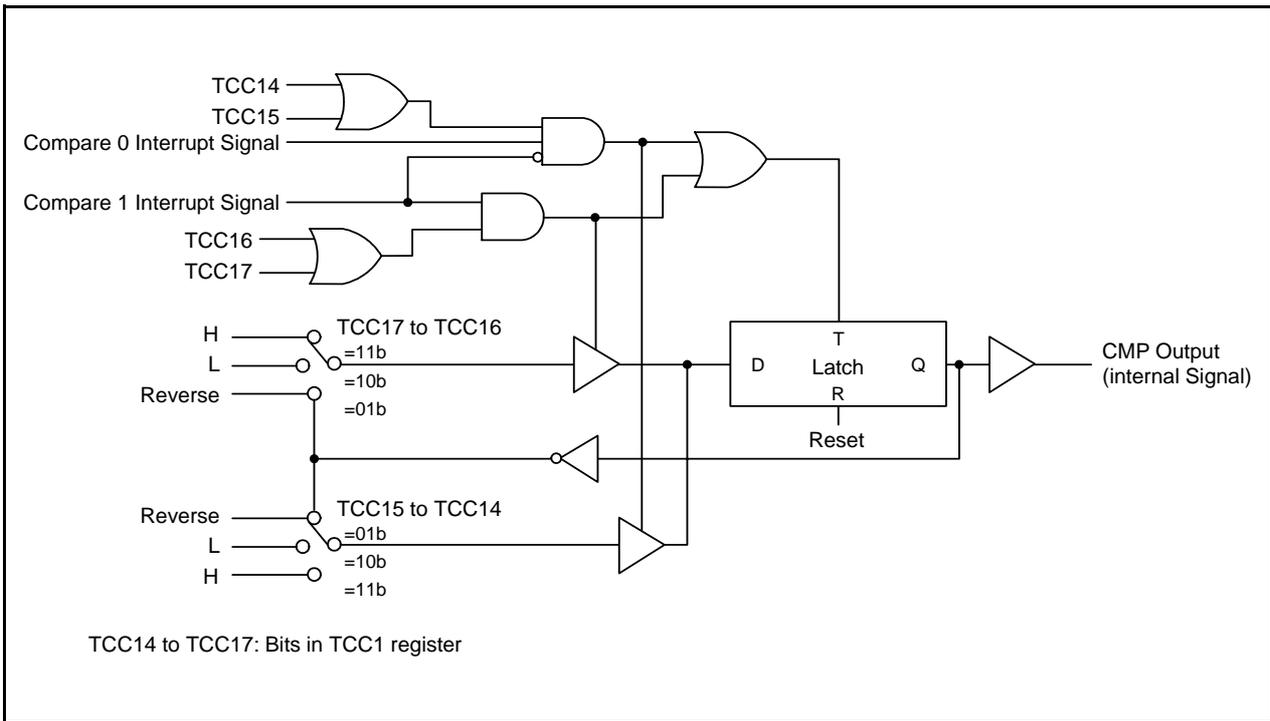


Figure 13.24 Block Diagram of CMP Waveform Generation Unit

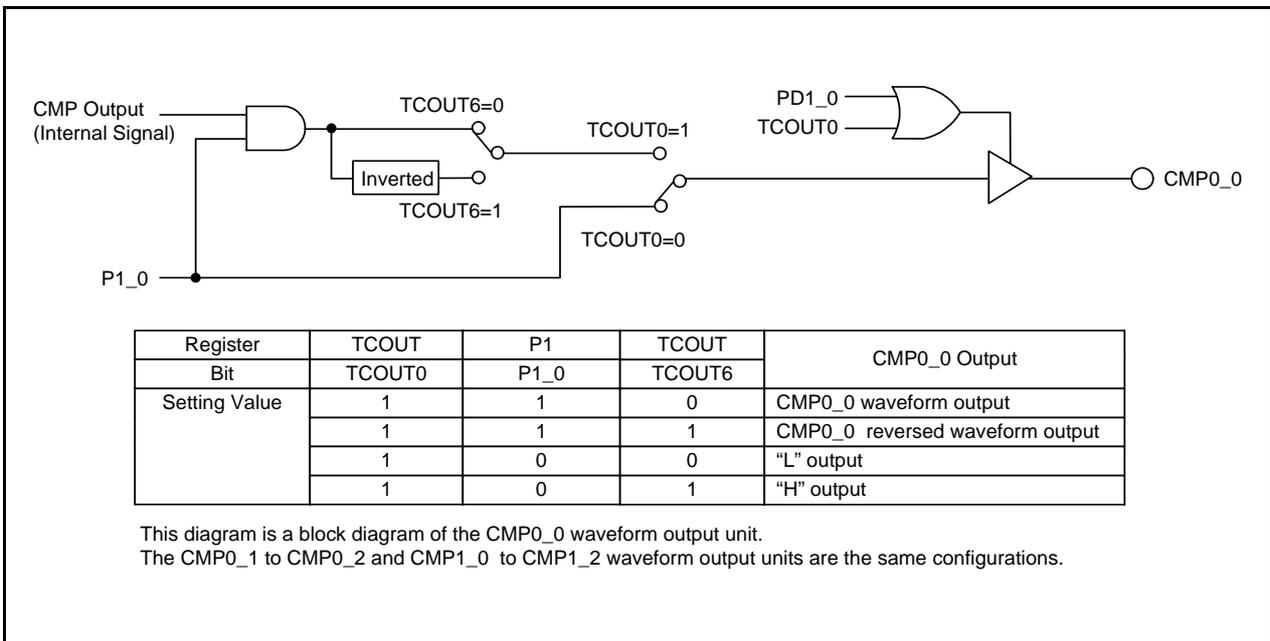


Figure 13.25 Block Diagram of CMP Waveform Output Unit

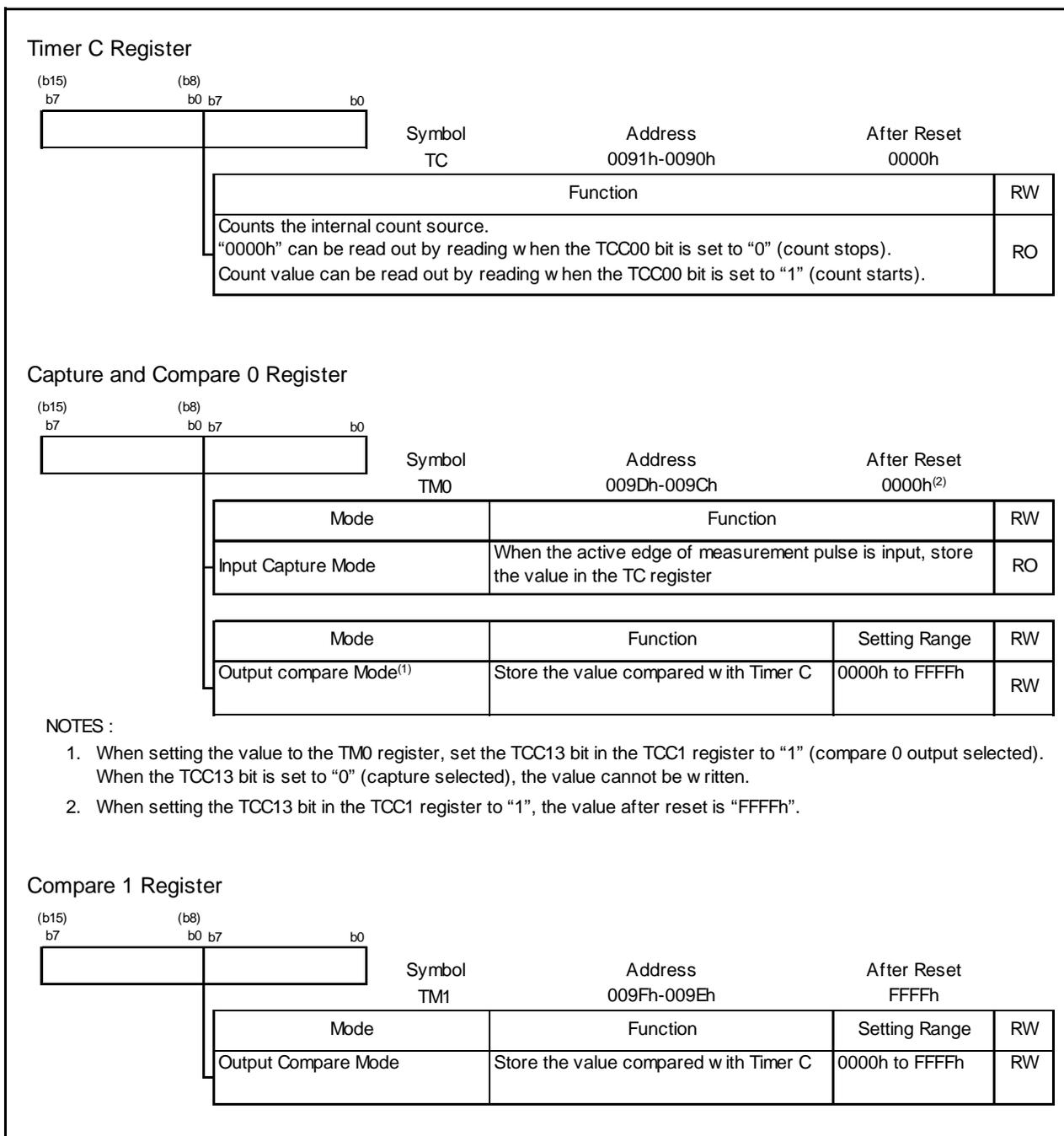


Figure 13.26 TC, TM0 and TM1 Registers

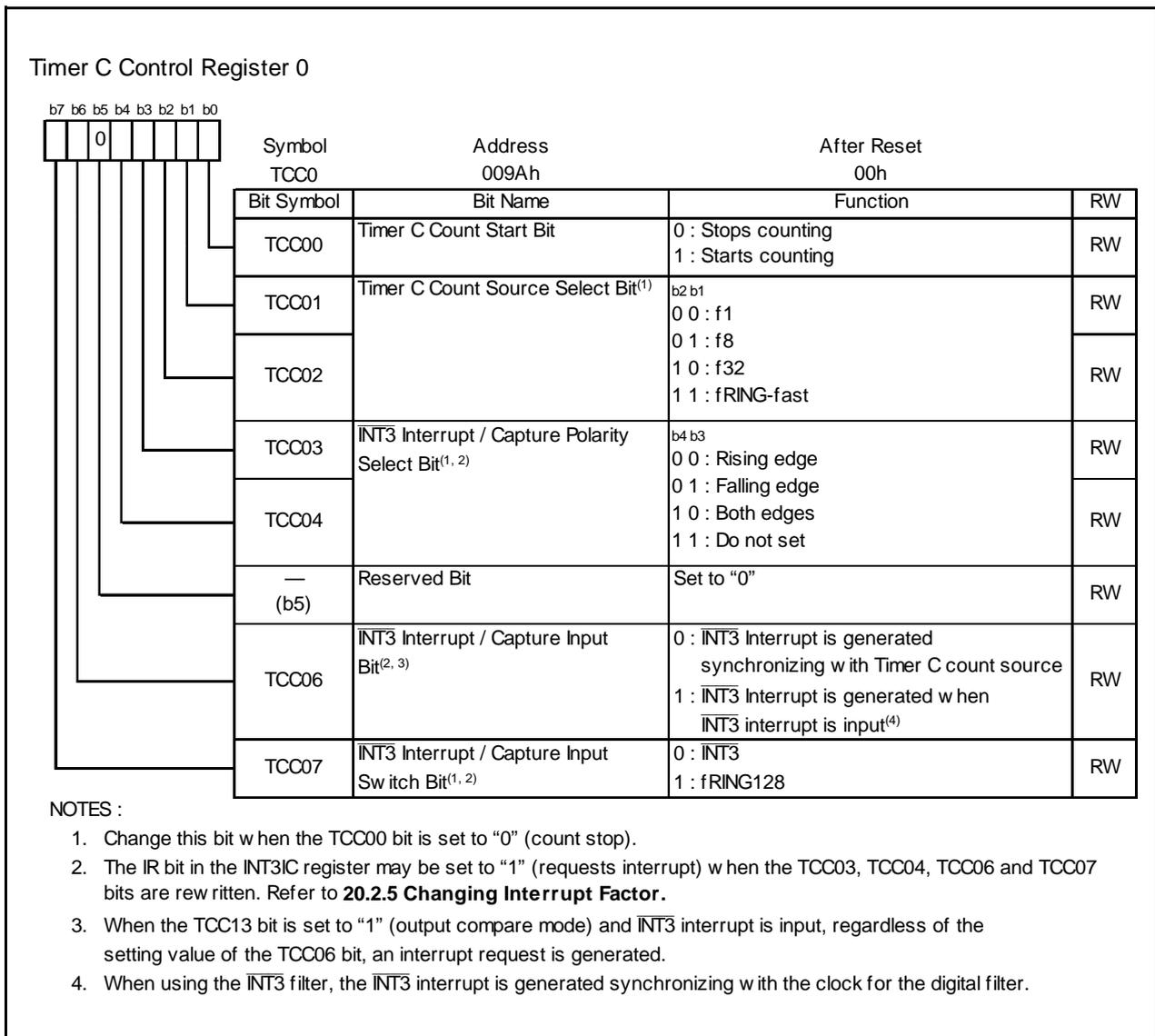


Figure 13.27 TCC0 Register

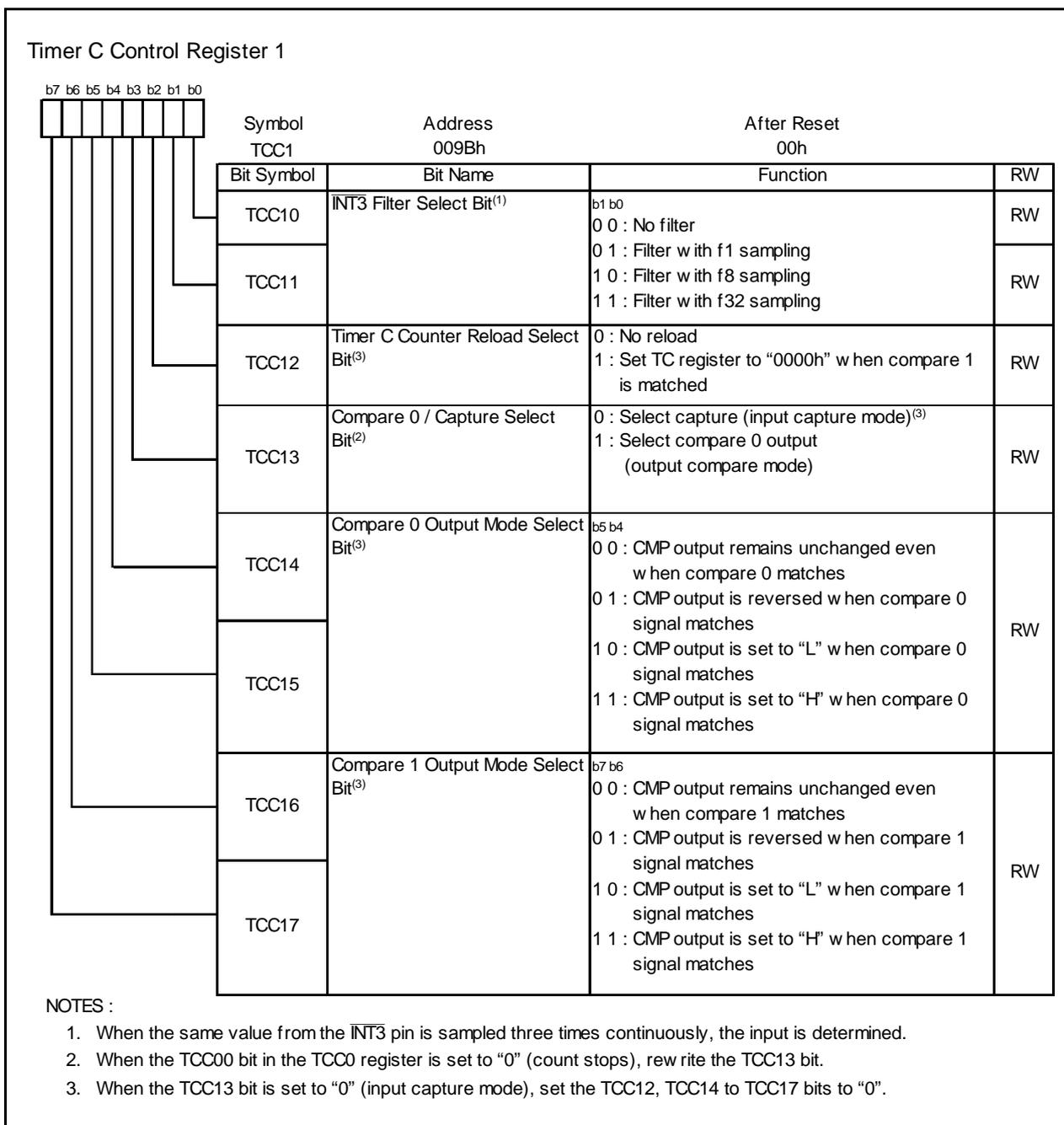


Figure 13.28 TCC1 Register

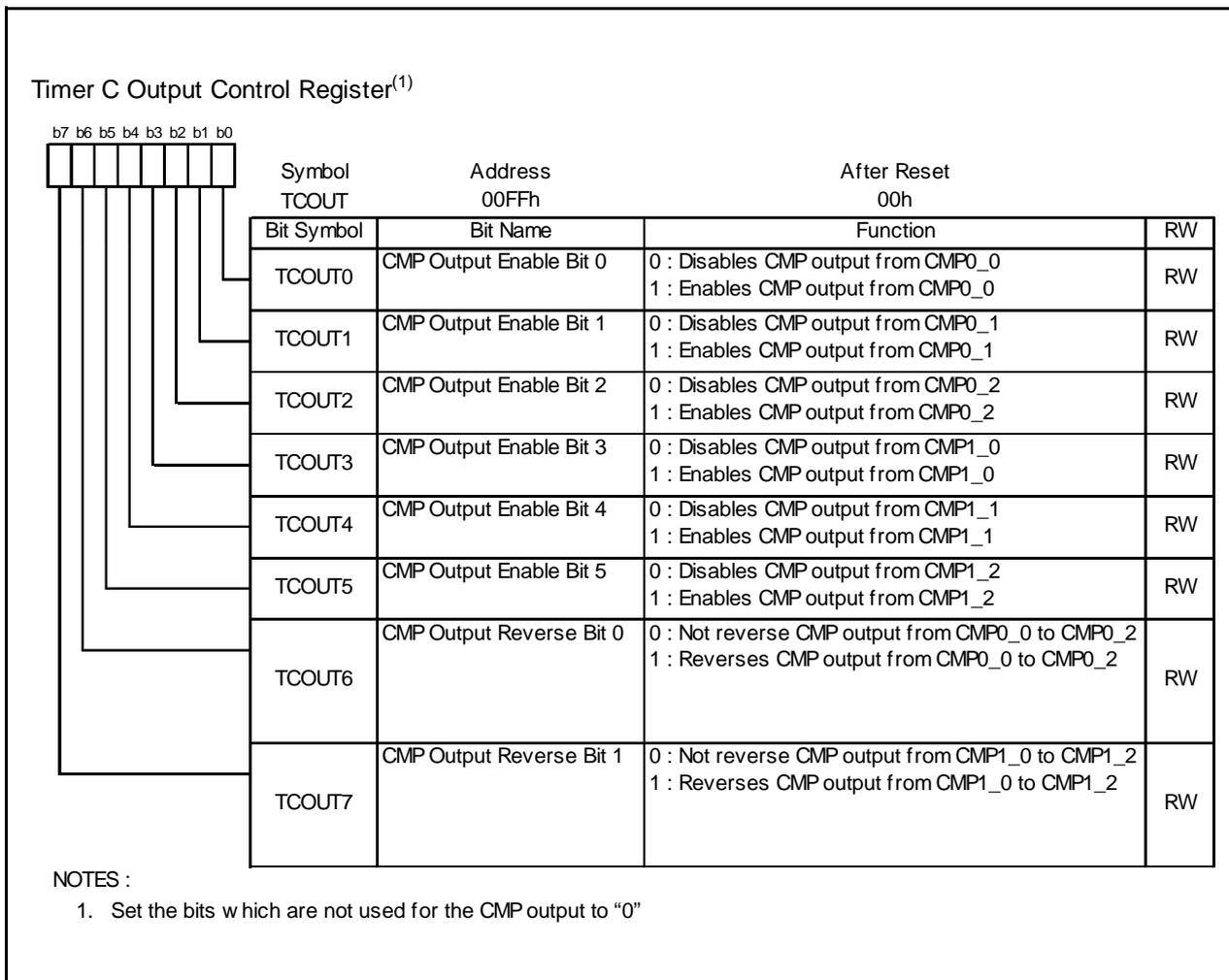


Figure 13.29 TCOUT Register

13.3.1 Input Capture Mode

Input capture mode is mode to input an edge to the TCIN pin or the fRING128 clock as trigger to latch the timer value and generates an interrupt request. The TCIN input contains a digital filter and this prevents an error caused by noise or so on from occurring. Table 13.11 shows Specification of Input Capture Mode. Figure 13.30 shows an Operating Example in Input Capture Mode.

Table 13.11 Specification of Input Capture Mode

| Item | Specification |
|---|--|
| Count Source | f1, f8, f32, fRING-fast |
| Count Operation | <ul style="list-style-type: none"> • Increment • Transfer the value in the TC register to the TM0 register at the active edge of measurement pulse • The value in the TC register is set to "0000h" when the count stops |
| Count Start Condition | The TCC00 bit in the TCC0 register is set to "1" (count starts) |
| Counter Stop Condition | The TCC00 bit in the TCC0 register is set to "0" (count stops) |
| Interrupt Request Generation Timing | <ul style="list-style-type: none"> • When the active edge of measurement pulse is input [$\overline{\text{INT3}}$ interrupt]⁽¹⁾ • When Timer C overflows [Timer C interrupt] |
| $\overline{\text{INT3}}$ /TCIN Pin Function | Programmable I/O port or measurement pulse input ($\overline{\text{INT3}}$ interrupt input) |
| P1_0 to P1_2, P3_3 to P3_5 Pin Function | Programmable I/O port |
| Counter Value Reset Timing | When the TCC00 bit in the TCC0 register is set to "0" (capture disabled) |
| Read from Timer ⁽²⁾ | <ul style="list-style-type: none"> • The count value can be read out by reading the TC register. • The count value at measurement pulse active edge input can be read out by reading the TM0 register |
| Write to Timer | Write to the TC and TM0 registers is disabled |
| Select Function | <ul style="list-style-type: none"> • $\overline{\text{INT3}}$/TCIN polarity select function The TCC03 to TCC04 bits can select the active edge of measurement pulse • Digital filter function The TCC11 to TCC10 bits can select the digital filter sampling frequency • Trigger select function The TCC07 bit can select the TCIN input or the fRING128 |

NOTES:

1. The digital filter delay and one count source (max.) delay are generated for the $\overline{\text{INT3}}$ interrupt.
2. Read the TC and TM0 registers in 16-bit unit.

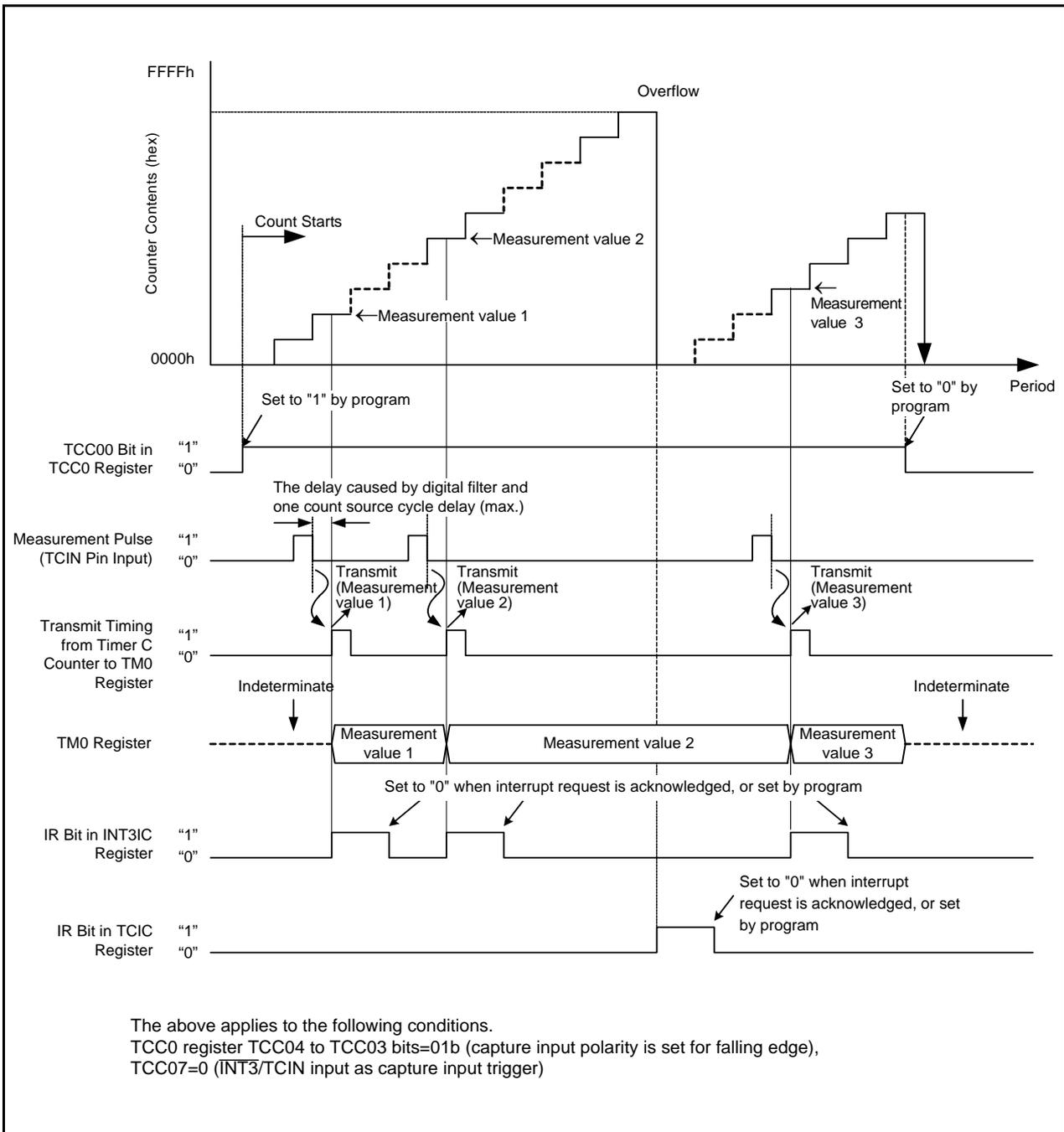


Figure 13.30 Operating Example in Input Capture Mode

13.3.2 Output Compare Mode

Output compare mode is mode to generate an interrupt request when the value of the TC register matches the value of the TM0 or TM1 register. Table 13.12 shows Specification of Output Compare Mode. Figure 13.31 shows an Operating Example in Output Compare Mode.

Table 13.12 Specification of Output Compare Mode

| Item | Specification |
|--|--|
| Count Source | f1, f8, f32, fRING-fast |
| Count Operation | <ul style="list-style-type: none"> • Increment • The value in the TC register is set to "0000h" when a count stops |
| Count Start Condition | The TCC00 bit in the TCC0 register is set to "1" (count starts) |
| Counter Stop Condition | The TCC00 bit in the TCC0 register is set to "0" (count stops) |
| Waveform Output Start Condition | The TCOUT0 to TCOUT5 bits in the TCOUT register is set to "1" (enables CMP output). ⁽²⁾ |
| Waveform Output Stop Condition | The TCOUT0 to TCOUT5 bits in the TCOUT register is set to "0" (disables CMP output). |
| Interrupt Request Generation Timing | <ul style="list-style-type: none"> • When a match occurs in the compare circuit 0 [compare 0 interrupt] • When a match occurs in the compare circuit 1 [compare 1 interrupt] • When Time C overflows [Timer C interrupt] |
| INT3/TCIN Pin Function | Programmable I/O port or INT3 interrupt input |
| P1_0 to P1_2 Pins and P3_0 to P3_2 Pins Function | Programmable I/O port or CMP output ⁽¹⁾ |
| Counter Value Reset Timing | When the TCC00 bit in the TCC0 register is set to "0" (count stops) |
| Read from Timer ⁽¹⁾ | <ul style="list-style-type: none"> • The value in the compare register can be read out by reading the TM0 and TM1 registers. • The count value can be read out by reading the TC register. |
| Write to Timer ⁽¹⁾ | <ul style="list-style-type: none"> • Write to the TC register is disabled. • The values written to the TM0 and TM1 registers are stored in the compare register at the following timings: <ul style="list-style-type: none"> - When the TM0 and TM1 registers are written if the TCC00 bit is set to "0" (count stops) - When the counter overflows if the TCC00 bit is set to "1" (during counting) and the TCC12 bit in the TCC1 register is set to "0" (free-run) - When the compare 1 matches a counter if the TCC00 bit is set to "1" and the TCC12 bit is set to "1" (set the TC register to "0000h" when the compare 1 matches) |
| Select Function | <ul style="list-style-type: none"> • Timer C counter reload select function The TCC12 bit in the TCC1 register can select whether the counter value in the TC register is set to "0000h" when the compare circuit 1 matches or not. • The TCC14 to TCC15 bits in the TCC1 register can select the output level when the compare circuit 0 matches. The TCC16 to TCC17 bits in the TCC1 register can select the output level when the compare circuit 1 matches. • The TCOUT6 to TCOUT7 bits in the TCOUT register can select whether the output is reversed or not. |

NOTES:

1. When the corresponding port data is "1", the waveform is output depending on the setting of the registers TCC1 and TCOUT. When the corresponding port data is "0", the fixed level is output (refer to **Figure 13.25 Block Diagram of CMP Waveform Output Unit**).
2. Access the TC, TM0, and TM1 registers in 16-bit units.

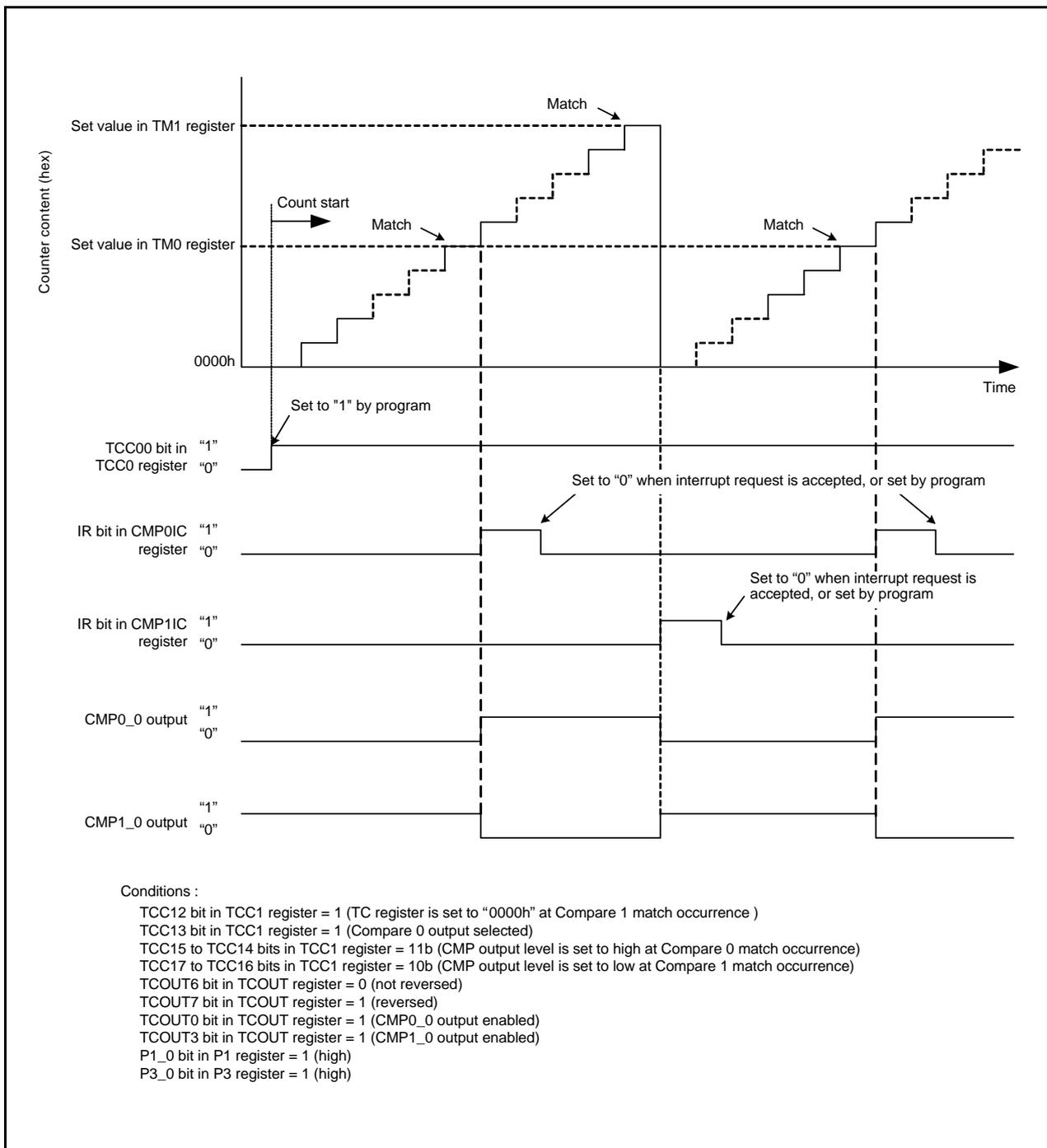


Figure 13.31 Operating Example in Output Compare Mode

14. Serial Interface

Serial Interface is configured with one channels: UART0. UART0 has an exclusive timer to generate a transfer clock.

Figure 14.1 shows a UART0 Block Diagram. Figure 14.2 shows a UART0 Transmit/Receive Unit.

UART0 has two modes: clock synchronous serial I/O mode, and clock asynchronous serial I/O mode (UART mode).

Figure 14.3 to 14.5 show the UART0-associated registers.

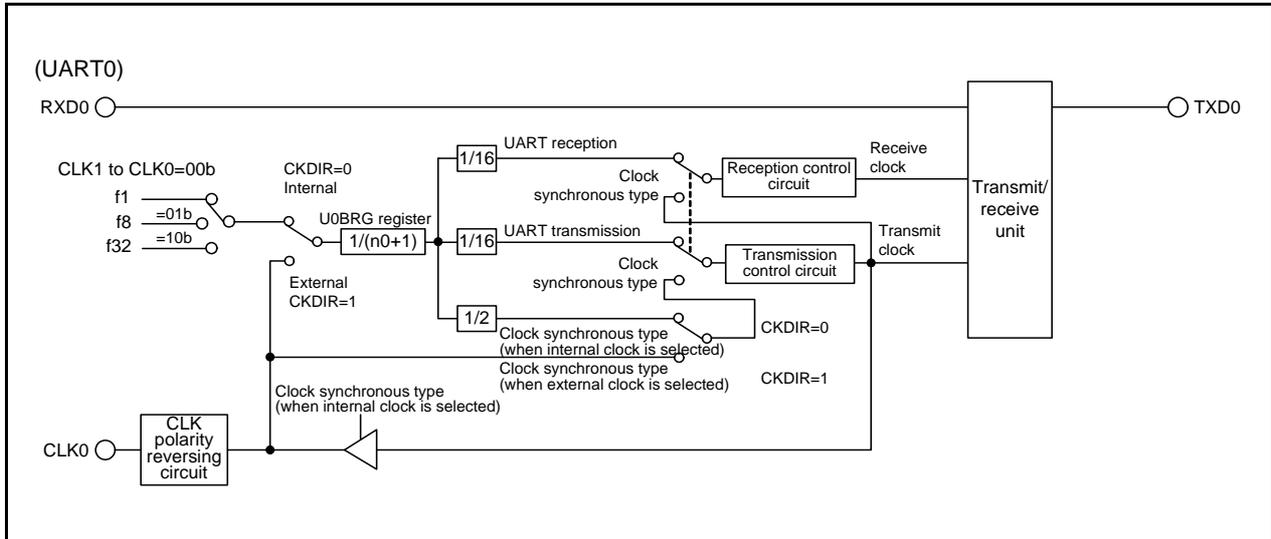


Figure 14.1 UART0 Block Diagram

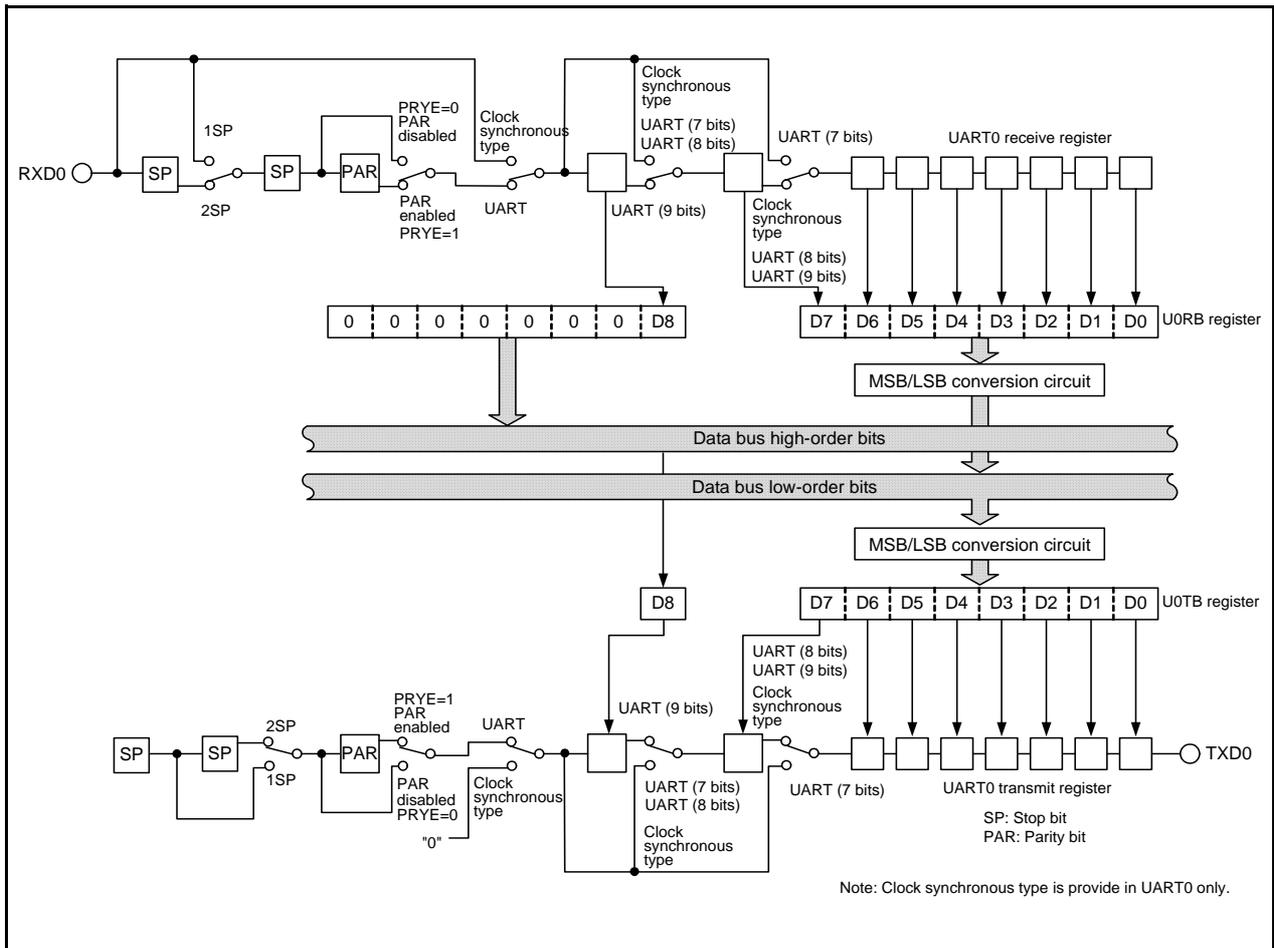


Figure 14.2 UART0 Transmit/Receive Unit

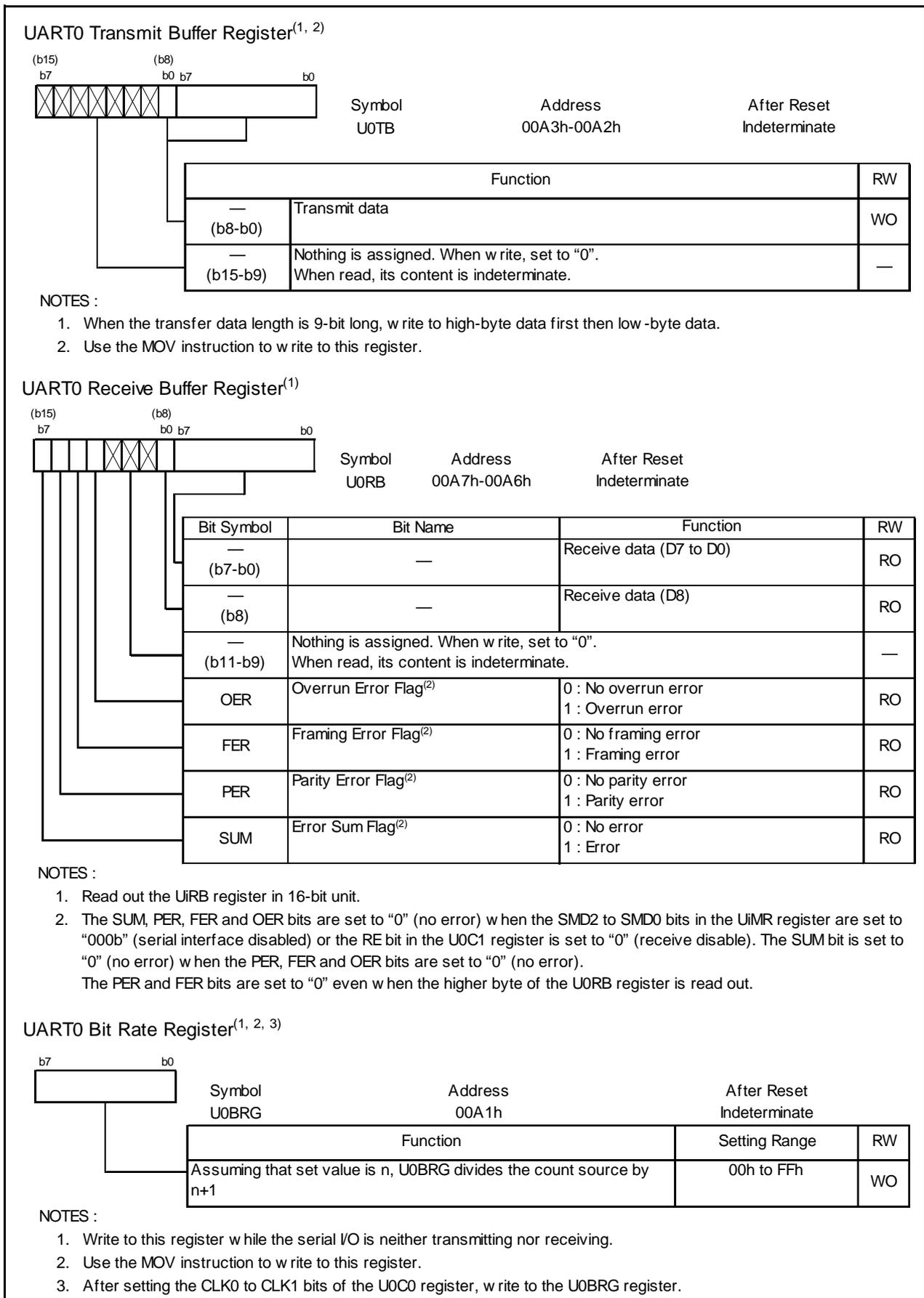


Figure 14.3 U0TB, U0RB and U0BRG Registers

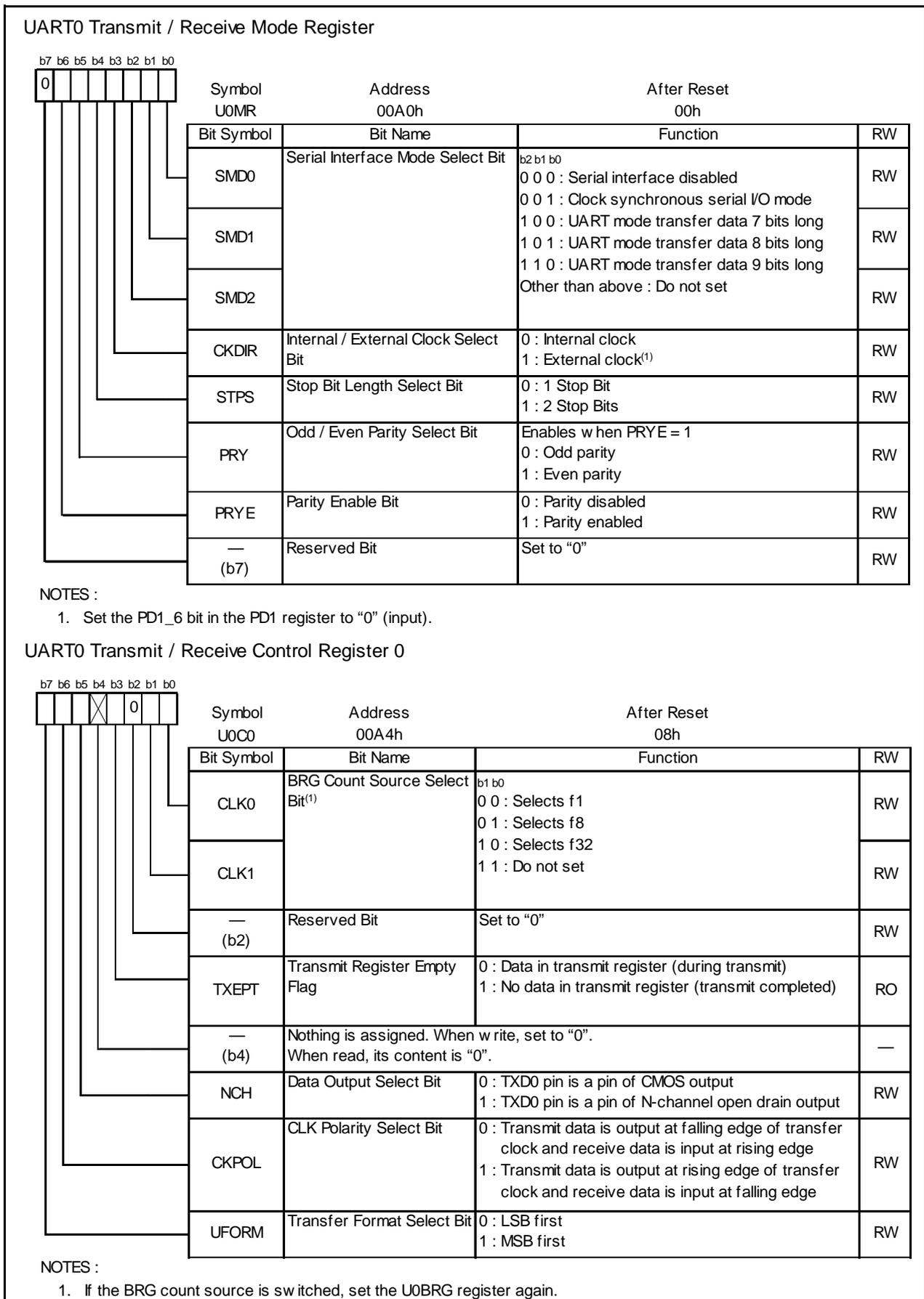


Figure 14.4 U0MR and U0C0 Registers

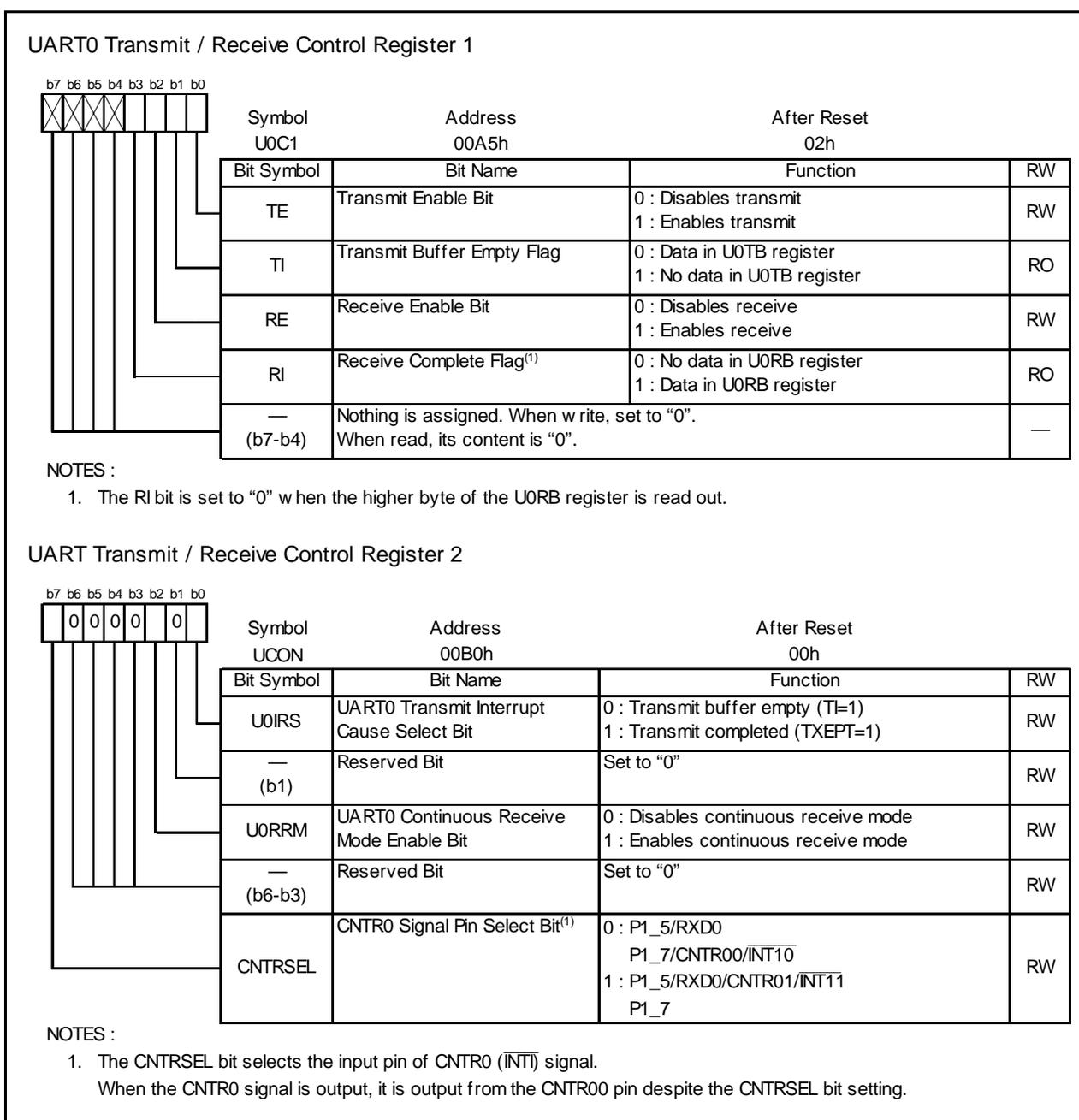


Figure 14.5 UOC1 and UCON Registers

14.1 Clock Synchronous Serial I/O Mode

The clock synchronous serial I/O mode is mode to transmit and receive data using a transfer clock. Table 14.1 lists the Specification of Clock Synchronous Serial I/O Mode. Table 14.2 lists the Registers to Be Used and Settings in Clock Synchronous Serial I/O Mode(1).

Table 14.1 Specification of Clock Synchronous Serial I/O Mode

| Item | Specification |
|-------------------------------------|---|
| Transfer Data Format | <ul style="list-style-type: none"> Transfer data length: 8 bits |
| Transfer Clock | <ul style="list-style-type: none"> CKDIR bit in U0MR register is set to "0" (internal clock): $f_i/(2(n+1))$ $f_i=f_1, f_8, f_{32}$ n=setting value in U0BRG register: 00h to FFh The CKDIR bit is set to "1" (external clock): input from CLK0 pin |
| Transmit Start Condition | <ul style="list-style-type: none"> Before transmit starts, the following requirements are required⁽¹⁾ <ul style="list-style-type: none"> The TE bit in the U0C1 register is set to "1" (transmit enabled) The TI bit in the U0C1 register is set to "0" (data in the U0TB register) |
| Receive Start Condition | <ul style="list-style-type: none"> Before receive starts, the following requirements are required⁽¹⁾ <ul style="list-style-type: none"> The RE bit in the U0C1 register is set to "1" (receive enabled) The TE bit in the U0C1 register is set to "1" (transmit enabled) The TI bit in the U0C1 register is set to "0" (data in the U0TB register) |
| Interrupt Request Generation Timing | <ul style="list-style-type: none"> When transmit, one of the following conditions can be selected <ul style="list-style-type: none"> The U0IRS bit is set to "0" (transmit buffer empty): when transferring data from the U0TB register to UART0 transmit register (when transmit starts) The U0IRS bit is set to "1" (transmit completes): when completing transmit data from UAR<i>T</i>_i transmit register When receive When transferring data from the UART0 receive register to the U0RB register (when receive completes) |
| Error Detection | <ul style="list-style-type: none"> Overrun error⁽²⁾ This error occurs if serial interface starts receiving the following data before reading the U0RB register and receives the 7th bit of the following data |
| Select Function | <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 beginning with the bit 0 or beginning with the bit 7 can be selected Continuous receive mode selection Receive is enabled immediately by reading the U0RB register |

NOTES:

- When an external clock is selected, meet the conditions while the CKPOL bit in the U0C0 register is set to "0" (transmit data output at the falling edge and the receive data input at the rising edge of the transfer clock), the external clock is held "H"; if the CKPOL bit in the U0C0 register is set to "1" (transmit data output at the rising edge and the receive data input at the falling edge of the transfer clock), the external clock is held "L".
- If an overrun error occurs, the value of the U0RB register will be indeterminate. The IR bit in the S0RIC register remains unchanged.

Table 14.2 Registers to Be Used and Settings in Clock Synchronous Serial I/O Mode⁽¹⁾

| Register | Bit | Function |
|----------|--------------|--|
| U0TB | 0 to 7 | Set transmit data |
| U0RB | 0 to 7 | Receive data can be read |
| | OER | Overrun error flag |
| U0BRG | 0 to 7 | Set bit rate |
| U0MR | SMD2 to SMD0 | Set to "001b" |
| | CKDIR | Select the internal clock or external clock |
| U0C0 | CLK1 to CLK0 | Select the count source in the U0BRG register |
| | TXEPT | Transmit register empty flag |
| | NCH | Select TXD0 pin output mode |
| | CKPOL | Select the transfer clock polarity |
| | UFORM | Select the LSB first or MSB first |
| U0C1 | TE | Set this bit to "1" to enable transmit/receive |
| | TI | Transmit buffer empty flag |
| | RE | Set this bit to "1" to enable reception |
| | RI | Reception complete flag |
| UCON | U0IRS | Select the factor of UART0 transmit interrupt |
| | U0RRM | Set this bit to "1" to use continuous receive mode |
| | CNTRSEL | Set this bit to "1" to select P1_5/RXD0/CNTR01/INT11 |

NOTES:

1. Set bits which are not in this table to "0" when writing to the registers in clock synchronous serial I/O mode.

Table 14.3 lists the I/O Pin Functions in Clock Synchronous Serial I/O Mode. The TXD0 pin outputs "H" level between the operating mode selection of UART0 and transfer start, an "H" (If the NCH bit is set to "1" (the N-channel open-drain output), this pin is in a high-impedance state.)

Table 14.3 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 receive 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 transmit 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 |

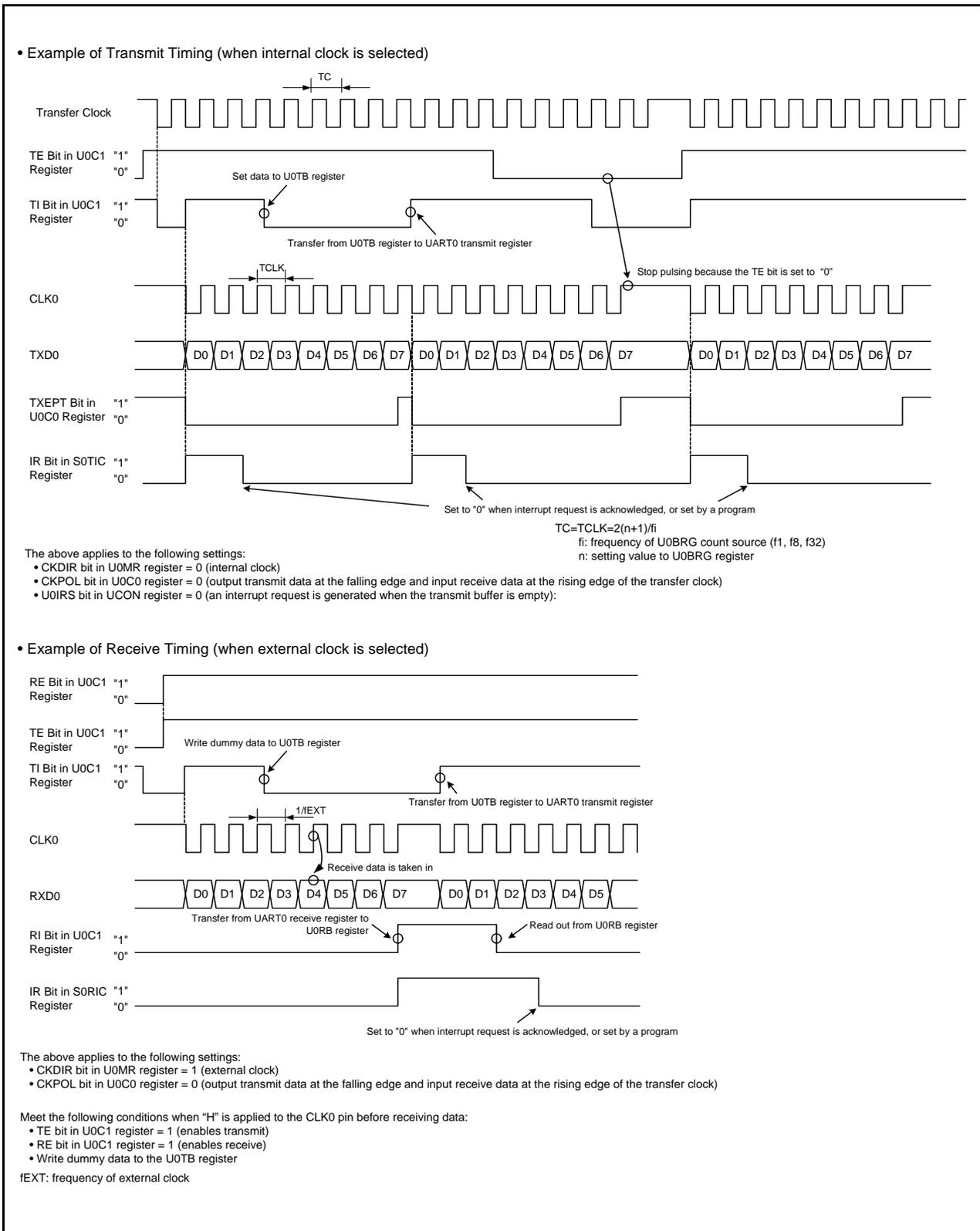


Figure 14.6 Transmit and Receive Operation

14.1.1 Polarity Select Function

Figure 14.7 shows the Transfer Clock Polarity. Use the CKPOL bit in the U0C0 register to select the transfer clock polarity.

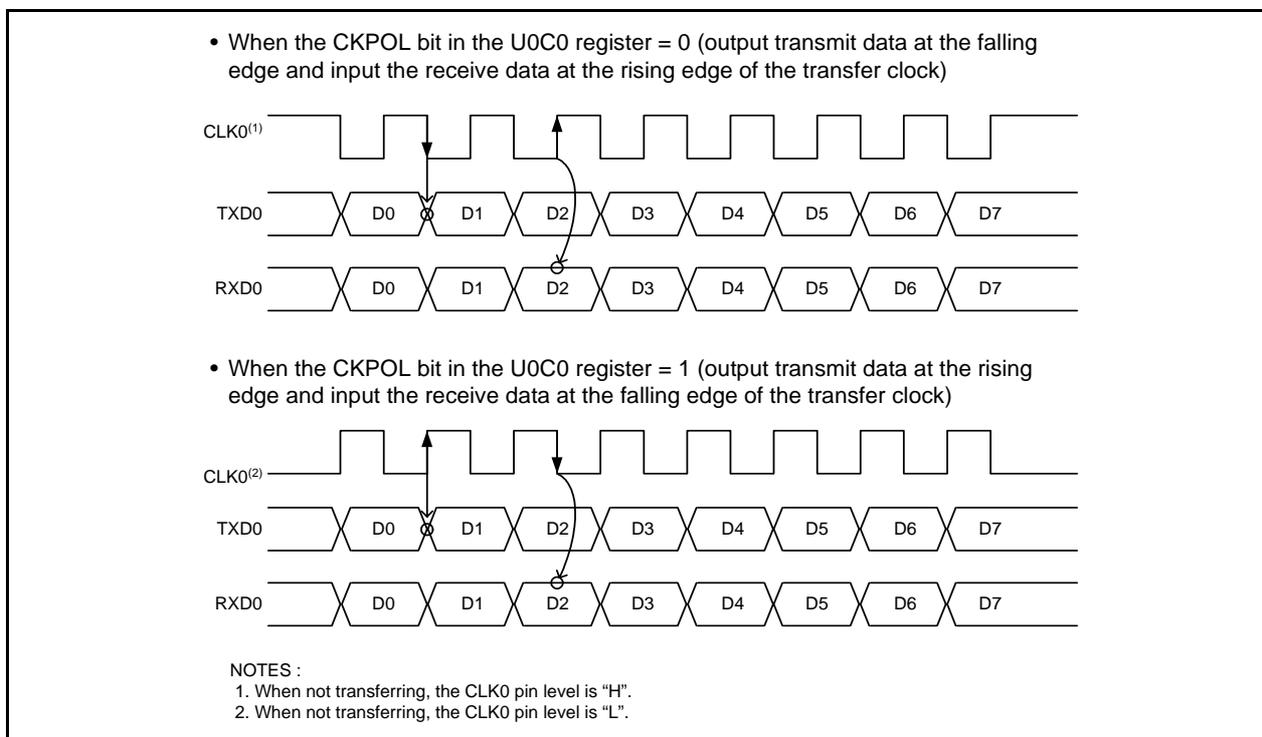


Figure 14.7 Transfer Clock Polarity

14.1.2 LSB First/MSB First Select Function

Figure 14.8 shows the Transfer Format. Use the UFORM bit in the U0C0 register to select the transfer format.

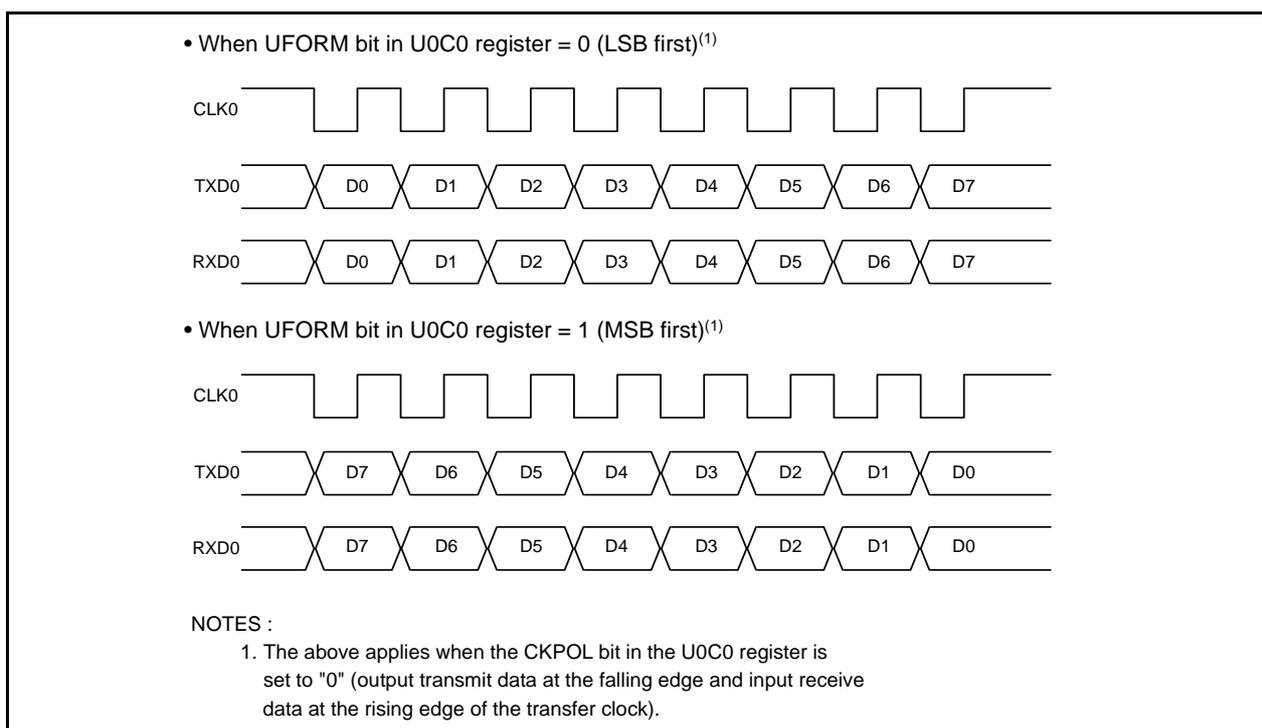


Figure 14.8 Transfer Format

14.1.3 Continuous Receive Mode

Continuous receive mode is held by setting the U0RRM bit in the UCON register to “1” (enables continuous receive mode). In this mode, reading U0RB register sets the TI bit in the U0C1 register to “0” (data in the U0TB register). When the U0RRM bit is set to “1”, do not write dummy data to the U0TB register in a program.

14.2 Clock Asynchronous Serial I/O (UART) Mode

The UART mode allows transmit and receive data after setting the desired bit rate and transfer data format. Table 14.4 lists the Specification of UART Mode. Table 14.5 lists the Registers to Be Used and Settings in UART Mode.

Table 14.4 Specification of UART Mode

| Item | Specification |
|-------------------------------------|--|
| Transfer Data Format | <ul style="list-style-type: none"> • Character bit (transfer data): selectable from 7, 8 or 9 bits • Start bit: 1 bit • Parity bit: selectable from odd, even, or none • Stop bit: selectable from 1 or 2 bits |
| Transfer Clock | <ul style="list-style-type: none"> • CKDIR bit in U0MR register is set to "0" (internal clock) : $f_j/(16(n+1))$ $f_j=f1, f8, f32$ n=setting value in U0BRG register: 00h to FFh • CKDIR bit is set to "1" (external clock) : $f_{EXT}/(16(n+1))$ f_{EXT}: input from CLK0 pin n=setting value in U0BRG register: 00h to FFh |
| Transmit Start Condition | <ul style="list-style-type: none"> • Before transmit starts, the following are required <ul style="list-style-type: none"> - TE bit in U0C1 register is set to "1" (transmit enabled) - TI bit in U0C1 register is set to "0" (data in U0TB register) |
| Receive Start Condition | <ul style="list-style-type: none"> • Before receive starts, the following are required <ul style="list-style-type: none"> - RE bit in U0C1 register is set to "1" (receive enabled) - Detects start bit |
| 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"> - U0IRS bit is set to "0" (transmit buffer empty): when transferring data from the U0TB register to UART0 transmit register (when transmit starts) - U0IRS bit is set to "1" (transfer ends): when serial interface completes transmitting data from the UART0 transmit register • When receiving When transferring data from the UART0 receive register to U0RB register (when receive ends) |
| Error Detection | <ul style="list-style-type: none"> • Overrun error⁽¹⁾ This error occurs if serial interface starts receiving the following data before reading the U0RB register and receiving the bit one before the last stop bit of the following data • Framing error This error occurs when the number of stop bits set are not detected • Parity error This error occurs when parity is enabled, 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 any of the overrun, framing, and parity errors is generated |

NOTES:

1. If an overrun error occurs, the value in the U0RB register will be indeterminate. The IR bit in the S0RIC register remains unchanged.

Table 14.5 Registers to Be Used and Settings in UART Mode

| Register | Bit | Function |
|----------|-----------------|--|
| U0TB | 0 to 8 | Set transmit data ⁽¹⁾ |
| U0RB | 0 to 8 | Receive data can be read ⁽¹⁾ |
| | OER,FER,PER,SUM | Error flag |
| U0BRG | 0 to 7 | Set a bit rate |
| U0MR | SMD2 to SMD0 | Set to "100b" when transfer data is 7-bit long Set to "101b" when transfer data is 8-bit long Set to "110b" when transfer data is 9-bit long |
| | CKDIR | Select the internal clock or external clock |
| | STPS | Select the stop bit |
| | PRY, PRYE | Select whether parity is included and odd or even |
| U0C0 | CLK0, CLK1 | Select the count source for the U0BRG register |
| | TXEPT | Transmit register empty flag |
| | NCH | Select TXD0 pin output mode |
| | CKPOL | Set to "0" |
| | UFORM | LSB first or MSB first can be selected when transfer data is 8-bit long. Set to "0" when transfer data is 7- or 9-bit long. |
| U0C1 | TE | Set to "1" to enable transmit |
| | TI | Transmit buffer empty flag |
| | RE | Set to "1" to enable receive |
| | RI | Receive complete flag |
| UCON | U0IRS, U1IRS | Select the factor of UART0 transmit interrupt |
| | U0RRM | Set to "0" |
| | CNTRSEL | Set to "1" to select P1_5/RXD0/CNTR01/ <u>INT11</u> |

NOTES:

1. The bits used for transmit/receive data are as follows: Bits 0 to 6 when transfer data is 7-bit long; bits 0 to 7 when transfer data is 8-bit long; bits 0 to 8 when transfer data is 9-bit long.

Table 14.6 lists the I/O Pin Functions in Clock Asynchronous Serial I/O Mode. After the UART0 operating mode is selected, the TXD0 pin outputs "H" level (If the NCH bit is set to "1" (N-channel open-drain outputs), this pin is in a high-impedance state) until transfer starts.

Table 14.6 I/O Pin Functions in Clock Asynchronous Serial I/O Mode

| Pin name | Function | Selection Method |
|------------|-----------------------|--|
| TXD0(P1_4) | Output serial data | (Cannot be used as a port when performing receive 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 transmit 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 |

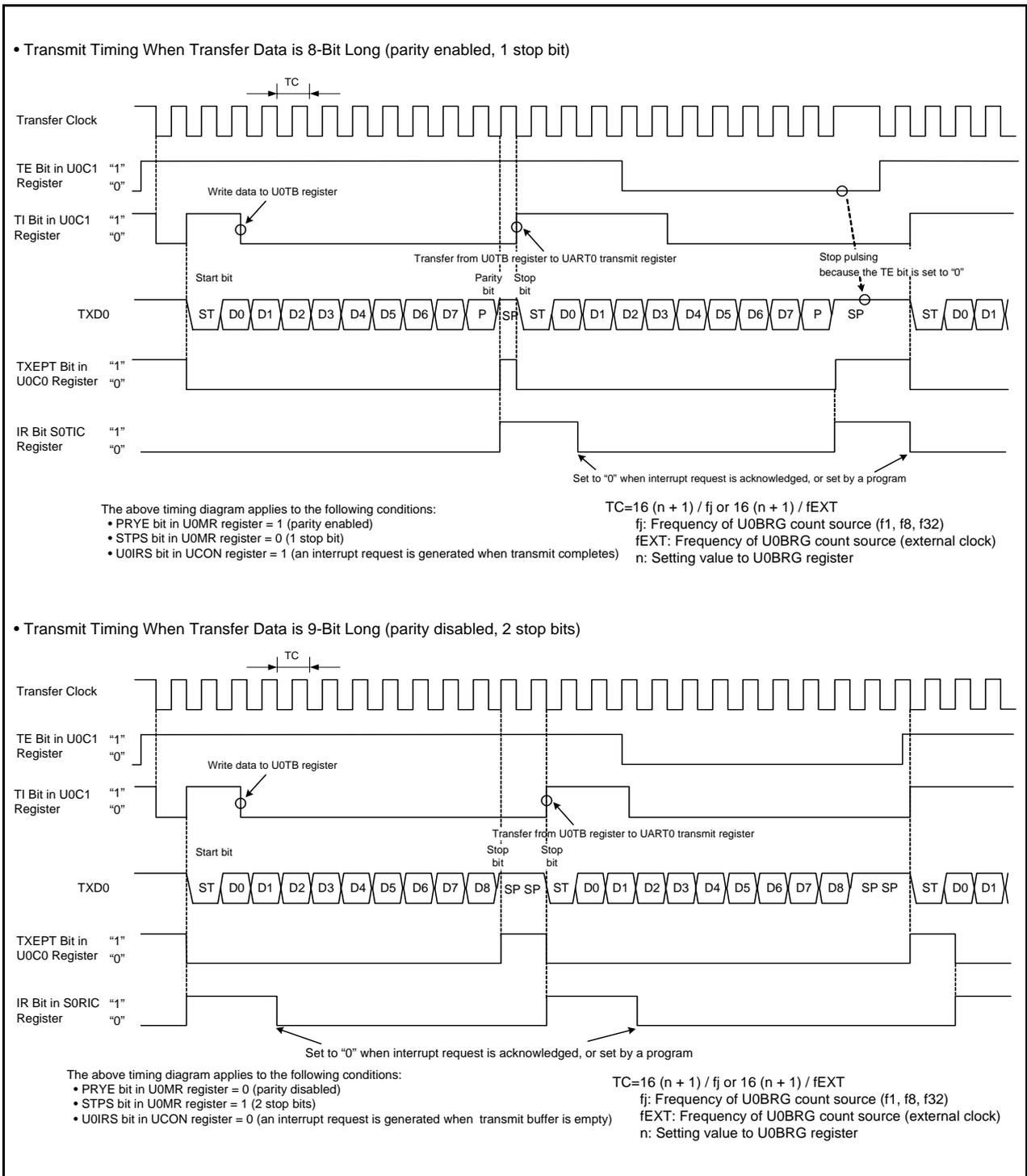


Figure 14.9 Transmit Timing in UART Mode

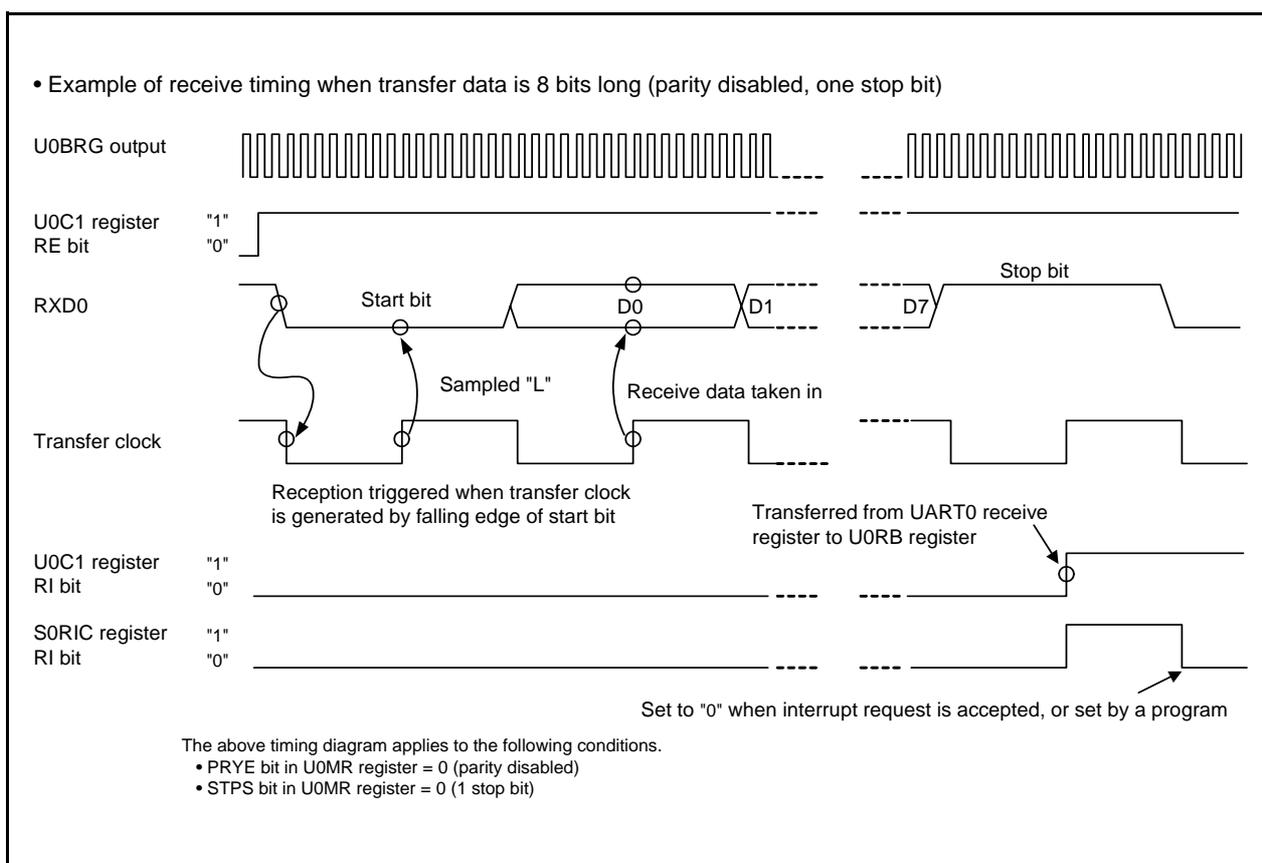


Figure 14.10 Receive Timing in UART Mode

14.2.1 CNTR0 Pin Select Function

The CNTRSEL bit in the UCON register selects whether P1_7 can be used as the CNTR00/ $\overline{\text{INT10}}$ input pin or P1_5 can be used as the CNTR01/ $\overline{\text{INT11}}$ input pin.

When the CNTRSEL bit is set to "0", P1_7 is used as the CNTR00/ $\overline{\text{INT10}}$ pin and when the CNTRSEL bit is set to "1", P1_5 is used as the CNTR01/ $\overline{\text{INT11}}$ pin.

14.2.2 Bit Rate

Divided-by-16 of frequency by the U0BRG register in UART mode is a bit rate.

<UART Mode>

- When selecting internal clock

$$\text{Setting value to the U0BRG register} = \frac{f_j}{\text{Bit Rate} \times 16} - 1$$

F_j : Count source frequency of the U0BRG register (f₁, f₈ and f₃₂)

- When selecting external clock

$$\text{Setting value to the U0BRG register} = \frac{f_{\text{EXT}}}{\text{Bit Rate} \times 16} - 1$$

f_{EXT} : Count source frequency of the U0BRG register (external clock)

Figure 14.11 Calculation Formula of U0BRG Register Setting Value

Table 14.7 Bit Rate Setting Example in UART Mode

| Bit Rate (bps) | BRG Count Source | System Clock = 20MHz | | | System Clock = 8MHz | | |
|----------------|------------------|----------------------|-------------------|----------|---------------------|-------------------|----------|
| | | BRG Setting Value | Actual Time (bps) | Error(%) | BRG 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 |

15. Clock Synchronous Serial I/O with Chip Select (SSU)

The serial data of the clock synchronous can communicate for the clock synchronous serial I/O with chip select (hereinafter referred to as SSU). Table 15.1 shows a SSU Specification and Figure 15.1 shows a Block Diagram of SSU.

Figure 15.2 to 15.8 show SSU Associated Registers.

Table 15.1 SSU Specification

| Item | Specification |
|-----------------------------|---|
| Transfer Data Format | <ul style="list-style-type: none"> Transfer-data length 8 bits Continuous transmit and receive of serial data are enabled since both transmitter and receiver have buffer structure.⁽²⁾ |
| Operating Mode | <ul style="list-style-type: none"> Clock synchronous communication mode 4-wire bus communication mode (including bidirectional communication) |
| Master / Slave Device | Selectable |
| I/O Pin | SSCK (I/O) : Clock I/O pin SSI (I/O) : Data I/O pin SSO (I/O) : Data I/O pin SCS (I/O) : Chip-select I/O pin |
| Transfer Clock | <ul style="list-style-type: none"> When the MSS bit in the SSCRH register is set to "0" (operates as slave device), external clock can be selected. When the MSS bit in the SSCRH register is set to "1" (operates as master device), internal clock (selects from $\phi/256$, $\phi/128$, $\phi/64$, $\phi/32$, $\phi/16$, $\phi/8$ and $\phi/4$ and outputs from SSCK pin) can be 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 receive and completes by error. While the RDRF bit in the SSSR register is set to "1" (data in the SSRDR register) and completing the next serial data receive, the ORER bit is set to "1". |
| Multimaster Error Detection | <ul style="list-style-type: none"> Conflict error While 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 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 SCS pin input changes state from "L" to "H", the CE bit in the SSSR register is set to "1". |
| Interrupt Request | 5 interrupt requests (transmit-end, transmit-data-empty, receive-data-full, overrun error and conflict error). ⁽¹⁾ |
| Select Function | <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 |

NOTES:

1. The interrupt vector table is one of the SSU.
2. When setting to the slave device, do not transmit continuously.

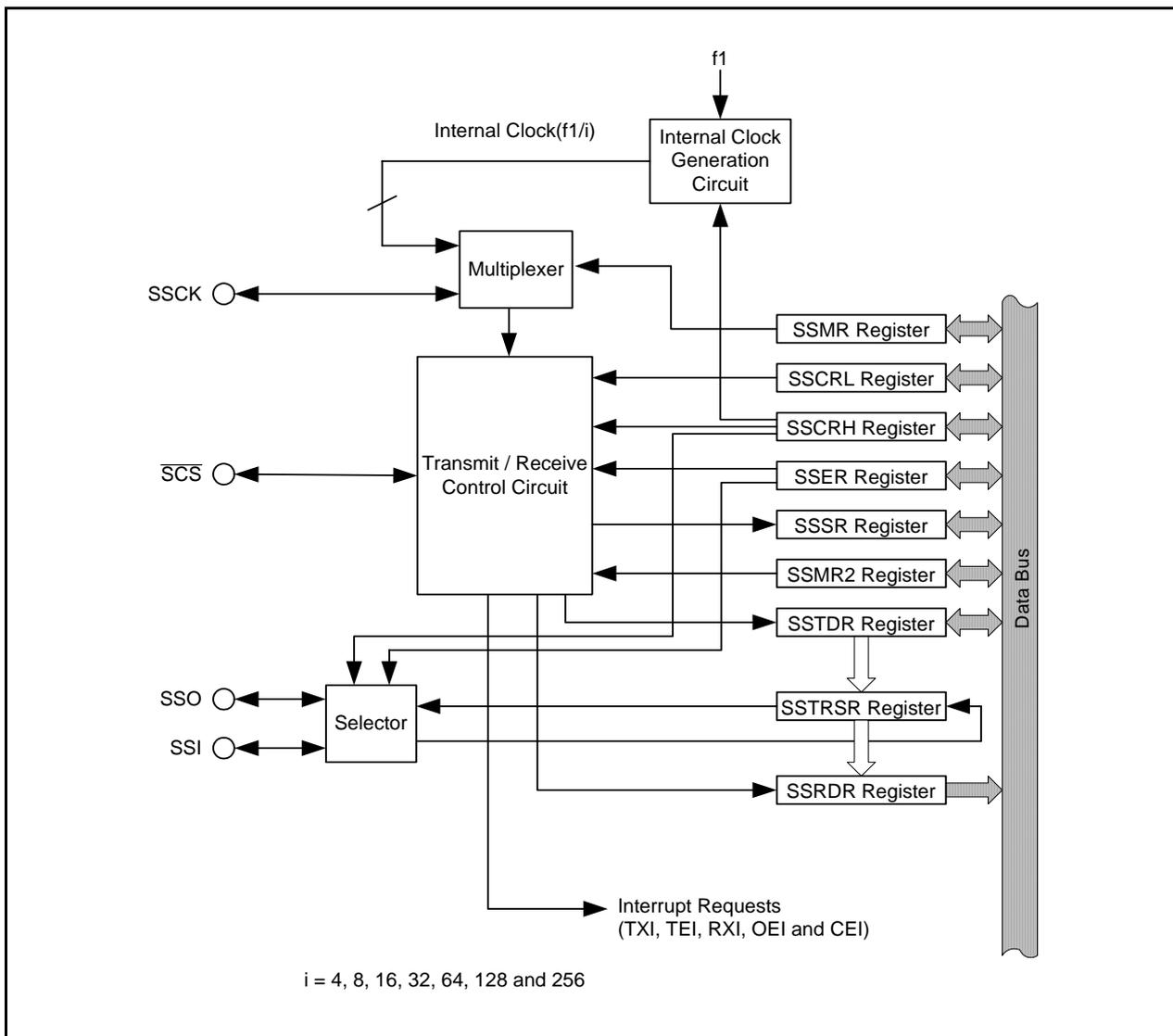


Figure 15.1 Block Diagram of SSU

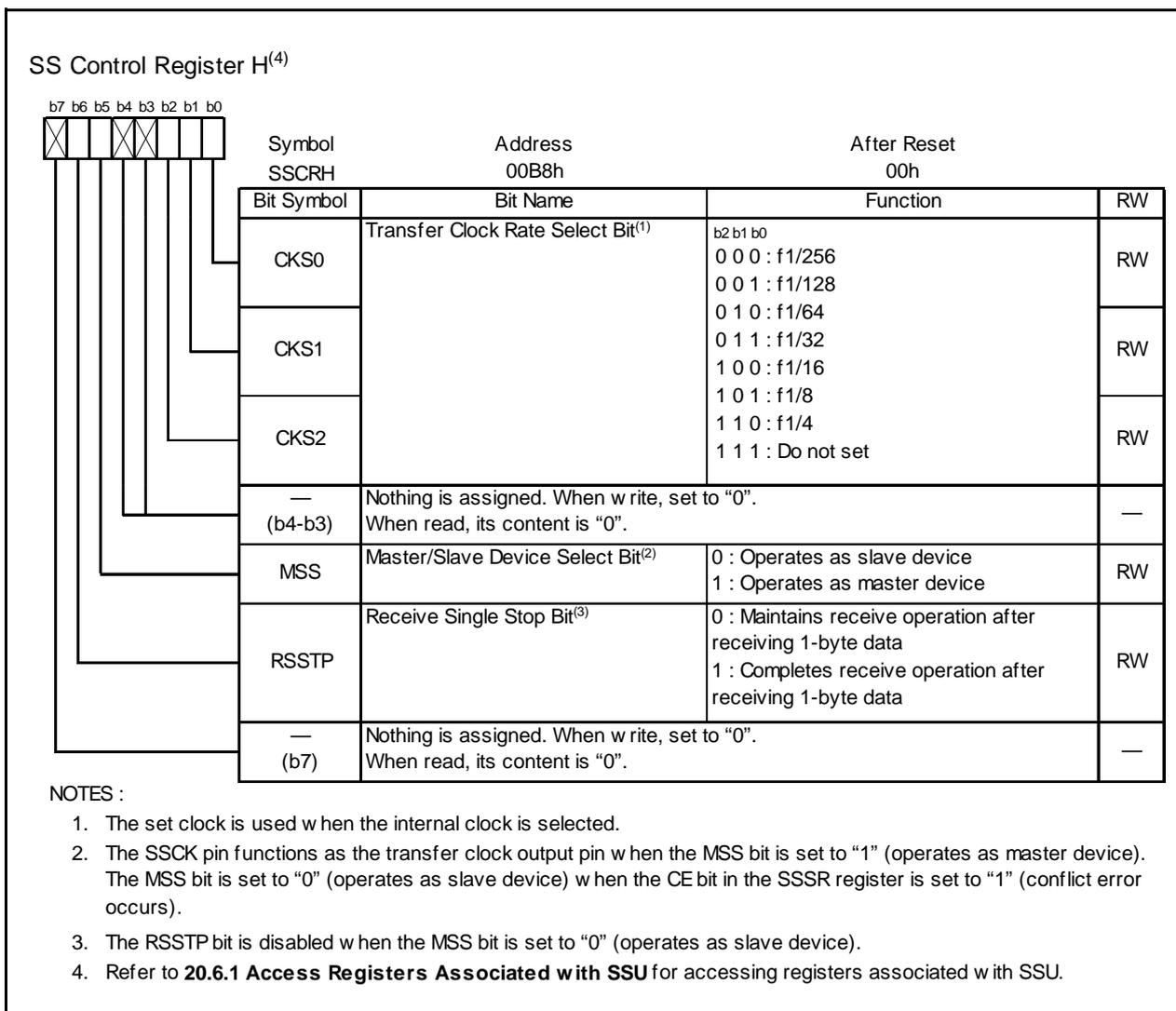


Figure 15.2 SSCRH Register

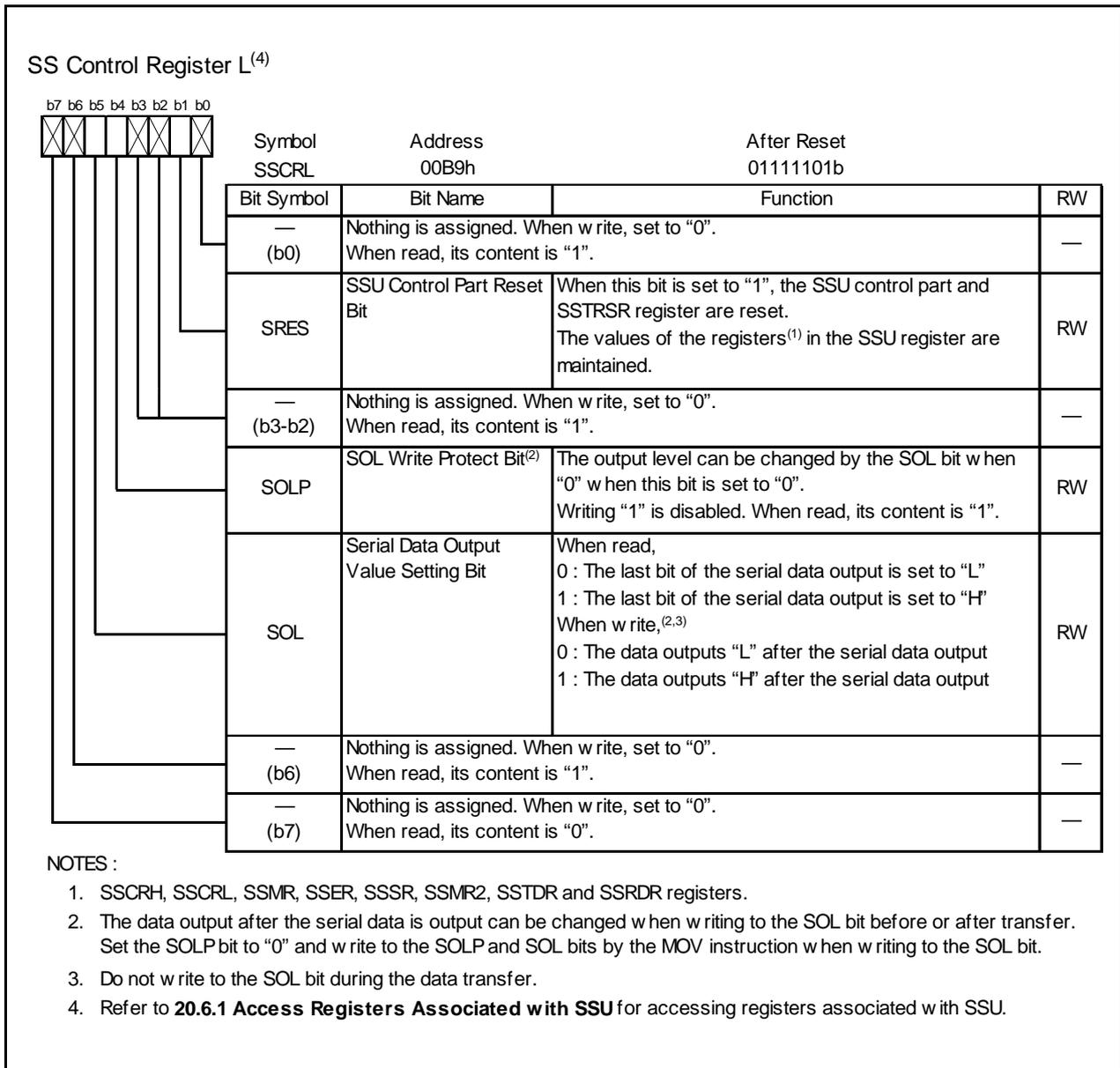


Figure 15.3 SSCRL Register

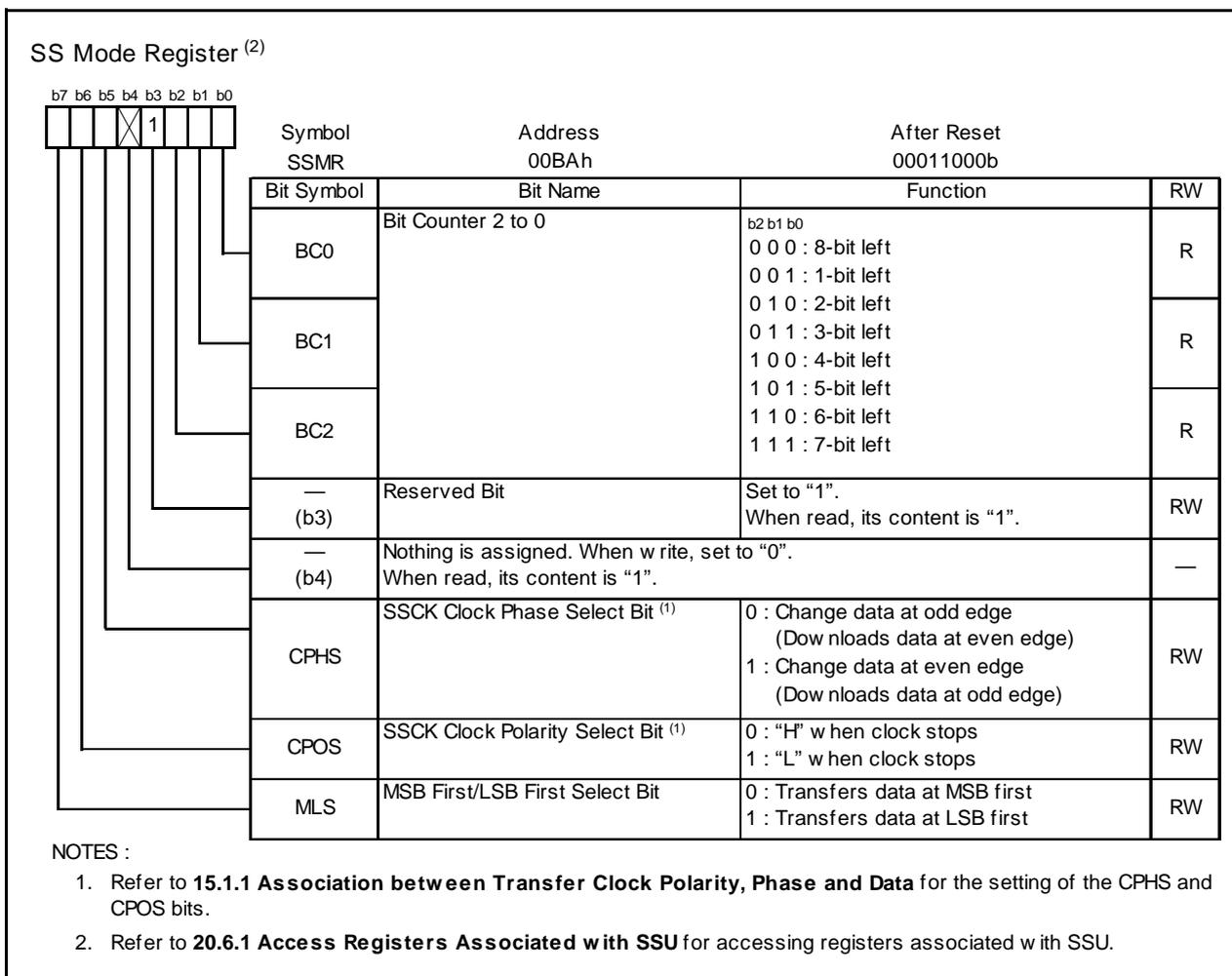


Figure 15.4 SSMR Register

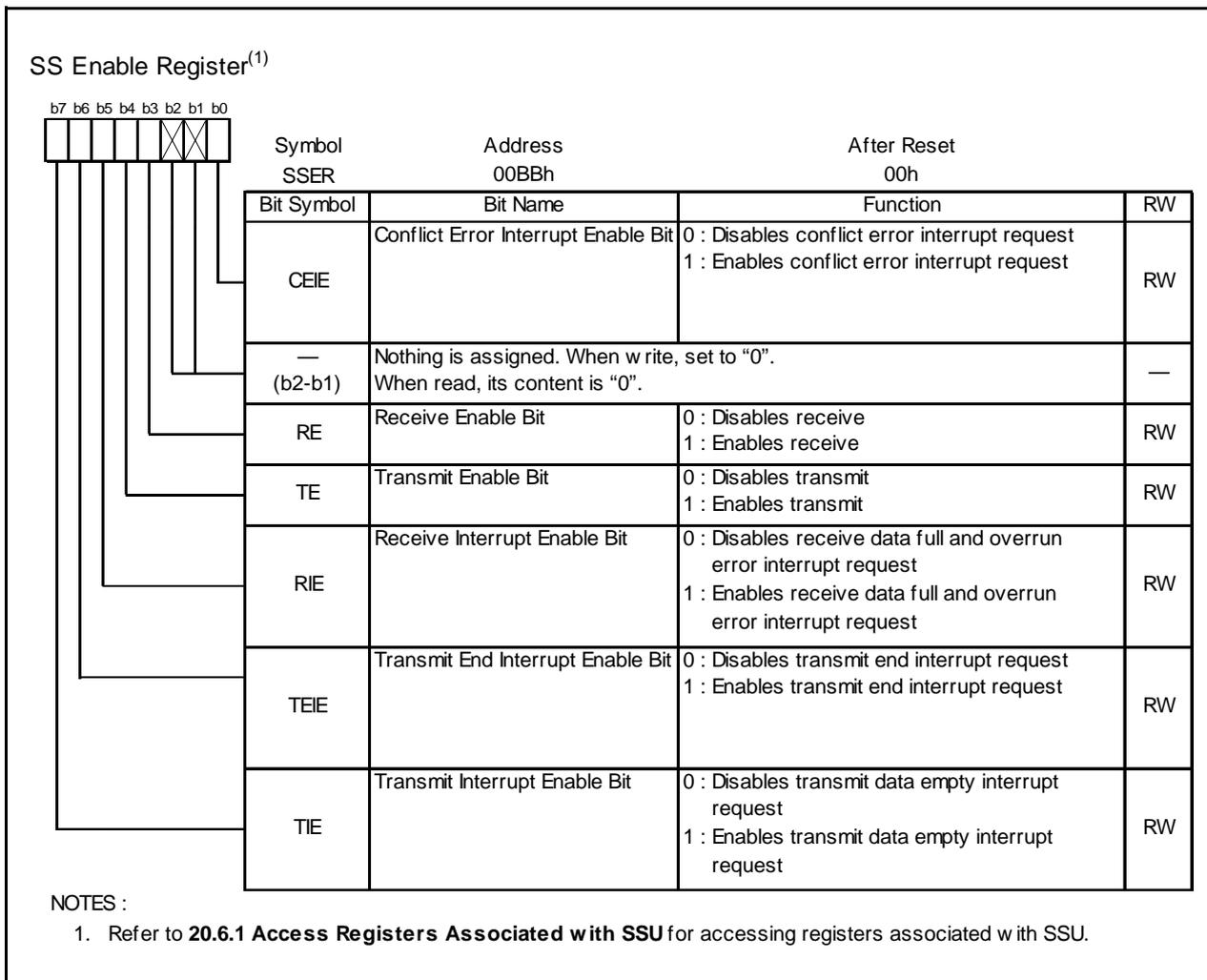


Figure 15.5 SSER Register

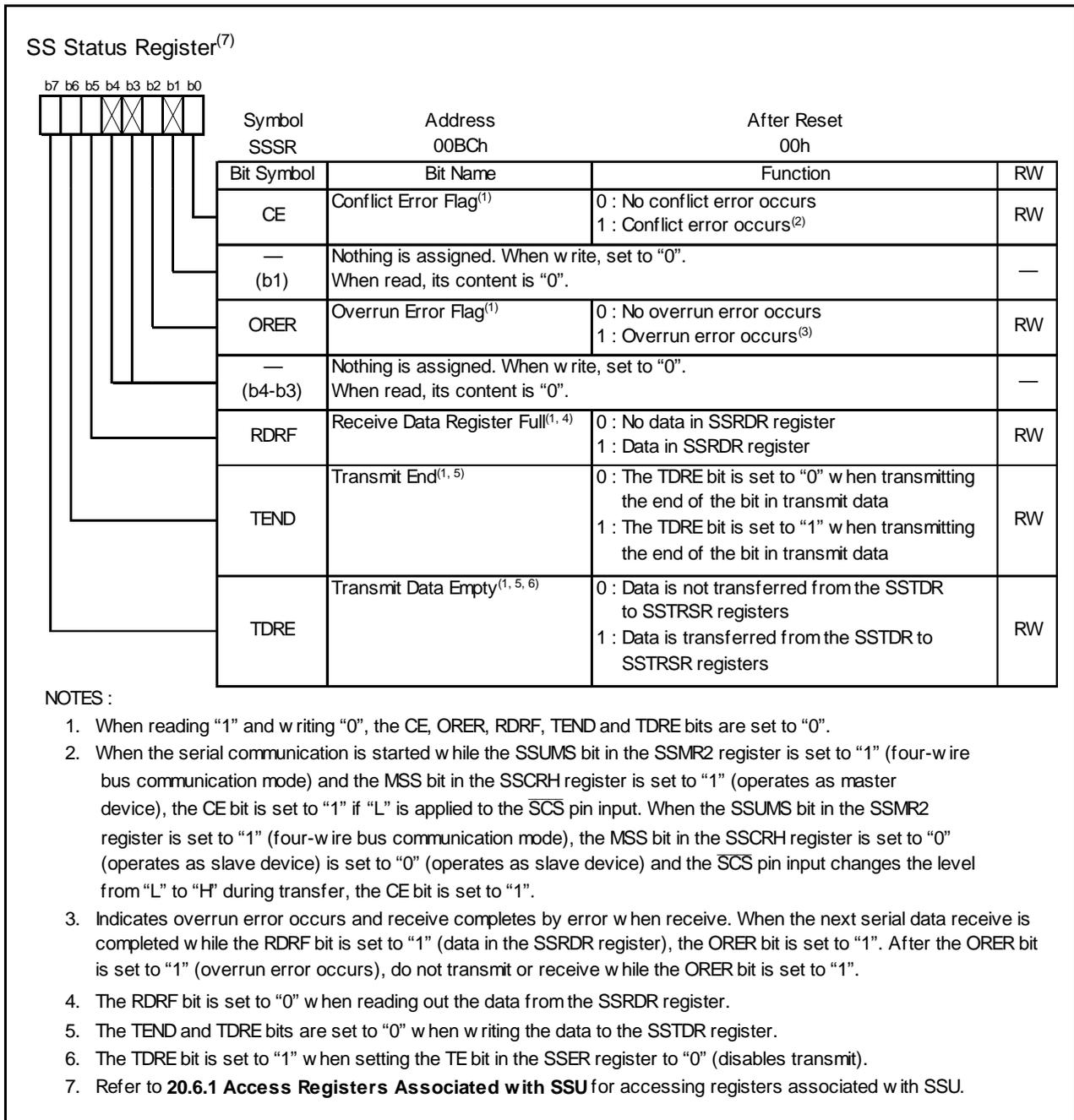


Figure 15.6 SSSR Register

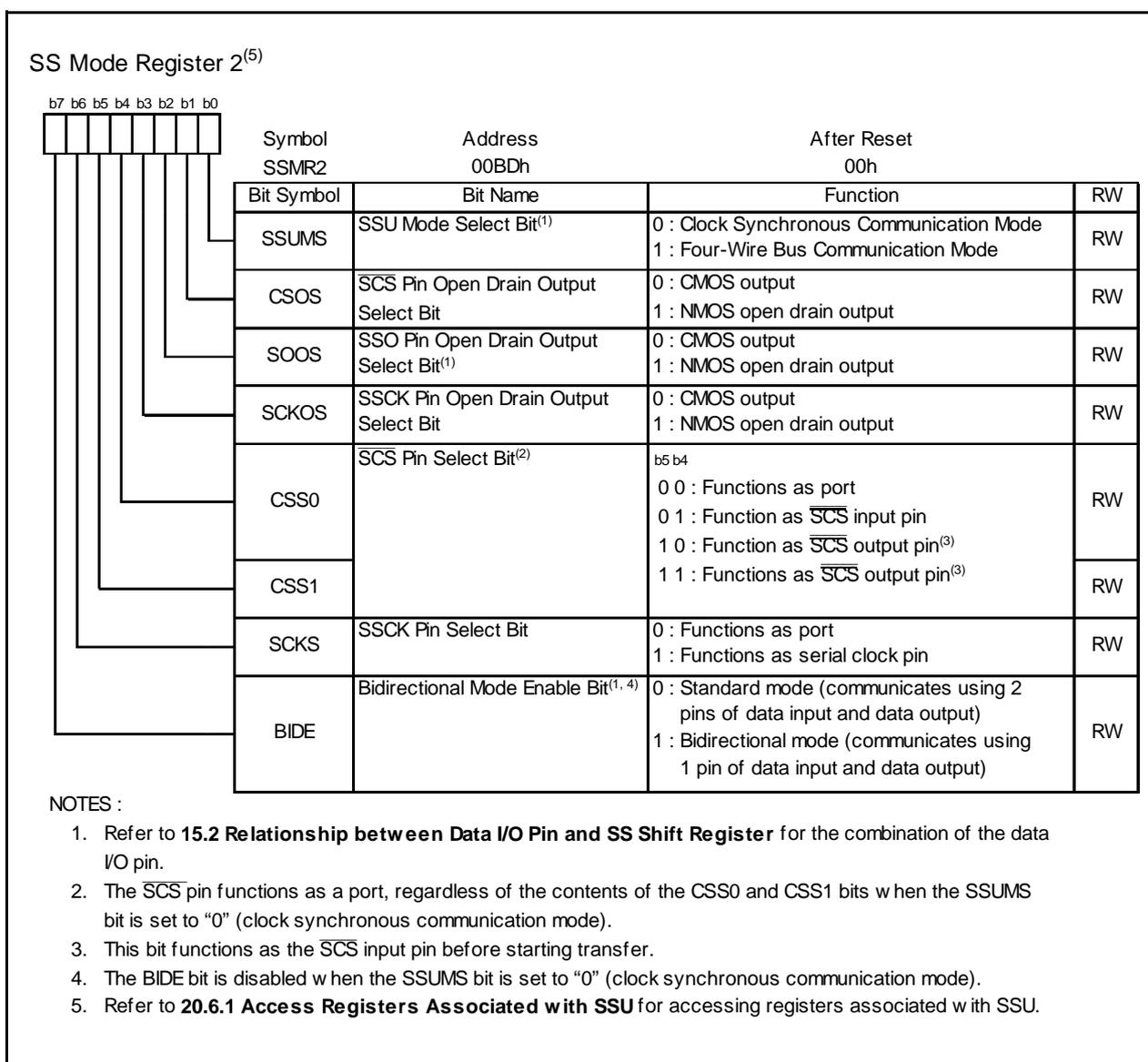


Figure 15.7 SSMR2 Register

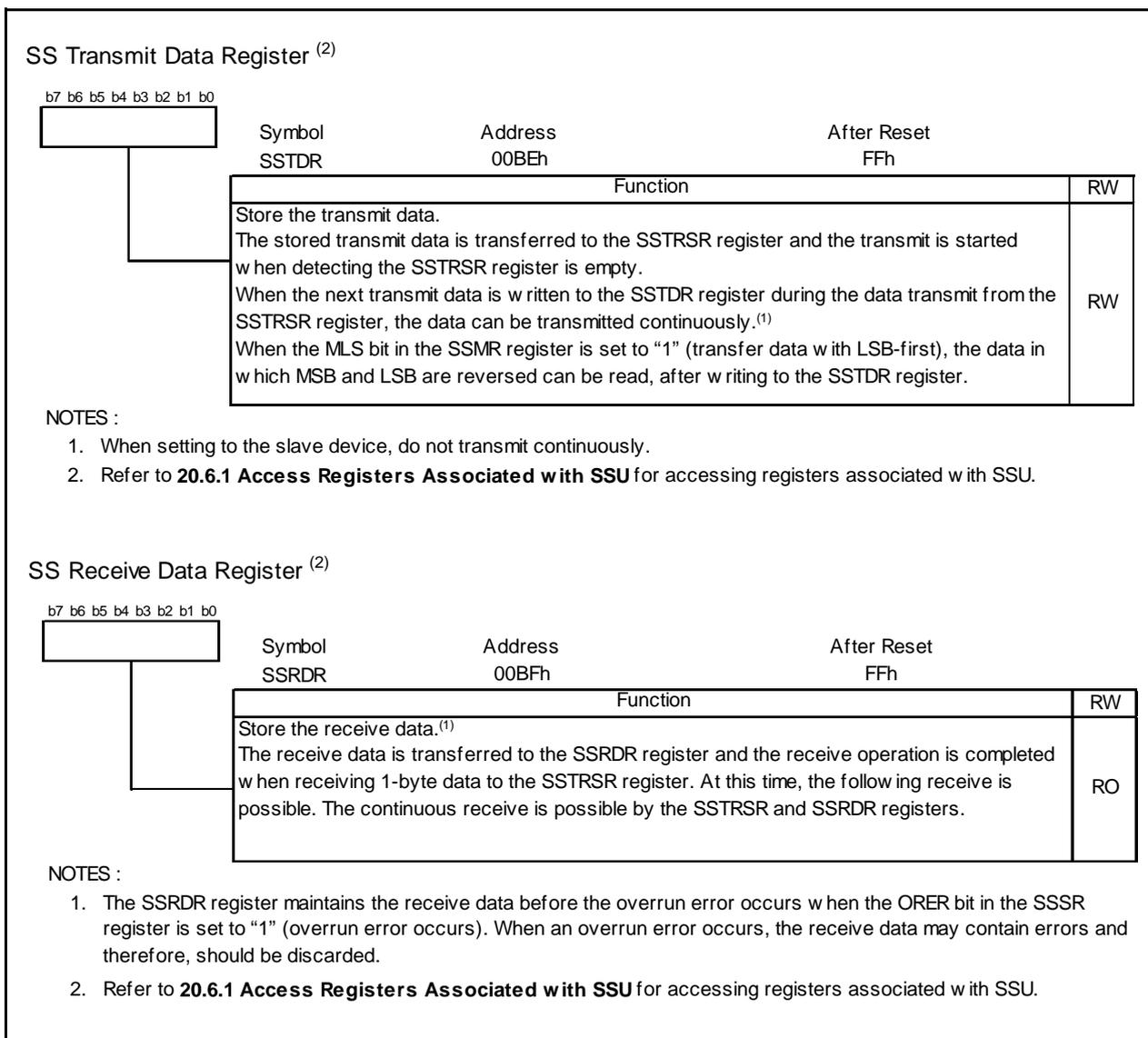


Figure 15.8 SSTDR and SSRDR Register

15.1 Transfer Clock

A transfer clock can be selected from 7 internal clocks ($\phi/256$, $\phi/128$, $\phi/64$, $\phi/32$, $\phi/16$, $\phi/8$ and $\phi/4$) and an external clock.

When using the SSU, 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 in the CKS0 to CKS2 bits 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.

15.1.1 Association between Transfer Clock Polarity, Phase and Data

Association between transfer clock polarity, phase and data changes according to a combination of the SSUMS bit in the SSMR2 register and the CPHS and CPOS bits in the SSMR register. Figure 15.9 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 to MSB. When the MLS bit is set to "0", transfer is started from the MSB to LSB.

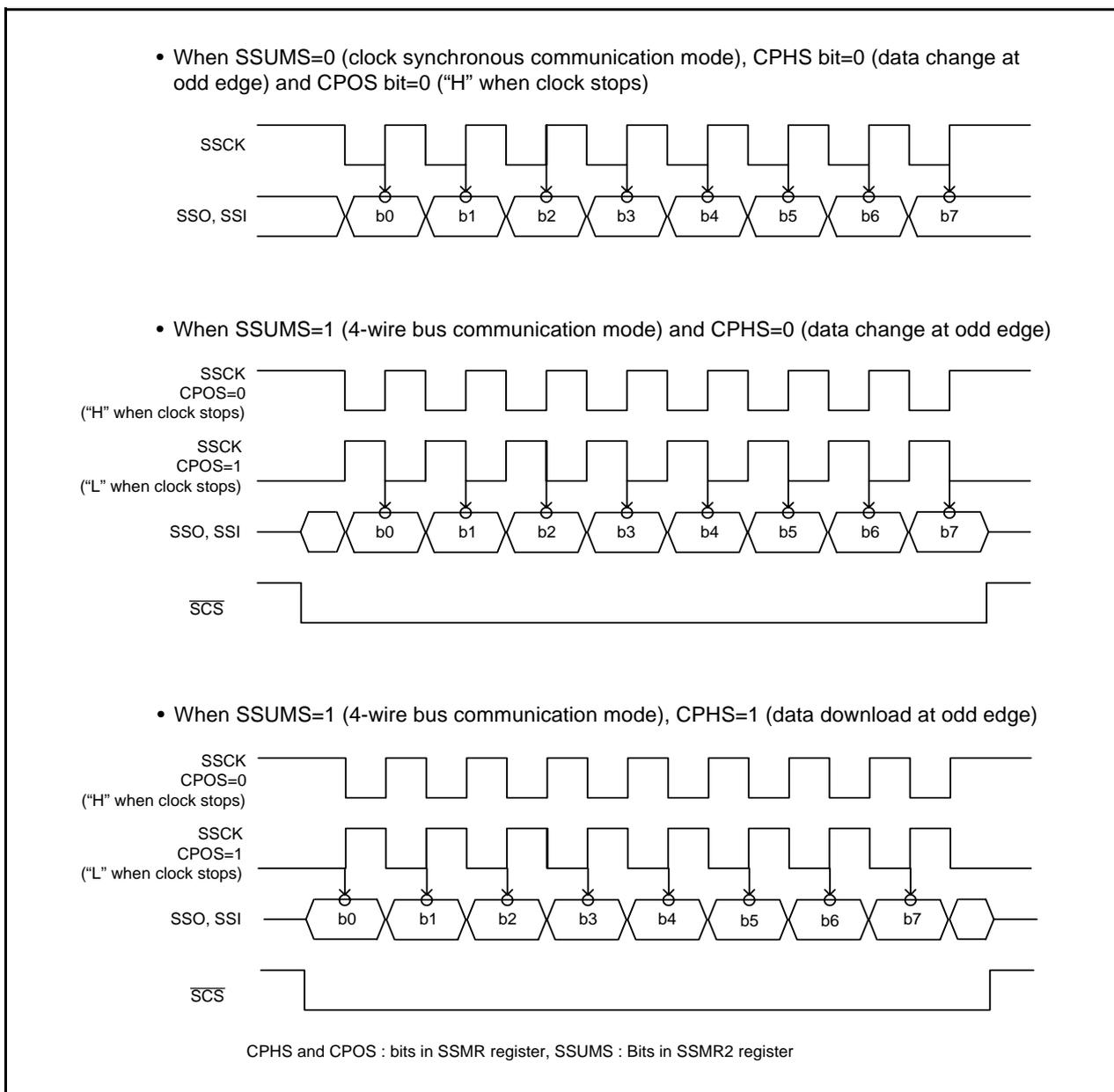


Figure 15.9 Association between Transfer Clock Polarity, Phase and Transfer Data

15.2 SS Shift Register (SSTRSR)

The SSTRSR register is the shift register to transmit and receive the serial data.

When the 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 the bit 0 in the SSTRSR register. When the MLS bit is set to "1" (LSB-first), the bit 7 in the SSTDR register is transferred to the bit 0 in the SSTRSR register.

15.2.1 Association between Data I/O Pin and SS Shift Register

Connecting association between the data I/O pin 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. Also, connecting association changes according to the BIDE bit in the SSMR2 register. Figure 15.10 shows a Connecting Association between Data I/O Pin and SSTRSR Register.

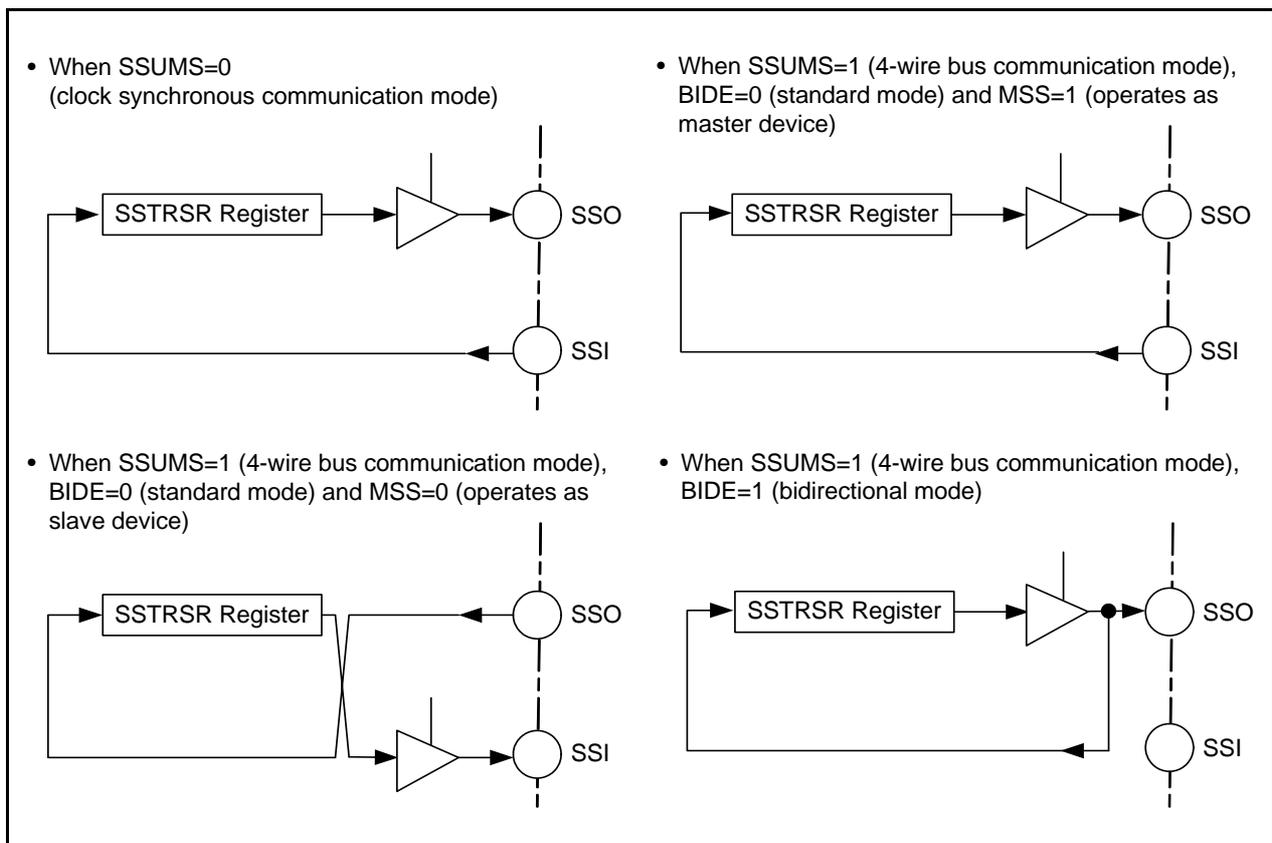


Figure 15.10 Connecting Association between Data I/O Pin and SSTRSR Register

15.3 Interrupt Requests

SSU has five interrupt requests : transmit data empty, transmit end, receive data full, overrun error and conflict error. Since these interrupt requests are assigned to the SSU interrupt vector table, determining interrupt sources by flags is required. Table 15.2 shows the SSU Interrupt Requests.

Table 15.2 SSU 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

Generation conditions of Table 15.2 are met, a SSU interrupt request is generated. Set the each interrupt source to "0" by a SSU interrupt routine.

However, the TDRE and TEND bits are automatically set to "0" by writing the transmit data to the SSTDR register and the RDRF bit is automatically set to "0" by reading the SSRDR register. When writing the transmit data to the SSTDR register, at the same time the TDRE bit is set to "1" (data is transmitted from the SSTDR to SSTRSR registers) again and when setting the TDRE bit to "0" (data is not transmitted from the SSTDR to SSTRSR registers), additional 1-byte data may be transmitted.

15.4 Communication Modes and Pin Functions

SSU switches functions of the I/O pin in each communication mode according to the setting of the MSS bit in the SSCRH register and the RE and TE bits in the SSER register. Table 15.3 shows the Association between Communication Modes and I/O Pins.

Table 15.3 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 programmable I/O port.
2. Do not set both the TE and RE bits 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

15.5 Clock Synchronous Communication Mode

15.5.1 Initialization in Clock Synchronous Communication Mode

Figure 15.11 shows an Initialization in Clock Synchronous Communication Mode. Set the TE bit in the SSER register to "0" (disables transmit) and the RE bit to "0" (disables receive) before data transmit / receive as an initialization.

When communication mode and format are changed, set the TE bit to "0" and the RE bit to "0" before changing.

Setting the RE bit to "0" does not change the contents of the RDRF and ORER flags, and the contents of the SSRDR register.

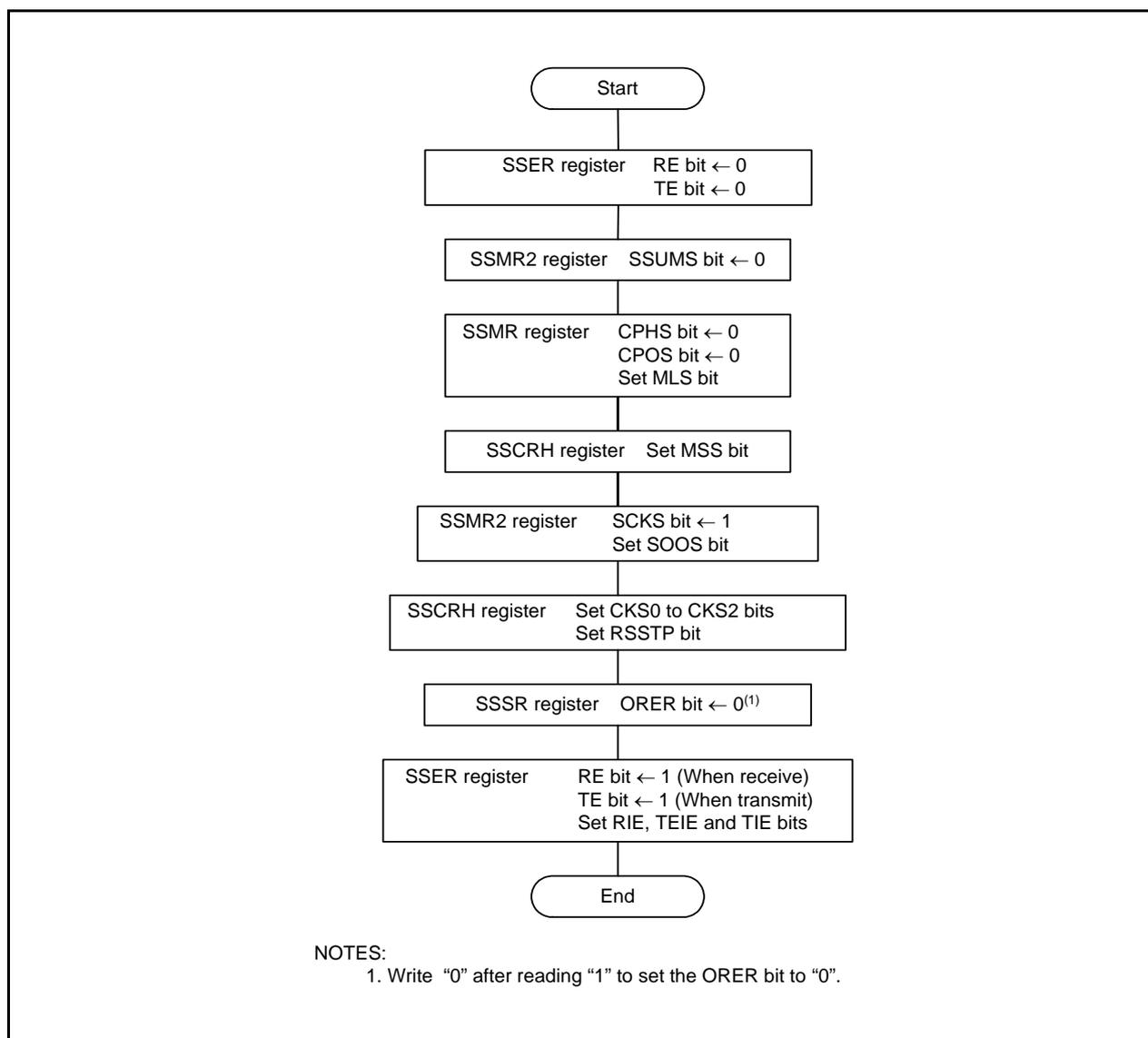


Figure 15.11 Initialization in Clock Synchronous Communication Mode

15.5.2 Data Transmit

Figure 15.12 shows an Example of SSU Operation for Data Transmit (Clock Synchronous Communication Mode). During the data transmit, the SSU operates as described below.

When the SSU is set as a master device, it outputs a synchronous clock and data.

When the SSU is set as a slave device, it outputs data synchronized with the input clock. When setting the TE bit to "1" (enables transmit) before writing the transmit data to the SSTDR register, the TDRE bit is automatically set to "0" (data is not transferred from the SSTDR to SSTRSR registers) and the data is transferred from the SSTDR to SSTRSR registers.

After the TDRE bit is set to "1" (data is transferred from the SSTDR to SSTRSR registers), a transmit is started. 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 the SSTDR to SSTRSR registers and a transmit 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" (enables transmit-end interrupt request). The SSCK pin is retained "H" after transmit-end.

Transmit can not be performed while the ORER bit in the SSRR register is set to "1" (overrun error occurs). Confirm that the ORER bit is set to "0" before transmit.

When setting the microcomputer to the slave device, ensure the TEND bit is set to "1" (data transmit ends) and write the following transmit data to the SSTDR register. When setting the microcomputer to the master device, continuous transmit is enabled.

Figure 15.13 shows a Sample Flowchart for Data Transmit (Clock Synchronous Communication Mode).

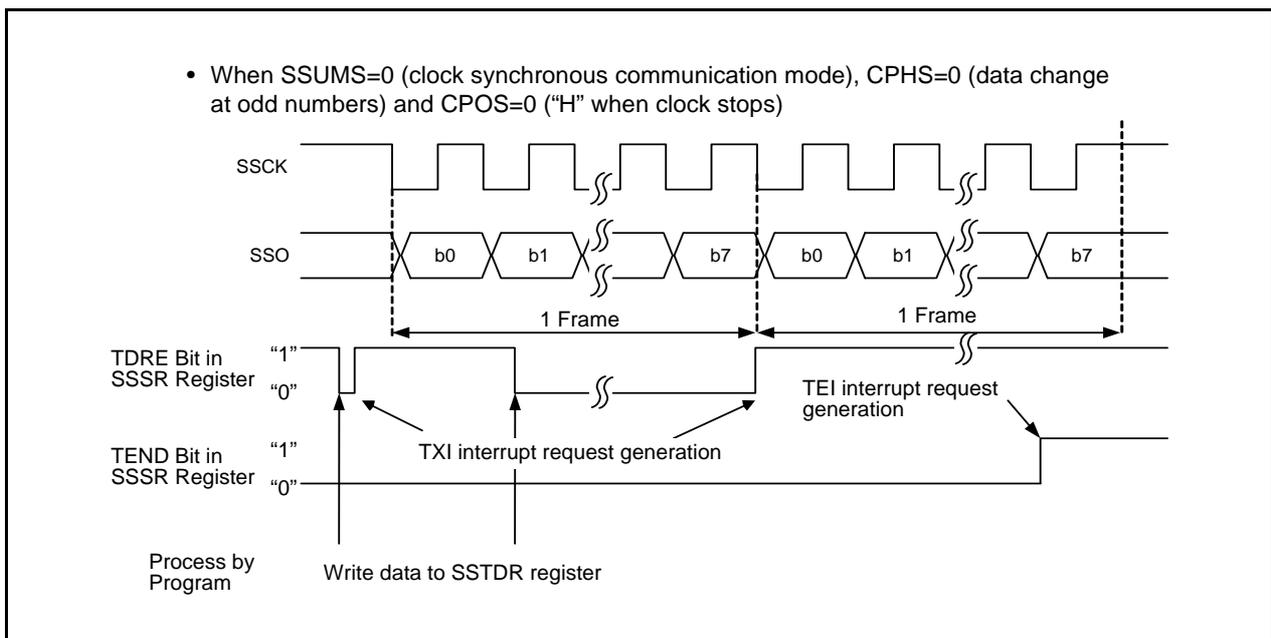


Figure 15.12 Example of SSU Operation for Data Transmit (Clock Synchronous Communication Mode)

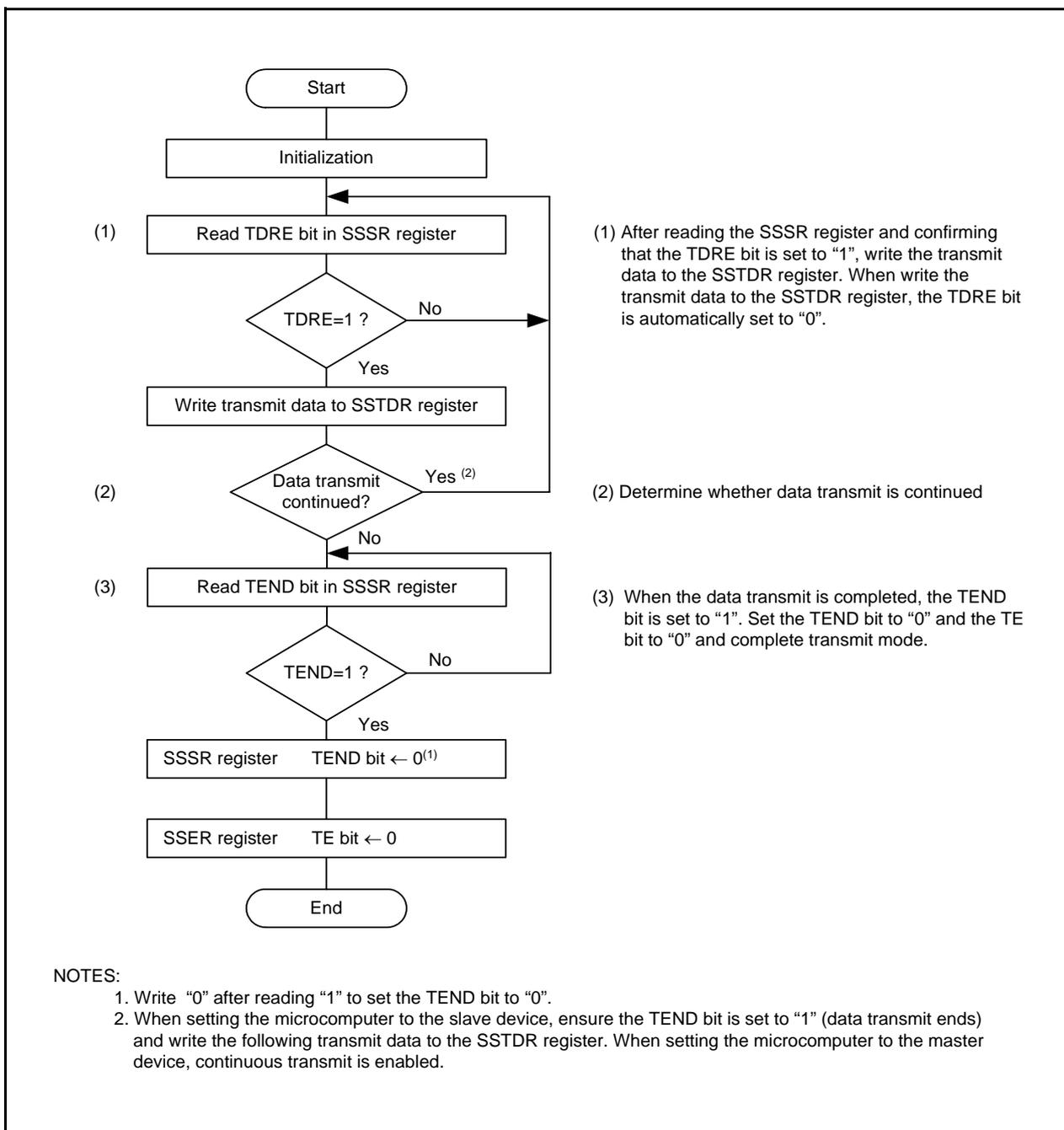


Figure 15.13 Sample Flowchart for Data Transmit (Clock Synchronous Communication Mode)

15.5.3 Data Receive

Figure 15.14 shows an Example of Operation for Data Receive (Clock Synchronous Communication Mode).

During the data receive, the SSU operates as described below. When the SSU is set as a master device, it outputs a synchronous clock and inputs data.

When the SSU is set as a slave device, it outputs data synchronized with the input clock. When the SSU is set as a master device, it outputs a receive clock and starts receiving by performing dummy read on the SSRDR register.

After the 8-bit data is 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" (enables RXI and OEI interrupt request), the RXI interrupt request is generated. If the SSSR 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). The SSU outputs a clock for receiving 8-bit data and stops. After that, set the RE bit in the SSER register to "0" (disables receive) and the RSSTP bit to "0" (receive operation is continued after receiving the 1-byte data) and read the receive data. If the SSRDR register is read while the RE bit is set to "1" (enables receive), 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" (overflow error occurs : OEI) and the operation is stopped. When the ORER bit is set to "1", receive can not be performed. Confirm that the ORER bit is set to "0" before restarting receive.

Figure 15.15 shows a Sample Flowchart for Data Receive (MSS=1) (Clock Synchronous Communication Mode).

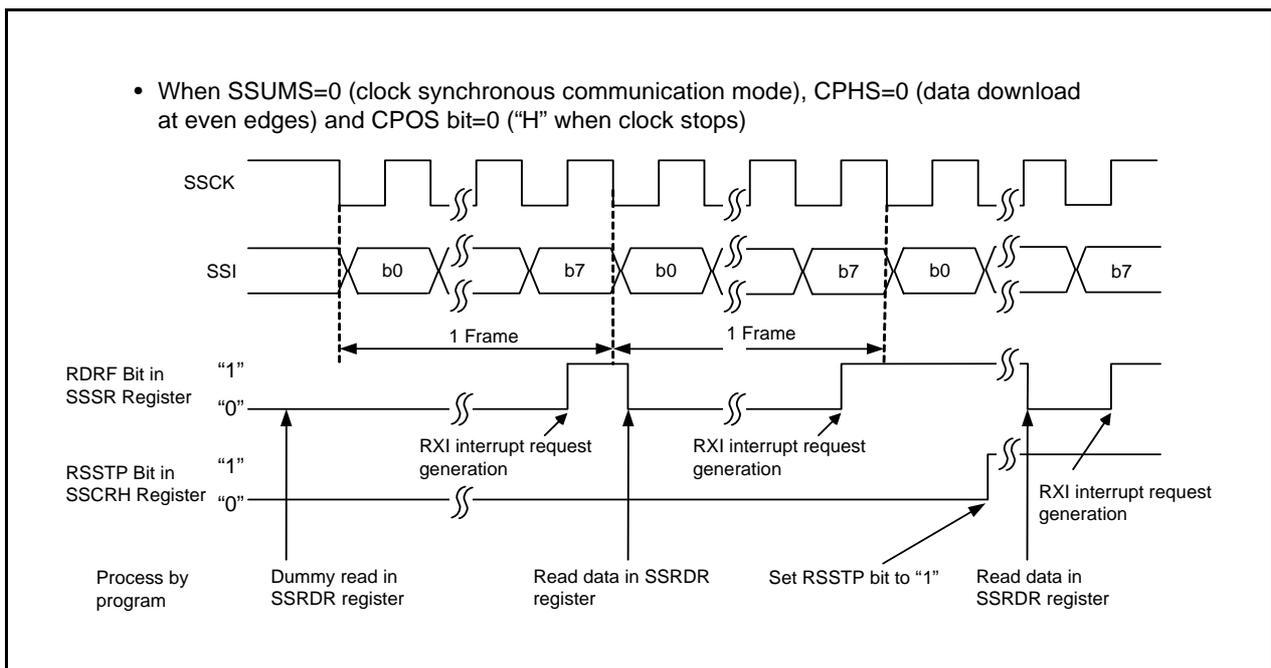


Figure 15.14 Example of Operation for Data Receive (Clock Synchronous Communication Mode)

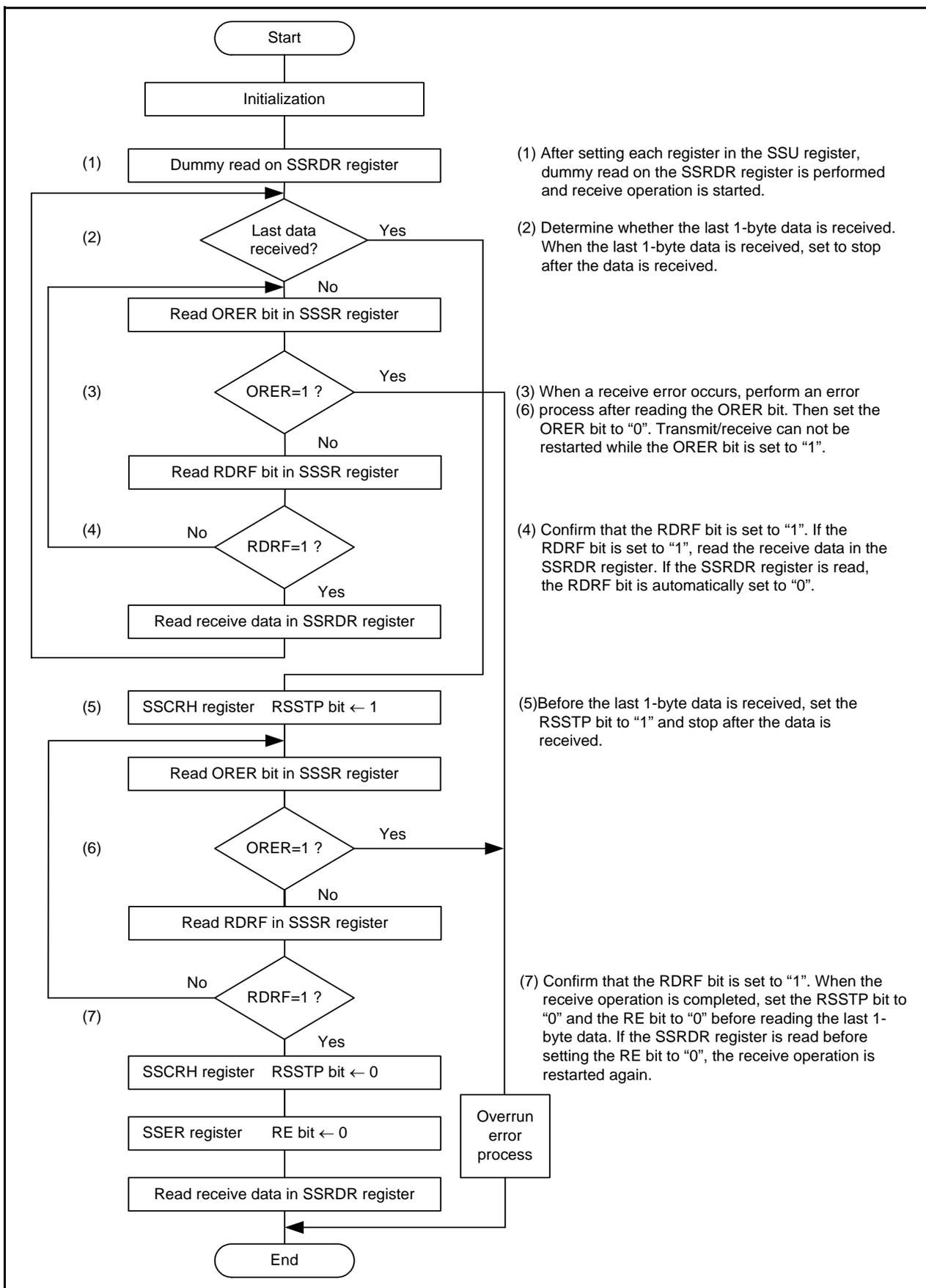


Figure 15.15 Sample Flowchart for Data Receive (MSS=1) (Clock Synchronous Communication Mode)

15.5.4 Data Transmit/Receive

Data transmit/receive is a combined operation of data transmit and receive which are described before. Transmit/receive is started by writing data in the SSTDR register.

When the 8th clock rises or the ORER bit is set to "1" (overrun error occurs) while the TDRE bit is set to "1" (data is transferred from the SSTDR to SSTRSR registers), 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 RERF bit is set to "0" (no data in the SSRDR register) and the ORER bit is set to "0" (no overrun error), set the TE and RE bits to "1".

When setting the microcomputer to the slave device, ensure the TEND bit is set to "1" (data transmit ends) and write the following transmit data to the SSTDR register. When setting the microcomputer to the master device, continuous transmit is enabled.

Figure 15.16 shows a Sample Flowchart for Data Transmit/Receive (Clock Synchronous Communication Mode).

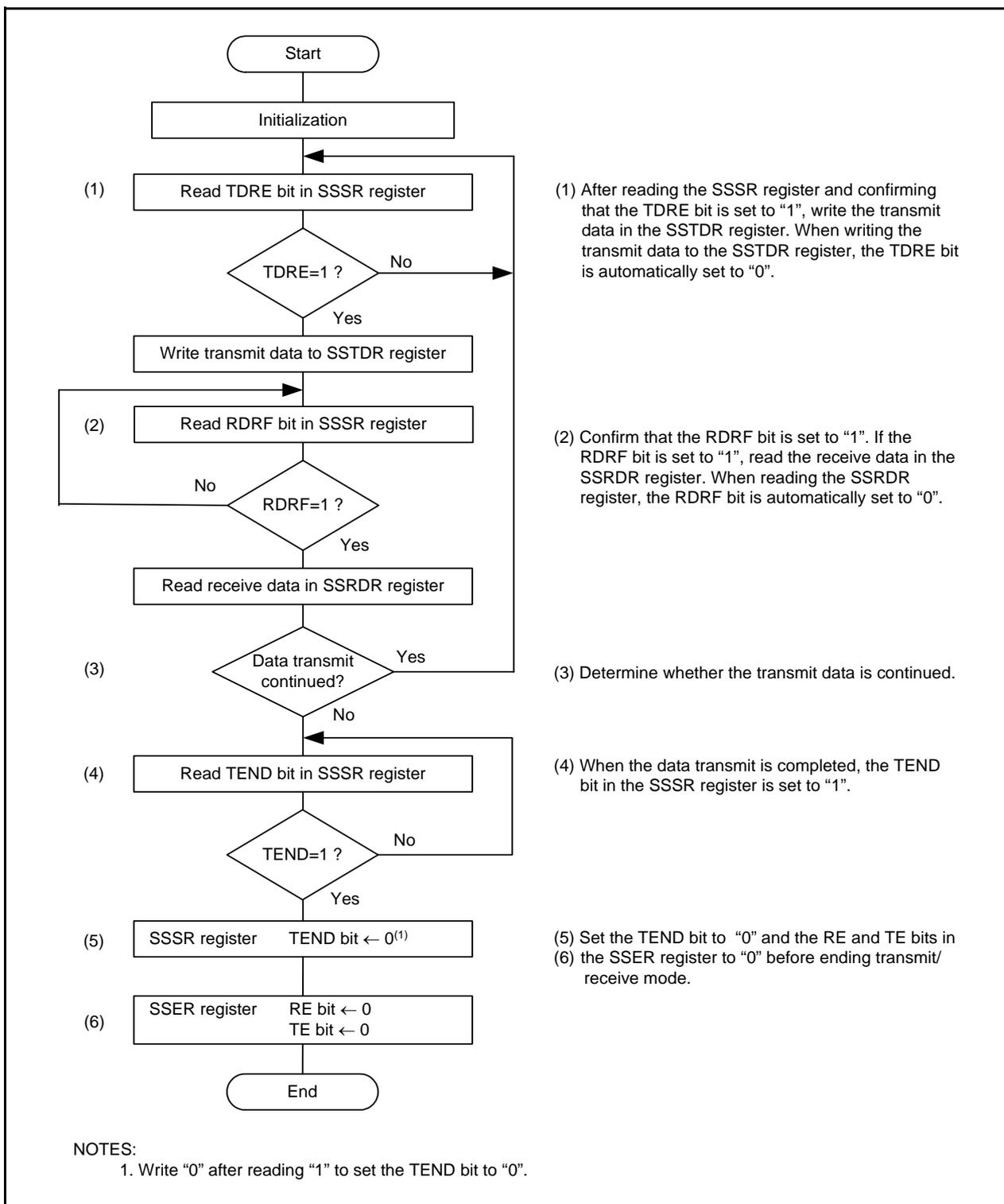


Figure 15.16 Sample Flowchart for Data Transmit/Receive (Clock Synchronous Communication Mode)

15.6 Operation in 4-Wire Bus Communication Mode

4-wire bus communication mode is a mode which communicates with the 4-wire bus; a clock line, data input line, data output line and chip select line. 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 are changed according to the setting of the MSS bit in the SSCRH register and the BIDE bit in the SSMR2 register. For details, refer to **15.2.1 Association between Data I/O Pin and SS Shift Register**. In this mode, association between the clock polarity, phase and data can be set by the CPOS and CPHS bits in the SSMR register. For details, refer to **15.1.1 Association between Transfer Clock Polarity, Phase and Data**.

When the SSU is set as a master device, the chip select line controls output. When the SSU is set as a slave device, the chip select line controls input. When the SSU is set as master device, the chip select line controls output of the $\overline{\text{SCS}}$ pin or controls output of a general port by setting the CSS1 bit in the SSMR2 register. When the SSU is set as a slave device, the chip select line set the $\overline{\text{SCS}}$ pin as an input pin by setting the CSS1 and CSS0 bits 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 using the MSB-first.

15.6.1 Initialization in 4-Wire Bus Communication Mode

Figure 15.17 shows an Initialization in 4-Wire Bus Communication Mode. Before the data transit/receive, set the TE bit in the SSER register to "0" (disables transmit) and the RE bit in the SSER register to "0" (disables receive) and initialize the SSU.

When communication mode and format are changed, set the TE bit to "0" and the RE bit to "0" before changing.

Setting the RE bit to "0" does not change the contents of the RDRF and ORER flags, and the contents of the SSRDR register.

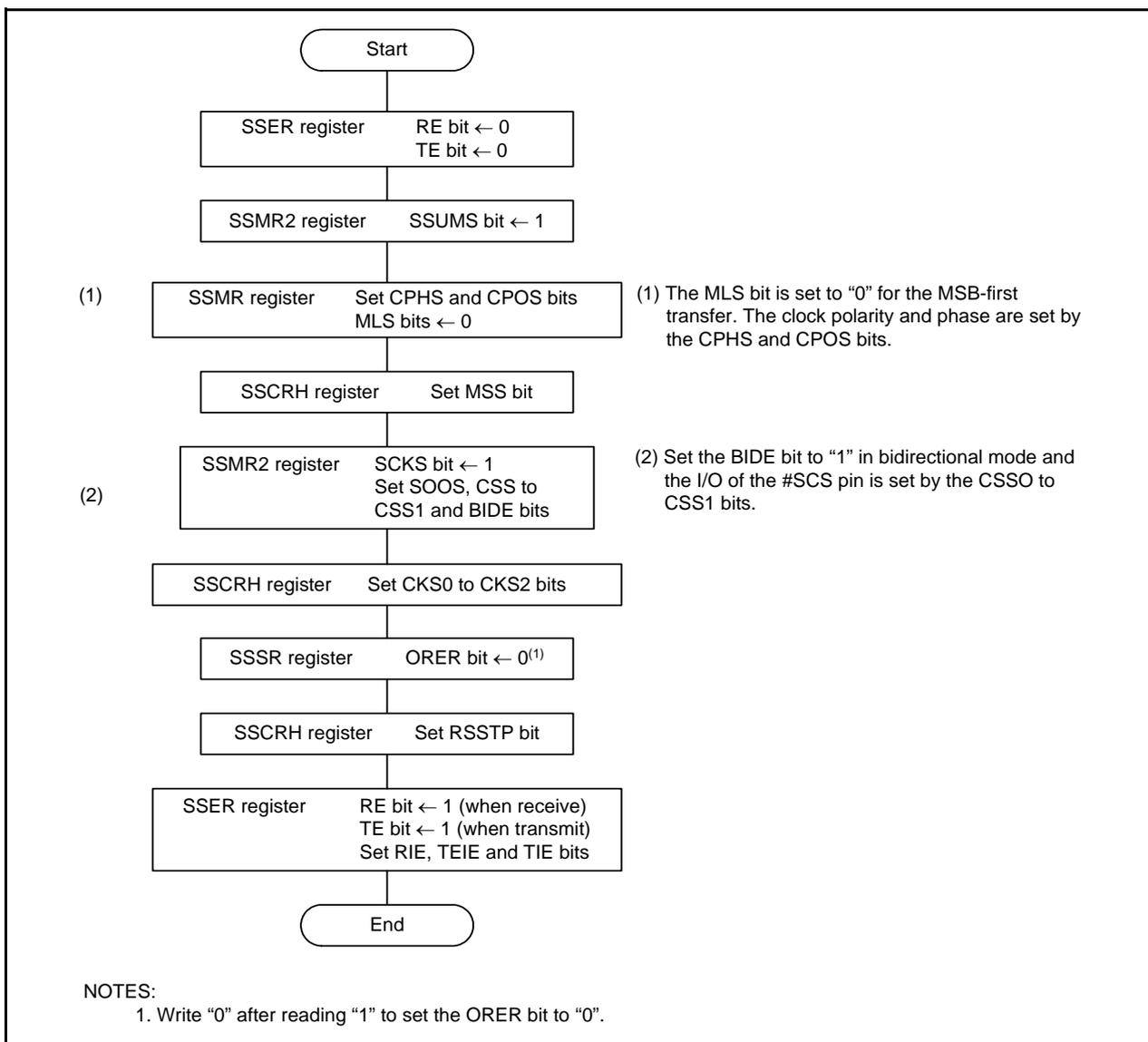


Figure 15.17 Initialization in 4-Wire Bus Communication Mode

15.6.2 Data Transmit

Figure 15.18 shows an Example of Operation in Data Transmit (4-Wire Bus Communication Mode). During the data transmit, the SSU operates as described below.

When the SSU is set as a master device, it outputs a synchronous clock and data. When the \overline{UUSA} is set as a slave device, it outputs data in synchronized with the input clock while "L" applies to the \overline{SCS} pin.

When writing the transmit data to the SSTDR register after setting the TE bit to "1" (enables transmit), the TDRE bit is automatically set to "0" (data is not transferred from the SSTDR to SSTRSR registers) and the data is transferred from the SSTDR to SSTRSR registers. After the TDRE bit is set to "1" (data is transferred from the SSTDR to SSTRSR registers), a transmit is started. When the TIE bit in the SSER register is set to "1", the TXI interrupt request is generated.

When the 1-frame data is transferred while the TDRE bit is set to "0", the data is transferred from the SSTDR to SSTRSR registers and the next frame transmit is started. If the 8th bit is transmitted while the TDRE is set to "1", the 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" (enables transmit-end interrupt request), the TEI interrupt request is generated. The SSCK pin is retained "H" after transmit-end and the \overline{SCS} pin is held "H". When the \overline{SCS} pin is transmitted 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.

Transmit can not be performed while the ORER bit in the SSSR register is set to "1" (overflow error occurs). Confirm that the ORER bit is set to "0" before transmit.

When setting the microcomputer to the slave device, ensure the TEND bit is set to "1" (data transmit ends) and write the following transmit data to the SSTDR register. When setting the microcomputer to the master device, continuous transmit is enabled.

The difference from the clock synchronous communication mode is that 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.

A sample flowchart is the same as the clock synchronous communication mode (Refer to **Figure 15.13 Sample Flowchart for Data Transmit (Clock Synchronous Communication Mode)**).

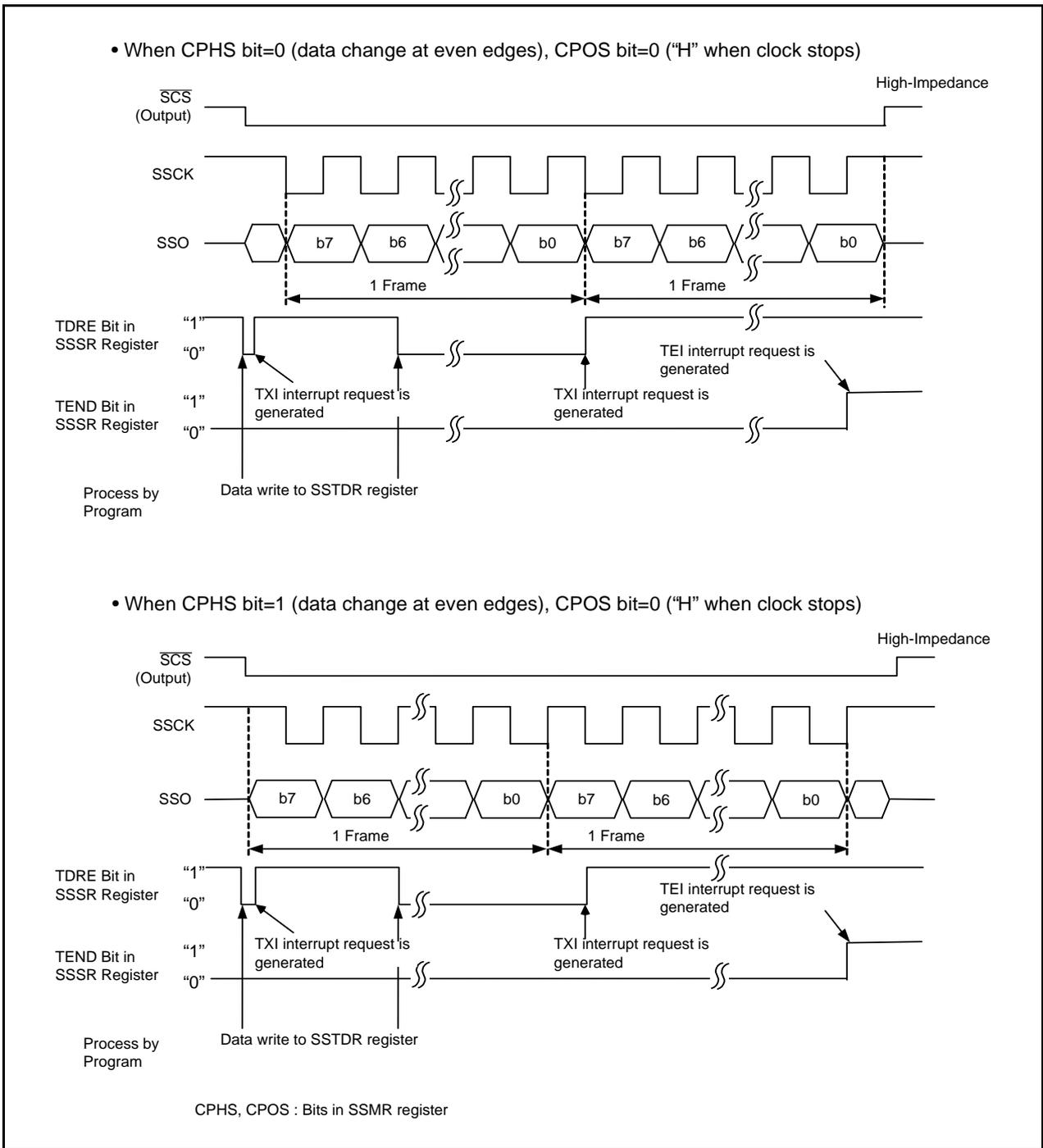


Figure 15.18 Example of Operation in Data Transmit (4-Wire Bus Communication Mode)

15.6.3 Data Receive

Figure 15.19 shows an example of the SSU operation for the data receive. During the data receive, the SSU operates as described below.

When the SSU is set as a master device, it outputs a synchronous clock and inputs data. When the SSU is set as a slave device, it outputs data synchronized with the input clock while the \overline{SCS} pin is held "L" input. When the SSU is set as a master device, it outputs a receive clock and starts receiving by performing dummy read on the SSRDR register.

After the 8-bit data is 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" (enables RXI and OEI interrupt request), 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 data, the receive operation is completed). The SSU outputs a clock for receiving 8-bit data and stops. After that, set the RE bit in the SSER register to "0" (disables receive) and the RSSTP bit to "0" (receive operation is continued after receiving 1-byte data) and read the receive data. If the SSRDR register is read while the RE bit is set to "1" (enables receive), 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" (overflow error occurs : OEI) and the operation is stopped. When the ORER bit is set to "1", receive can not be performed. Confirm that the ORER bit is set to "0" before restarting receive.

When the RDRF and ORER bits are set to "1", it varies depending on setting the CPHS bit in the SSMR register. Figure 15.19 shows when the RDRF and ORER bits are set to "1".

When the CPHS bit is set to "1" (data download at the odd edges), the RDRF and ORER bits are set to "1" at one point of a frame.

A sample flowchart is the same as the clock synchronous communication mode (Refer to **Figure 15.15 Sample Flowchart for Data Receive (MSS=1) (Clock Synchronous Communication Mode)**).

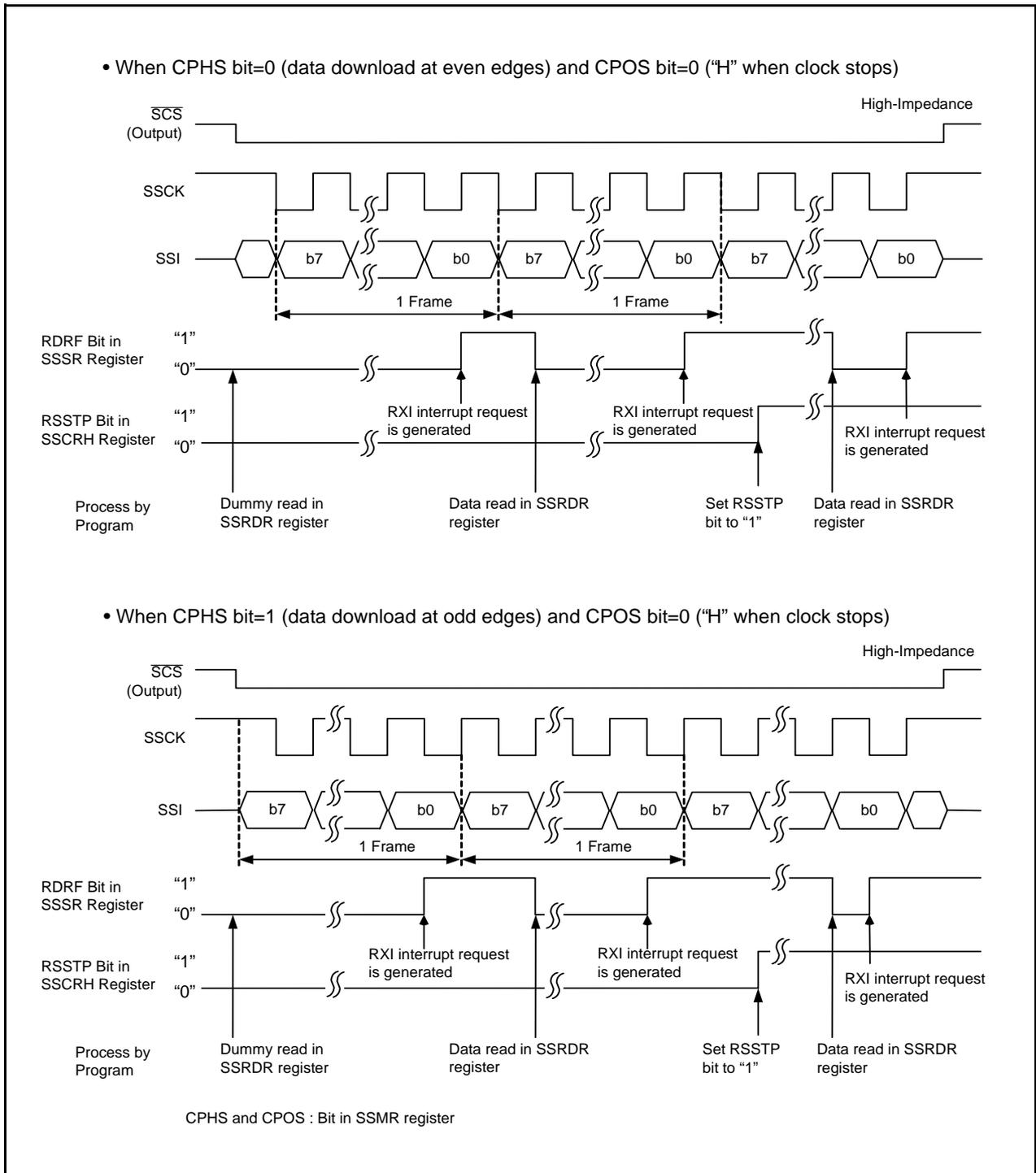


Figure 15.19 Example of Operation in Data Receive (4-Wire Bus Communication Mode)

15.6.4 $\overline{\text{SCS}}$ 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 a master device) and check the arbitration of the $\overline{\text{SCS}}$ pin before starting serial transfer. If the SSU detects that the synchronized internal $\overline{\text{SCS}}$ signal is held "L" in this period, the CE bit in the SSSR register to "1" (a conflict error occurs) and the MSS bit is automatically set to "0" (operates as a slave device).

Figure 15.20 shows an Arbitration Check Timing.

A future transmit operation is not performed while the CE bit is set to "1". Set the CE bit to "0" (a conflict error does not occur) before a transmit is started.

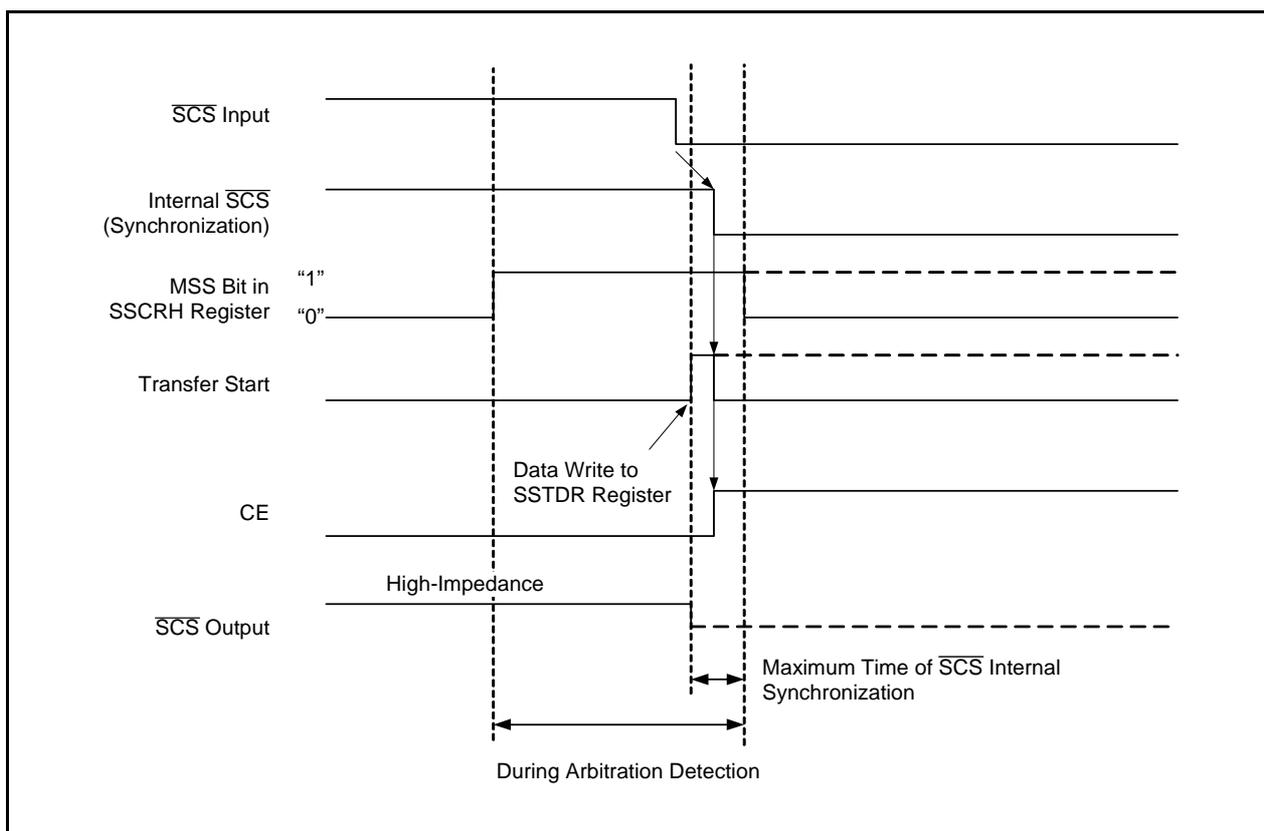


Figure 15.20 Arbitration Check Timing

16. 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 the pins with P1_0 to P1_3. Therefore, when using these pins, ensure 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, helping to reduce the power consumption of the chip.

The result of A/D conversion is stored in the AD register.

Table 16.1 lists the Performance of A/D converter. Figure 16.1 shows the Block Diagram of A/D Converter. Figures 16.2 and 16.3 show the A/D converter-related registers.

Table 16.1 Performance of A/D converter

| Item | Performance |
|--|---|
| A/D Conversion Method | Successive approximation (with capacitive coupling amplifier) |
| Analog Input Voltage ⁽¹⁾ | 0V to Vref |
| Operating Clock ϕ_{AD} ⁽²⁾ | $4.2V \leq AVCC \leq 5.5V$ f1, f2, f4 $2.7V \leq AVCC < 4.2V$ f2, f4 |
| Resolution | 8 bit or 10 bit is selectable |
| Absolute Accuracy | $AVCC = Vref = 5V$ <ul style="list-style-type: none"> • 8-bit resolution ± 2 LSB • 10-bit resolution ± 3 LSB $AVCC = Vref = 3.3V$ <ul style="list-style-type: none"> • 8-bit resolution ± 2 LSB • 10-bit resolution ± 5 LSB |
| Operating Mode | One-shot and repeat modes ⁽³⁾ |
| Analog Input Pin | 4 pins (AN8 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 Z 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. Analog input voltage does not depend on use of sample and hold function.
2. The frequency of ϕ_{AD} must be 10 MHz or below.
Without sample and hold function, the ϕ_{AD} frequency should be 250 kHz or above.
With the sample and hold function, the ϕ_{AD} frequency should be 1 MHz or above.
3. In repeat mode, only 8-bit mode can be used.

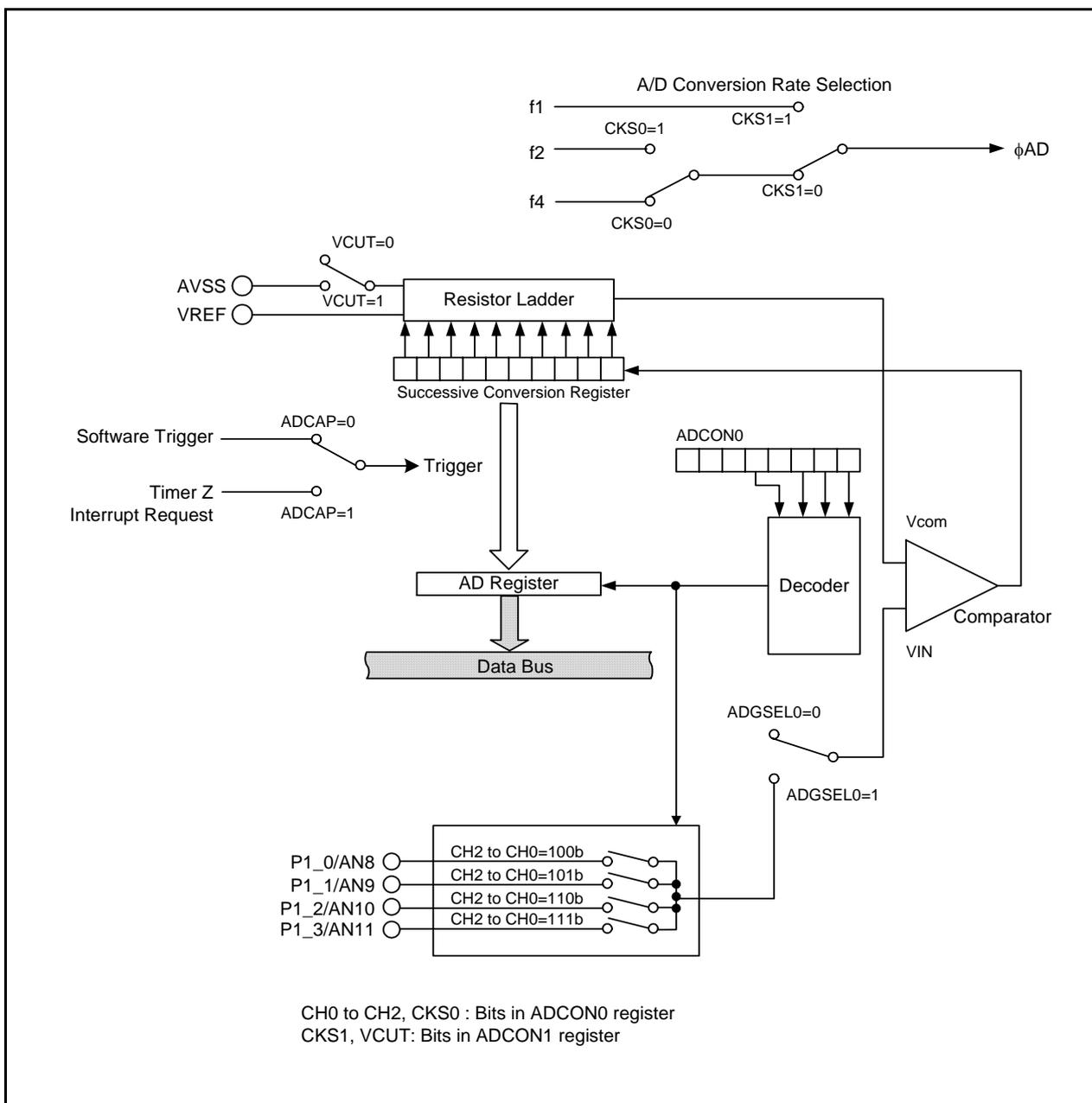


Figure 16.1 Block Diagram of A/D Converter

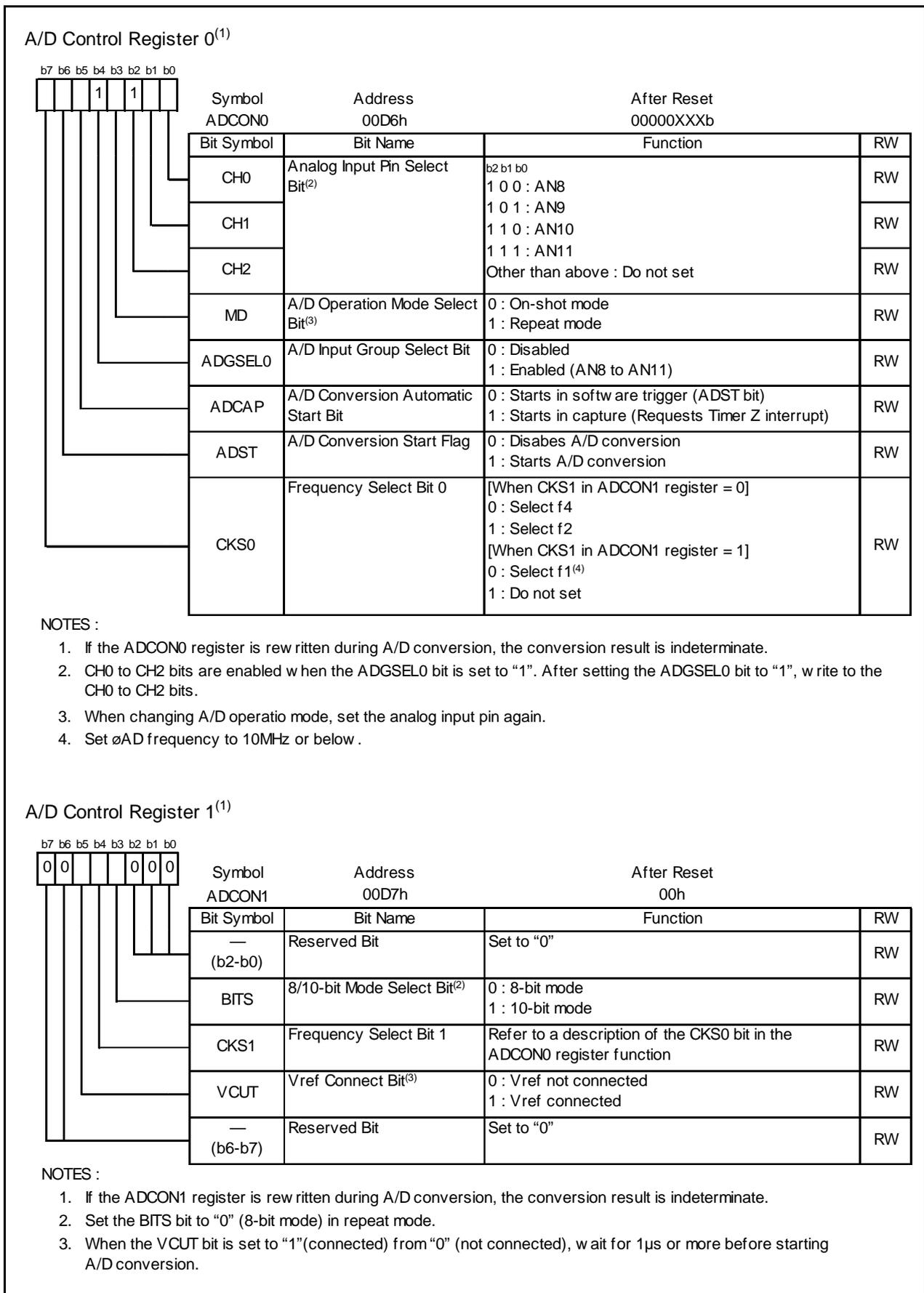


Figure 16.2 ADCON0 and ADCON1 Registers

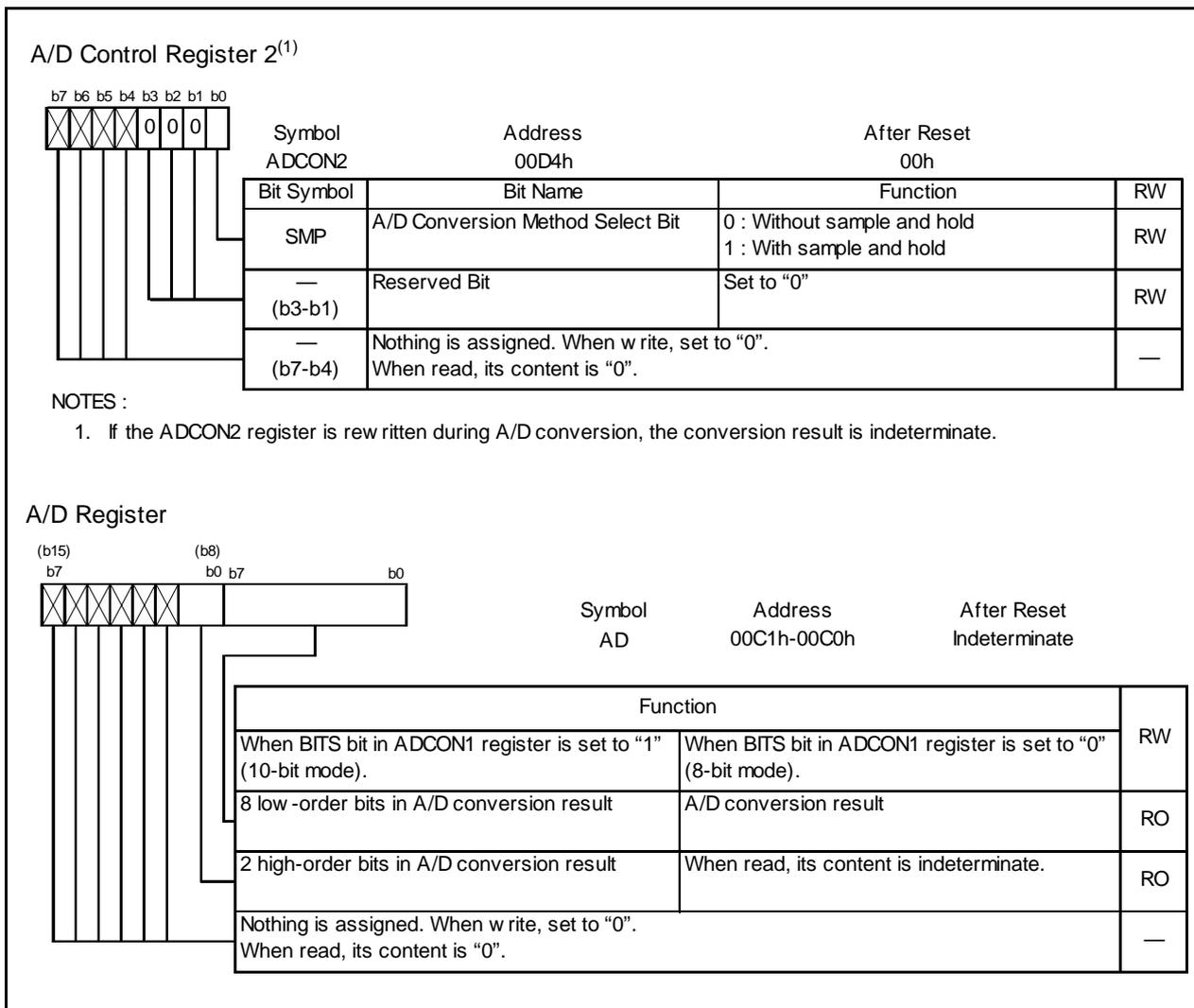


Figure 16.3 ADCON2 and AD Registers

16.1 One-Shot Mode

In one-shot mode, the input voltage on one selected pin is A/D converted once. Table 16.2 lists the Specification of One-Shot Mode. Figure 16.4 shows the ADCON0 and ADCON1 Registers in One-Shot Mode.

Table 16.2 Specification of One-Shot Mode

| Item | Specification |
|-------------------------------------|---|
| Function | The input voltage on one selected pin by the CH2 to CH0 bits 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" (capture), Timer Z interrupt request is generated while the ADST bit is set to "1" |
| Stop Condition | <ul style="list-style-type: none"> • A/D conversion completes (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 AN8 to AN11 |
| Reading of A/D Conversion Result | Read AD register |

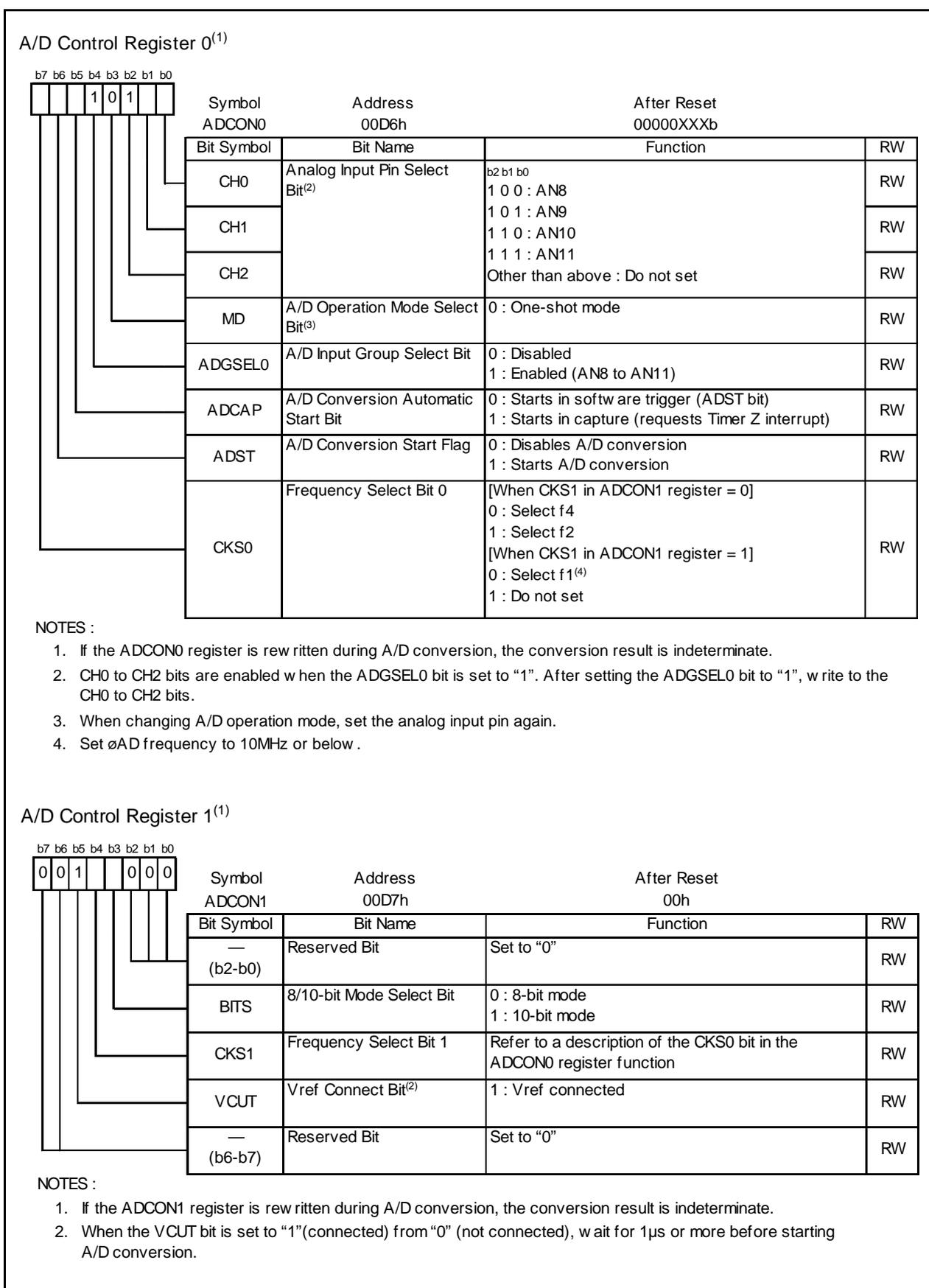


Figure 16.4 ADCON0 and ADCON1 Registers in One-Shot Mode

16.2 Repeat Mode

In repeat mode, the input voltage on one selected pin is A/D converted repeatedly. Table 16.3 lists the Repeat Mode Specifications. Figure 16.5 shows the ADCON0 and ADCON1 Registers in Repeat Mode.

Table 16.3 Repeat Mode Specifications

| Item | Specification |
|-------------------------------------|---|
| Function | The Input voltage on one pin selected by CH2 to CH0 and ADGSEL0 bits is A/D converted repeatedly |
| 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" (capture) Timer Z interrupt request 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 AN8 to AN11 |
| Reading of result of A/D converter | Read AD register |

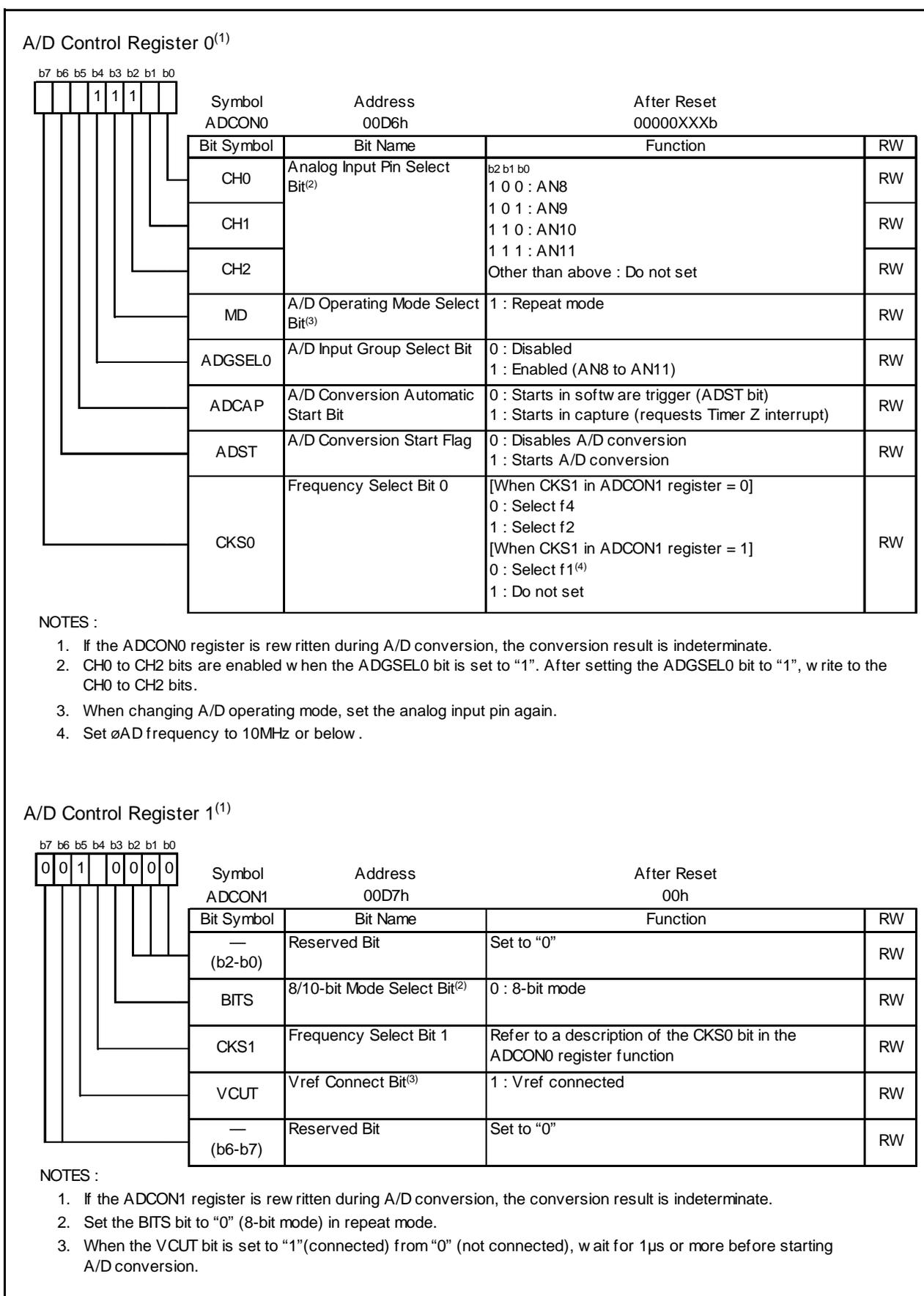


Figure 16.5 ADCON0 and ADCON1 Registers in Repeat Mode

16.3 Sample and Hold

When the SMP bit in the ADCON2 register is set to "1" (with sample and hold function), A/D conversion rate per pin increases to $28\phi_{AD}$ cycles for 8-bit resolution or $33\phi_{AD}$ cycles for 10-bit resolution. The sample and hold function is available in all operating modes. Start the A/D conversion after selecting whether the sample and hold circuit is to be used or not.

When performing the A/D conversion, charge the comparator capacitor in the microcomputer.

Figure 16.6 shows the Timing Diagram of A/D Conversion.

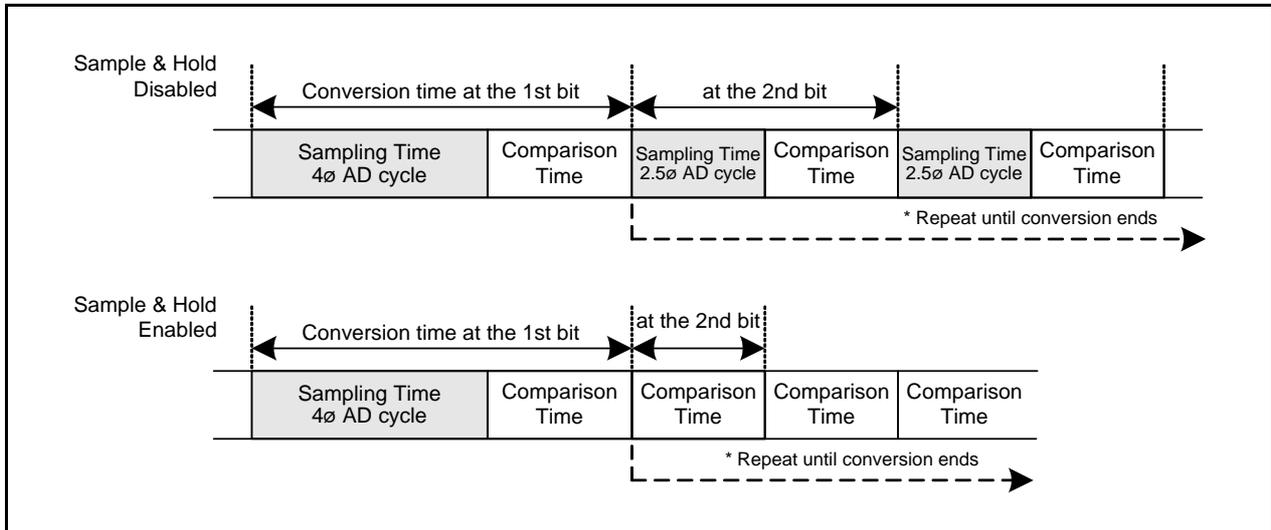


Figure 16.6 Timing Diagram of A/D Conversion

16.4 A/D Conversion Cycles

Figure 16.7 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 | $49\phi_{AD}$ | $4\phi_{AD}$ | $2.0\phi_{AD}$ | $2.5\phi_{AD}$ | $2.5\phi_{AD}$ | $8.0\phi_{AD}$ |
| Without Sample & Hold | 10 bits | $59\phi_{AD}$ | $4\phi_{AD}$ | $2.0\phi_{AD}$ | $2.5\phi_{AD}$ | $2.5\phi_{AD}$ | $8.0\phi_{AD}$ |
| With Sample & Hold | 8 bits | $28\phi_{AD}$ | $4\phi_{AD}$ | $2.5\phi_{AD}$ | $0.0\phi_{AD}$ | $2.5\phi_{AD}$ | $4.0\phi_{AD}$ |
| With Sample & Hold | 10 bits | $33\phi_{AD}$ | $4\phi_{AD}$ | $2.5\phi_{AD}$ | $0.0\phi_{AD}$ | $2.5\phi_{AD}$ | $4.0\phi_{AD}$ |

Figure 16.7 A/D Conversion Cycles

16.5 Internal Equivalent Circuit of Analog Input

Figure 16.8 shows the Internal Equivalent Circuit of Analog Input.

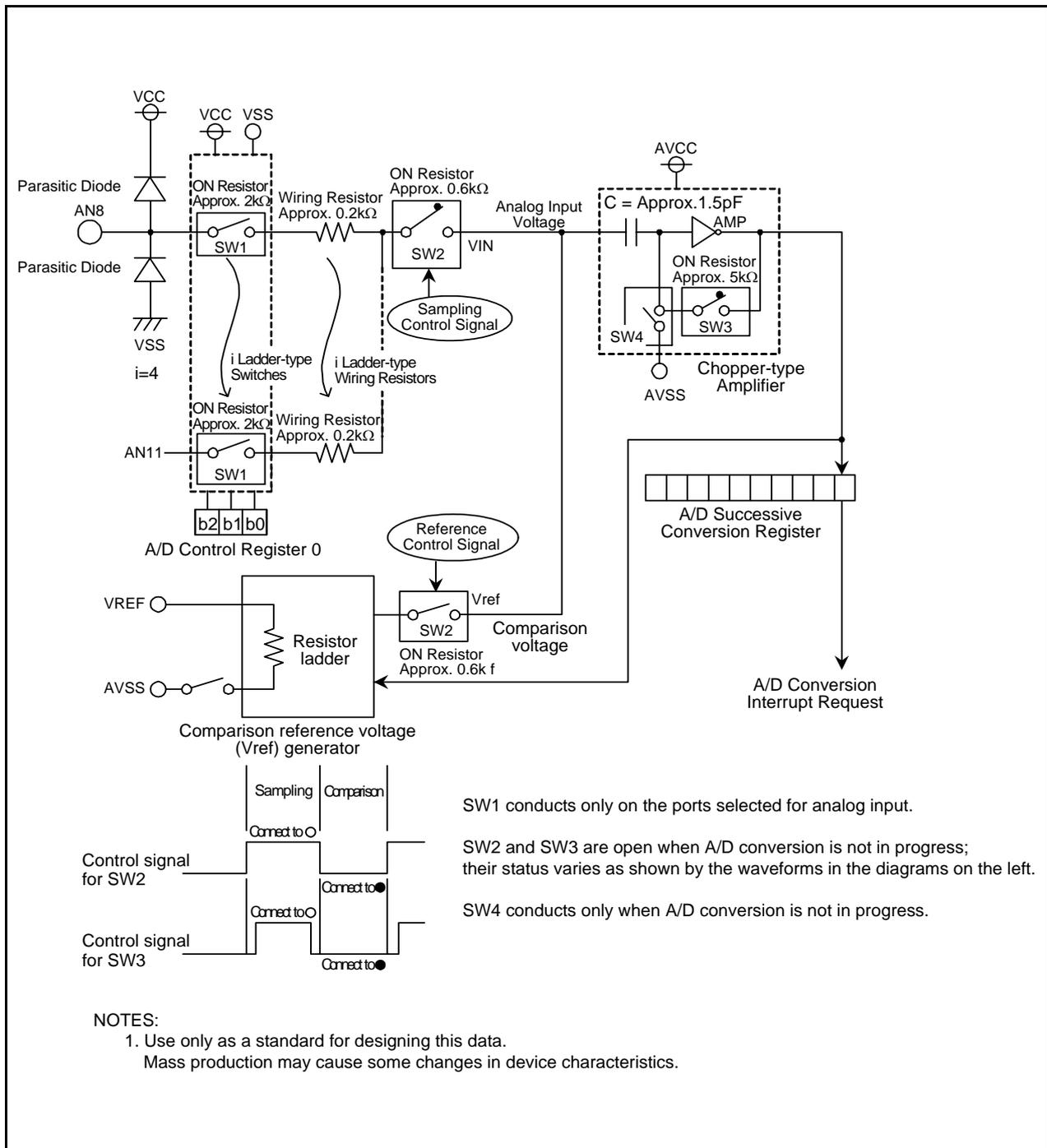


Figure 16.8 Internal Equivalent Circuit of Analog Input

16.6 Inflow Current Bypass Circuit

Figure 16.9 shows the Configuration of the Inflow Current Bypass Circuit, Figure 16.10 shows the Example of an Inflow Current Bypass Circuit where VCC or More is Applied.

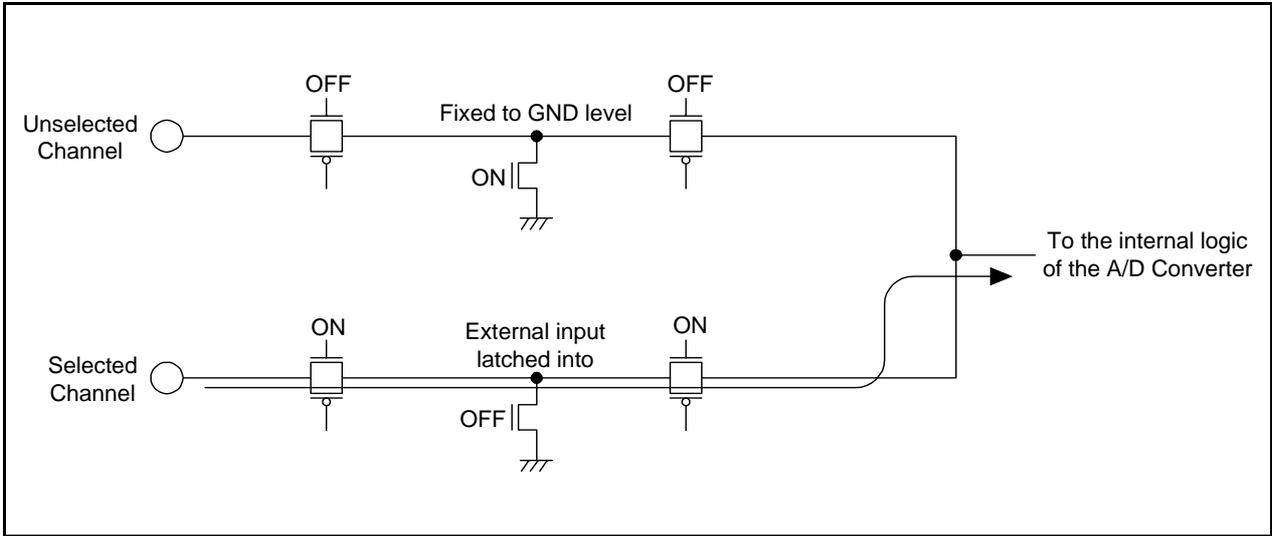


Figure 16.9 Configuration of the Inflow Current Bypass Circuit

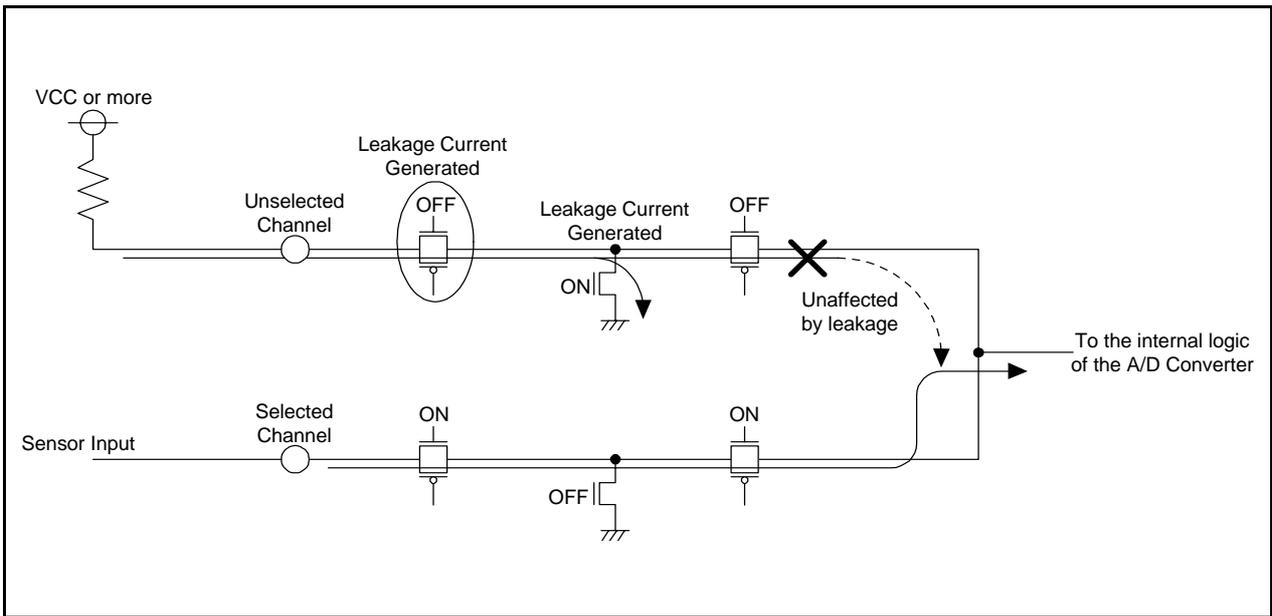


Figure 16.10 Example of an Inflow Current Bypass Circuit where VCC or More is Applied

17. Programmable I/O Ports

Programmable Input/Output ports (hereafter referred to as “I/O ports”) have 13 ports of the P1, P3_3 to P3_5, P3_7, and P4_5. Also, the main clock oscillation circuit is not used, the P4_6 and P4_7 can be used as the input port only. Table 17.1 lists the Overview of Programmable I/O Ports.

Table 17.1 Overview of Programmable I/O Ports

| Ports | I/O | Output Form | I/O Setting | Internal Pull-Up Resistor | Drive Capacity Selection |
|---------------------------|-----|---------------------------|---------------|---------------------------------|--|
| P1 | I/O | CMOS3 State | Set every bit | Set every 4 bits ⁽¹⁾ | Set every bit ⁽²⁾ of P1_0 to P1_3 |
| P3_3, P4_5 | I/O | CMOS3 State | Set every bit | Set every bit ⁽¹⁾ | None |
| P3_4, P3_5, P3_7 | I/O | CMOS3 State | Set every bit | Set every 3 bits ⁽¹⁾ | None |
| P4_6, P4_7 ⁽³⁾ | I | (Without output function) | None | None | None |

NOTES:

1. In input mode, whether the internal pull-up resistor is connected or not can be selected by the PUR0 and PUR1 registers.
2. This port can be used as the LED drive port by setting the DRR register to “1” (High).
3. When the main clock oscillation circuit is not used, these ports can be used as the input port only.

17.1 Functions of Programmable I/O Ports

The PDi_j (j=0 to 7) bit in the PDi (i=1,3 and 4) register controls I/O of the ports P1, P3_3 to P3_5, P3_7 and P4_5. The Pi register consists of a port latch to hold output data and a circuit to read pin state. Figures 17.1 to 17.3 show the Configurations of Programmable I/O Ports.

Table 17.2 lists the Functions of Programmable I/O Ports. Also, Figure 17.5 shows the PD1, PD3 and PD4 Registers. Figure 17.6 shows the P1, P3 and P4 Registers, Figure 17.7 shows the PUR0 and PUR1 Registers and Figure 17.8 shows the DRR Register.

Table 17.2 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 in the port latch, it is output from the pin. |

NOTES:

1. Nothing is assigned to the PD3_0 to PD3_2, PD3_6, PD4_0 to PD4_4, PD4_6 and PD4_7 bits.

17.2 Effect on Peripheral Functions

Programmable I/O ports function as I/O of peripheral functions (Refer to **Table 1.6** Pin Name Information by Pin Number). Table 17.3 lists the Setting of PDi_j Bit When Functioning as I/O of Peripheral Functions. Refer to descriptions of each function for how to set peripheral functions.

Table 17.3 Setting of PDi_j Bit When Functioning as I/O of Peripheral Functions

| I/O of Peripheral Functions | PDi _j Bit Setting of Port shared with Pin |
|-----------------------------|---|
| Input | Set this bit to “0” (input mode). |
| Output | This bit can be set to both “0” and “1” (output regardless of the port setting) |

17.3 Pins Other than Programmable I/O Ports

Figure 17.4 shows the Configuration of I/O Pins.

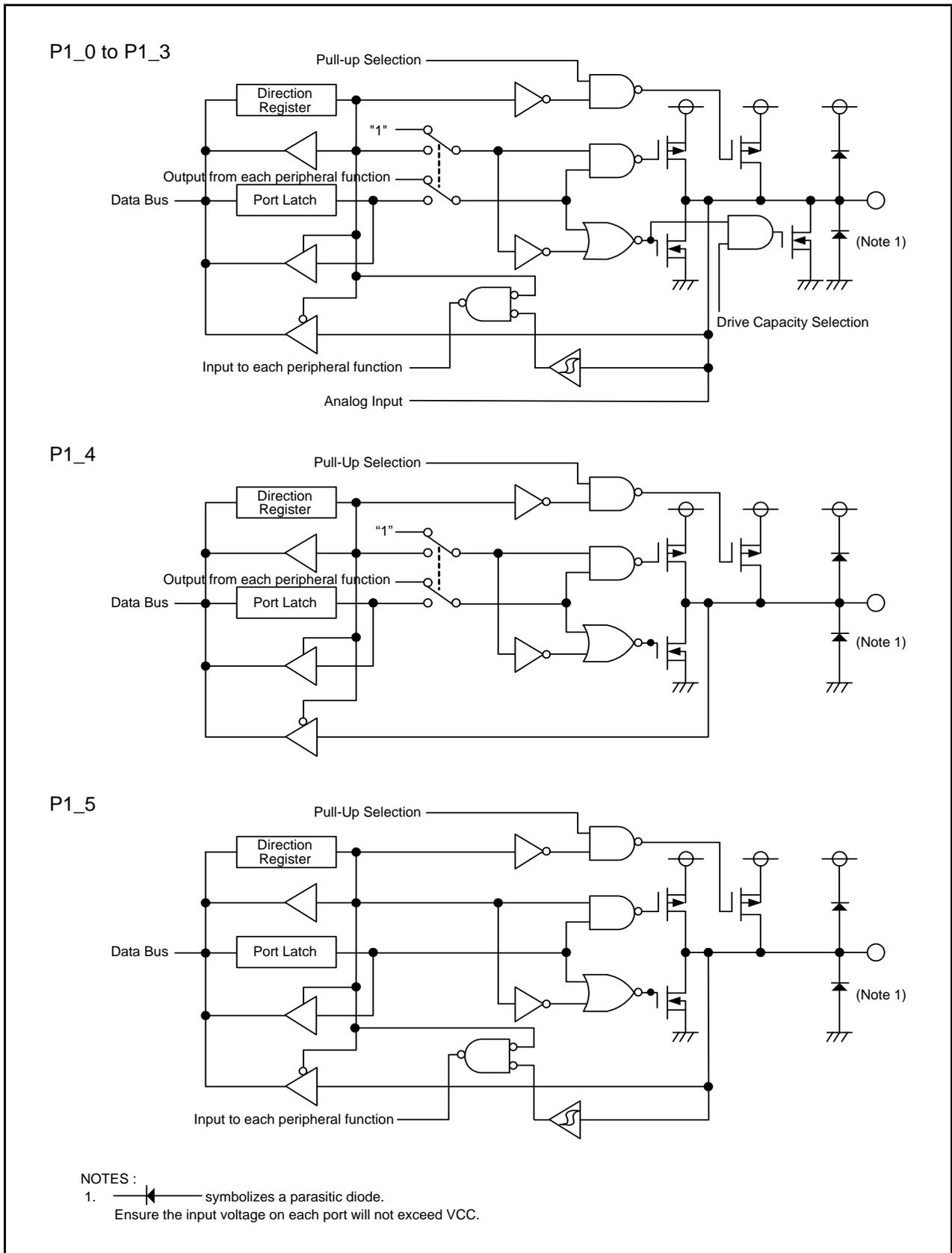


Figure 17.1 Configuration of Programmable I/O Ports (1)

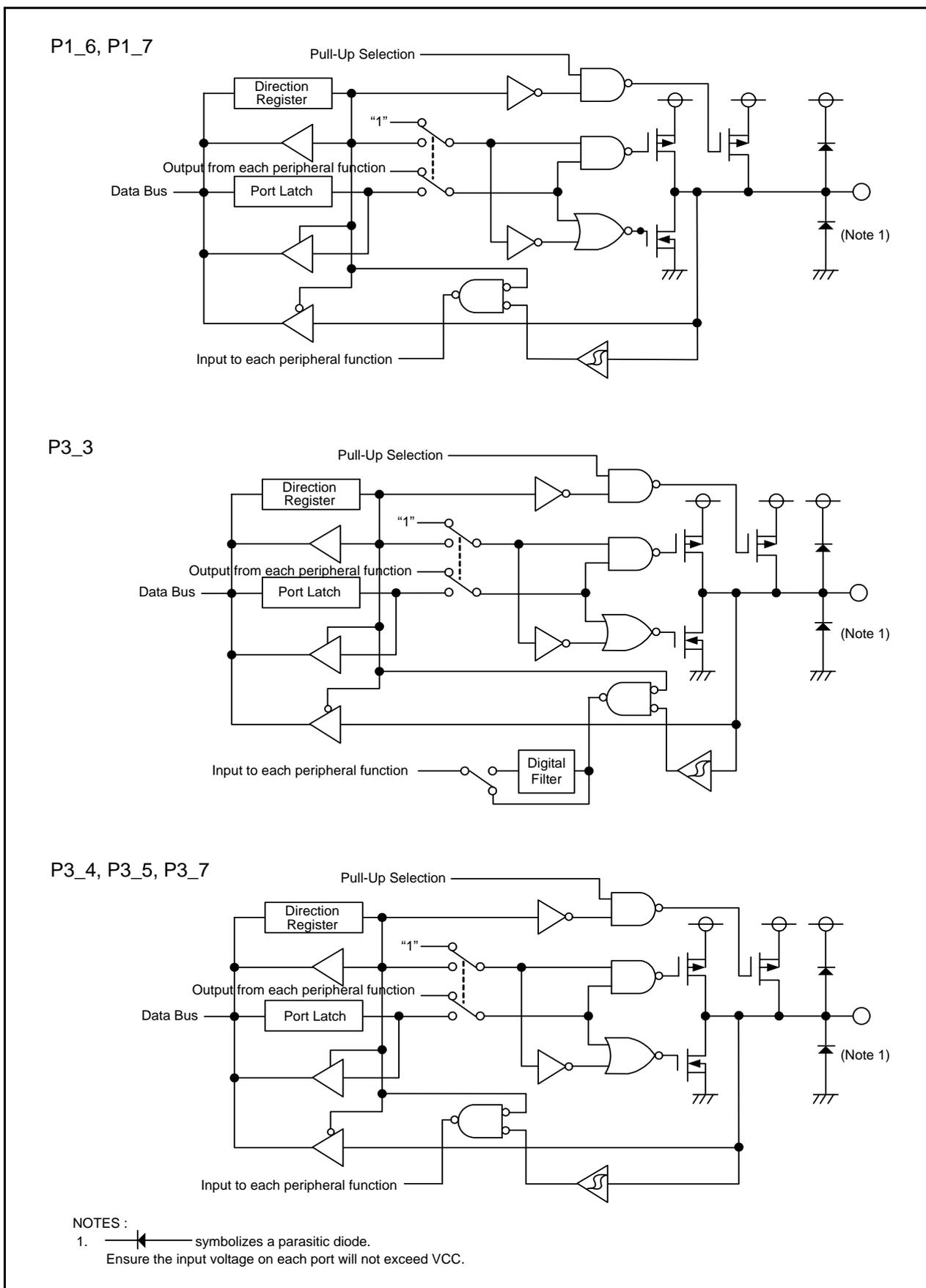


Figure 17.2 Configuration of Programmable I/O Ports (2)

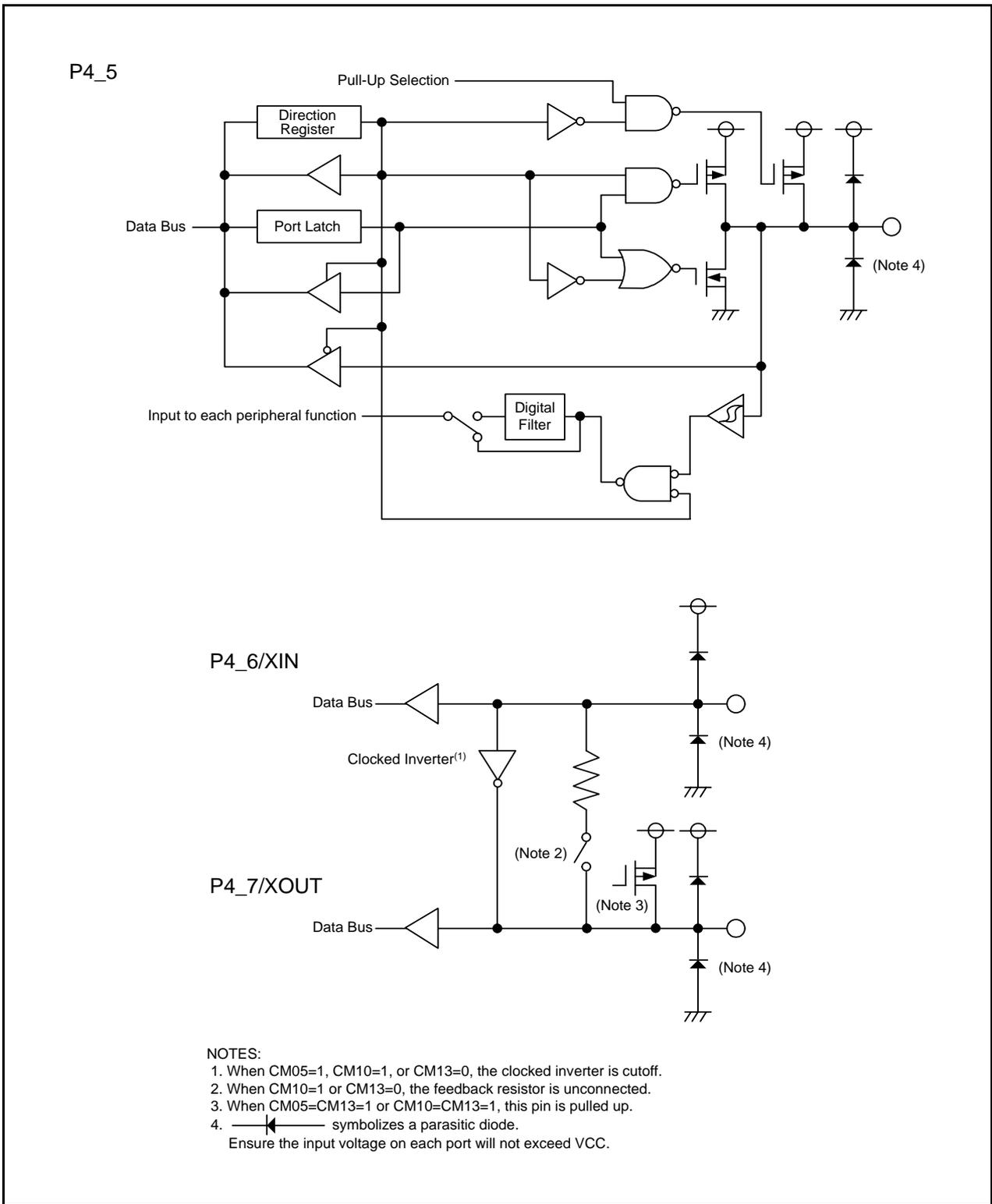
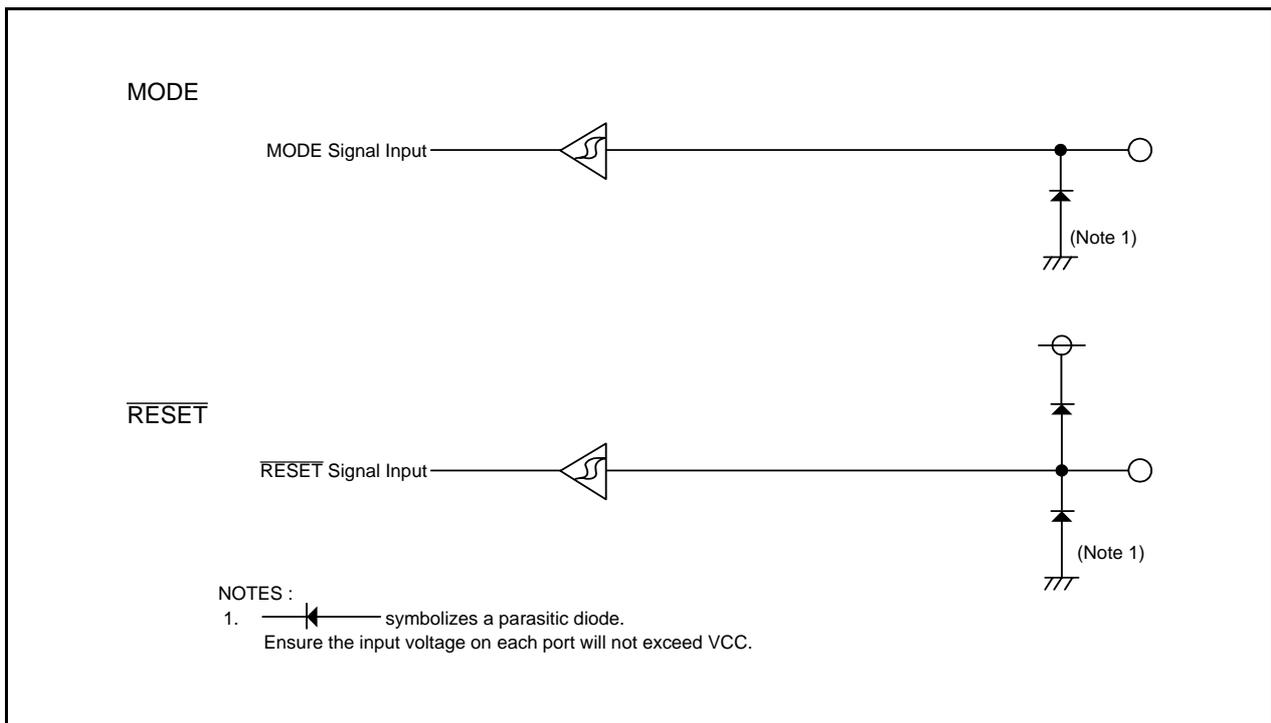


Figure 17.3 Configuration of Programmable I/O Ports (3)

**Figure 17.4 Configuration of I/O Pins**

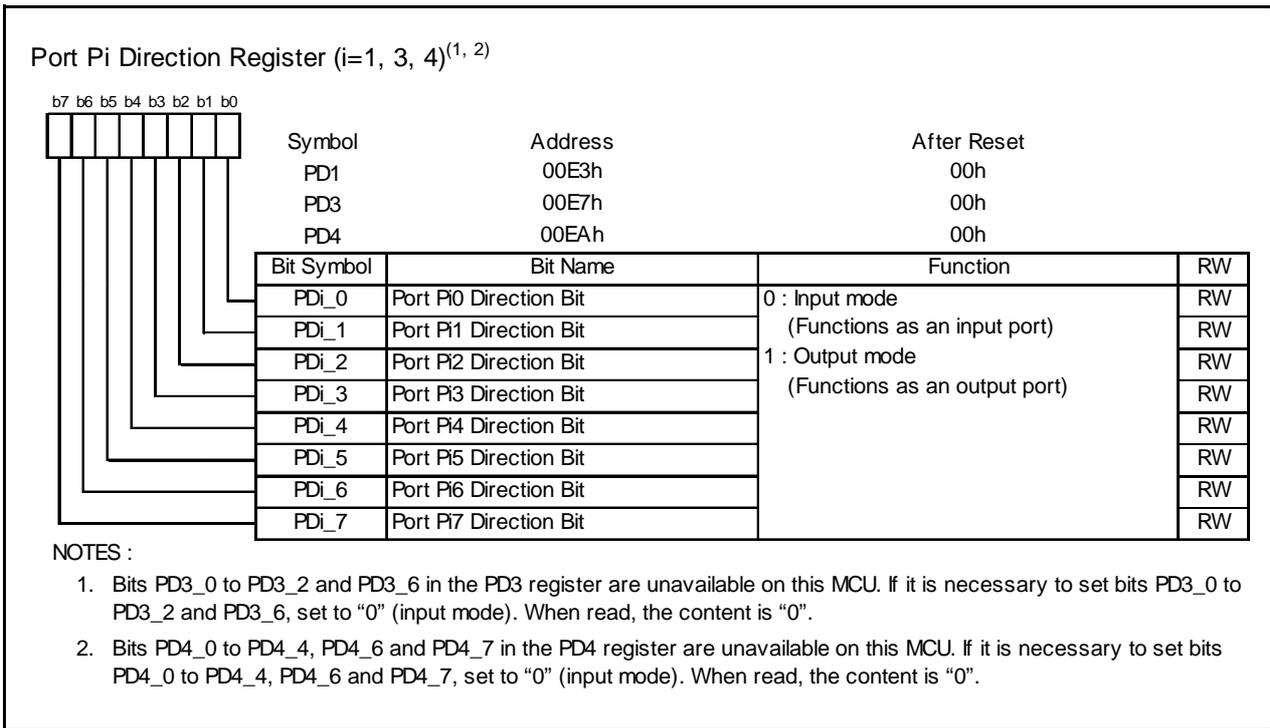


Figure 17.5 PD1, PD3 and PD4 Registers

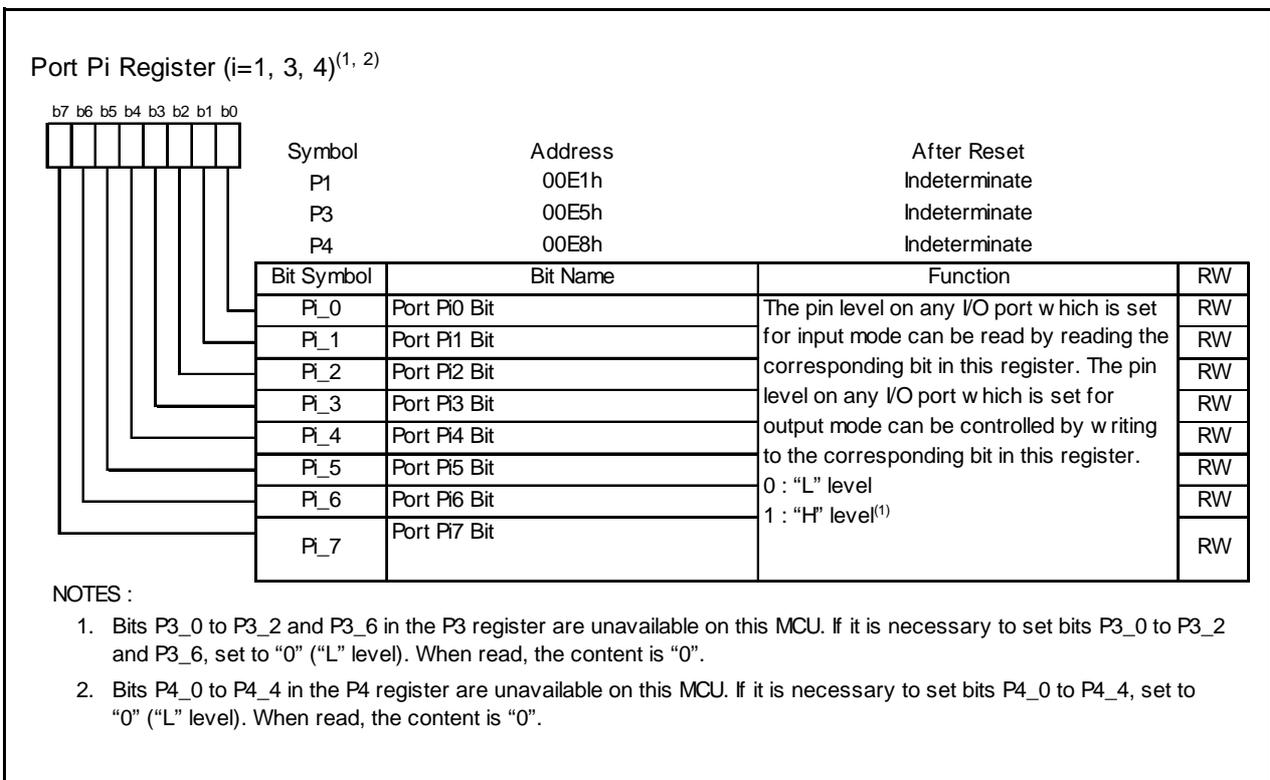


Figure 17.6 P1, P3 and P4 Registers

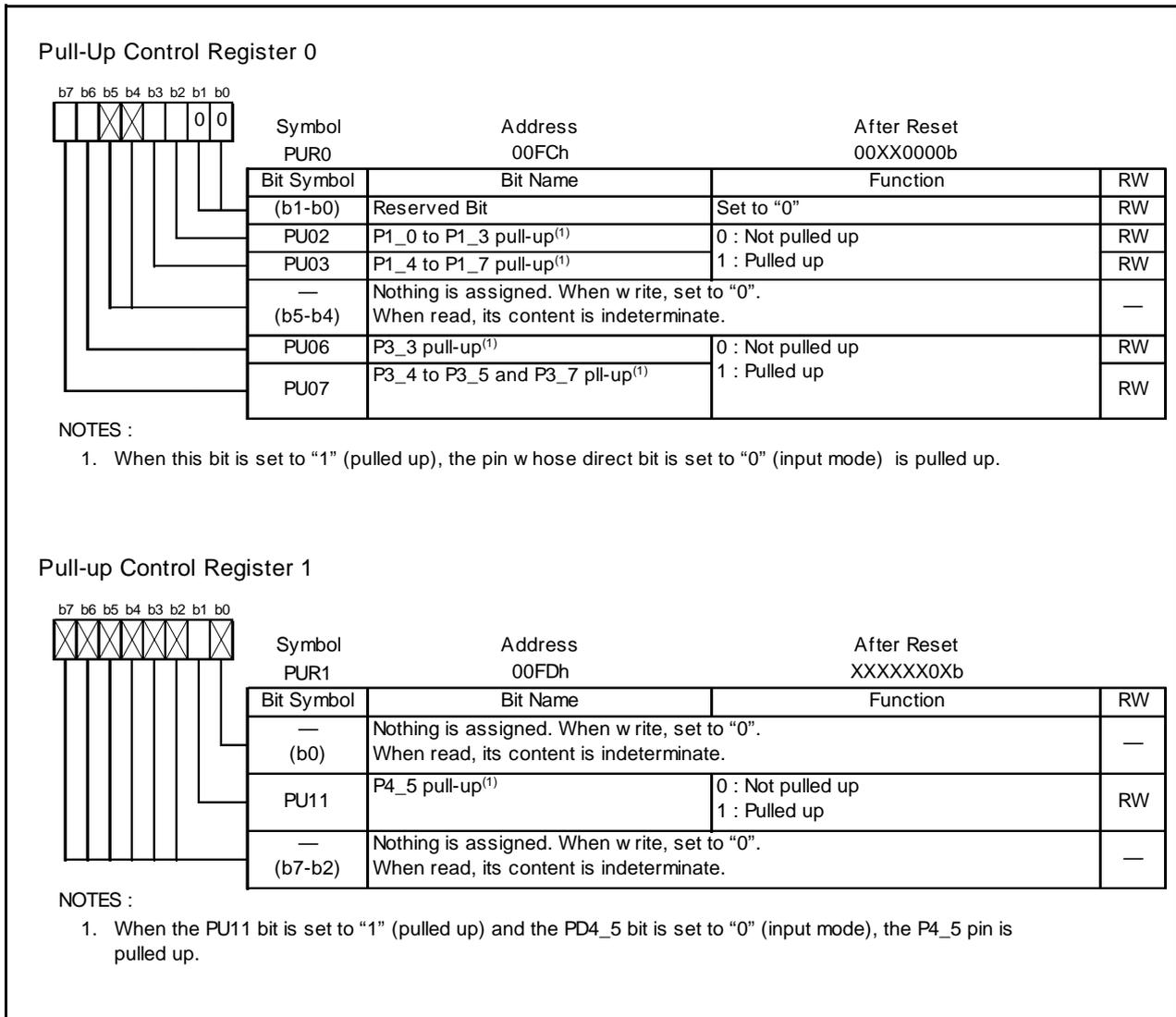


Figure 17.7 PUR0 and PUR1 Registers

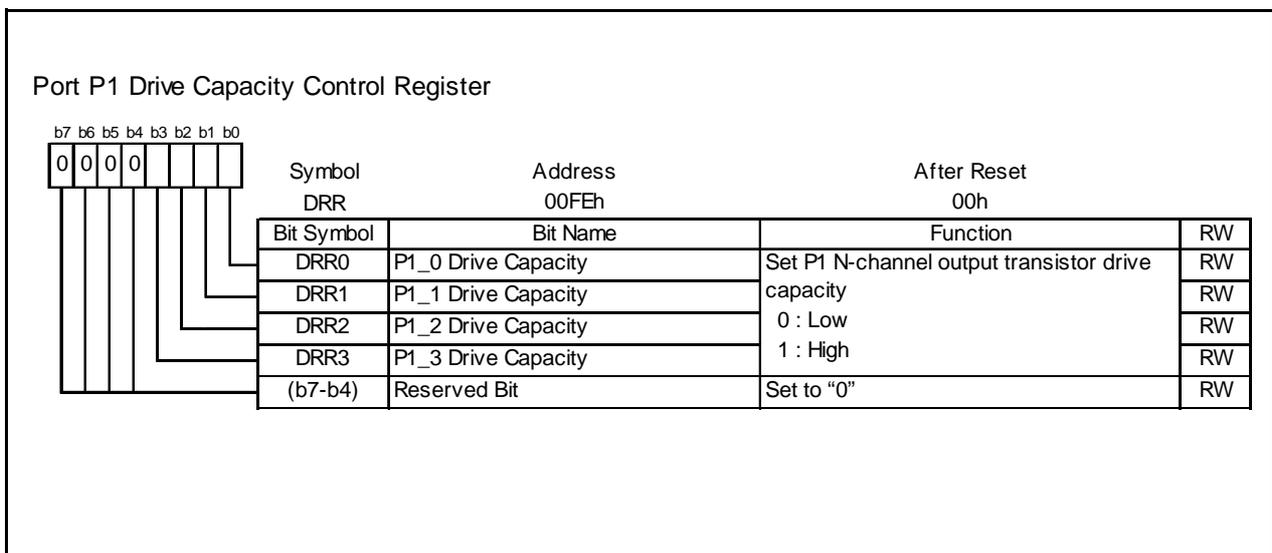


Figure 17.8 DRR Register

17.4 Port setting

Table 17.4 to Table 17.17 list the port setting.

Table 17.4 Port P1_0/ $\overline{\text{KI0}}$ /AN8/CMP0_0 Setting

| Register | PD1 | PUR0 | DRR | KIEN | ADCON0 | TCOUT | Function |
|---------------|-------|------|------|-------|------------------------|--------|-------------------------------|
| Bit | PD1_0 | PU02 | DRR0 | KI0EN | CH2, CH1, CH0, ADGSEL0 | TCOUT0 | |
| Setting Value | 0 | 0 | X | X | XXXXb | 0 | Input port (not pulled up) |
| | 0 | 1 | X | X | XXXXb | 0 | Input port (pulled up) |
| | 0 | 0 | X | 1 | XXXXb | 0 | $\overline{\text{KI0}}$ input |
| | 0 | 0 | X | X | 1001b | 0 | A/D Converter input (AN8) |
| | 1 | X | 0 | X | XXXXb | 0 | Output port |
| | 1 | X | 1 | X | XXXXb | 0 | Output port (High drive) |
| | X | X | X | X | XXXXb | 1 | CMP0_0 output |

X: "0" or "1"

Table 17.5 Port P1_1/ $\overline{\text{KI1}}$ /AN9/CMP0_1 Setting

| Register | PD1 | PUR0 | DRR | KIEN | ADCON0 | TCOUT | Function |
|---------------|-------|------|------|-------|------------------------|--------|-------------------------------|
| Bit | PD1_1 | PU02 | DRR1 | KI1EN | CH2, CH1, CH0, ADGSEL0 | TCOUT1 | |
| Setting Value | 0 | 0 | X | X | XXXXb | 0 | Input port (not pulled up) |
| | 0 | 1 | X | X | XXXXb | 0 | Input port (pulled up) |
| | 0 | 0 | X | 1 | XXXXb | 0 | $\overline{\text{KI1}}$ input |
| | 0 | 0 | X | X | 1011b | 0 | A/D Converter input (AN9) |
| | 1 | X | 0 | X | XXXXb | 0 | Output port |
| | 1 | X | 1 | X | XXXXb | 0 | Output port (High drive) |
| | X | X | X | X | XXXXb | 1 | CMP0_1 output |

X: "0" or "1"

Table 17.6 Port P1_2/ $\overline{\text{KI2}}$ /AN10/CMP0_2 Setting

| Register | PD1 | PUR0 | DRR | KIEN | ADCON0 | TCOUT | Function |
|---------------|-------|------|------|-------|------------------------|--------|-------------------------------|
| Bit | PD1_2 | PU02 | DRR2 | KI2EN | CH2, CH1, CH0, ADGSEL0 | TCOUT2 | |
| Setting Value | 0 | 0 | X | X | XXXXb | 0 | Input port (not pulled up) |
| | 0 | 1 | X | X | XXXXb | 0 | Input port (pulled up) |
| | 0 | 0 | X | 1 | XXXXb | 0 | $\overline{\text{KI2}}$ input |
| | 0 | 0 | X | X | 1101b | 0 | A/D Converter input (AN10) |
| | 1 | X | 0 | X | XXXXb | 0 | Output port |
| | 1 | X | 1 | X | XXXXb | 0 | Output port (High drive) |
| | X | X | X | X | XXXXb | 1 | CMP0_2 input |

X: "0" or "1"

Table 17.7 Port P1_3/ $\overline{\text{KI3}}$ /AN11/TZOUT Setting

| Register | PD1 | PUR0 | DRR | KIEN | ADCON0 | TZMR | TZOC | Function |
|---------------|-------|------|------|-------|---------------------------|-------------------|--------------|-------------------------------|
| Bit | PD1_3 | PU02 | DRR3 | KI3EN | CH2, CH1, CH0, ADGSEL0 | TZMOD1, TZMOD0 | TZOCNT | |
| Setting Value | 0 | 0 | X | X | XXXXb | 00b | X | Input port (not pulled up) |
| | 0 | 1 | X | X | XXXXb | 00b | X | Input port (pulled up) |
| | 0 | 0 | X | 1 | XXXXb | 00b | X | $\overline{\text{KI3}}$ input |
| | 0 | 0 | X | X | 1111b | 00b | X | A/D Converter input (AN11) |
| | 1 | X | 0 | X | XXXXb | 00b | X | Output port |
| | 1 | X | 1 | X | XXXXb | 00b | X | Output port (High drive) |
| | X | X | 0 | X | XXXXb | 01b | 1 | Output port |
| | X | X | 1 | X | XXXXb | 01b | 1 | Output port (High drive) |
| | X | X | X | X | XXXXb | 01b | 0 | TZOUT output |
| X | X | X | X | XXXXb | 1Xb | X | TZOUT output | |

X: "0" or "1"

Table 17.8 Port P1_4/TXD0 Setting

| Register | PD1 | PUR0 | U0MR | U0C0 | Function | |
|---------------|-------|------|------------------|------|----------------------------|------------------------------------|
| Bit | PD1_4 | PU03 | SMD2, SMD1, SMD0 | NCH | | |
| Setting Value | 0 | 0 | 000b | X | Input port (not pulled up) | |
| | 0 | 1 | 000b | X | Input port (pulled up) | |
| | 1 | X | 000b | X | Output port | |
| | X | X | X | 001b | 0 | TXD0 output, CMOS output |
| | | | | 100b | | |
| | | | | 101b | | |
| | | | | 110b | | |
| | X | X | X | 001b | 1 | TXD0 output, N-channel open output |
| | | | | 100b | | |
| | | | | 101b | | |
| | | | | 110b | | |

X: "0" or "1"

Table 17.9 Port P1_5/RXD0/ $\overline{\text{CNTR01}}$ /INT11 Setting

| Register | PD1 | PUR0 | UCON | TXMR | Function |
|---------------|-------|------|---------|----------------|---|
| Bit | PD1_5 | PU03 | CNTRSEL | TXMOD1, TXMOD0 | |
| Setting Value | 0 | 0 | X | XXb | Input port (not pulled up) |
| | 0 | 1 | X | XXb | Input port (pulled up) |
| | 0 | X | X | Other than 01b | RXD0 input |
| | 0 | X | 1 | Other than 01b | $\overline{\text{CNTR01}}$ /INT11 input |
| | 1 | X | X | Other than 01b | Output port |
| | 1 | X | 1 | Other than 01b | CNTR01 output |

Table 17.10 Port P1_6/CLK0/SSI Setting

| Register | PD1 | PUR0 | U0MR | Function |
|---------------|-------|------|-------------------------|------------------------------|
| Bit | PD1_6 | PU03 | SMD2, SMD1, SMD0, CKDIR | |
| Setting Value | 0 | 0 | Other than 0X10b | Input port (not pulled up) |
| | 0 | 1 | Other than 0X10b | Input port (pulled up) |
| | 0 | 0 | XXX1b | CLK0 (external clock) input |
| | 1 | X | Other than 0X10b | Output port |
| | X | X | 0X10b | CLK0 (internal clock) output |

X: "0" or "1"

Table 17.11 Port P1_7/CNTR00/INT10 Setting

| Register | PD1 | PUR0 | TXMR | UCON | Function |
|---------------|-------|------|----------------|---------|---|
| Bit | PD1_7 | PU03 | TXMOD1, TXMOD0 | CNTRSEL | |
| Setting Value | 0 | 0 | Other than 01b | X | Input port (not pulled up) |
| | 0 | 1 | Other than 01b | X | Input port (pulled up) |
| | 0 | 0 | Other than 01b | 0 | CNTR00/ $\overline{\text{INT10}}$ input |
| | 1 | X | Other than 01b | X | Output port |
| | X | X | Other than 01b | 0 | CNTR00 output |

X: "0" or "1"

Table 17.12 Port P3_3/TCIN/INT3/SSI/CMP1_0 Setting

| Register | PD3 | PUR0 | SSU (Refer to Table 15.3 Association between Communication Modes and I/O Pins) | | TCOUT | Function |
|---------------|-------|------|---|-------------------|--------|--------------------------------------|
| Bit | PD3_3 | PU06 | SSI Output Control | SSI Input Control | TCOUT3 | |
| Setting Value | 0 | 0 | 0 | 0 | 0 | Input port (not pulled up) |
| | 0 | 1 | 0 | 0 | 0 | Input port (pulled up) |
| | X | 0 | 0 | 1 | X | SSI input |
| | 1 | X | 0 | 0 | 0 | Output port |
| | X | X | 0 | 0 | 1 | CMP1_0 output |
| | X | X | 1 | 0 | X | SSI output |
| | 0 | X | 1 | 1 | 0 | TCIN input/ $\overline{\text{INT3}}$ |

X: "0" or "1"

Table 17.13 Port P3_4/SCS/CMP1_1 Setting

| Register | PD3 | PUR0 | SSU (Refer to Table 15.3 Association between Communication Modes and I/O Pins) | | TCOUT | Function |
|---------------|-------|------|---|---------------------------------------|--------|--------------------------------|
| Bit | PD3_4 | PU07 | $\overline{\text{SCS}}$ Output Control | $\overline{\text{SCS}}$ Input Control | TCOUT4 | |
| Setting Value | 0 | 0 | 0 | 0 | 0 | Input port (not pulled up) |
| | 0 | 1 | 0 | 0 | 0 | Input port (pulled up) |
| | 0 | 0 | 0 | 1 | 0 | $\overline{\text{SCS}}$ input |
| | 1 | X | 0 | 0 | 0 | Output port |
| | X | X | 0 | 0 | 1 | CMP1_1 output |
| | X | X | 1 | 0 | X | $\overline{\text{SCS}}$ output |

X: "0" or "1"

Table 17.14 Port P3_5/SSCK/CMP1_2 Setting

| Register | PD3 | PUR0 | SSU (Refer to Table 15.3 Association between Communication Modes and I/O Pins) | | TCOUT | Function |
|---------------|-------|------|---|--------------------|--------|----------------------------|
| Bit | PD3_5 | PU07 | SSCK Output Control | SSCK Input Control | TCOUT5 | |
| Setting Value | 0 | 0 | 0 | 0 | 0 | Input port (not pulled up) |
| | 0 | 1 | 0 | 0 | 0 | Input port (pulled up) |
| | 0 | 0 | 0 | 1 | 0 | SSCK input |
| | 1 | X | 0 | 0 | 0 | Output port |
| | X | X | 0 | 0 | 1 | CMP1_2 output |
| | X | X | 1 | 0 | X | SSCK output |

X: "0" or "1"

Table 17.15 Port P3_7/CNTR0/SSO Setting

| Register | PD3 | PUR0 | SSU (Refer to Table 15.3 Association between Communication Modes and I/O Pins) | | TXMR | UCON | Function |
|---------------|-------|------|---|-------------------|--------|----------------|----------------------------|
| Bit | PD3_7 | PU07 | SSO Output Control | SSO Input Control | TXOCNT | U1SEL1, U1SEL0 | |
| Setting Value | 0 | 0 | 0 | 0 | 0 | 0Xb | Input port (not pulled up) |
| | 0 | 1 | 0 | 0 | 0 | 0Xb | Input port (pulled up) |
| | 1 | X | 0 | 0 | 0 | 0Xb | Output port |
| | X | X | 0 | 0 | 1 | XXb | CNTR0 output pin |
| | X | X | 0 | 1 | X | XXb | SSO input pin |
| | X | X | 1 | 0 | X | XXb | SSO output pin |

X: "0" or "1"

Table 17.16 Port XIN/P4_6, XOUT/P4_7 Setting

| Register | CM1 | CM1 | CM0 | Circuit Specification | | Function |
|---------------|------|------|------|-----------------------|---------------------|---|
| Bit | CM13 | CM10 | CM05 | Oscillation Buffer | Feedback Resistance | |
| Setting Value | 1 | 1 | 1 | OFF | OFF | XIN-XOUT oscillation stop |
| | 1 | 0 | 1 | OFF | ON | External input to XIN pin, "H" output from XOUT pin |
| | 1 | 0 | 1 | OFF | ON | XIN-XOUT oscillation stop |
| | 1 | 0 | 0 | ON | ON | XIN-XOUT oscillation |
| | 0 | X | X | OFF | OFF | Input port |

X: "0" or "1"

Table 17.17 Port P4_5/INT0 Setting

| Register | PD4 | PUR1 | INTEN | Function |
|---------------|-------|------|--------|----------------------------|
| Bit | PD4_5 | PU11 | INT0EN | |
| Setting Value | 0 | 0 | 0 | Input port (not pulled up) |
| | 0 | 1 | 0 | Input port (pulled up) |
| | 0 | 0 | 1 | INT0 input |
| | 1 | X | X | Output port |

X: "0" or "1"

17.5 Unassigned Pin Handling

Table 17.18 lists the Unassigned Pin Handling. Figure 17.9 show the Unassigned Pin Handling.

Table 17.18 Unassigned Pin Handling

| Pin Name | Connection |
|------------------------------------|---|
| Ports P1, P3_3 to P3_5, P3_7, P4_5 | <ul style="list-style-type: none"> • After setting to input mode, connect every pin to VSS via a resistor (pull-down) or connect every 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 resistor (pull-up)(2) |
| AVCC, VREF | Connect to VCC |
| RESET ⁽³⁾ | Connect to VCC via a resistor (pull-up)(2) |

NOTES:

1. When setting these ports to output mode and leaving them open, they remain input mode until they are switched to output mode by a program. The voltage level of these pins may be indeterminate and the power current may increase while the ports remain input mode. The content of the direction registers may change due to noise or out of control caused by noise. In order to enhance program reliability, set the direction registers periodically by a program.
2. Connect these unassigned pins to the microcomputer using the shortest wire length (within 2 cm) as possible.
3. When power-on reset function is used.

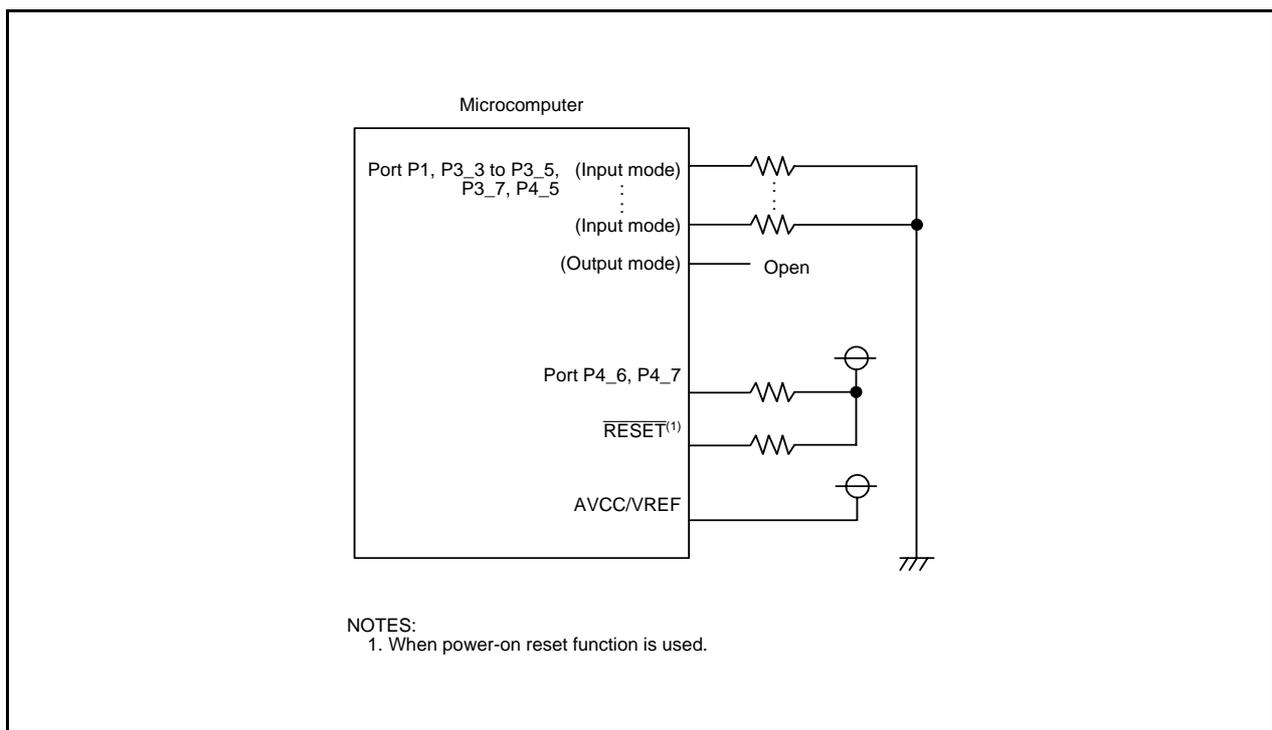


Figure 17.9 Unassigned Pin Handling

18. Flash Memory Version

18.1 Overview

In the flash memory version, rewrite operations to the flash memory can be performed in three modes; CPU rewrite, standard serial I/O, parallel I/O modes.

Table 18.1 lists the Flash Memory Version Performance (see **Table 1.1** Performance Outline of the R8C/14 Group and **Table 1.2** Performance Outline of the R8C/15 Group for the items not listed on Table 18.1).

Table 18.1 Flash Memory Version Performance

| Item | | Specification |
|--|--|---|
| Flash Memory Operating Mode | | 3 modes (CPU rewrite, standard serial I/O, and parallel I/O mode) |
| Division of Erase Block | | See Figure 18.1 and Figure 18.2 |
| Program Method | | Byte unit |
| Erase Method | | Block erase |
| Program, Erase Control Method | | Program and erase control by software command |
| Rewrite Control Method | | Rewrite control for Block 0 and 1 by FMR02 bit in FMR0 register |
| | | Rewrite control for Block 0 by FMR16 bit and Block 1 by FMR16 bit |
| Number of Commands | | 5 commands |
| Program and Erase Endurance ⁽¹⁾ | Block0 and 1 (Program ROM) | R8C/14 Group : 100 times ; R8C/15 Group : 1,000 times |
| | BlockA and B (Data flash) ⁽²⁾ | 10,000 times |
| ID Code Check Function | | Standard serial I/O mode supported |
| ROM Code Protect | | For parallel I/O mode supported |

NOTES:

1. Definition of program and erase endurance.

The program and erase endurance is defined to be per-block. When the program and erase endurance is n times (n=100 or 10,000 times), to erase n times per block is possible. For example, if performing one-byte write to the distinct addresses on Block A of 1K-byte block 1,024 times and then erasing that block, the program and erase endurance is counted as one time. If rewriting more than 100 times, execute the program until the blank areas are all used to reduce the substantial rewrite endurance and then erase. Do not rewrite only particular blocks and rewrite to average the program and erase endurance to each block. Also keep the erase endurance as information and set up the limit endurance.

2. Blocks A and B are embedded only in the R8C/15 group.

Table 18.2 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 any area other than flash memory EW1 mode: Rewritable in flash memory | User ROM area is rewritten by using a dedicated serial programmer. | User ROM area is rewritten by using 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 |

18.2 Memory Map

The flash memory contains a user ROM area and a boot ROM area (reserved area). Figure 18.1 shows the Flash Memory Block Diagram for R8C/14 Group. Figure 18.2 shows the Flash Memory Block Diagram for R8C/15 Group.

The user ROM area of R8C/15 group contains an area (program ROM) which stores a microcomputer operating program and the 1-Kbyte Block A and B (data flash).

The user ROM area is divided into several blocks. The user ROM area can be rewritten in CPU rewrite and standard serial I/O and parallel I/O modes.

When rewriting the Block 0 and Block 1 in CPU rewrite mode, set the FMR02 bit in the FMR0 register to "1" (rewrite enables), and when setting the FMR15 bit in the FMR1 register to "0" (rewrite enables), Block 0 is rewritable. When setting the FMR16 bit to "0" (rewrite enables), Block 1 is rewritable.

The rewrite control program for standard serial I/O mode is stored in boot ROM area before shipment. The boot ROM area and the user ROM area share the same address, but have an another memory.

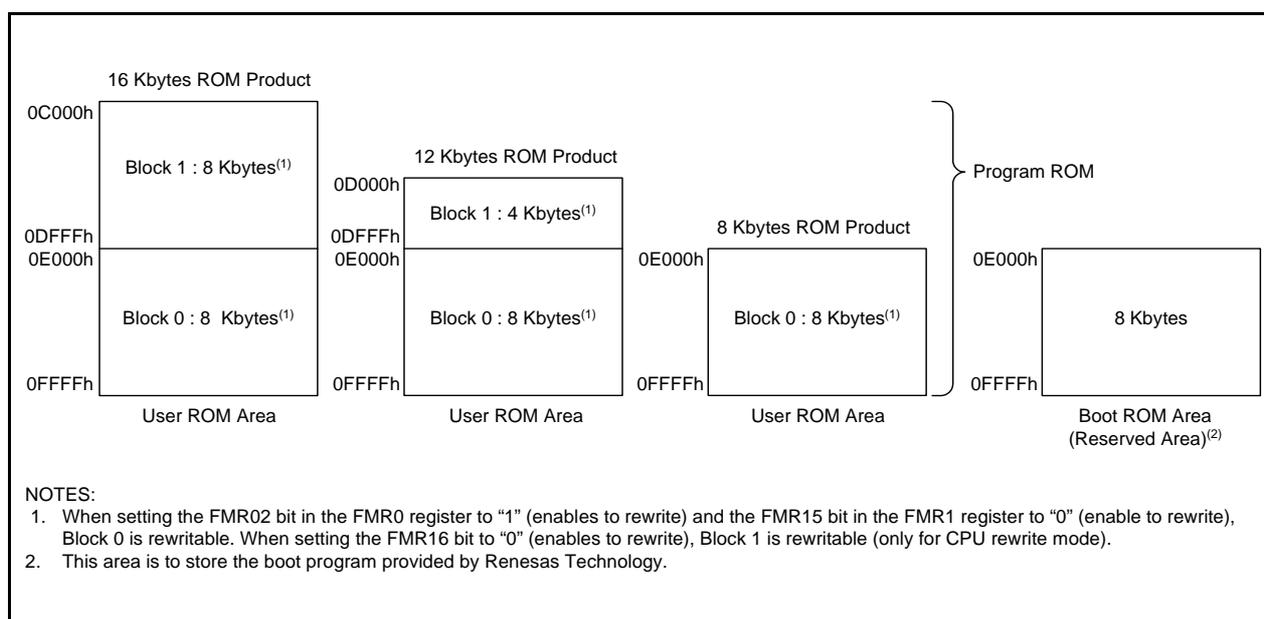


Figure 18.1 Flash Memory Block Diagram for R8C/14 Group

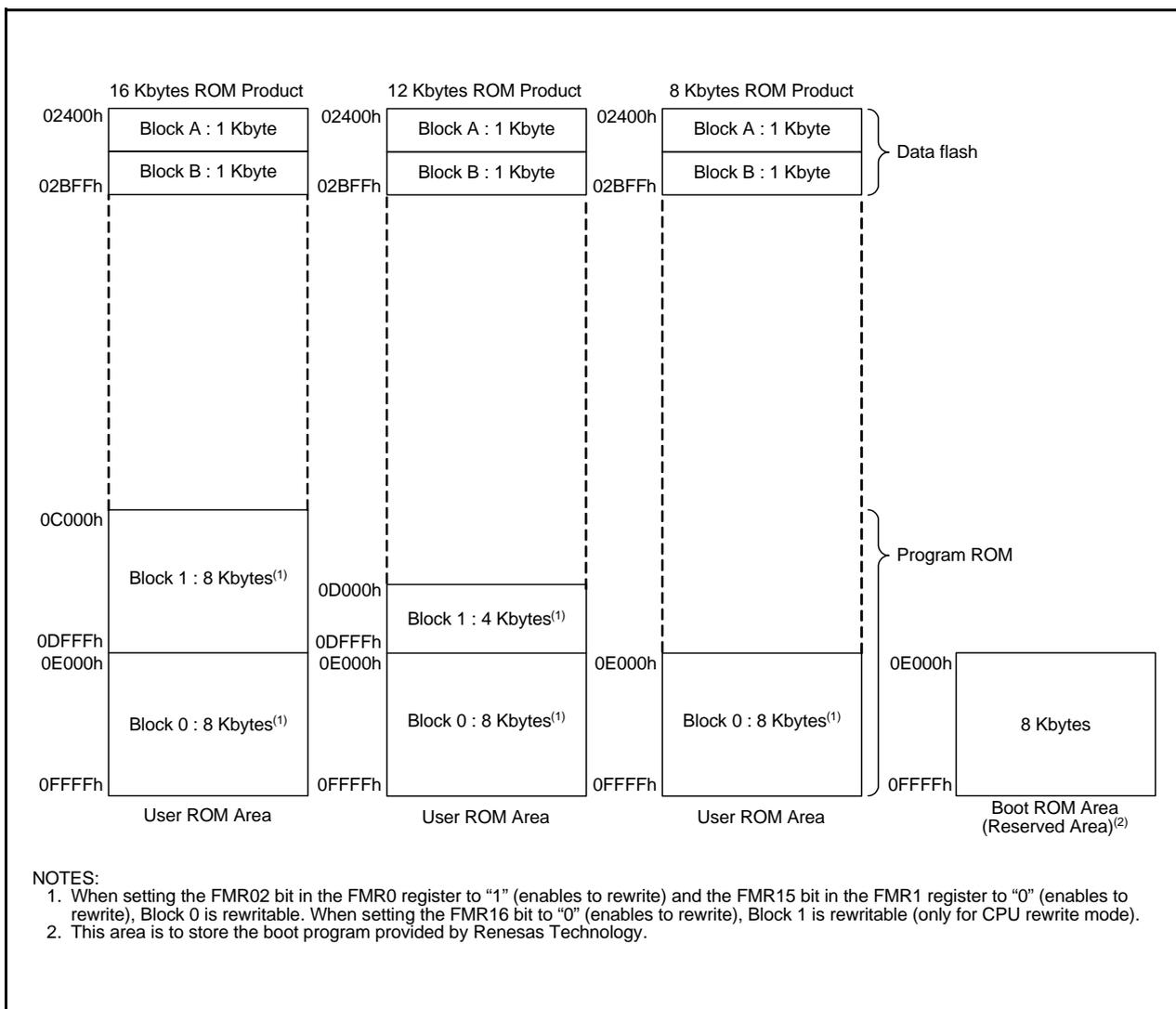


Figure 18.2 Flash Memory Block Diagram for R8C/15 Group

18.3 Functions To Prevent Flash Memory from Rewriting

Standard serial I/O mode contains an ID code check function, and the parallel I/O mode contains a ROM code protect function to prevent the flash memory from reading or rewriting easily.

18.3.1 ID Code Check Function

Use this function 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 determined whether they match. If the ID codes do not match, the commands sent from the programmer are not acknowledged. The ID code consists of 8-bit data, the areas of which, beginning with the first byte, are 00FFDFh, 00FFE3h, 00FFE7h, 00FEFh, 00FFF3h, 00FFF7h, and 00FFFBh. Write a program in which the ID codes are set at these addresses and write it in the flash memory.

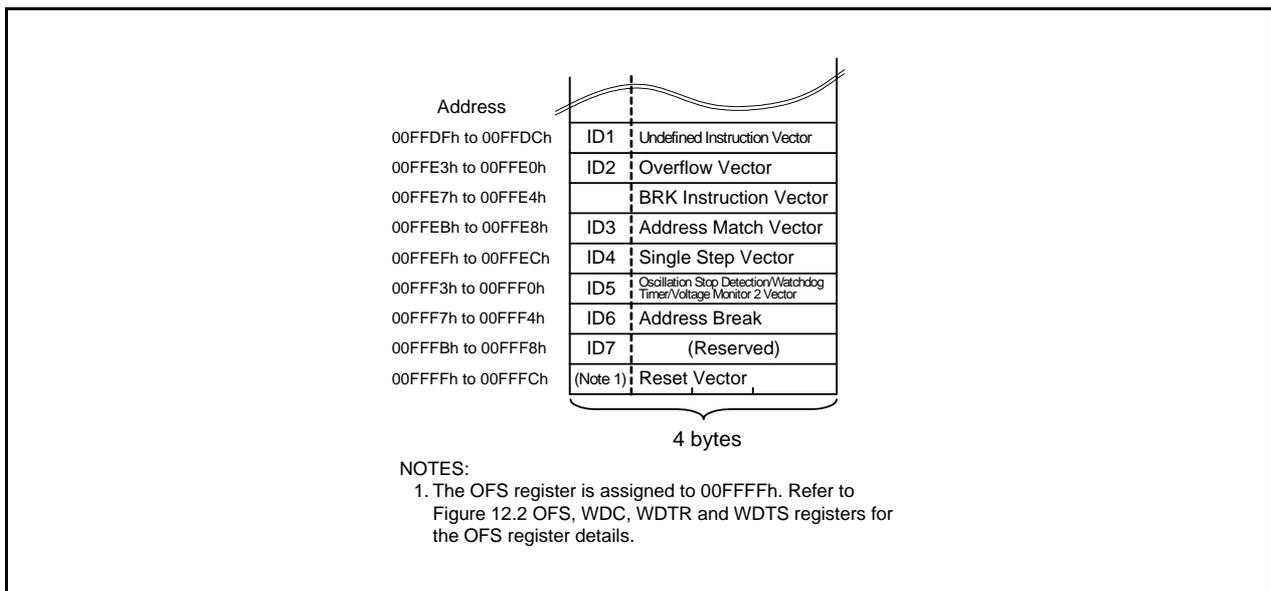


Figure 18.3 Address for ID Code Stored

18.3.2 ROM Code Protect Function

The ROM code protect function disables to read and change the internal flash memory by the OFS register in parallel I/O mode. Figure 18.4 shows the OFS Register.

The ROM code protect function is enabled by writing “0” to the ROMCP1 bit and “1” to the ROMCR bit and disables to read and change the internal flash memory. Once the 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.

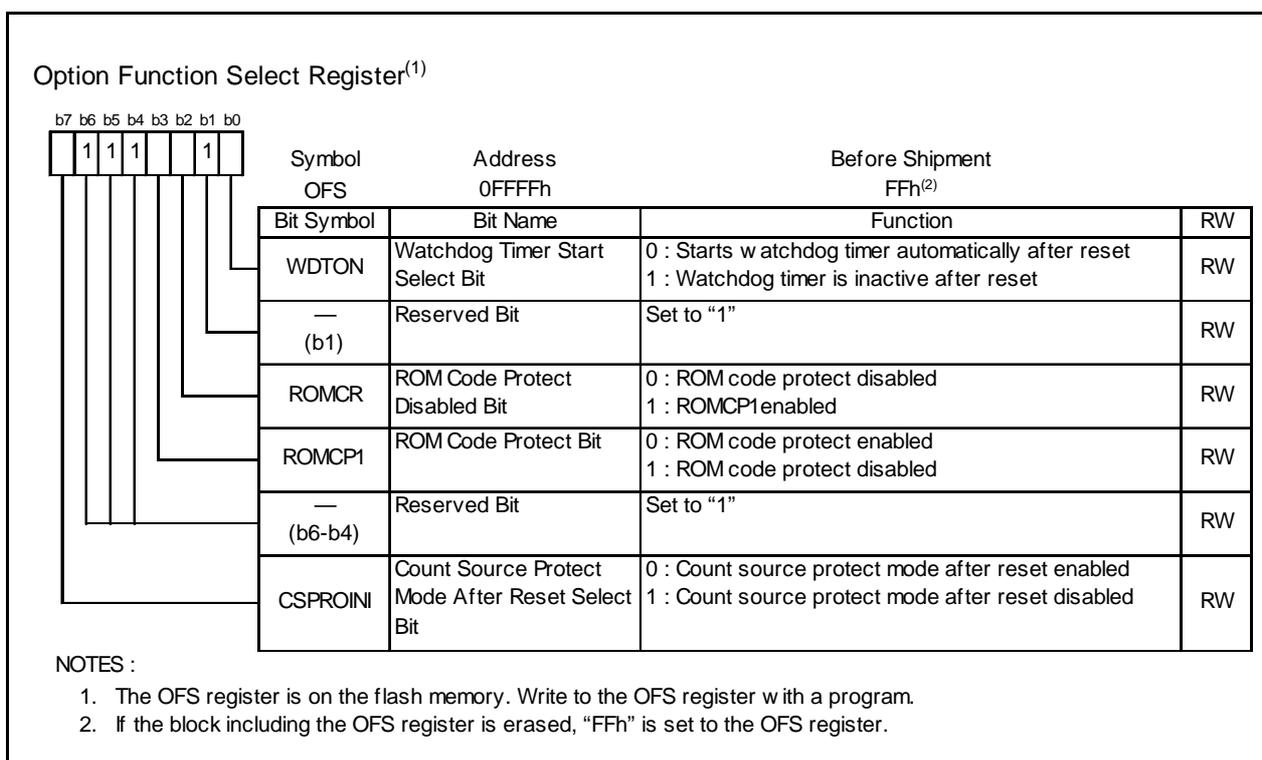


Figure 18.4 OFS Register

18.4 CPU Rewrite Mode

In CPU rewrite mode, user ROM area can be rewritten by executing software commands from the CPU. Therefore, the user ROM area can be rewritten directly while the microcomputer is mounted on a board without using such as a ROM programmer. Execute the program and block erase commands only to each block in user ROM area.

When an interrupt request is generated during an erase operation in CPU rewrite mode, the flash module contains an erase-suspend function which performs the interrupt process after the erase operation is halted temporarily. During the erase-suspend, user ROM area can be read by a program. CPU rewrite mode contains erase write 0 mode(EW0 mode) and erase write 1 mode(EW1 mode). Table 18.3 lists the Differences between EW0 Mode and EW1 Mode.

Table 18.3 Differences between EW0 Mode and EW1 Mode

| Item | EW0 Mode | EW1 Mode |
|---|---|---|
| Operating Mode | Single chip mode | Single chip mode |
| Area in which rewrite control program can be located | User ROM area | User ROM area |
| Area in which rewrite control program can be executed | Necessary to transfer to any areas other than the flash memory (e.g., RAM) before executing | Executing directly on user ROM area is possible |
| Area which can be rewritten | User ROM area | User ROM area However, other than the blocks which contain a rewrite control program ⁽¹⁾ |
| Software Command Restriction | None | <ul style="list-style-type: none"> • Program, block erase command Disable to execute on any block which contains a rewrite control program • Disables to execute the read status register command |
| Mode after Program or Erase | Read status register mode | Read array mode |
| 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 the FMR00, FMR06, and FMR07 bits in the FMR0 register by a program • Execute the read status register command and read the SR7, SR5, and SR4 bits in the status register. | Read the FMR00, FMR06, and FMR07 bits in the FMR0 register by a program |
| Condition for Transition to Erase-Suspend | Set the FMR40 and FMR41 bits 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 | 5MHz or below | No restriction to the following (clock frequency to be used) |

NOTES:

1. When setting the FMR02 bit in the FMR0 register to "1" (rewrite enables) and rewriting Block 0 is enabled by setting the FMR15 bit in the FMR1 register to "0" (rewrite enables). Rewriting Block 1 is enabled by setting the FMR16 bit to "0" (rewrite enables).

18.4.1 EW0 Mode

The microcomputer 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 a program and erase operations. The FMR0 register or the status register can determine status when program and erase operation complete.

When entering an erase-suspend, set the FMR40 bit to "1" (enables erase-suspend) and the FMR41 bit to "1" (requests erase-suspend). Wait for $t_d(SR-ES)$ and ensure that the FMR46 bit is set to "1" (enables reading) before accessing the user ROM area. The auto-erase operation restarts by setting the FMR41 bit to "0" (erase restarts).

18.4.2 EW1 Mode

The microcomputer enters 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 determine status when program and erase operation complete. Do not execute the read status register command in EW1 mode.

To enable the erase-suspend function, execute the block erase command after setting the FMR40 bit to "1" (enables erase-suspend). The interrupt to enter an erase-suspend should be in interrupt enabled status. After passing $t_d(SR-ES)$ since the block erase command is executed, an 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 is halted. If the auto-erase operation does not complete (FMR00 bit is "0") when the interrupt process completes, the auto-erase operation restarts by setting the FMR41 bit to "0" (erase restarts).

Figure 18.5 shows the FMR0 Register. Figure 18.6 shows the FMR1 and FMR4 Registers.

18.4.2.1 FMR00 Bit

This bit indicates the operating status of the flash memory. The bit is “0” during programming, erasing, or erase-suspend mode; otherwise, the bit is “1”.

18.4.2.2 FMR01 Bit

The microcomputer is made ready to accept commands by setting the FMR01 bit to “1” (CPU rewrite mode).

18.4.2.3 FMR02 Bit

The Block1 and Block0 do not accept the Program and Block Erase commands if the FMR02 bit is set to “0” (rewrite disabled).

The Block0 and Block1 are controlled rewriting in the FMR15 and FMR16 bits if the FMR02 bit is set to “1” (rewrite enabled).

18.4.2.4 FMSTP Bit

This bit is provided for initializing the flash memory control circuits, as well as for reducing the amount of current consumed in the flash memory. The flash memory is disabled against access by setting the FMSTP bit to “1”. Therefore, the FMSTP bit must be written to by a program in other than the flash memory.

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))
- When entering on-chip oscillator mode (main clock stop)

Figure 18.10 shows a flow chart to be followed before and after entering on-chip oscillator mode (main clock stop). 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.

18.4.2.5 FMR06 Bit

This is a read-only bit indicating the status of auto program operation. The bit is set to “1” when a program error occurs; otherwise, it is cleared to “0”. For details, refer to the description of the **18.4.5 Full Status Check**.

18.4.2.6 FMR07 Bit

This is a read-only bit indicating the status of auto erase operation. The bit is set to “1” when an erase error occurs; otherwise, it is set to “0”. Refer to **18.4.5 Full Status Check** for the details.

18.4.2.7 FMR11 Bit

Setting this bit to “1” (EW1 mode) places the microcomputer in EW1 mode.

18.4.2.8 FMR15 Bit

When the FMR02 bit is set to “1” (rewrite enabled) and the FMR15 bit is set to “0” (rewrite enabled), the Block0 accepts the program command and block erase command.

18.4.2.9 FMR16 Bit

When the FMR02 bit is set to “1” (rewrite enabled) and the FMR16 bit is set to “0” (rewrite enabled), the Block1 accepts the program command and block erase command.

18.4.2.10 FMR40 bit

The erase-suspend function is enabled by setting the FMR40 bit to “1” (enable).

18.4.2.11 FMR41 bit

In EW0 mode, the microcomputer enters erase-suspend mode when setting the FMR41 bit to “1” by a program. The FMR41 bit is automatically set to “1” (requests erase-suspend) when an interrupt request of an enabled interrupt is generated in EW1 mode, and then the microcomputer enters erase-suspend mode.

Set the FMR41 bit to “0” (erase restart) when the auto-erase operation restarts.

18.4.2.12 FMR46 bit

The FMR46 bit is set to “0” (disable reading) during auto-erase execution and set to “1” (enables reading) in erase-suspend mode. Do not access to the flash memory while this bit is set to “0”.

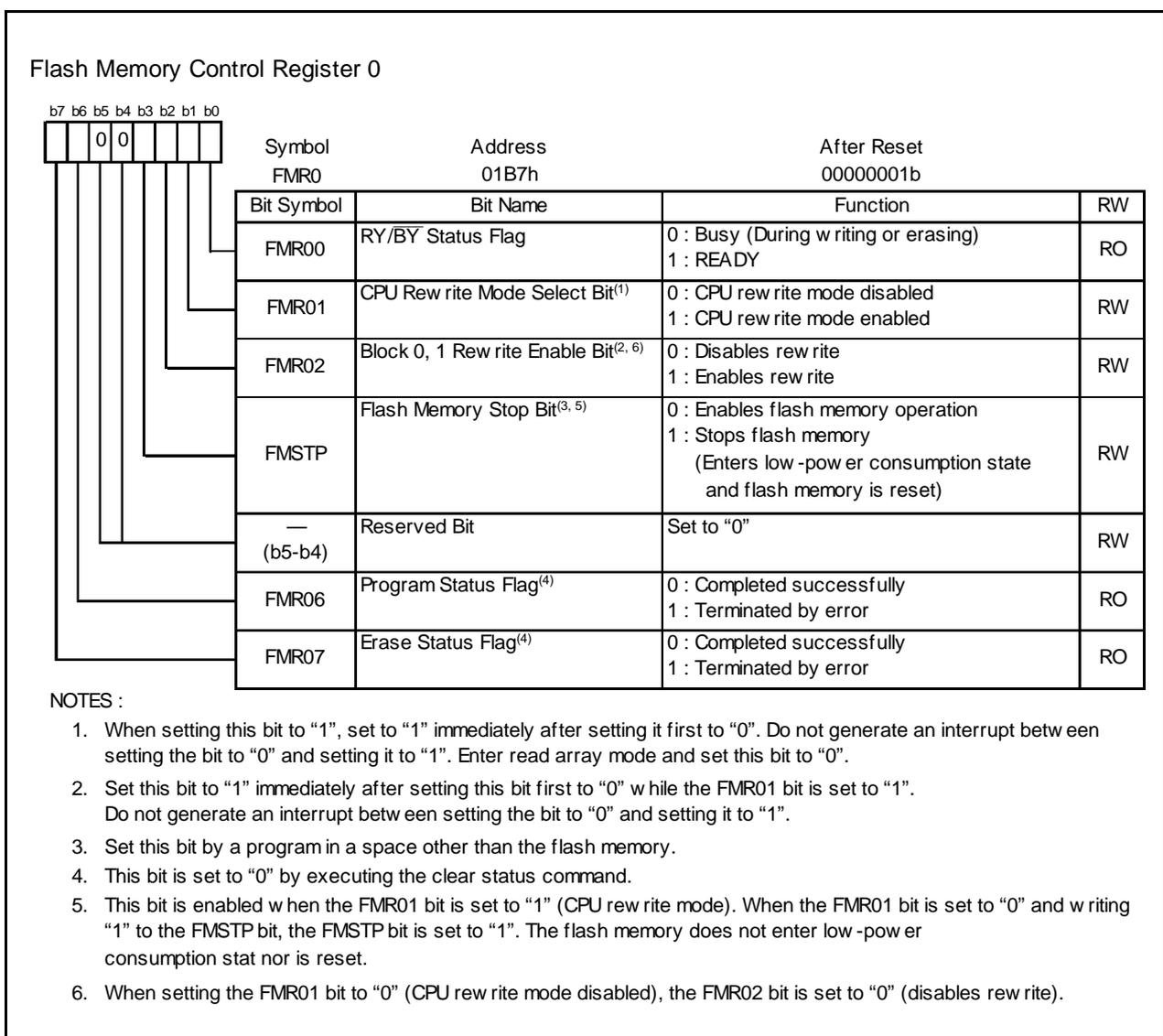


Figure 18.5 FMR0 Register

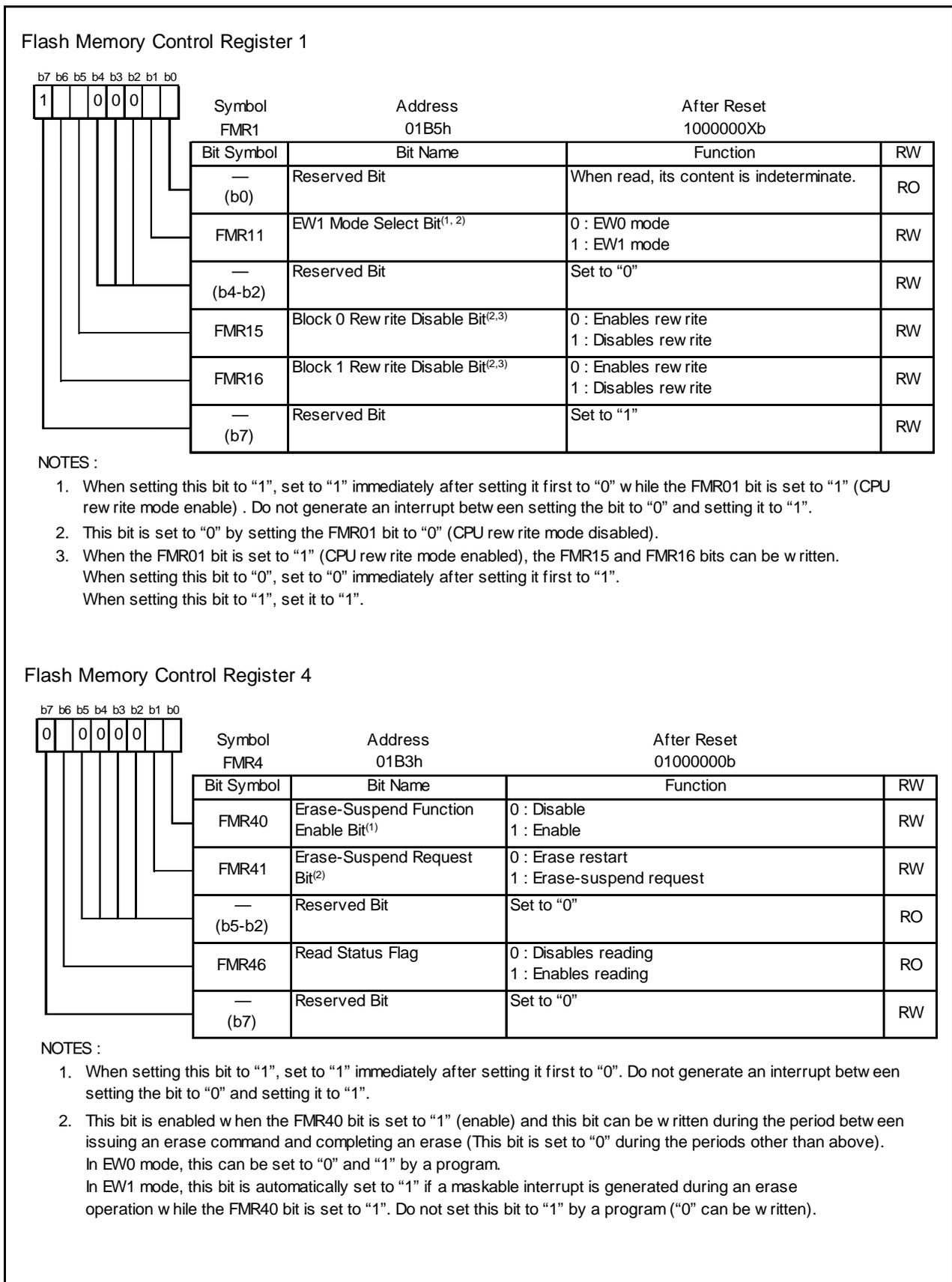


Figure 18.6 FMR1 and FMR4 Registers

Figure 18.7 shows the Timing on Suspend Operation.

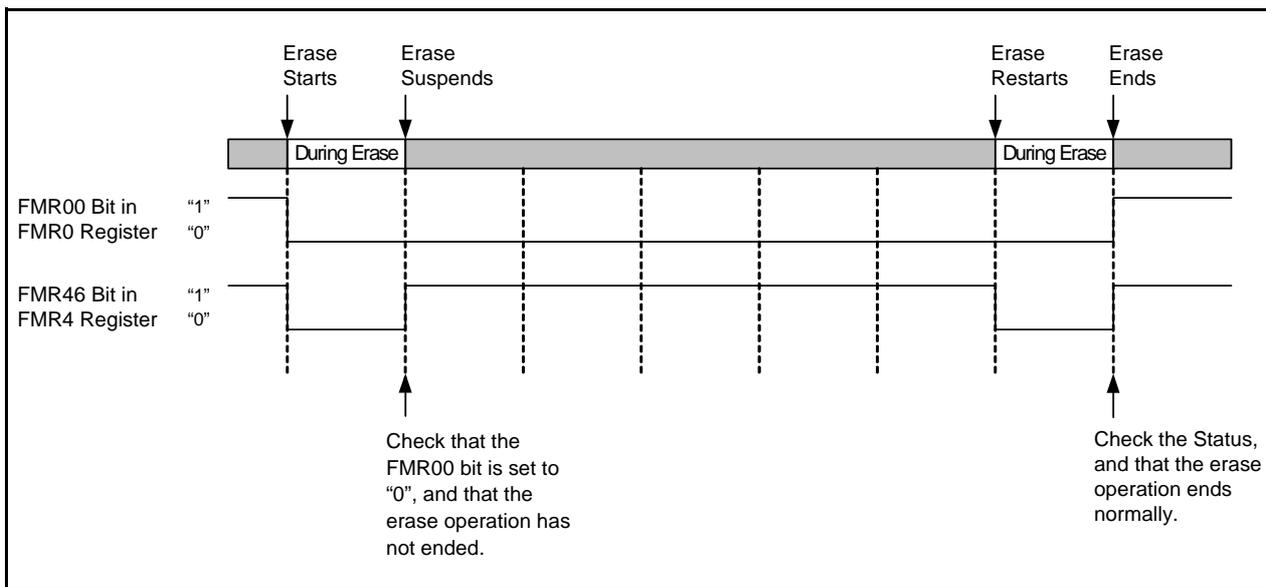


Figure 18.7 Timing on Suspend Operation

Figure 18.8 shows the How to Set and Exit EW0 Mode. Figure 18.9 shows the How to Set and Exit EW1 Mode.

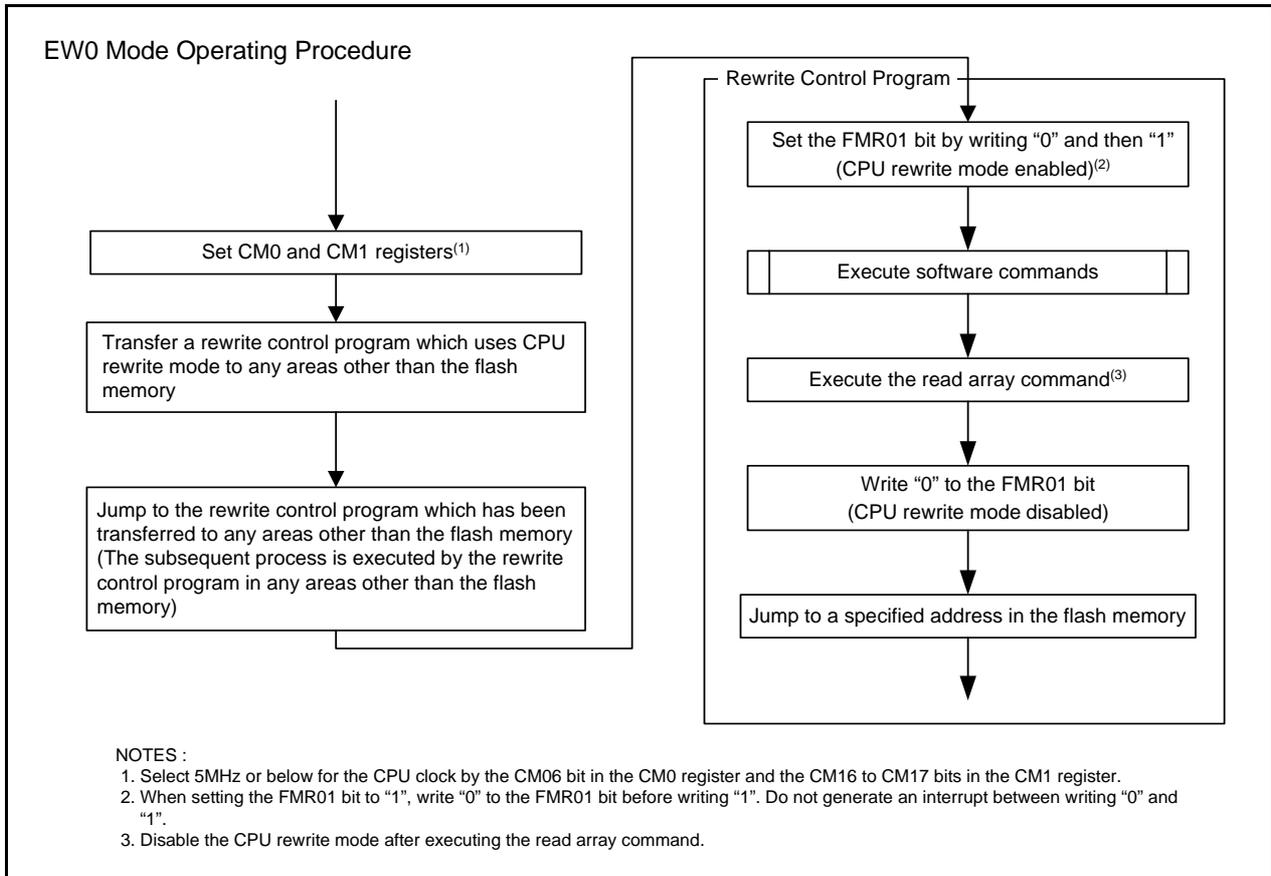


Figure 18.8 How to Set and Exit EW0 Mode

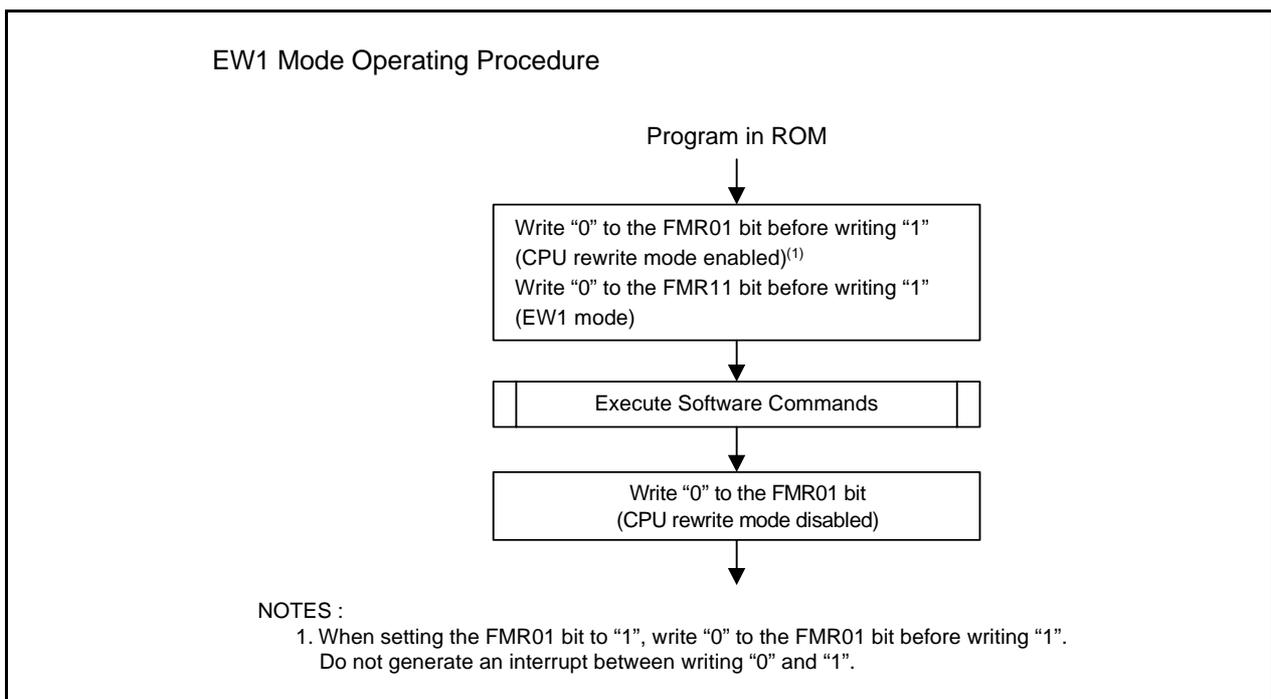


Figure 18.9 How to Set and Exit EW1 Mode

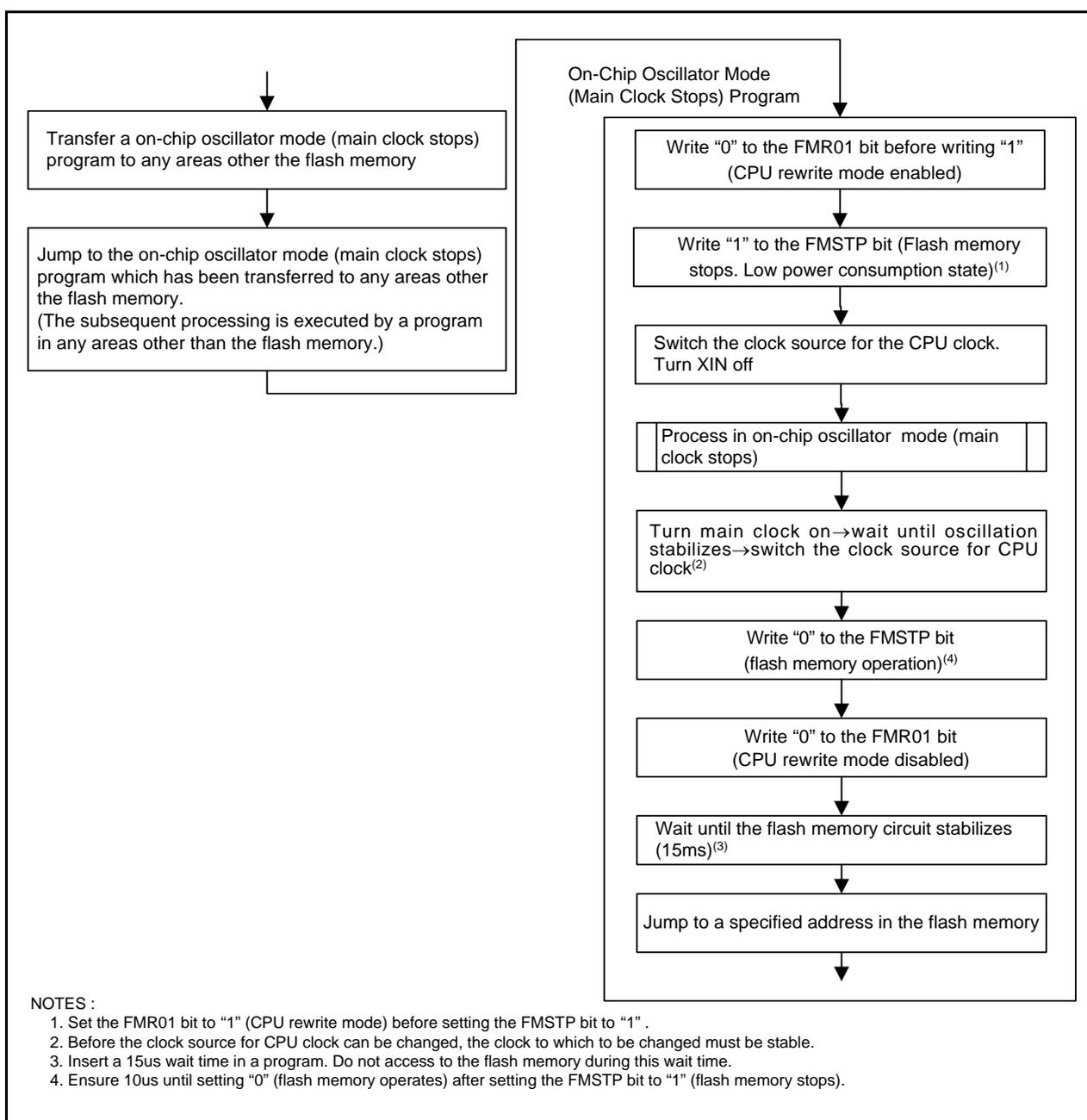


Figure 18.10 Process to Reduce Power Consumption in On-Chip Oscillator Mode (Main Clock Stops)

18.4.3 Software Commands

Software commands are described below. Read or write commands and data from or to in 8-bit units.

Table 18.4 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 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

18.4.3.1 Read Array Command

The read array command reads the flash memory.

The microcomputer enters read array mode by writing "FFh" in the first bus cycle. If entering the read address after the following bus cycles, the content of the specified address can be read in 8-bit units. Since the microcomputer remains in read array mode until another command is written, the contents of multiple addresses can be read continuously.

18.4.3.2 Read Status Register Command

The read status register command reads the status register.

If writing "70h" in the first bus cycle, the status register can be read in the second bus cycle. (Refer to **18.4.4 Status Register**) When reading the status register, specify an address in the user ROM area. Do not execute this command in EW1 mode.

18.4.3.3 Clear Status Register Command

The clear status register command sets the status register to "0".

If writing "50h" in the first bus cycle, the FMR06 to FMR07 bits in the FMR0 register and SR4 to SR5 in the status register will be set to "0".

18.4.3.4 Program Command

The program command writes data to the flash memory in 1-byte units.

Write "40h" in the first bus cycle and write data to the write address in the second bus cycle, and 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 determine whether auto programming has completed. The FMR00 bit is set to "0" during auto programming and set to "1" when auto programming completes.

The FMR06 bit in the FMR0 register can determine the result of auto programming after it has been finished. (Refer to **18.4.5 Full Status Check**)

Do not write additions to the already programmed address.

When the FMR02 bit in the FMR0 register is set to "0" (disable rewriting), or the FMR02 bit is set to "1" (rewrite enables) and the FMR15 bit in the FMR1 register is set to "1" (disable rewriting), the program command on Block 0 is not acknowledged. When the FMR16 bit is set to "1" (disable rewriting), the program command on Block 1 is not acknowledged.

In EW1 mode, do not execute this command on any address at which the rewrite control program is allocated.

In EW0 mode, the microcomputer 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 microcomputer remains in read status register mode until a read array command is written next. Reading the status register can determine the result of auto programming after auto programming has completed.

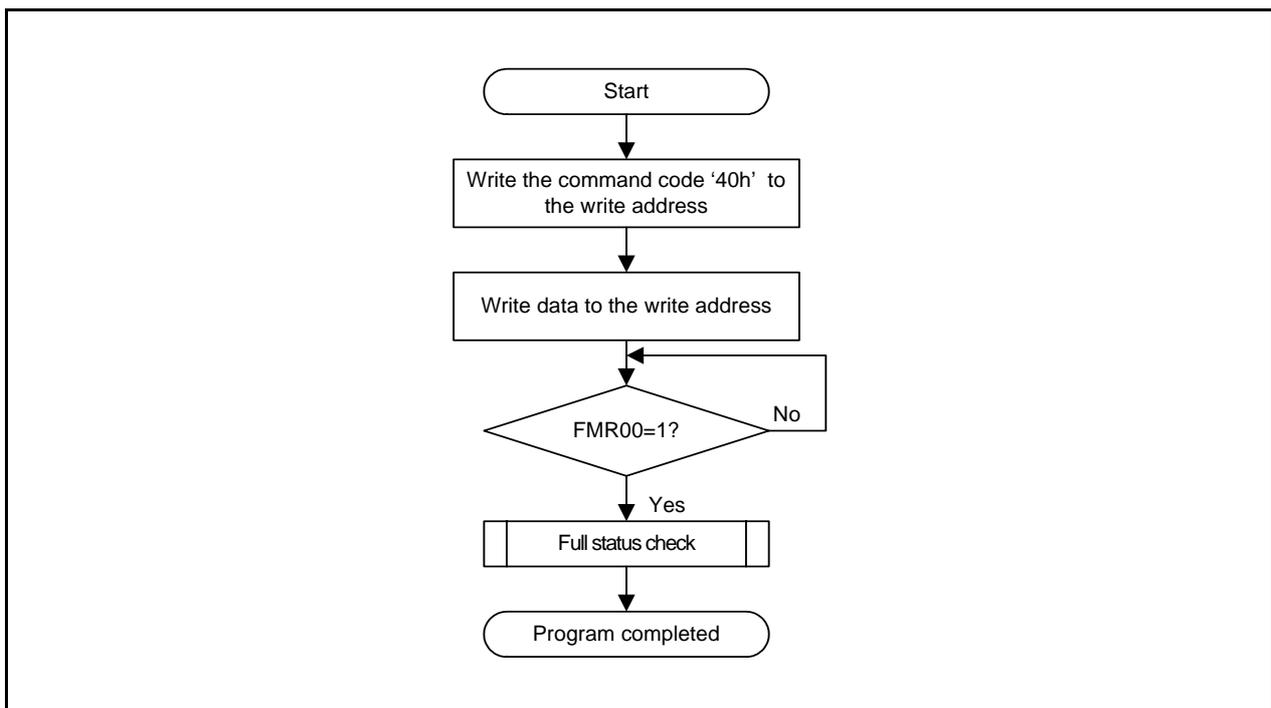


Figure 18.11 Program Command

18.4.3.5 Block Erase

If writing "20h" in the first bus cycle and "D0h" to the given address of a block in the second bus cycle, and an auto erase operation (erase and verify) will start.

The FMR00 bit in the FMR0 register can determine whether auto erasing has completed.

The FMR00 bit is set to "0" during auto erasing and set to "1" when auto erasing completes.

The FMR07 bit in the FMR0 register can determine the result of auto erasing after auto erasing has completed. (Refer to **18.4.5 Full Status Check**)

When the FMR02 bit in the FMR0 register is set to "0" (disable rewriting) or the FMR02 bit is set to "1" (rewrite enables) and the FMR15 bit in the FMR1 register is set to "1" (disable rewriting), the block erase command on Block 0 is not acknowledged. When the FMR16 bit is set to "1" (disable rewriting), the block erase command on Block 1 is not acknowledged.

Figure 18.12 shows the Block Erase Command (When Not Using Erase-Suspend Function). Figure 18.13 shows the Block Erase Command (When Using Erase-Suspend Function).

In EW1 mode, do not execute this command on any address at which the rewrite control program is allocated.

In EW0 mode, the microcomputer enters read status register mode at the same time auto erasing starts and the status register can be read. The status register bit 7 (SR7) is set to "0" at the same time auto erasing starts and set back to "1" when auto erasing completes. In this case, the microcomputer remains in read status register mode until the read array command is written next.

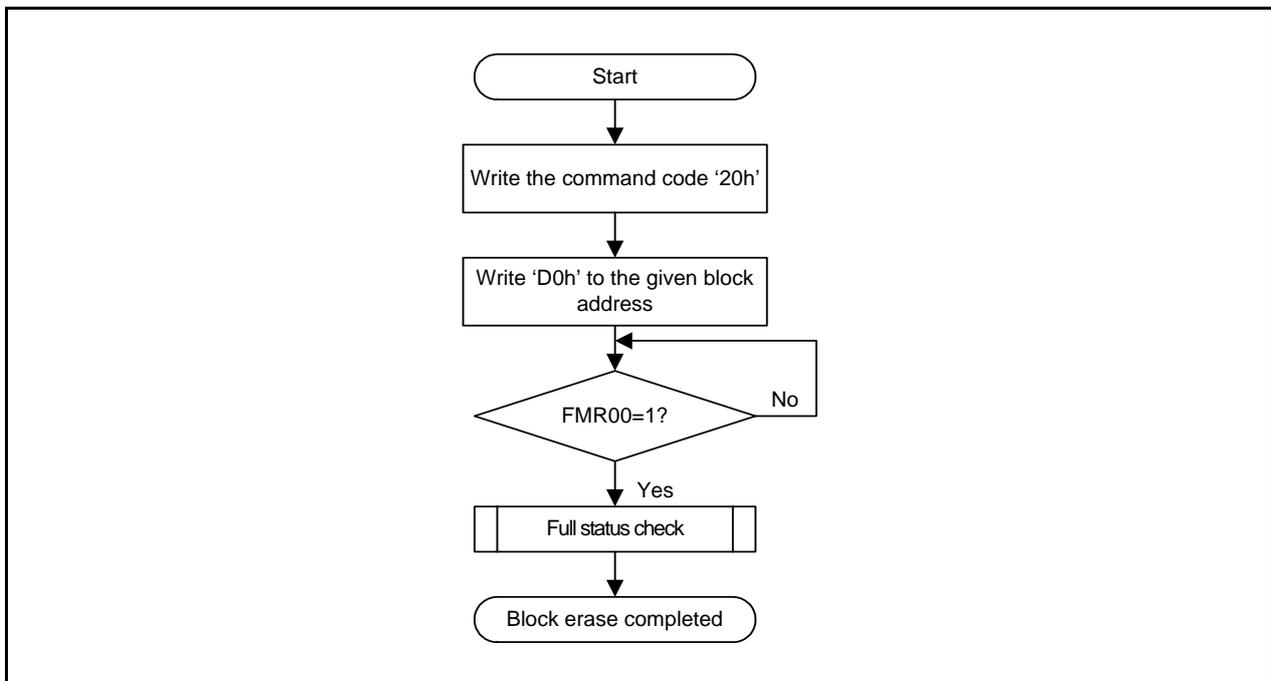


Figure 18.12 Block Erase Command (When Not Using Erase-Suspend Function)

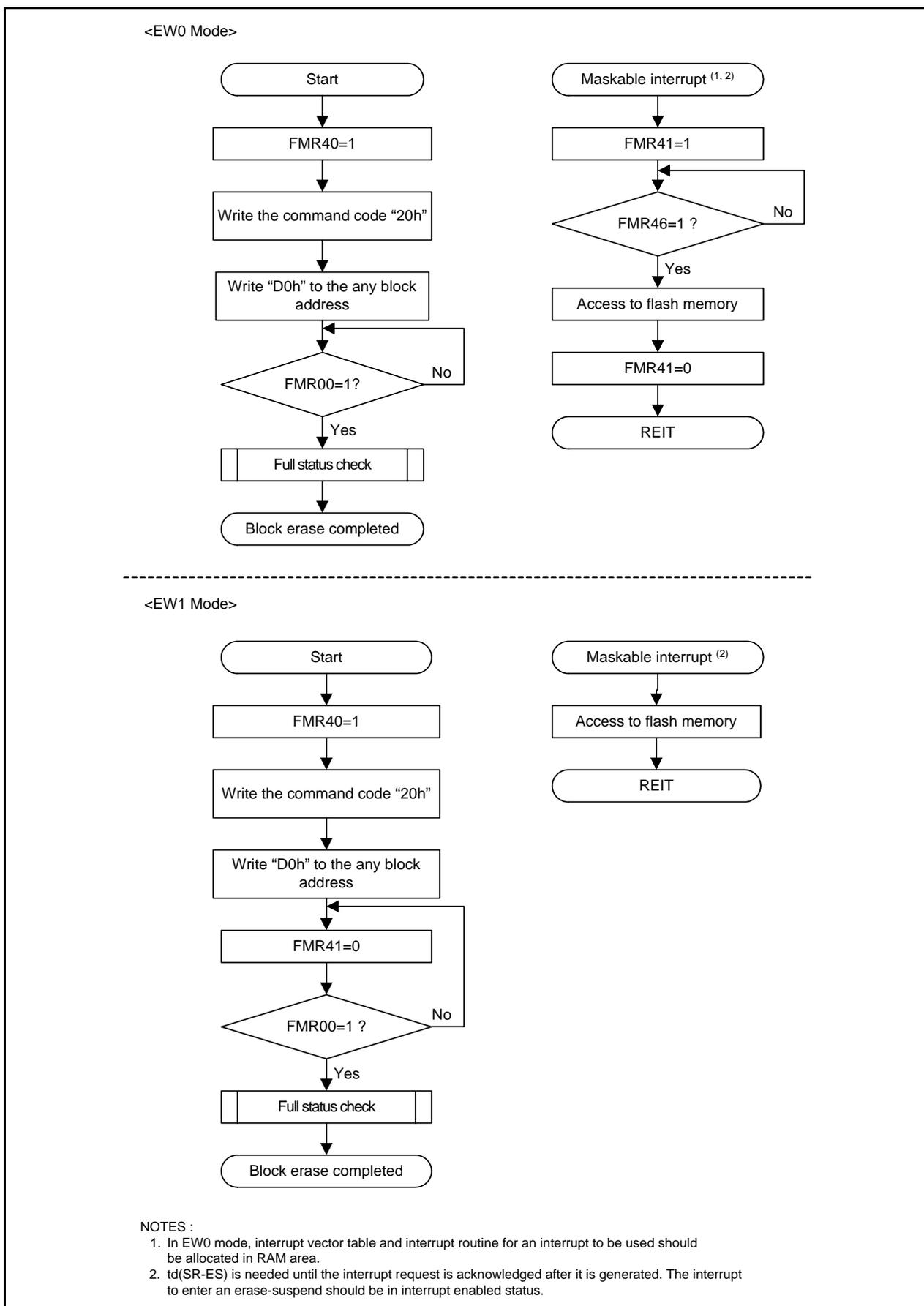


Figure 18.13 Block Erase Command (When Using Erase-Suspend Function)

18.4.4 Status Register

The status register indicates the operating status of the flash memory and whether an erasing or programming operation completes normally or in error. Status of the status register can be read by the FMR00, FMR06, and FMR07 bits in the FMR0 register.

Table 18.5 lists the Status Register.

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 the program or block erase command but before executing the read array command.

18.4.4.1 Sequencer Status (SR7 and FMR00 Bits)

The sequencer status indicates operating status of the flash memory. SR7 = 0 (busy) during auto programming and auto erasing, and is set to "1" (ready) at the same time the operation completes.

18.4.4.2 Erase Status (SR5 and FMR07 Bits)

Refer to 18.4.5 Full Status Check.

18.4.4.3 Program Status (SR4 and FMR06 Bits)

Refer to 18.4.5 Full Status Check.

Table 18.5 Status Register

| Status Register Bit | FMR0 Register Bit | Status Name | Contents | | 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 | 0 |

- D0 to D7: Indicates the data bus which is read when the read status register command is executed.
- The FMR07 (SR5) to FMR06 bits (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 command cannot be accepted.

18.4.5 Full Status Check

When an error occurs, the FMR06 to FMR07 bits in the FMR0 register are set to "1", indicating occurrence of each specific error. Therefore, Checking these status bits (full status check) can determine the executed result.

Table 18.6 lists the Errors and FMR0 Register Status. Figure 18.14 shows the Full Status Check and Handling Procedure for Each Error.

Table 18.6 Errors and FMR0 Register Status

| FRM00 Register (Status Register) Status | | Error | Error occurrence condition |
|---|------------|------------------------|---|
| FMR07(SR5) | FMR06(SR4) | | |
| 1 | 1 | Command Sequence Error | <ul style="list-style-type: none"> • When any command is not written correctly • When invalid data other than those that can be written in the second bus cycle of the block erase command is written (i.e., other than "D0h" or "FFh")(1) • When executing the program command or block erase command while rewriting is disabled using the FMR02 bit in the FMR0 register, the FMR15 or FMR16 bit in the FMR1 register. • When inputting and erasing the address in which the Flash memory is not allocated during the erase command input • When executing to erase the block which disables rewriting during the erase command input. • When inputting and writing the address in which the Flash memory is not allocated during the write command input. • When executing to write the block which disables rewriting during the write command input. |
| 1 | 0 | Erase Error | <ul style="list-style-type: none"> • When the block erase command is executed but not automatically erased correctly |
| 0 | 1 | Program Error | <ul style="list-style-type: none"> • When the program command is executed but not automatically programmed correctly. |

NOTES:

1. The microcomputer enters read array mode by writing "FFh" in the second bus cycle of these commands, at the same time the command code written in the first bus cycle will disabled.

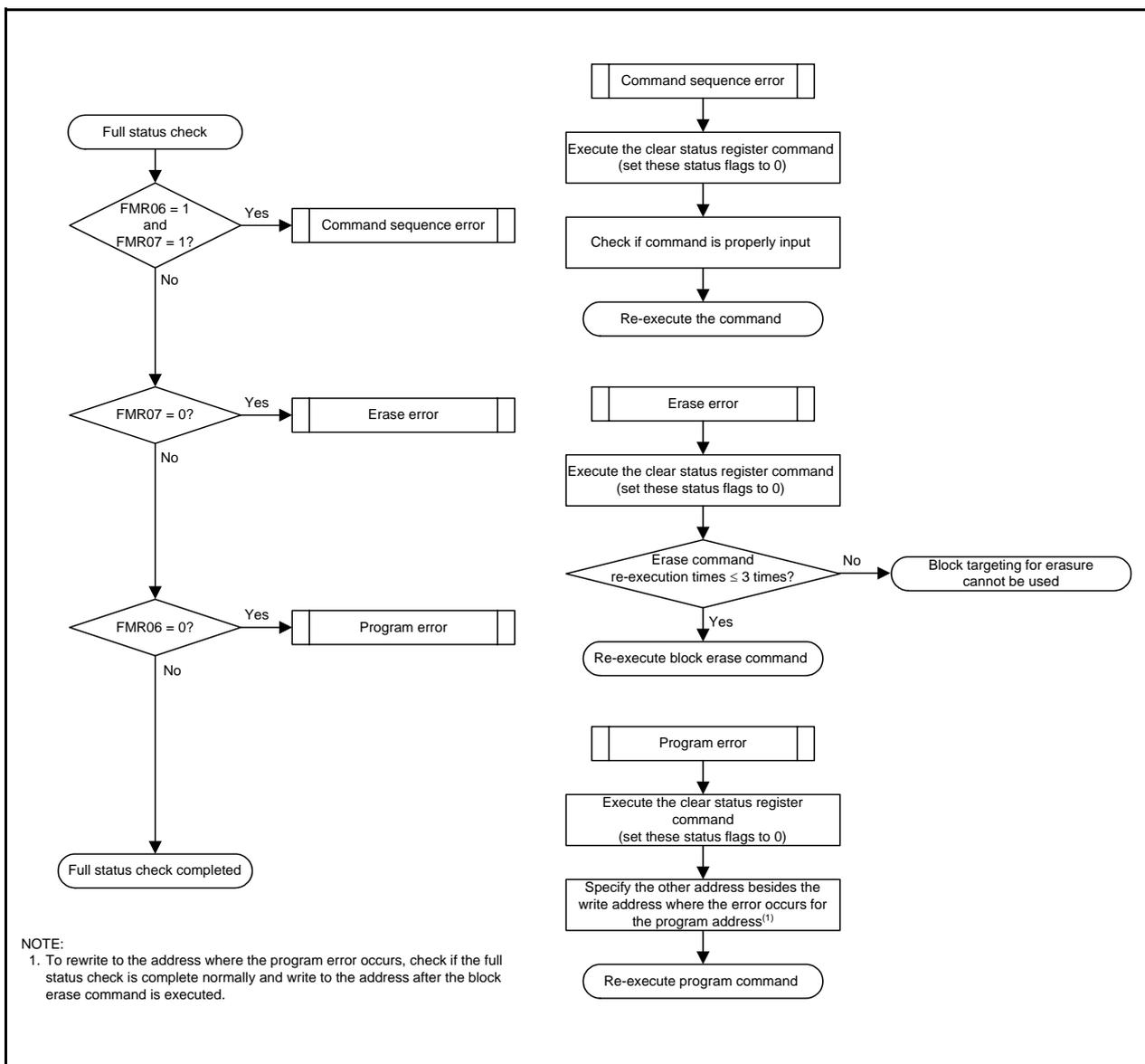


Figure 18.14 Full Status Check and Handling Procedure for Each Error

18.5 Standard Serial I/O Mode

In standard serial I/O mode, the user ROM area can be rewritten while the microcomputer is mounted on-board by using a serial programmer which is applicable for this microcomputer.

Standard serial I/O mode is used to connect with a serial writer using a special clock asynchronous serial I/O.

There are three types of Standard serial I/O modes:

- Standard serial I/O mode 1 Clock synchronous serial I/O used to connect with a serial programmer
- Standard serial I/O mode 2 Clock asynchronous serial I/O used to connect with a serial programmer
- Standard serial I/O mode 3 Special clock asynchronous serial I/O used to connect with a serial programmer

This microcomputer uses Standard serial I/O mode 2 and Standard serial I/O mode 3.

Refer to **Appendix 2. Connecting Example between Serial Writer and On-Chip Debugging Emulator**. Contact the manufacturer of your serial programmer for serial programmer. Refer to the user's manual of your serial programmer for details on how to use it.

Table 18.7 lists the Pin Functions (Flash Memory Standard Serial I/O Mode 2), Table 18.8 lists the Pin Functions (Flash Memory Standard Serial I/O Mode 3). Figure 18.15 show Pin Connections for Standard Serial I/O Mode 3.

After processing the pins shown in Table 18.8 and rewriting a flash memory using a writer, apply "H" to the MODE pin and reset a hardware if a program is operated on the flash memory in single-chip mode.

18.5.1 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 **18.3 Functions To Prevent Flash Memory from Rewriting**).

Table 18.7 Pin Functions (Flash Memory Standard Serial I/O Mode 2)

| Pin | Name | I/O | Description |
|--------------|---------------------------|-----|--|
| VCC,VSS | Power input | | Apply the voltage guaranteed for program and erase to VCC pin and 0V to VSS pin. |
| RESET | Reset input | I | Reset input pin. |
| P4_6/XIN | P4_6 input/clock input | I | Connect ceramic resonator or crystal oscillator between XIN and XOUT pins. |
| P4_7/XOUT | P4_7 input/clock output | I/O | |
| AVCC, AVSS | Analog power supply input | I | Connect AVSS to VSS and AVCC to VCC, respectively. |
| P1_0 to P1_7 | Input port P1 | I | Input "H" or "L" level signal or leave the pin open. |
| VREF | Reference voltage input | I | Reference voltage input pin to A/D converter. |
| P3_3 to P3_5 | Input port P3 | I | Input "H" or "L" level signal or leave the pin open. |
| MODE | MODE | I/O | Input "L". |
| P3_7 | TXD output | O | Serial data output pin. |
| P4_5 | RXD input | I | Serial data input pin. |

Table 18.8 Pin Functions (Flash Memory Standard Serial I/O Mode 3)

| Pin | Name | I/O | Description |
|-----------------------|---------------------------|-----|---|
| VCC,VSS | Power input | | Apply the voltage guaranteed for program and erase to VCC pin and 0V to VSS pin. |
| RESET | Reset input | I | Reset input pin. |
| P4_6/XIN | P4_6 input/clock input | I | Connect ceramic resonator or crystal oscillator between 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 | |
| AVCC, AVSS | Analog power supply input | I | Connect AVSS to VSS and AVCC to VCC, respectively. |
| VREF | Reference voltage input | I | Reference voltage input pin to A/D converter. |
| P1_0 to P1_7 | Input port P1 | I | Input "H" or "L" level signal or leave the pin open. |
| P3_3 to P3_5, P3_7 | Input port P3 | I | Input "H" or "L" level signal or leave the pin open. |
| P4_5 | Input port P4 | I | Input "H" or "L" level signal or leave the pin open. |
| MODE | MODE | I/O | Serial data I/O pin. Connect to the flash programmer. |

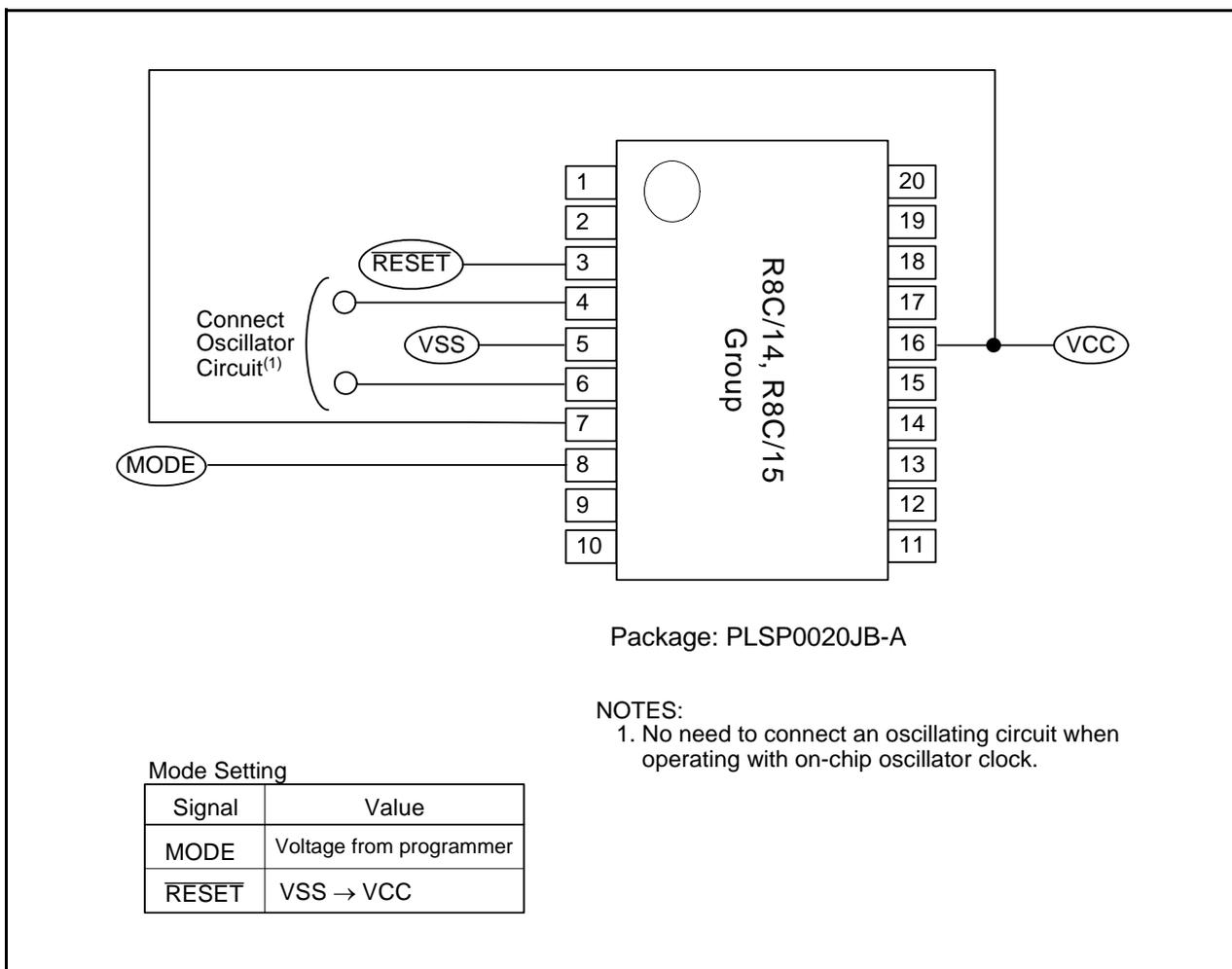


Figure 18.15 Pin Connections for Standard Serial I/O Mode 3

18.5.1.1 Example of Circuit Application in the Standard Serial I/O Mode

Figure 18.16 show Pin Process in Standard Serial I/O Mode 2, Figure 18.17 show Pin Process in Standard Serial I/O Mode 3. Since the controlled pins vary depending on the programmer, refer to the manual of your serial programmer.

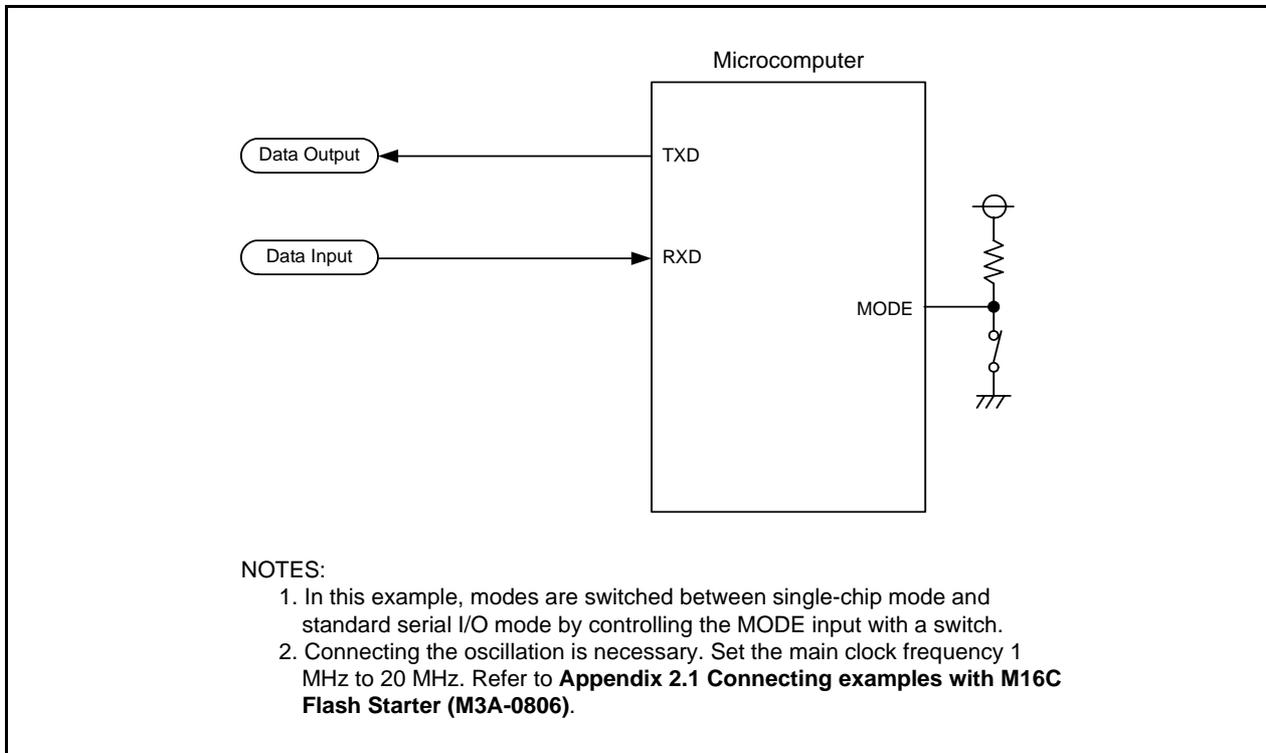


Figure 18.16 Pin Process in Standard Serial I/O Mode 2

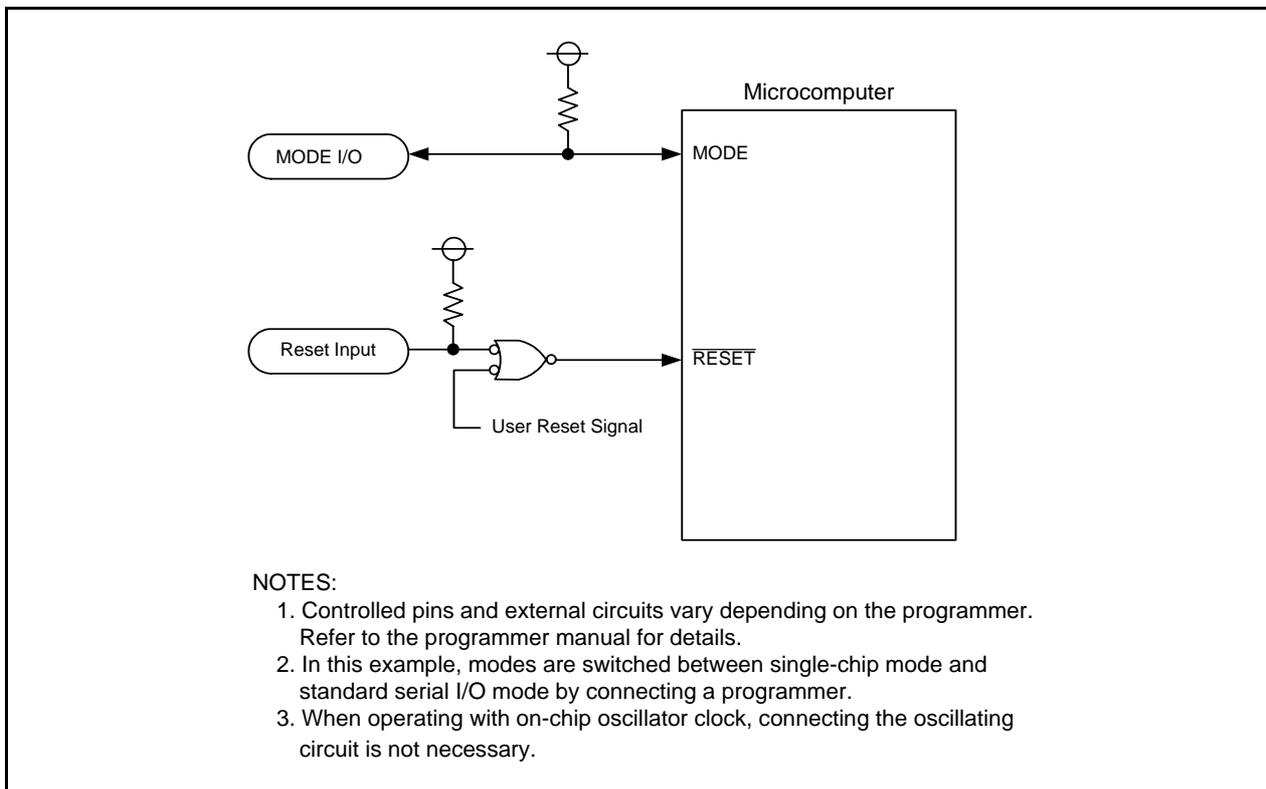


Figure 18.17 Pin Process in Standard Serial I/O Mode 3

18.6 Parallel I/O Mode

Parallel I/O mode is used to input and output the required software command, address and data parallel to controls (read, program and erase) for internal flash memory. Use a parallel programmer which supports this microcomputer. Contact the manufacturer of your parallel programmer about the parallel programmer and refer to the user's manual of your parallel programmer for details on how to use it.

User ROM area can be rewritten shown in Figures 18.1 and 18.2 in parallel I/O mode.

18.6.1 ROM Code Protect Function

The ROM code protect function disables to read and rewrite the flash memory. (Refer to the **18.3 Functions To Prevent Flash Memory from Rewriting.**)

19. Electrical Characteristics

Table 19.1 Absolute Maximum Ratings

| Symbol | Parameter | Condition | Rated value | Unit |
|------------------|-------------------------------|------------------------------------|-----------------------------------|------|
| V _{CC} | Supply Voltage | V _{CC} = AV _{CC} | -0.3 to 6.5 | V |
| AV _{CC} | Analog Supply Voltage | V _{CC} = AV _{CC} | -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°C | 300 | mW |
| T _{opr} | Operating Ambient Temperature | | -20 to 85 / -40 to 85 (D version) | °C |
| T _{stg} | Storage Temperature | | -65 to 150 | °C |

Table 19.2 Recommended Operating Conditions

| Symbol | Parameter | | Conditions | Standard | | | Unit |
|-----------------------|--|--|-------------------------------|--------------------|----------------------------------|--------------------|------|
| | | | | Min. | Typ. | Max. | |
| V _{CC} | Supply Voltage | | | 2.7 | – | 5.5 | V |
| AV _{CC} | Analog Supply Voltage | | | – | V _{CC} (³) | – | V |
| V _{SS} | Supply Voltage | | | – | 0 | – | V |
| AV _{SS} | Analog Supply Voltage | | | – | 0 | – | V |
| V _{IH} | Input “H” Voltage | | | 0.8V _{CC} | – | V _{CC} | V |
| V _{IL} | Input “L” Voltage | | | 0 | – | 0.2V _{CC} | V |
| I _{OH(sum)} | Peak Sum Output “H” Current | Sum of All Pins I _{OH} (peak) | | – | – | -60 | mA |
| I _{OH(peak)} | Peak Output “H” Current | | | – | – | -10 | mA |
| I _{OH(avg)} | Average Output “H” Current | | | – | – | -5 | mA |
| I _{OL(sum)} | Peak Sum Output “L” Currents | Sum of All Pins I _{OL} (peak) | | – | – | 60 | mA |
| I _{OL(peak)} | Peak Output “L” Currents | Except P1_0 to P1_3 | | – | – | 10 | mA |
| | | P1_0 to P1_3 | Drive Capacity HIGH | – | – | 30 | mA |
| | | | Drive Capacity LOW | – | – | 10 | mA |
| I _{OL(avg)} | Average Output “L” Current | Except P1_0 to P1_3 | | – | – | 5 | mA |
| | | P1_0 to P1_3 | Drive Capacity HIGH | – | – | 15 | mA |
| | | | Drive Capacity LOW | – | – | 5 | mA |
| f _(XIN) | Main Clock Input Oscillation Frequency | | 3.0V ≤ V _{CC} ≤ 5.5V | 0 | – | 20 | MHz |
| | | | 2.7V ≤ V _{CC} < 3.0V | 0 | – | 10 | MHz |

NOTES:

1. V_{CC} = AV_{CC} = 2.7 to 5.5V at T_{opr} = -20 to 85 °C / -40 to 85 °C, unless otherwise specified.
2. The typical values when average output current is 100ms.
3. Hold V_{CC} = AV_{CC}.

Table 19.3 A/D Converter Characteristics

| Symbol | Parameter | | Conditions | Standard | | | Unit |
|--------------|--|-----------------------|---|----------|----------------|-----------|---------------|
| | | | | Min. | Typ. | Max. | |
| – | Resolution | | $V_{ref} = V_{CC}$ | – | – | 10 | Bits |
| – | Absolute Accuracy | 10-Bit Mode | $\phi_{AD} = 10\text{MHz}$, $V_{ref} = V_{CC} = 5.0\text{V}$ | – | – | ± 3 | LSB |
| | | 8-Bit Mode | $\phi_{AD} = 10\text{MHz}$, $V_{ref} = V_{CC} = 5.0\text{V}$ | – | – | ± 2 | LSB |
| | | 10-Bit Mode | $\phi_{AD} = 10\text{MHz}$, $V_{ref} = V_{CC} = 3.3\text{V}^{(3)}$ | – | – | ± 5 | LSB |
| | | 8-Bit Mode | $\phi_{AD} = 10\text{MHz}$, $V_{ref} = V_{CC} = 3.3\text{V}^{(3)}$ | – | – | ± 2 | LSB |
| R_{ladder} | Resistor Ladder | | $V_{ref} = V_{CC}$ | 10 | – | 40 | $k\Omega$ |
| t_{conv} | Conversion Time | 10-Bit Mode | $\phi_{AD} = 10\text{MHz}$, $V_{ref} = V_{CC} = 5.0\text{V}$ | 3.3 | – | – | μs |
| | | 8-Bit Mode | $\phi_{AD} = 10\text{MHz}$, $V_{ref} = V_{CC} = 5.0\text{V}$ | 2.8 | – | – | μs |
| V_{ref} | Reference voltage | | | – | $V_{CC}^{(4)}$ | – | V |
| V_{IA} | Analog Input Voltage | | | 0 | – | V_{ref} | V |
| – | A/D Operating Clock Frequency ⁽²⁾ | Without Sample & Hold | | 0.25 | – | 10 | MHz |
| | | With Sample & Hold | | 1 | – | 10 | MHz |

NOTES:

1. $V_{CC} = AV_{CC} = 2.7$ to 5.5V at $T_{opr} = -20$ to $85\text{ }^\circ\text{C}$ / -40 to $85\text{ }^\circ\text{C}$, unless otherwise specified.
2. If f_1 exceeds 10MHz , divide the f_1 and hold A/D operating clock frequency (ϕ_{AD}) 10MHz or below.
3. If the AV_{CC} is less than 4.2V , divide the f_1 and hold A/D operating clock frequency (ϕ_{AD}) $f_1/2$ or below.
4. Hold $V_{CC} = V_{ref}$

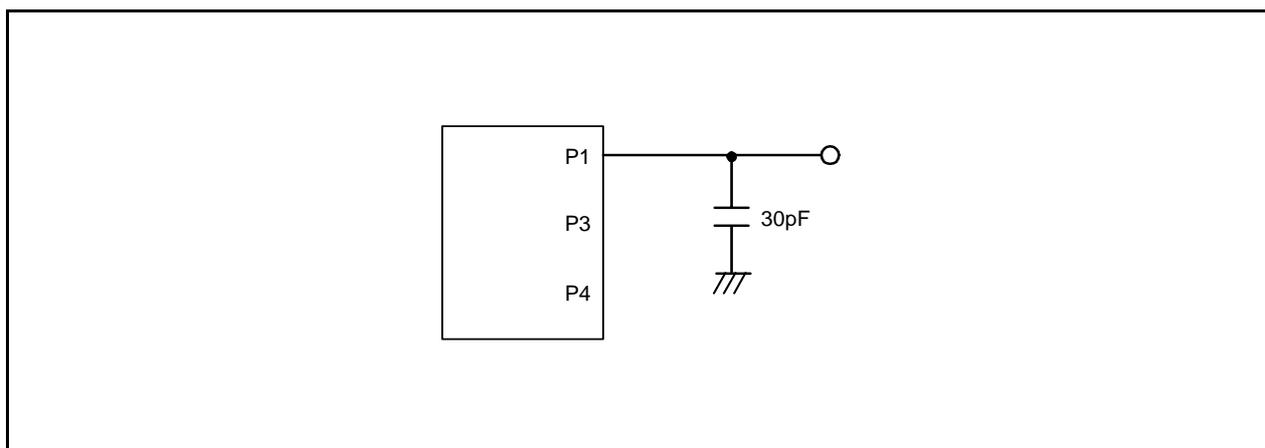
**Figure 19.1 Port P1, P3 and P4 Measurement Circuit**

Table 19.4 Flash Memory (Program ROM) Electrical Characteristics

| Symbol | Parameter | Conditions | Standard | | | Unit |
|------------------------|---|---|----------------------|------|------|-------|
| | | | Min. | Typ. | Max. | |
| – | Program/Erase Endurance ⁽²⁾ | R8C/14 Group | 100 ⁽³⁾ | – | – | times |
| | | R8C/15 Group | 1,000 ⁽³⁾ | – | – | times |
| – | Byte Program Time | V _{CC} = 5.0 V at T _{opr} = 25 °C | – | 50 | 400 | μs |
| – | Block Erase Time | V _{CC} = 5.0 V at T _{opr} = 25 °C | – | 0.4 | 9 | s |
| t _d (SR-ES) | Time Delay from Suspend Request until Erase Suspend | | – | – | 8 | ms |
| – | Erase Suspend Request Interval | | 10 | – | – | ms |
| – | Program, Erase Voltage | | 2.7 | – | 5.5 | V |
| – | Read Voltage | | 2.7 | – | 5.5 | V |
| – | Program, Erase Temperature | | 0 | – | 60 | °C |
| – | Data Hold Time ⁽⁷⁾ | Ambient temperature = 55 °C | 20 | – | – | year |

NOTES:

- V_{CC} = AV_{CC} = 2.7 to 5.5V at T_{opr} = 0 to 60 °C, unless otherwise specified.
- Definition of program and erase
The program and erase endurance shows an erase endurance for every block.
If the program and erase endurance is “n” times (n = 100, 10000), “n” times erase can be performed for every block.
For example, if performing 1-byte write to the distinct addresses on Block A of 1Kbyte block 1,024 times and then erasing that block, program and erase endurance is counted as one time.
However, do not perform multiple programs to the same address for one time erase.(disable overwriting).
- Endurance to guarantee all electrical characteristics after program and erase.(1 to “Min.” value can be guaranteed).
- In the case of a system to execute multiple programs, perform one erase after programming as reducing effective reprogram endurance not to leave blank area as possible such as programming write addresses in turn . If programming a set of 16 bytes, programming up to 128 sets and then erasing them one time can reduce effective reprogram endurance. Additionally, averaging erase endurance for Block A and B can reduce effective reprogram endurance more. To leave erase endurance for every block as information and determine the restricted endurance are recommended.
- If error occurs during block erase, attempt to execute the clear status register command, then 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.
- The data hold time includes time that the power supply is off or the clock is not supplied.

Table 19.5 Flash Memory (Data flash Block A, Block B) Electrical Characteristics

| 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) | V _{CC} = 5.0 V at T _{opr} = 25 °C | – | 50 | 400 | μs |
| – | Byte Program Time (Program/Erase Endurance > 1,000 Times) | V _{CC} = 5.0 V at T _{opr} = 25 °C | – | 65 | – | μs |
| – | Block Erase Time (Program/Erase Endurance ≤ 1,000 Times) | V _{CC} = 5.0 V at T _{opr} = 25 °C | – | 0.2 | 9 | s |
| – | Block Erase Time (Program/Erase Endurance > 1,000 Times) | V _{CC} = 5.0 V at T _{opr} = 25 °C | – | 0.3 | – | s |
| t _d (SR-ES) | Time Delay from Suspend Request until Erase Suspend | | – | – | 8 | ms |
| – | Erase Suspend Request Interval | | 10 | – | – | ms |
| – | Program, Erase Voltage | | 2.7 | – | 5.5 | V |
| – | Read Voltage | | 2.7 | – | 5.5 | V |
| – | Program, Erase Temperature | | -20 ⁽⁸⁾ | – | 85 | °C |
| – | Data Hold Time ⁽⁹⁾ | Ambient temperature = 55 °C | 20 | – | – | year |

NOTES:

- V_{CC} = AV_{CC} = 2.7 to 5.5V at T_{opr} = –20 to 85 °C / –40 to 85 °C, unless otherwise specified.
- Definition of program and erase
The program and erase endurance shows an erase endurance for every block.
If the program and erase endurance is “n” times (n = 100, 10000), “n” times erase can be performed for every block.
For example, if performing 1-byte write to the distinct addresses on Block A of 1Kbyte block 1,024 times and then erasing that block, program and erase endurance is counted as one time.
However, do not perform multiple programs to the same address for one time erase.(disable overwriting).
- 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 aer the same as that in program area.
- In the case of a system to execute multiple programs, perform one erase after programming as reducing effective reprogram endurance not to leave blank area as possible such as programming write addresses in turn . If programming a set of 16 bytes, programming up to 128 sets and then erasing them one time can reduce effective reprogram endurance. Additionally, averaging erase endurance for Block A and B can reduce effective reprogram endurance more. To leave erase endurance for every block as information and determine the restricted endurance are recommended.
- If error occurs during block erase, attempt to execute the clear status register command, then 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.

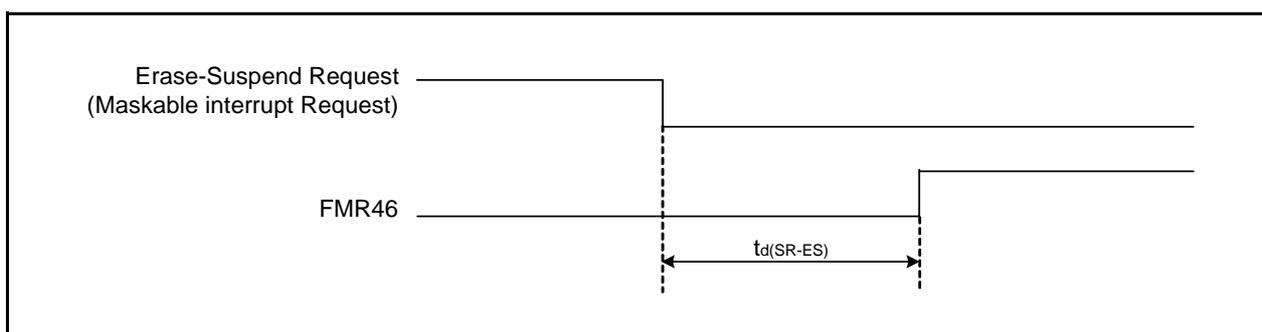


Figure 19.2 Time delay from Suspend Request until Erase Suspend

Table 19.6 Voltage Detection 1 Circuit Electrical Characteristics

| Symbol | Parameter | Condition | Standard | | | Unit |
|---------------------|--|-----------------------------------|----------|------|------|------|
| | | | Min. | Typ. | Max. | |
| V _{det1} | Voltage Detection Level ⁽³⁾ | | 2.70 | 2.85 | 3.00 | V |
| – | Voltage Detection Circuit Self Power Consumption | VCA26 = 1, V _{CC} = 5.0V | – | 600 | – | nA |
| t _{d(E-A)} | Waiting Time until Voltage Detection Circuit Operation Starts ⁽²⁾ | | – | – | 100 | μs |
| V _{ccmin} | Microcomputer Operating Voltage Minimum Value | | 2.7 | – | – | V |

NOTES:

1. The measurement condition is V_{CC} = AV_{CC} = 2.7V to 5.5V and T_{opr} = -40°C to 85 °C.
2. Necessary time until the voltage detection circuit operates when setting to “1” again after setting the VCA26 bit in the VCA2 register to “0”.
3. Hold V_{det2} > V_{det1}.

Table 19.7 Voltage Detection 2 Circuit Electrical Characteristics

| Symbol | Parameter | Condition | Standard | | | Unit |
|---------------------|--|-----------------------------------|----------|------|------|------|
| | | | Min. | Typ. | Max. | |
| V _{det2} | Voltage Detection Level ⁽⁴⁾ | | 3.00 | 3.30 | 3.60 | V |
| – | Voltage Monitor 2 Interrupt Request Generation Time ⁽²⁾ | | – | 40 | – | μs |
| – | Voltage Detection Circuit Self Power Consumption | VCA27 = 1, V _{CC} = 5.0V | – | 600 | – | nA |
| t _{d(E-A)} | Waiting Time until Voltage Detection Circuit Operation Starts ⁽³⁾ | | – | – | 100 | μs |

NOTES:

1. The measurement condition is V_{CC} = AV_{CC} = 2.7V to 5.5V and T_{opr} = -40°C to 85 °C.
2. Time until the voltage monitor 2 interrupt request is generated since the voltage passes V_{det1}.
3. Necessary time until the voltage detection circuit operates when setting to “1” again after setting the VCA27 bit in the VCA2 register to “0”.
4. Hold V_{det2} > V_{det1}.

Table 19.8 Reset Circuit Electrical Characteristics (When Using Voltage Monitor 1 Reset)

| Symbol | Parameter | Condition | Standard | | | Unit |
|--|---|---|----------|------|-------------------|------|
| | | | Min. | Typ. | Max. | |
| V _{por2} | Power-On Reset Valid Voltage | -20°C ≤ Topr < 85°C | - | - | V _{det1} | V |
| t _w (V _{por2} -V _{det1}) | Supply Voltage Rising Time When Power-On Reset is Deasserted ⁽¹⁾ | -20°C ≤ Topr < 85°C, t _w (por2) ≥ 0s ⁽³⁾ | - | - | 100 | ms |

NOTES:

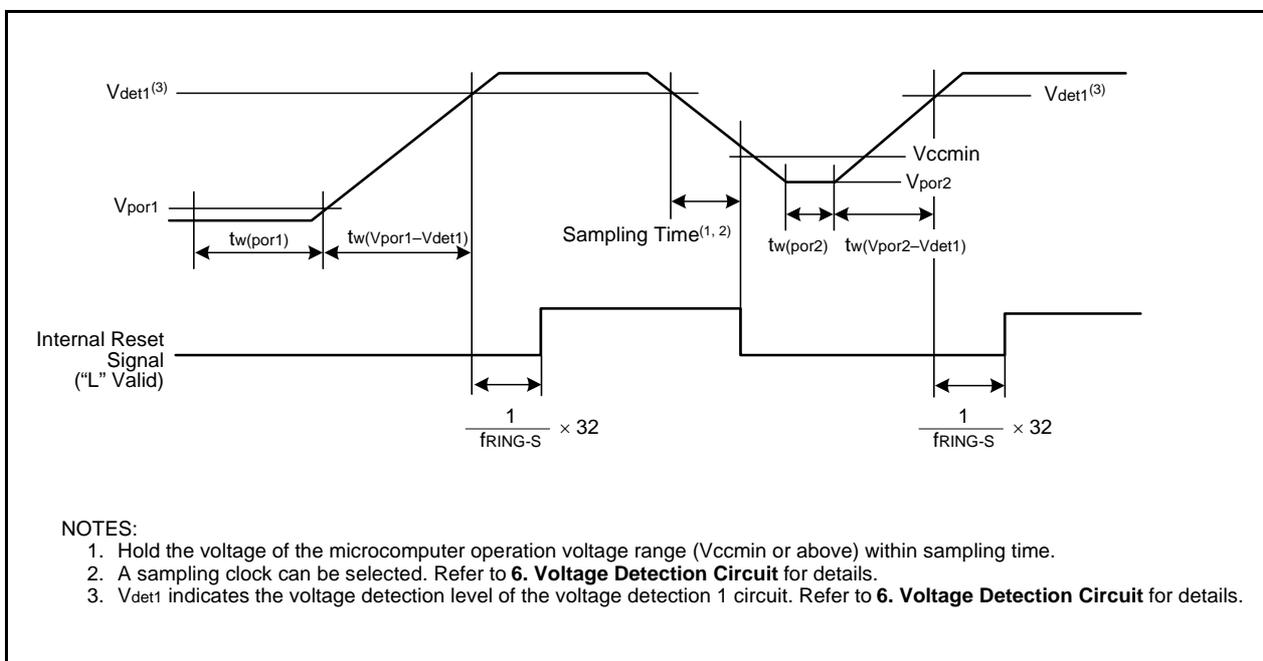
1. This condition is not applicable when using with V_{cc} ≥ 1.0V.
2. When turning power on after the time to hold the external power below effective voltage (V_{por1}) exceeds 10s, refer to **Table 19.9 Reset Circuit Electrical Characteristics (When Not Using Voltage Monitor 1 Reset)**.
3. t_w(por2) is time to hold the external power below effective voltage (V_{por2}).

Table 19.9 Reset Circuit Electrical Characteristics (When Not Using Voltage Monitor 1 Reset)

| Symbol | Parameter | Condition | Standard | | | Unit |
|--|--|---|----------|------|------|------|
| | | | Min. | Typ. | Max. | |
| V _{por1} | Power-On Reset Valid Voltage | -20°C ≤ Topr < 85°C | - | - | 0.1 | V |
| t _w (V _{por1} -V _{det1}) | Supply Voltage Rising Time When Power-On Reset is Deasserted | 0°C ≤ Topr ≤ 85°C, t _w (por1) ≥ 10s ⁽²⁾ | - | - | 100 | ms |
| t _w (V _{por1} -V _{det1}) | Supply Voltage Rising Time When Power-On Reset is Deasserted | -20°C ≤ Topr < 0°C, t _w (por1) ≥ 30s ⁽²⁾ | - | - | 100 | ms |
| t _w (V _{por1} -V _{det1}) | Supply Voltage Rising Time When Power-On Reset is Deasserted | -20°C ≤ Topr < 0°C, t _w (por1) ≥ 10s ⁽²⁾ | - | - | 1 | ms |
| t _w (V _{por1} -V _{det1}) | Supply Voltage Rising Time When Power-On Reset is Deasserted | 0°C ≤ Topr ≤ 85°C, t _w (por1) ≥ 1s ⁽²⁾ | - | - | 0.5 | ms |

NOTES:

1. When not using the voltage monitor 1 reset, use with V_{cc} ≥ 2.7V.
2. t_w(por1) is time to hold the external power below effective voltage (V_{por1}).



NOTES:

1. Hold the voltage of the microcomputer operation voltage range (V_{ccmin} or above) within sampling time.
2. A sampling clock can be selected. Refer to **6. Voltage Detection Circuit** for details.
3. V_{det1} indicates the voltage detection level of the voltage detection 1 circuit. Refer to **6. Voltage Detection Circuit** for details.

Figure 19.3 Reset Circuit Electrical Characteristics

Table 19.10 High-speed On-Chip Oscillator Circuit Electrical Characteristics

| Symbol | Parameter | Condition | Standard | | | Unit |
|--------|---|---|----------|------|------|------|
| | | | Min. | Typ. | Max. | |
| – | High-Speed On-Chip Oscillator Frequency When the Reset is Deasserted | $V_{CC} = 5.0V, T_{opr} = 25\text{ }^{\circ}\text{C}$ | – | 8 | – | MHz |
| – | High-Speed On-Chip Oscillator Frequency Temperature • Supply Voltage Dependence | 0 to +60 °C / 5 V \pm 5 % ⁽²⁾ | 7.44 | – | 8.56 | MHz |
| | | –20 to +85 °C / 2.7 to 5.5 V ⁽²⁾ | 7.04 | – | 8.96 | MHz |
| | | –40 to +85 °C / 2.7 to 5.5 V ⁽²⁾ | 6.80 | – | 9.20 | MHz |

NOTES:

1. The measurement condition is $V_{CC} = AV_{CC} = 5.0V$ and $T_{opr} = 25\text{ }^{\circ}\text{C}$.
2. The standard value shows when the HRA1 register is assumed as the value in shipping and the HRA2 register value is set to 00h.

Table 19.11 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:

1. The measurement condition is $V_{CC} = AV_{CC} = 2.7$ to $5.5V$ and $T_{opr} = 25\text{ }^{\circ}\text{C}$.
2. Waiting time until the internal power supply generation circuit stabilizes during power-on.
3. Time until CPU clock supply starts since the interrupt is acknowledged to exit stop mode.

Table 19.12 Timing Requirements of Clock Synchronous Serial I/O (SSU) with Chip Select⁽¹⁾

| Symbol | Parameter | | Conditions | Standard | | | Unit |
|-------------|---------------------------------|--------|------------|---------------|------|------------------|-----------------|
| | | | | Min. | Typ. | Max. | |
| t_{SUCYC} | SSCK Clock Cycle Time | | | 4 | – | – | $t_{CYC}^{(2)}$ |
| t_{HI} | SSCK Clock "H" Width | | | 0.4 | – | 0.6 | t_{SUCYC} |
| t_{LO} | SSCK Clock "L" Width | | | 0.4 | – | 0.6 | t_{SUCYC} |
| t_{RISE} | SSCK Clock Rising Time | Master | | – | – | 1 | $t_{CYC}^{(2)}$ |
| | | Slave | | – | – | 1 | μs |
| t_{FALL} | SSCK Clock Falling Time | Master | | – | – | 1 | $t_{CYC}^{(2)}$ |
| | | Slave | | – | – | 1 | μs |
| t_{SU} | SSO, SSI Data Input Setup Time | | | 100 | – | – | ns |
| t_{H} | SSO, SSI Data Input Hold Time | | | 1 | – | – | $t_{CYC}^{(2)}$ |
| t_{LEAD} | \overline{SCS} Setup Time | Slave | | $1t_{CYC}+50$ | – | – | ns |
| t_{LAG} | \overline{SCS} Hold Time | Slave | | $1t_{CYC}+50$ | – | – | ns |
| t_{OD} | SSO, SSI Data Output Delay Time | | | – | – | 1 | $t_{CYC}^{(2)}$ |
| t_{SA} | SSI Slave Access Time | | | – | – | $1.5t_{CYC}+100$ | ns |
| t_{OR} | SSI Slave Out Open Time | | | – | – | $1.5t_{CYC}+100$ | ns |

NOTES:

1. $V_{CC} = AV_{CC} = 2.7$ to $5.5V$, $V_{SS} = 0V$ at $T_{opr} = -20$ to $85\text{ }^{\circ}\text{C}$ / -40 to $85\text{ }^{\circ}\text{C}$, unless otherwise specified.
2. $1t_{CYC} = 1/f_1(\text{s})$

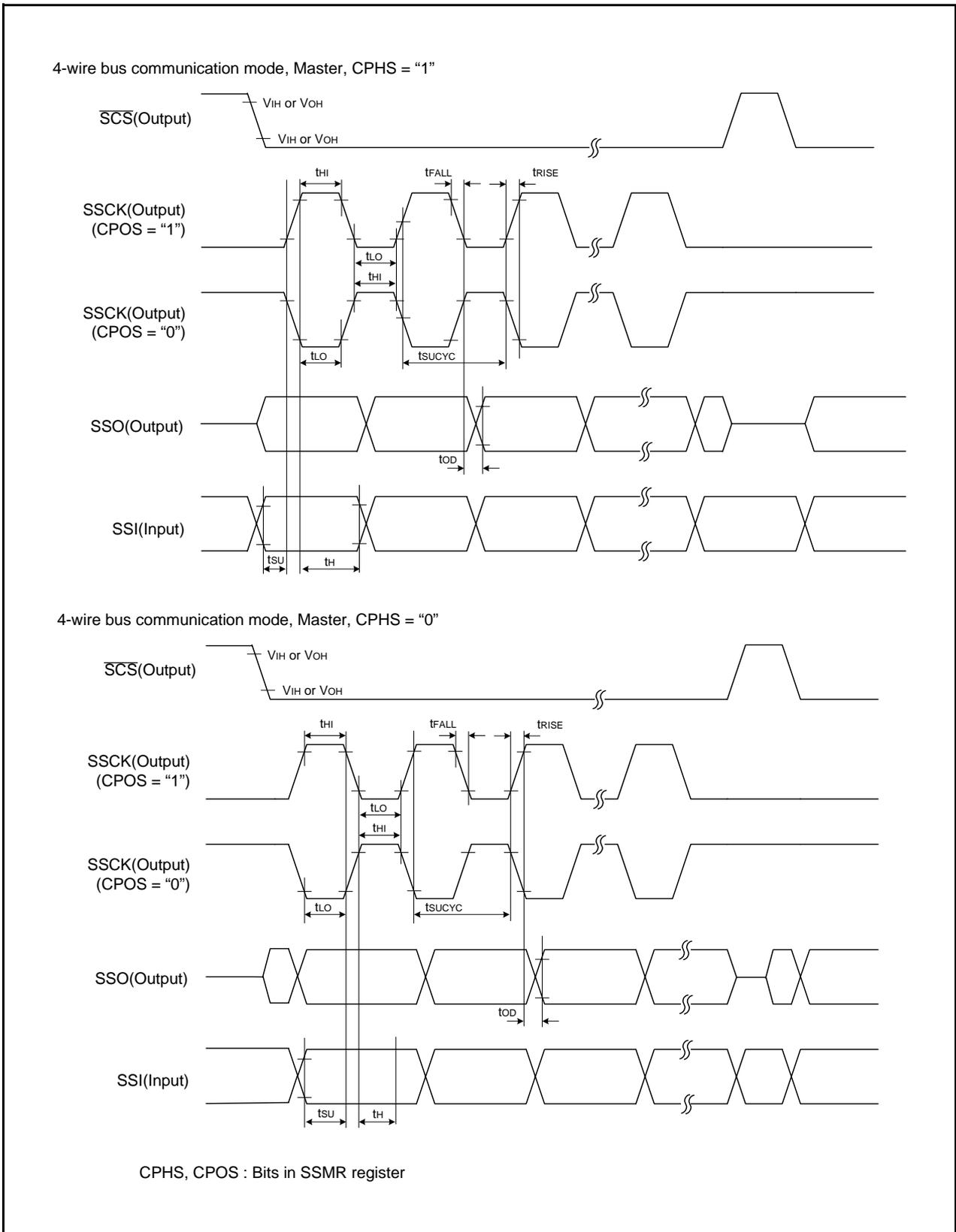


Figure 19.4 I/O Timing of Clock Synchronous Serial I/O (SSU) with Chip Select (Master)

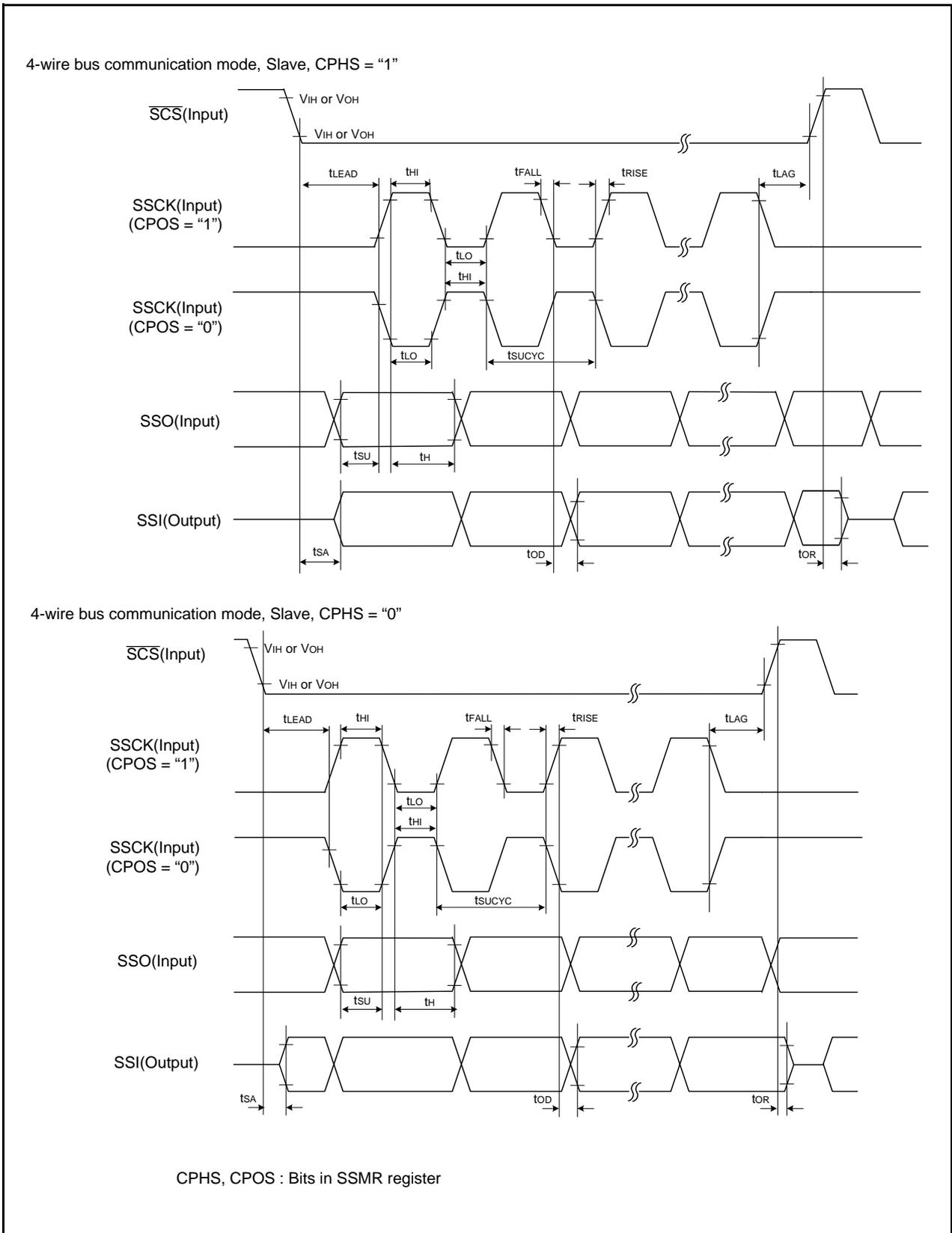


Figure 19.5 I/O Timing of Clock Synchronous Serial I/O (SSU) with Chip Select (Slave)

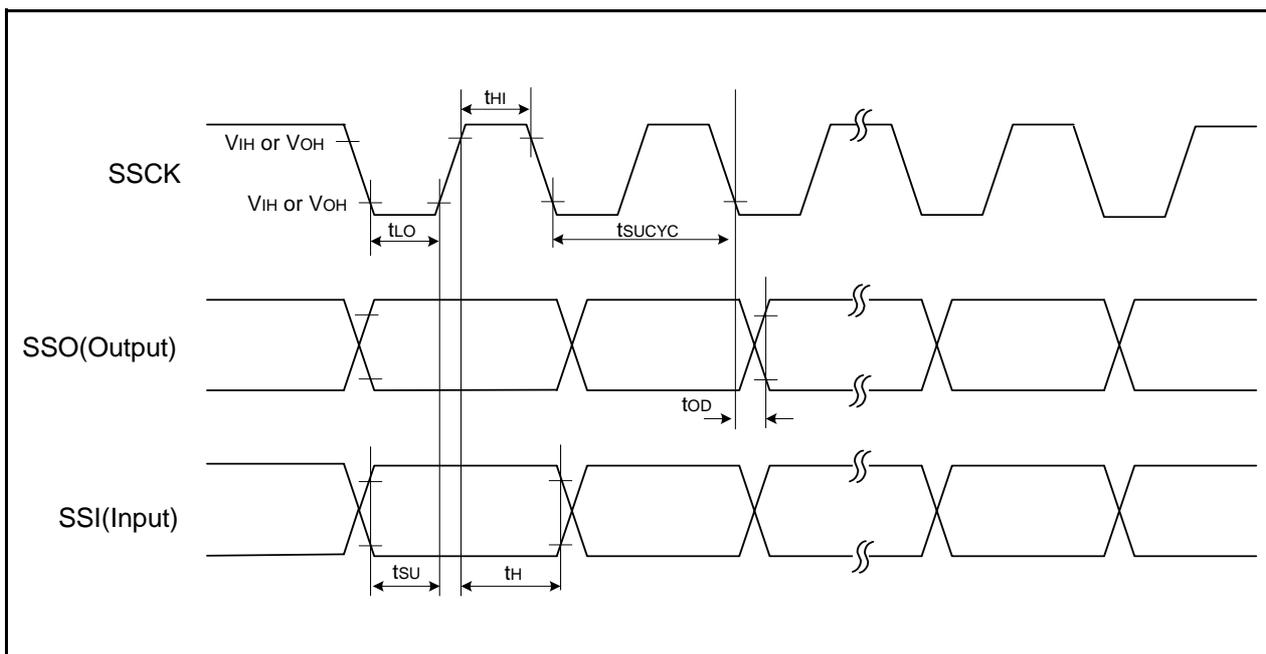


Figure 19.6 I/O Timing of Clock Synchronous Serial I/O (SSU) with Chip Select (Clock Synchronous Communication Mode)

Table 19.13 Electrical Characteristics (1) [Vcc = 5V]

| Symbol | Parameter | | Condition | | Standard | | | Unit |
|------------------|--|---------------------------|---------------------|---|-----------|------|------|------|
| | | | | | Min. | Typ. | Max. | |
| VOH | Output "H" Voltage | Except XOUT | IOH = -5mA | | Vcc - 2.0 | - | Vcc | V |
| | | | IOH = -200μA | | Vcc - 0.3 | - | Vcc | V |
| | | XOUT | Drive capacity HIGH | IOH = -1mA | Vcc - 2.0 | - | Vcc | V |
| | | | Drive capacity LOW | IOH = -500μA | Vcc - 2.0 | - | Vcc | V |
| VOL | Output "L" Voltage | Except P1_0 to P1_3, XOUT | IOL = 5mA | | - | - | 2.0 | V |
| | | | IOL = 200μA | | - | - | 0.45 | V |
| | | P1_0 to P1_3 | Drive capacity HIGH | IOL = 15mA | - | - | 2.0 | V |
| | | | Drive capacity LOW | IOL = 5mA | - | - | 2.0 | V |
| | | | Drive capacity LOW | IOL = 200μA | - | - | 0.45 | V |
| | | XOUT | Drive capacity HIGH | IOL = 1mA | - | - | 2.0 | V |
| | | | Drive capacity LOW | IOL = 500μA | - | - | 2.0 | V |
| | | VT+-VT- | Hysteresis | INT0, INT1, INT3, KI0, KI1, KI2, KI3, CNTR0, CNTR1, TCIN, RXD0, SSO | | | 0.2 | - |
| RESET | | | | 0.2 | - | 2.2 | V | |
| IiH | Input "H" current | | VI = 5V | | - | - | 5.0 | μA |
| IiL | Input "L" current | | VI = 0V | | - | - | -5.0 | μA |
| RPULLUP | Pull-Up Resistance | | VI = 0V | | 30 | 50 | 167 | kΩ |
| RfXIN | Feedback Resistance | XIN | | | - | 1.0 | - | MΩ |
| fRING-S | Low-Speed On-Chip Oscillator Frequency | | | | 40 | 125 | 250 | kHz |
| V _{RAM} | RAM Hold Voltage | | During stop mode | | 2.0 | - | - | V |

NOTES:

1. Vcc = AVcc = 4.2 to 5.5V at Topr = -20 to 85 °C / -40 to 85 °C, f(XIN)=20MHz, unless otherwise specified.

Table 19.14 Electrical Characteristics (2) [Vcc = 5V] (Topr = -40 to 85 °C, unless otherwise specified.)

| Symbol | Parameter | Condition | Standard | | | Unit | |
|--------|--|------------------------------------|--|------|------|------|----|
| | | | Min. | Typ. | Max. | | |
| Icc | Power Supply Current (Vcc=3.3 to 5.5V) In single-chip mode, the output pins are open and other pins are Vss | High-Speed Mode | XIN = 20MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on=125kHz No division | – | 9 | 15 | mA |
| | | | XIN = 16MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on=125kHz No division | – | 8 | 14 | mA |
| | | | XIN = 10MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on=125kHz No division | – | 5 | – | mA |
| | | Medium-Speed Mode | XIN = 20MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on=125kHz Divide-by-8 | – | 4 | – | mA |
| | | | XIN = 16MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on=125kHz Divide-by-8 | – | 3 | – | mA |
| | | | XIN = 10MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on=125kHz Divide-by-8 | – | 2 | – | mA |
| | | High-Speed On-Chip Oscillator Mode | Main clock off High-speed on-chip oscillator on=8MHz Low-speed on-chip oscillator on=125kHz No division | – | 4 | 8 | mA |
| | | | Main clock off High-speed on-chip oscillator on=8MHz Low-speed on-chip oscillator on=125kHz Divide-by-8 | – | 1.5 | – | mA |
| | | Low-Speed On-Chip Oscillator Mode | Main clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on=125kHz Divide-by-8 | – | 470 | 900 | μA |
| | | Wait Mode | Main clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on=125kHz While a WAIT instruction is executed Peripheral clock operation VCA26 = VCA27 = 0 | – | 40 | 80 | μA |
| | | Wait Mode | Main clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on=125kHz While a WAIT instruction is executed Peripheral clock off VCA26 = VCA27 = 0 | – | 38 | 76 | μA |
| | | Stop Mode | Main clock off, Topr = 25 °C High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA26 = VCA27 = 0 | – | 0.8 | 3.0 | μA |

Timing Requirements (Unless otherwise specified: $V_{CC} = 5V$, $V_{SS} = 0V$ at $T_{op} = 25\text{ }^{\circ}\text{C}$) [$V_{CC} = 5V$]**Table 19.15 XIN 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 |

Table 19.16 CNTR0 Input, CNTR1 Input, $\overline{INT1}$ Input

| Symbol | Parameter | Standard | | Unit |
|-----------------|------------------------|----------|------|------|
| | | Min. | Max. | |
| $t_{c(CNTR0)}$ | CNTR0 Input Cycle Time | 100 | – | ns |
| $t_{WH(CNTR0)}$ | CNTR0 Input "H" Width | 40 | – | ns |
| $t_{WL(CNTR0)}$ | CNTR0 input "L" Width | 40 | – | ns |

Table 19.17 TCIN Input, $\overline{INT3}$ Input

| Symbol | Parameter | Standard | | Unit |
|----------------|-----------------------|--------------------|------|------|
| | | Min. | Max. | |
| $t_{c(TCIN)}$ | TCIN Input Cycle Time | 400 ⁽¹⁾ | – | ns |
| $t_{WH(TCIN)}$ | TCIN Input "H" Width | 200 ⁽²⁾ | – | ns |
| $t_{WL(TCIN)}$ | TCIN input "L" Width | 200 ⁽²⁾ | – | ns |

NOTES:

1. When using Timer C input capture mode, adjust the cycle time ($1/\text{Timer C count source frequency} \times 3$) or above.
2. When using Timer C input capture mode, adjust the width ($1/\text{Timer C count source frequency} \times 1.5$) or above.

Table 19.18 Serial Interface

| Symbol | Parameter | Standard | | Unit |
|---------------|------------------------|----------|------|------|
| | | Min. | Max. | |
| $t_{c(CLK)}$ | CLKi Input Cycle Time | 200 | – | ns |
| $t_{W(CLKH)}$ | CLKi Input "H" Width | 100 | – | ns |
| $t_{W(CLKL)}$ | 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 |
| $t_{h(C-D)}$ | RCDi Input Hold Time | 90 | – | ns |

Table 19.19 External Interrupt $\overline{INT0}$ 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 $\overline{INT0}$ input filter select bit, use the $\overline{INT0}$ input HIGH width to the greater value, either ($1/\text{digital filter clock frequency} \times 3$) or the minimum value of standard.
2. When selecting the digital filter by the $\overline{INT0}$ input filter select bit, use the $\overline{INT0}$ input LOW width to the greater value, either ($1/\text{digital filter clock frequency} \times 3$) or the minimum value of standard.

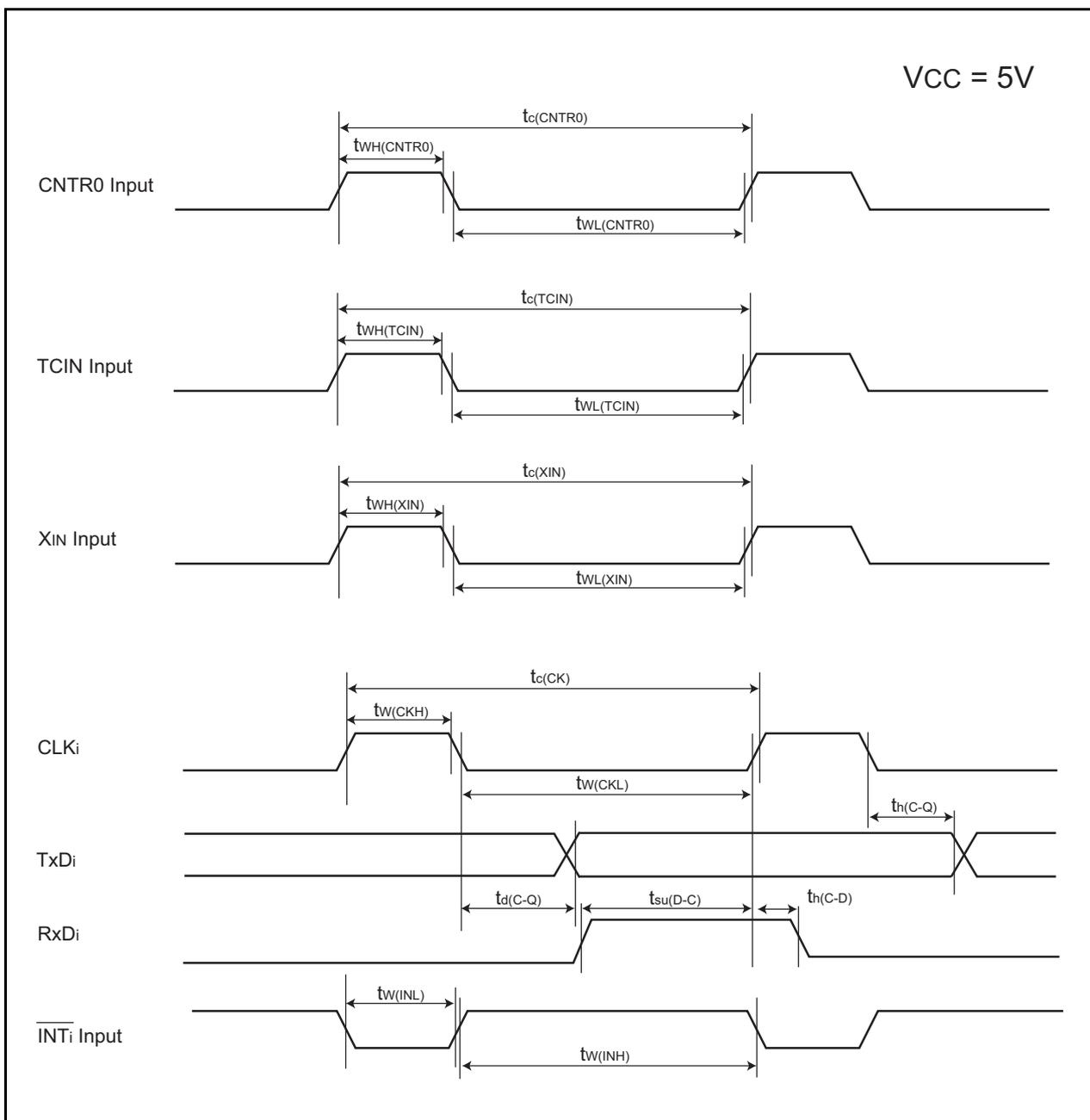


Figure 19.7 Timing Diagram When $V_{CC} = 5V$

Table 19.20 Electrical Characteristics (3) [Vcc = 3V]

| Symbol | Parameter | | Condition | | Standard | | | Unit |
|----------------------|--|--|---------------------|--------------|-----------|------|------|------|
| | | | | | Min. | Typ. | Max. | |
| VOH | Output "H" Voltage | Except XOUT | IOH = -1mA | | Vcc - 0.5 | - | Vcc | V |
| | | XOUT | Drive capacity HIGH | IOH = -0.1mA | Vcc - 0.5 | - | Vcc | V |
| | | | Drive capacity LOW | IOH = -50μA | Vcc - 0.5 | - | Vcc | V |
| VOL | Output "L" Voltage | Except P1_0 to P1_3, XOUT | IOL = 1mA | | - | - | 0.5 | V |
| | | P1_0 to P1_3 | Drive capacity HIGH | IOL = 2mA | - | - | 0.5 | V |
| | | | Drive capacity LOW | IOL = 1mA | - | - | 0.5 | V |
| | | XOUT | Drive capacity HIGH | IOL = 0.1mA | - | - | 0.5 | V |
| | | | Drive capacity LOW | IOL = 50μA | - | - | 0.5 | V |
| VT+-VT- | Hysteresis | $\overline{\text{INT0}}, \overline{\text{INT1}}, \overline{\text{INT3}},$ K10, K11, K12, K13, CNTR0, CNTR1, TCIN, RXD0, SSO | | | 0.2 | - | 0.8 | V |
| | | $\overline{\text{RESET}}$ | | | 0.2 | - | 1.8 | V |
| I _{IH} | Input "H" Current | | VI = 3V | | - | - | 4.0 | μA |
| I _{IL} | Input "L" Current | | VI = 0V | | - | - | -4.0 | μA |
| RPULLUP | Pull-Up Resistance | | VI = 0V | | 66 | 160 | 500 | kΩ |
| R _{IXIN} | Feedback Resistance | XIN | | | - | 3.0 | - | MΩ |
| f _{FRING-S} | Low-Speed On-Chip Oscillator Frequency | | | | 40 | 125 | 250 | kHz |
| V _{RAM} | RAM Hold Voltage | | During stop mode | | 2.0 | - | - | V |

NOTES:

- Vcc = AVcc = 2.7 to 3.3V at Topr = -20 to 85 °C / -40 to 85 °C, f(XIN)=10MHz, unless otherwise specified.

Table 19.21 Electrical Characteristics (4) [V_{CC} = 3V] (Topr = -40 to 85 °C, unless otherwise specified.)

| Symbol | Parameter | Condition | Standard | | | Unit | |
|-----------------|---|------------------------------------|--|------|------|------|----|
| | | | Min. | Typ. | Max. | | |
| I _{CC} | Power Supply Current (V _{CC} =2.7 to 3.3V) In single-chip mode, the output pins are open and other pins are V _{SS} | High-Speed Mode | XIN = 20MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on=125kHz No division | – | 8 | 13 | mA |
| | | | XIN = 16MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on=125kHz No division | – | 7 | 12 | mA |
| | | | XIN = 10MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on=125kHz No division | – | 5 | – | mA |
| | | Medium-Speed Mode | XIN = 20MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on=125kHz Divide-by-8 | – | 3 | – | mA |
| | | | XIN = 16MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on=125kHz Divide-by-8 | – | 2.5 | – | mA |
| | | | XIN = 10MHz (square wave) High-speed on-chip oscillator off Low-speed on-chip oscillator on=125kHz Divide-by-8 | – | 1.6 | – | mA |
| | | High-Speed On-Chip Oscillator Mode | Main clock off High-speed on-chip oscillator on=8MHz Low-speed on-chip oscillator on=125kHz No division | – | 3.5 | 7.5 | mA |
| | | | Main clock off High-speed on-chip oscillator on=8MHz Low-speed on-chip oscillator on=125kHz Divide-by-8 | – | 1.5 | – | mA |
| | | Low-Speed On-Chip Oscillator Mode | Main clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on=125kHz Divide-by-8 | – | 420 | 800 | μA |
| | | Wait Mode | Main clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on=125kHz While a WAIT instruction is executed Peripheral clock operation VCA26 = VCA27 = 0 | – | 37 | 74 | μA |
| | | Wait Mode | Main clock off High-speed on-chip oscillator off Low-speed on-chip oscillator on=125kHz While a WAIT instruction is executed Peripheral clock off VCA26 = VCA27 = 0 | – | 35 | 70 | μA |
| | | Stop Mode | Main clock off, Topr = 25 °C High-speed on-chip oscillator off Low-speed on-chip oscillator off CM10 = 1 Peripheral clock off VCA26 = VCA27 = 0 | – | 0.7 | 3.0 | μA |

Timing requirements (Unless otherwise specified: Vcc = 3V, Vss = 0V at Topr = 25 °C) [Vcc = 3V]**Table 19.22 XIN Input**

| Symbol | Parameter | Standard | | Unit |
|-----------------------|----------------------|----------|------|------|
| | | Min. | Max. | |
| t _c (XIN) | XIN Input Cycle Time | 100 | – | ns |
| t _{WH} (XIN) | XIN Input “H” Width | 40 | – | ns |
| t _{WL} (XIN) | XIN Input “L” Width | 40 | – | ns |

Table 19.23 CNTR0 Input, CNTR1 Input, $\overline{\text{INT1}}$ Input

| Symbol | Parameter | Standard | | Unit |
|-------------------------|------------------------|----------|------|------|
| | | Min. | Max. | |
| t _c (CNTR0) | CNTR0 Input Cycle Time | 300 | – | ns |
| t _{WH} (CNTR0) | CNTR0 Input “H” Width | 120 | – | ns |
| t _{WL} (CNTR0) | CNTR0 Input “L” Width | 120 | – | ns |

Table 19.24 TCIN Input, $\overline{\text{INT3}}$ Input

| Symbol | Parameter | Standard | | Unit |
|------------------------|-----------------------|----------------------|------|------|
| | | Min. | Max. | |
| t _c (TCIN) | TCIN Input Cycle Time | 1,200 ⁽¹⁾ | – | ns |
| t _{WH} (TCIN) | TCIN Input “H” Width | 600 ⁽²⁾ | – | ns |
| t _{WL} (TCIN) | TCIN Input “L” Width | 600 ⁽²⁾ | – | ns |

NOTES:

1. When using the Timer C input capture mode, adjust the cycle time (1/Timer C count source frequency x 3) or above.
2. When using the Timer C input capture mode, adjust the width (1/Timer C count source frequency x 1.5) or above.

Table 19.25 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) | RXD _i Input Setup Time | 70 | – | ns |
| t _h (C-D) | RCD _i Input Hold Time | 90 | – | ns |

Table 19.26 External Interrupt $\overline{\text{INT0}}$ Input

| Symbol | Parameter | Standard | | Unit |
|----------------------|--|--------------------|------|------|
| | | Min. | Max. | |
| t _W (INH) | $\overline{\text{INT0}}$ Input “H” Width | 380 ⁽¹⁾ | – | ns |
| t _W (INL) | $\overline{\text{INT0}}$ Input “L” Width | 380 ⁽²⁾ | – | ns |

NOTES:

1. When selecting the digital filter by the $\overline{\text{INT0}}$ input filter select bit, use the $\overline{\text{INT0}}$ input HIGH width to the greater value, either (1/digital filter clock frequency x 3) or the minimum value of standard.
2. When selecting the digital filter by the $\overline{\text{INT0}}$ input filter select bit, use the $\overline{\text{INT0}}$ input LOW width to the greater value, either (1/digital filter clock frequency x 3) or the minimum value of standard.

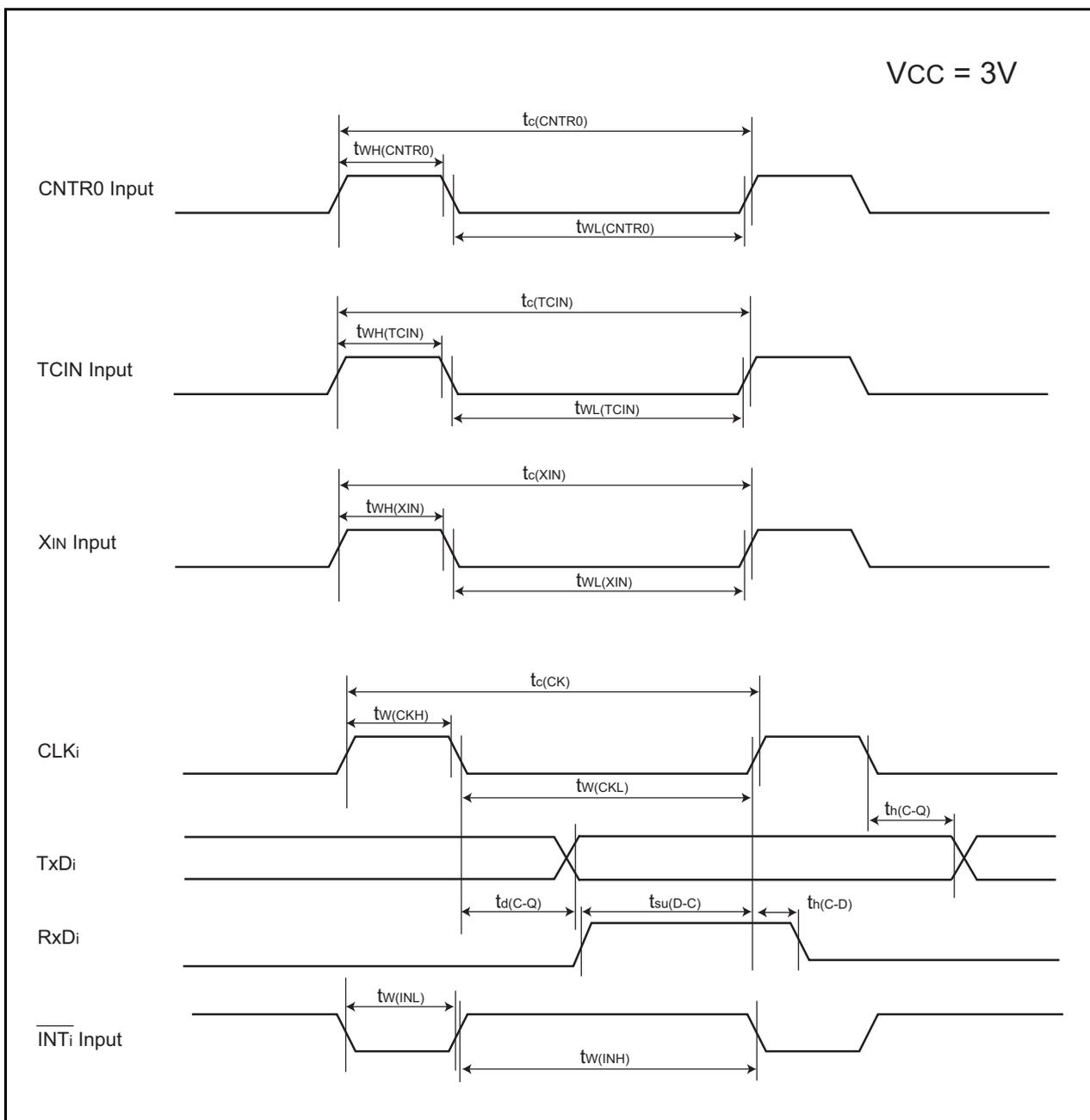


Figure 19.8 Timing Diagram When $V_{CC} = 3V$

20. Precautions

20.1 Stop Mode and Wait Mode

20.1.1 Stop Mode

When entering stop mode, set the FMR01 bit to "0" (CPU rewrite mode disabled) and the CM10 bit to "1" (stop mode). An instruction queue pre-reads 4 bytes from the instruction which sets the CM10 bit in the CM1 register to "1" (stop mode) and the program stops. Insert at least 4 NOP instructions after inserting the JMP.B instruction immediately after the instruction which sets the CM10 bit to "1". Use the next program to enter stop mode.

- Program to enter stop mode

```

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

```

20.1.2 Wait Mode

When entering wait mode, set the FMR01 bit 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.

Also, the value in the specific internal RAM area may be rewritten when exiting wait mode if writing to the internal RAM area before executing the WAIT instruction and entering wait mode. The area for a maximum of 3 bytes is rewritten from the following address of the internal RAM in which the writing is performed before the WAIT instruction. The rewritten value is the same value as the one which was written before the WAIT instruction. If this causes a problem, avoid by inserting the JMP.B instruction between the writing instruction to the internal RAM area and WAIT instruction as shown in the following program example.

- Example to execute the WAIT instruction

```

Program Example    MOV.B    #055h,0601h    ; Write to internal RAM area
...
JMP.B             LABEL_001
LABEL_001 :
FSET              I                ; Enable interrupt
BCLR              1,FMR0           ; CPU rewrite mode disabled
WAIT              ; Wait mode
NOP
NOP
NOP
NOP

```

When accessing any area other than the internal RAM area between the writing instruction to the internal RAM area and execution of the WAIT instruction, this situation will not occur.

20.2 Interrupts

20.2.1 Reading Address 00000h

Do not read the 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 the address 00000h is read in a program, the IR bit for the interrupt which has the highest priority among the enabled interrupts is set to "0". This may cause a problem that the interrupt is canceled, or an unexpected interrupt is generated.

20.2.2 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 any value in the SP, the program may run out of control.

20.2.3 External Interrupt and Key Input Interrupt

Either an "L" level or an "H" level of at least 250ns width is necessary for the signal input to the $\overline{\text{INT0}}$ to INT3 pins and $\overline{\text{KI0}}$ to $\overline{\text{KI3}}$ pins regardless of the CPU clock.s

20.2.4 Watchdog Timer Interrupt

Reset the watchdog timer after a watchdog timer interrupt is generated.

20.2.5 Changing Interrupt Factor

The IR bit in the interrupt control register may be set to "1" (interrupt requested) when the interrupt factor changes. When using an interrupt, set the IR bit to "0" (no interrupt requested) after changing the interrupt factor.

In addition, the changes of interrupt factors include all factors that change the interrupt factors assigned to individual software interrupt numbers, polarities, and timing. Therefore, when a mode change of the peripheral functions involves interrupt factors, edge polarities, and timing, Set the IR bit to "0" (no interrupt requested) after the change. Refer to each peripheral function for the interrupts caused by the peripheral functions.

Figure 20.1 shows an Example of Procedure for Changing Interrupt Factor.

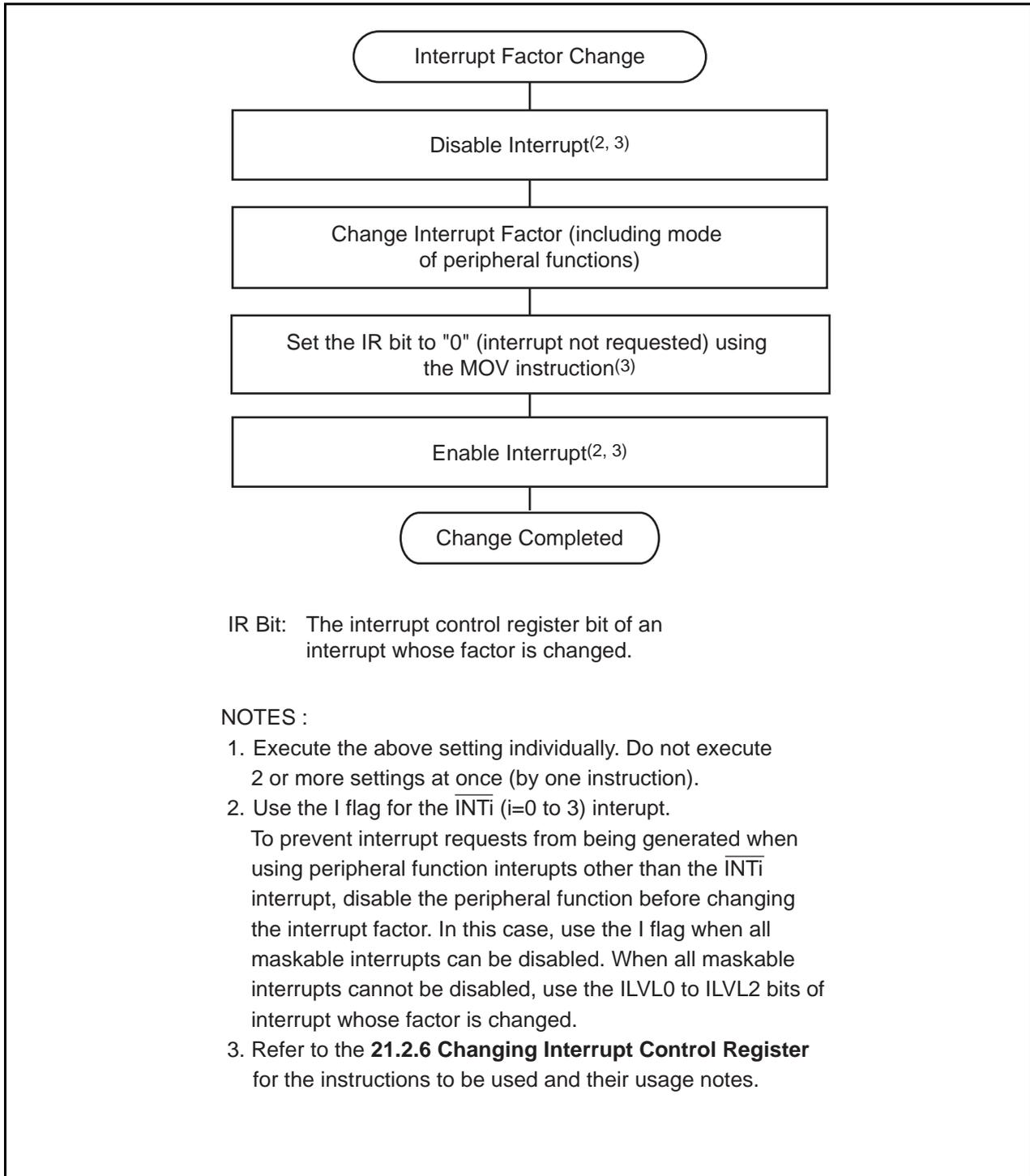


Figure 20.1 Example of Procedure for Changing Interrupt Factor

20.2.6 Changing Interrupt Control Register

- (a) Each interrupt control register can only be changed while interrupt requests corresponding to that register are not generated. If interrupt requests may be generated, disable the interrupts before changing the interrupt control register.
- (b) When changing any interrupt control register after disabling interrupts, be careful with the instructions to be used.
When changing any bit other than IR bit
 If an interrupt request corresponding to that 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.
 Instructions to use: AND, OR, BCLR, BSET
When 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 to be 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 according to the following sample programs. Refer to (b) for the change of interrupt control registers in the sample programs.

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

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

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

Example 2: Use dummy read to have FSET instruction wait

```
INT_SWITCH2:
  FCLR   I           ; Disable interrupts
  AND.B  #00H,0056H ; Set TXIC 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 TXIC register to "00h"
  POPC   FLG        ; Enable interrupts
```

20.3 Clock Generation Circuit

20.3.1 Oscillation Stop Detection Function

Since the oscillation stop detection function cannot be used if the main clock frequency is below 2MHz, set the OCD1 to OCD0 bits to "00b" (oscillation stop detection function disabled).

20.3.2 Oscillation Circuit Constants

Ask the maker of the oscillator to specify the best oscillation circuit constants on your system.

20.4 Timers

20.4.1 Timers X and Z

- Timers X and Z stop counting after reset. Set the value to these timers and prescalers before the count starts.
- Even if the prescalers and timers are read out in 16-bit units, these registers are read by 1 byte in the microcomputer. Consequently, the timer value may be updated during the period these two registers are being read.

20.4.2 Timer X

- Do not rewrite the TXMOD0 to TXMOD1 bits, the TXMOD2 and TXS bits simultaneously.
- In pulse period measurement mode, the TXEDG bit and TXUND bit in the TXMR register can be set to "0" by writing "0" to these bits by a program. However, these bits remain unchanged when "1" is written. When using the READ-MODIFY-WRITE instruction for the TXMR register, the TXEDG or TXUND bit may be set to "0" although these bits are set to while the instruction is executed. At the time, write "1" to the TXEDG or TXUND bit which is not supposed to be set to "0" with the MOV instruction.
- When changing to pulse period measurement mode from other mode, the contents of the TXEDG and TXUND bits are indeterminate. Write "0" to the TXEDG and TXUND bits before the count starts.
- The TXEDG bit may be set to "1" by the prescaler X underflow which is generated for the first time since the count starts.
- When using the pulse period measurement mode, leave two periods or more of the prescaler X immediately after count starts, and set the TXEDG bit to "0".
- The TXS bit in the TXMR register has a function to instruct Timer X to start or stop counting, and a function to indicate the count starts or stops.
"0" (count stops) can be read until the following count source is applied after "1" (count starts) is written to the TXS bit while the count is being stopped. If the following count source is applied, "1" can be read from the TXS bit. Do not access registers associated with Timer X (TXMR, PREX, TX, TCSS, TXIC registers) except for the TXS bit until "1" can be read from the TXS bit. The count starts at the following count source after the TXS bit is set to "1".
Also, when writing "0" (count stops) to the TXS bit during the count, Timer X stops counting at the following count source.
"1" (count starts) can be read by reading the TXS bit until the count stops after writing "0" to the TXS bit. Do not access registers associated with Timer X other than the TXS bit until "0" can be read by the TXS bit after writing "0" to the TXS bit.

20.4.3 Timer Z

- Do not rewrite the TZMOD0 to TZMOD1 bits and the TZS bit simultaneously.
- In programmable one-shot generation mode and programmable wait one-shot generation mode, when setting the TZS bit in the TZMR register to "0" (stops counting) or setting the TZOS bit in the TZOC register to "0" (stops one-shot), the timer reloads the value of reload register and stops. Therefore, read the timer count value in programmable one-shot generation mode and programmable wait one-shot generation mode before the timer stops.
- The TZS bit in the TZMR register has a function to instruct Timer Z to start or stop counting, and a function to indicate the count starts or stops.
"0" (count stops) can be read until the following count source is applied after "1" (count starts) is written to the TZS bit while the count is being stopped. If the following count source is applied, "1" can be read from the TZS bit. Do not access registers associated with Timer Z (TZMR, PREZ, TZSC, TZPR, TZOC, PUM, TCSC, TZIC registers) except for the TZS bit until "1" can be read from the TZS bit. The count starts at the following count source after the TZS bit is set to "1".
Also, when writing "0" (count stops) to the TZS bit during the count, Timer Z stops counting at the following count source.
"1" (count starts) can be read by reading the TZS bit until the count stops after writing "0" to the TZS bit. Do not access registers associated with Timer Z other than the TZS bit until "0" can be read by the TZS bit after writing "0" to the TZS bit.

20.4.4 Timer C

Access the TC, TM0 and TM1 registers in 16-bit units.

The TC register can be read in 16-bit units. This prevents the timer value from being updated between the low-order byte and high-order byte are being read.

Example (when Timer C is read):

```
MOV.W    0090H,R0    ; Read out timer C
```

20.5 Serial Interface

- When reading data from the U0RB (i = 0, 1) register even in the clock asynchronous serial I/O mode or in the clock synchronous serial I/O mode. Ensure to read data in 16-bit unit. When the high-order byte of the U0RB register is read, the PER and FER bits in the U0RB register and the RI bit in the U0C1 register are set to "0".

Example (when reading receive buffer register):

```
MOV.W    00A6H,R0    ; Read the U0RB register
```

- When writing data to the U0TB register in the clock asynchronous serial I/O mode with 9-bit transfer data length, write data high-order byte first, then 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
```

20.6 Clock Synchronous Serial I/O (SSU) with Chip Select

20.6.1 Access Registers Associated with SSU

After the conditions of “3 instructions or more after writing to the registers associated with SSU (00B8h to 00BFh)” or “4 cycles or more after writing to them” are met, read those registers.

- An example to wait for 3 instructions or more

```

Program Example      MOV.B      #00h,00BBh      ; Set the SSER register to "00h".
                    NOP
                    NOP
                    NOP
                    MOV.B      00BBh,R0L
  
```

- An example to wait for 4 cycles or more

```

Program Example      BCLR      4,00BBh      : Disable transmit
                    JMP.B      NEXT
NEXT:
                    BEST       3,00BBh      : Enable receive
  
```

20.7 A/D Converter

- Write to each bit (other than bit 6) in the ADCON0 register, each bit in the ADCON1 register, or the SMP bit in the ADCON2 register when the A/D conversion stops (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 or longer before the A/D conversion starts.
- When changing A/D operating mode, select an analog input pin again.
- When using in one-shot mode. Ensure that the A/D conversion is completed and read the AD register. The IR bit in the ADIC register or the ADST bit in the ADCON0 register can determine whether the A/D conversion is completed.
- When using In repeat mode, use the undivided main clock for the CPU clock.
- If setting the ADST bit in the ADCON0 register to “0” (A/D conversion stops) by a program and the A/D conversion is forcibly terminated during the A/D conversion operation, the conversion result of the A/D converter will be indeterminate. If the ADST bit is set to “0” by a program, do not use the value of AD register.
- Connect 0.1 μ F capacitor between the AVCC/VREF pin and AVSS pin.

20.8 Flash Memory Version

20.8.1 CPU Rewrite Mode

20.8.1.1 Operating Speed

Before entering CPU rewrite mode (EW0 mode), select 5MHz or below for the CPU clock using the CM06 bit in the CM0 register and the CM16 to CM17 bits in the CM1 register. This usage note is not needed for EW1 mode.

20.8.1.2 Instructions Inhibited Against Use

The following instructions cannot be used in EW0 mode because the flash memory internal data is referenced: UND, INTO, and BRK instructions.

20.8.1.3 Interrupts

Table 20.1 lists the Interrupt in EW0 Mode and Table 20.2 lists the Interrupt in EW1 Mode.

Table 20.1 Interrupt in EW0 Mode

| Mode | Status | When maskable interrupt request is acknowledged | When watchdog timer, oscillation stop detection and voltage monitor 2 interrupt request are acknowledged |
|------|--------------------------|---|--|
| EW0 | During automatic erasing | Any interrupt can be used by allocating a vector to RAM | Once an interrupt request is acknowledged, the auto-programming or auto-erasing is forcibly stopped immediately and resets the flash memory. An interrupt process starts after the fixed period and the flash memory restarts. Since the block during the auto-erasing or the address during the auto-programming is forcibly stopped, the normal value may not be read. Execute the auto-erasing again and ensure the auto-erasing is completed normally. Since the watchdog timer does not stop during the command operation, the interrupt request may be generated. Reset the watchdog timer regularly. |
| | Automatic writing | | |

NOTES:

1. Do not use the address match interrupt while the command is executed because the vector of the address match interrupt is allocated on ROM.
2. Do not use the non-maskable interrupt while Block 0 is automatically erased because the fixed vector is allocated Block 0.

Table 20.2 Interrupt in EW1 Mode

| Mode | Status | When maskable interrupt request is acknowledged | When watchdog timer, oscillation stop detection and voltage monitor 2 interrupt request are acknowledged |
|------|---|---|--|
| EW1 | During automatic erasing (erase-suspend function is enabled) | The auto-erasing is suspended after td(SR-ES) and the interrupt process is executed. The auto-erasing can be restarted by setting the FMR41 bit in the FMR4 register to "0"(erase restart) after the interrupt process completes. | Once an interrupt request is acknowledged, the auto-programming or auto-erasing is forcibly stopped immediately and resets the flash memory. An interrupt process starts after the fixed period and the flash memory restarts. Since the block during the auto-erasing or the address during the auto-programming is forcibly stopped, the normal value may not be read. Execute the auto-erasing again and ensure the auto-erasing is completed normally. Since the watchdog timer does not stop during the command operation, the interrupt request may be generated. Reset the watchdog timer regularly using the erase-suspend function. |
| | During automatic erasing (erase-suspend function is disabled) | The auto-erasing has a priority and the interrupt request acknowledgement is waited. The interrupt process is executed after the auto-erasing completes. Refer to 20.8.1.9 Interrupt Request Generation during Auto-erase Operation in EW1 Mode. | |
| | Auto programming | The auto-programming has a priority and the interrupt request acknowledgement is waited. The interrupt process is executed after the auto-programming completes. | |

NOTES:

1. Do not use the address match interrupt while the command is executed because the vector of the address match interrupt is allocated on ROM.
2. Do not use the non-maskable interrupt while Block 0 is automatically erased because the fixed vector is allocated Block 0.

20.8.1.4 How to Access

Write "0" to the corresponding bits before writing "1" when setting the FMR01, FMR02, or FMR11 bit to "1". Do not generate an interrupt between writing "0" and "1".

20.8.1.5 Rewriting User ROM Area

In EW0 Mode, if the power supply voltage drops while rewriting any block in which the rewrite control program is stored, the flash memory may not be able to be rewritten because the rewrite control program cannot be rewritten correctly. In this case, use standard serial I/O mode.

20.8.1.6 Program

Do not write additions to the already programmed address.

20.8.1.7 Reset Flash Memory

When setting the FMSTP bit in the FMR0 register to "1" (flash memory stops) during erase-suspend in EW1 mode, a CPU stops and cannot return. Do not set the FMSTP bit to "1".

20.8.1.8 Entering Stop Mode or Wait Mode

Do not enter stop mode or wait mode during erase-suspend.

20.8.1.9 Interrupt Request Generation during Auto-erase Operation in EW1 Mode

When an interrupt request is generated during erasing with FMR01 = 1 (CPU rewrite mode enabled) in FMR0 register, FMR11 = 1 (EW1 mode) in FMR1 register and FMR40 = 0 (disable erase suspend function) in FMR4 register, the CPU may not operate properly.

Select any of the following 3 processes as a software countermeasure:

- (a) Disable an interrupt by setting the priority level of all maskable interrupts to level 0. Note that disabling the interrupts by the I flag will not be in the software countermeasure
- (b) Set the FMR40 = 1 (enable erase suspend function) and the I flag = 1 (enable interrupt) when using the FMR11 = 1 (EW1 mode)
- (c) Use EW0 mode.

20.9 Noise

20.9.1 Insert a bypass capacitor between VCC and VSS pins as the countermeasures against noise and latch-up

Connect the bypass capacitor (at least 0.1 μ F) using the shortest and thickest as possible.

20.9.2 Countermeasures against Noise Error of Port Control Registers

During severe noise testing, mainly power supply system noise, and introduction of external noise, the data of port related registers may be changed.

As a firmware countermeasure, it is recommended to periodically reset the port registers, port direction registers and pull-up control registers. However, examine fully before introducing the reset routine as conflicts may be created between this reset routine and interrupt routines.

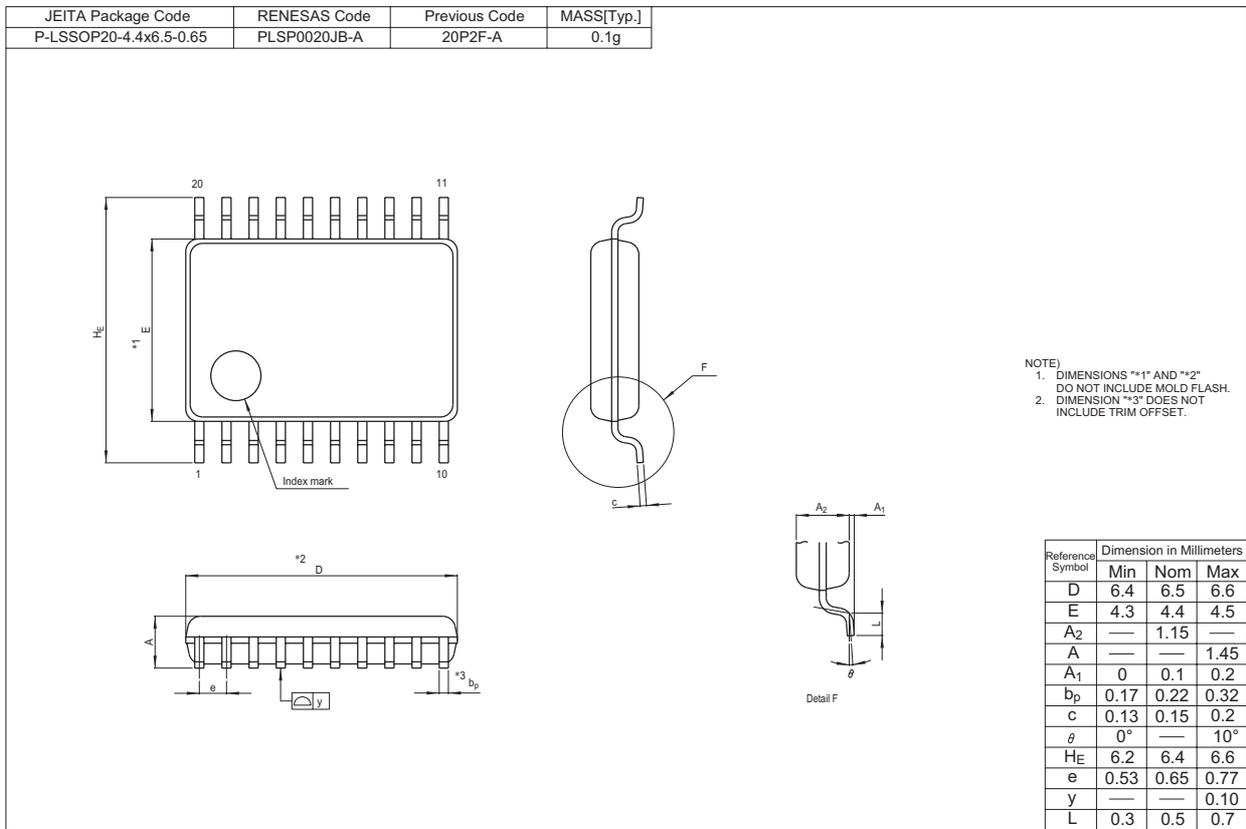
21. Precaution for On-chip Debugger

When using the on-chip debugger to develop the R8C/14 and R8C/15 groups program and debug, pay the following attention.

- (1) Do not use from OC000h address to OC7FFh because the on-chip debugger uses these addresses.
- (2) Do not set the address match interrupt (the registers of AIER, RMAD0, RMAD1 and the fixed vector tables) in a user system.
- (3) Do not use the BRK instruction in a user system.
- (4) The stack pointer with up to 8 bytes is used during the user program break. Therefore, save space of 8 bytes for the stack area.

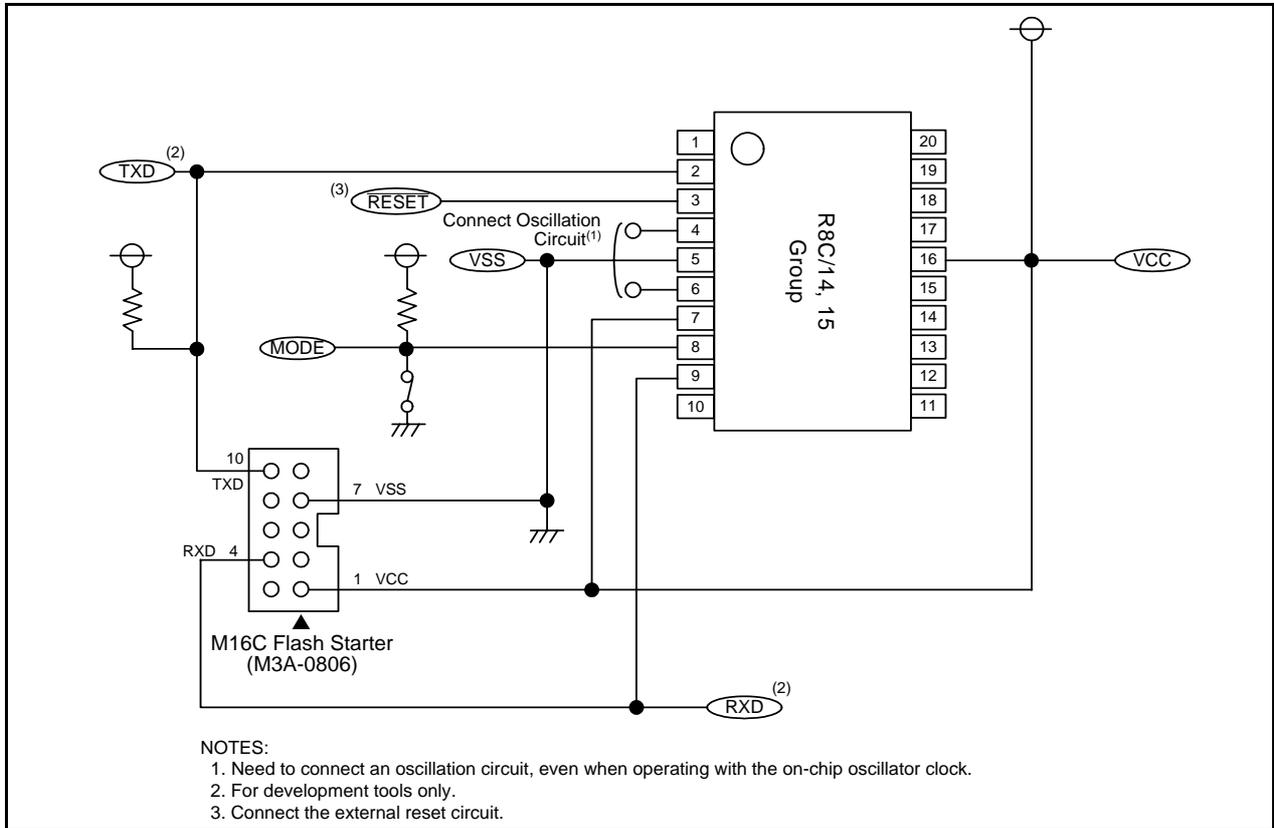
Connecting and using the on-chip debugger has some peculiar restrictions. Refer to each on-chip debugger manual for on-chip debugger details.

Appendix 1. Package Dimensions

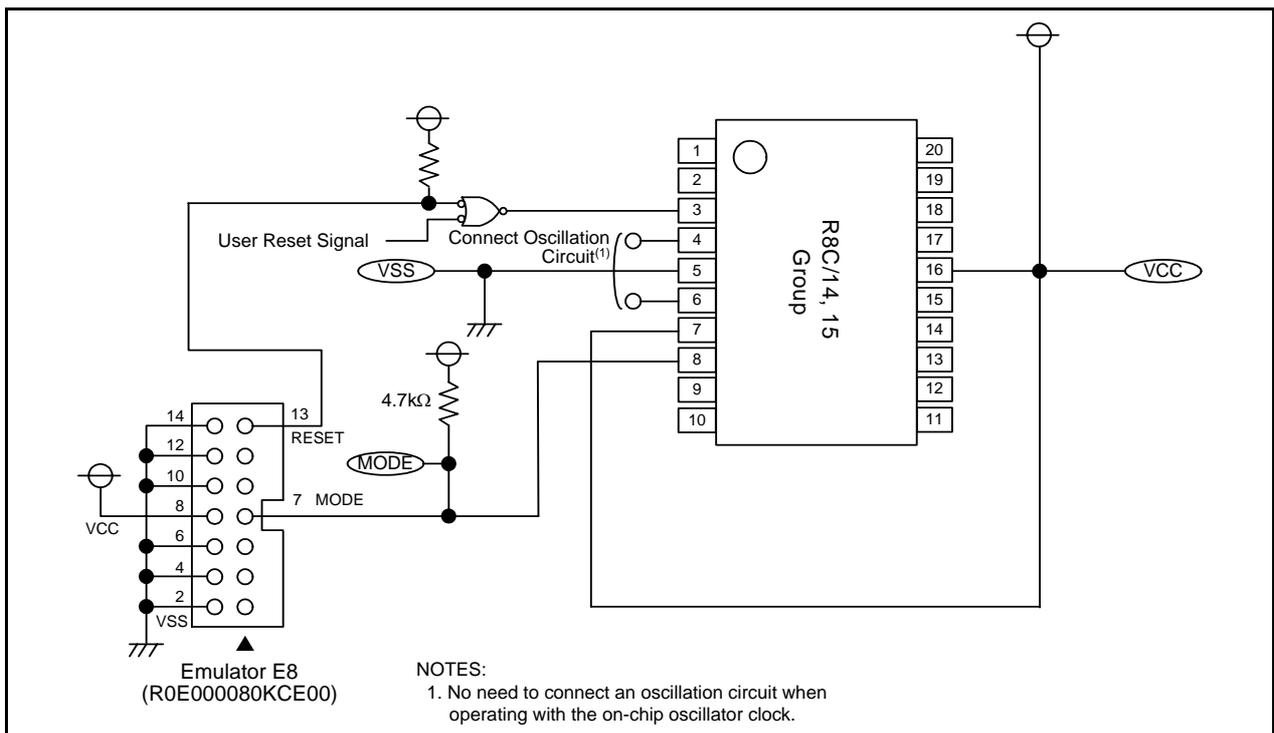


Appendix 2. Connecting Example between Serial Writer and On-Chip Debugging Emulator

Appendix Figure 2.1 shows the Connecting Example with M16C Flash Starter (M3A-0806) and Appendix Figure 2.2 shows the Connecting Example with Emulator E8 (R0E000080KCE00).



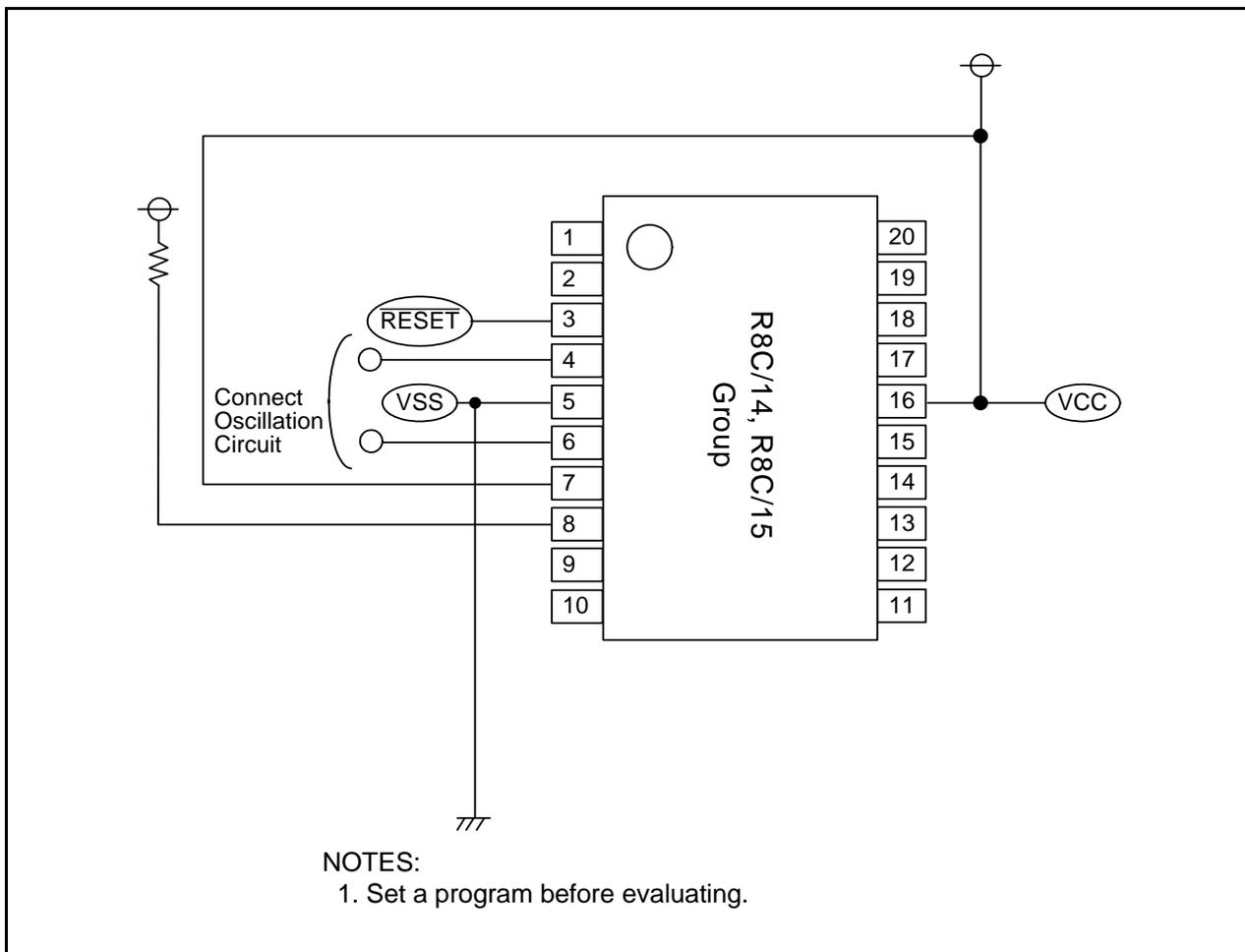
Appendix Figure 2.1 Connecting Example with M16C Flash Starter (M3A-0806)



Appendix Figure 2.2 Connecting Example with Emulator E8 (R0E000080KCE00)

Appendix 3. Example of Oscillation Evaluation Circuit

Appendix Figure 3.1 shows the Example of Oscillation Evaluation Circuit.



Appendix Figure 3.1 Example of Oscillation Evaluation Circuit

Register Index

| A | | P | | U | |
|--------------|---------|--------------|---------|-------------|-----|
| AD | 171 | P1 | 184 | U0BRG | 127 |
| ADCON0 | 170 | P3 | 184 | U0C0 | 128 |
| ADCON1 | 170 | P4 | 184 | U0C1 | 129 |
| ADCON2 | 171 | PD1 | 184 | U0MR | 128 |
| ADIC | 61 | PD3 | 184 | U0RB | 127 |
| AIER | 77 | PD4 | 184 | U0TB | 127 |
| | | PM0 | 35 | UCON | 129 |
| | | PM1 | 36 | | |
| | | PRCR | 55 | V | |
| | | PREX | 86 | VCA1 | 28 |
| | | PREZ | 100 | VCA2 | 28 |
| | | PUM | 101 | VW1C | 29 |
| | | PUR0 | 185 | VW2C | 30 |
| | | PUR1 | 185 | | |
| C | | R | | W | |
| CM0 | 40 | RMAD0 | 77 | WDC | 79 |
| CM1 | 41 | RMAD1 | 77 | WDTR | 80 |
| CMP0IC | 61 | | | WDTS | 80 |
| CMP1IC | 61 | | | | |
| CSPR | 80 | | | | |
| D | | S | | | |
| DRR | 185 | S0RIC | 61 | | |
| | | S0TIC | 61 | | |
| | | SSCRH | 142 | | |
| | | SSCRL | 143 | | |
| | | SSER | 145 | | |
| | | SSMR | 144 | | |
| | | SSMR2 | 147 | | |
| | | SSRDR | 148 | | |
| | | SSSR | 146 | | |
| | | SSTDTR | 148 | | |
| | | SSUAIC | 61 | | |
| F | | T | | | |
| FMR0 | 200 | TC | 117 | | |
| FMR1 | 201 | TCC0 | 118 | | |
| FMR4 | 201 | TCC1 | 119 | | |
| H | | TCIC | 61 | | |
| HRA0 | 43 | TCOUT | 120 | | |
| HRA1 | 44 | TCSS | 86, 102 | | |
| HRA2 | 44 | TM0 | 117 | | |
| | | TM1 | 117 | | |
| | | TX | 86 | | |
| | | TXIC | 61 | | |
| | | TXMR | 85 | | |
| | | TZIC | 61 | | |
| | | TZMR | 99 | | |
| | | TZOC | 101 | | |
| | | TZPR | 100 | | |
| | | TZSC | 100 | | |
| I | | | | | |
| INT0F | 69 | | | | |
| INT0IC | 62 | | | | |
| INT1IC | 61 | | | | |
| INT3IC | 61 | | | | |
| INTEN | 69 | | | | |
| K | | | | | |
| KIEN | 75 | | | | |
| KUPIC | 61 | | | | |
| O | | | | | |
| OCD | 42 | | | | |
| OFS | 79, 196 | | | | |

REVISION HISTORY

R8C/14 Group, R8C/15 Group Hardware

| Rev. | Date | Description | |
|------|--------------|-------------|---|
| | | Page | Summary |
| 0.10 | May 17, 2004 | – | First Edition issued |
| 0.20 | Jul 12, 2004 | – | Rev.0.20 issued |
| 0.30 | Aug 06, 2004 | all pages | Words standardized (on-chip oscillator, serial interface, SSU) |
| | | 2 | Table 1.1 revised |
| | | 3 | Table 1.2 revised |
| | | 9 | Table 1.5 revised |
| | | 10 | Table 1.6 added |
| | | 14,15 | “Address Break” in Figures 3.1 and 3.2 ; notes added |
| | | 16 | Table 4.1, HRA2 Register at 0022h added ; NOTE2 to 6 revised |
| | | 18 | Table 4.3 the value after reset to FFh at 009Ch to 009Fh revised |
| | | 19 | Tabel 4.4, the value after reset to FFh at 009Ch to 009Fh revised ; NOTES added |
| | | 20-25 | Compositions and contents of “5. Reset” modified |
| | | 26-35 | Compositions and contents of “6. Voltage Detection Circuit” modified |
| | | 37 | Figure 7.2, function of b0 revised |
| | | 40 | Figure 9.1 revised |
| | | 41 | Figure 9.2, “System” at CM06 bit added |
| | | 42 | Figure 9.3, “System” at CM16 and CM17 bits added |
| | | 44 | Figure 9.5 revised |
| | | 47 | 9.2.2, “The oscillation starts...HRA2 registers” added |
| | | 48 | 9.3.1 added |
| | | | 9.3.3 “The clock...divided-by-i”added |
| | | 52 | Table 9.4 revised |
| | | 60 | 11.1.3.4, “Address Break Interrup” added ; the referred distination to “20. On-Chip Debugger” revised |
| | | 61 | Table 11.1, some referred distinations revised |
| | | 62 | Table 11.2, some referred distinations revised |
| | | 68 | Figures 11.7 and 11.8 added |
| | | 72 | 11.2.1, “The INTO pin...timer Z” added |
| | | 74 | 11.2.3, “The INTO pin...CNTR01 pin” added |
| | | 75 | 11.2.4, “The INT3 pin is used with the TCIN pin” added |
| | | 80-84 | Compositions and contents of “12. Watchdog Timer” modified |
| | | 87 | Figure 13.2 revised |
| | | 89 | Table 13.2 revised |
| | | 90 | Table 13.3 revised |
| | | 91 | Figure 13.5 revised |
| | | 92 | Table 13.4 revised |
| | | 93 | Figure 13.6 revised |
| | | 94 | Table 13.5 revised |
| | | 95 | Figure 13.7 revised |
| | | 97 | Table 13.6 revised |
| | | 98 | Figure 13.9 revised |
| | | 99 | Figure 13.10 revised |
| | | 100 | 13.2 revised |
| | | 105 | Table 13.7 revised |
| | | 107 | Table 13.8 revised |

REVISION HISTORY

R8C/14 Group, R8C/15 Group Hardware

| Rev. | Date | Description | |
|---------|---|-------------|---|
| | | Page | Summary |
| 0.30 | Aug 06, 2004 | 110 | Table 13.9 revised |
| | | 112 | Figure 13.20 revised |
| | | 114 | Table 13.10 revised |
| | | 118 | Figure 13.25 revised |
| | | 119 | Figure 13.26 revised |
| | | 121 | Figure 13.28 revised |
| | | 123 | Table 13.11 revised |
| | | 125 | Table 13.12 revised |
| | | 130 | Figure 14.4 revised |
| | | 131 | Figure 14.5 revised |
| | | 136 | 14.1.3 revised |
| | | 138 | Table 14.5, NOTES revised |
| | | 140 | Figure 14.10 revised ; 14.2.1 "input" added |
| | | 141 | 14.2.2 added |
| | | 143 | Figure 15.1 revised |
| | | 144 | Figure 15.2 revised |
| | | 146 | Figure 15.4 revised |
| | | 148 | Figure 15.6 revised |
| | | 152 | Figure 15.9 revised |
| | | 154 | 15.3 revised |
| | | 159 | Figure 15.14 revised |
| | | 162 | 15.6 revised |
| | | 164 | 15.6.2 revised |
| | | 165 | Figure 15.18 revised |
| | | 167 | Figure 15.19 revised |
| | | 171 | Figure 16.2 revised |
| | | 174 | Figure 16.4 revised |
| | | 175 | Table 16.3 revised |
| | | 176 | Figure 16.5 revised |
| | | 178 | 17.1.4 revised |
| | | 179 | Figure 17.1 revised |
| | | 180 | Figure 17.2 revised |
| | | 184 | Figure 17.8 revised |
| | | 185 | Table 17.1 revised |
| | | 186 | Table 18.1 revised |
| | | 188 | 18.2 revised |
| | | 89 | Figure 18.2, NOTES revised |
| | | 190 | Figure 18.3 ID5 and 6 revised |
| | | 191 | 18.3.2 revised ; "After Reset" revised to "Before Shipment" |
| | | 193 | 18.4.1 and 18.4.2 revised |
| 195 | 18.4.2.11 and 18.4.2.12 revised | | |
| | Figure 18.5 revised | | |
| 196 | Figure 18.6 revised | | |
| 198 | Figure 18.9 revised | | |
| 204 | Table 18.6 revised | | |
| 210-223 | "19. Electrical Characteristics" added | | |
| 230 | 21.1 "Stop Mode and Wait Mode" revised | | |
| 240 | 21.7.1.8 revised | | |
| | 21.7.1.9 added | | |
| 244 | "Appendix 2. Connecting Example between Serial Writer and On-Chip Debugging Emulator" added | | |
| 247 | "Appendix 3. Example of Oscillation Evaluation Circuit" added | | |

REVISION HISTORY

R8C/14 Group, R8C/15 Group Hardware

| Rev. | Date | Description | |
|---------|---|-------------|--|
| | | Page | Summary |
| 1.00 | Feb 25, 2005 | 2-3 | Tables 1.1 and 1.2 revised |
| | | 5 | Table 1.3 and figure 1.2 revised |
| | | 6 | Table 1.4 and figure 1.3 revised |
| | | 7-8 | Figures 1.4 and 1.5 revised |
| | | 16 | Tabel 4.1, The value after reset to 000XXXXXb to 00011111b at 000Fh; and the value after reset to 00001000b to 0000X000b and 01001001b to 0100X001b at 0036h revised |
| | | 18 | Tabel 4.3, The value after reset to 0000h at 009Ch to 009Dh revised; NOTES2 added, and the value after reset to 00h at 00BCh revised |
| | | 20 | Figure 5.1 revised |
| | | 22 | 5.1.1 (2) and 5.1.2 (4) revised |
| | | 24 | 5.2 revised |
| | | | Figure 5.6 revised |
| | | 25 | 5.3 revised |
| | | 26 | Table 6.1 revised |
| | | 27 | Figures 6.1 and 6.2 revised |
| | | 29 | Figure 6.4 revised |
| | | 30 | Figure 6.5 revised |
| | | 31 | Figure 6.6 revised |
| | | 32 | 6.1.1 revised |
| | | 33 | Table 6.2 and figure 6.7 revised |
| | | 34 | Table 6.3 revised |
| | | 35 | Figure 6.8 revised |
| | | 37 | Figure 7.2 revised |
| | | 39 | Table 9.1 revised; NOTE2 added |
| | | 40 | Figure 9.1 revised |
| | | 41 | Figure 9.2 revised |
| | | 42 | Figure 9.3 revised |
| | | 44 | Figure 9.5 revised |
| | | 51 | Table 9.3 revised |
| | | 52 | Table 9.4 revised |
| | | 55 | 9.5 and 9.5.1 revised |
| | | | Table 9.5 revised |
| | | 60 | 11.1.3.5 revised |
| | | 61 | Table 11.1 revised |
| | | 68 | 11.1.6.7 revised |
| | | 71 | Figure 11.11 "INTEN Register" revised |
| 78-79 | 11.4 "Address Match Interrupt", Table 11.6, 11.7 and Figure 11.19 added | | |
| 81 | Figure 12.2 "WDC Register" revised | | |
| 83 | Table 12.1 revised | | |
| 89-96 | Table 13.2, 13.3, 13.4, 13.5 and 13.6 revised; "Write to Timer" revised | | |
| 104 | Table 13.7 revised | | |
| 106-113 | Table 13.8, 13.9 and 13.10 revised | | |
| 118 | Figure 13.26 revised | | |
| 126 | Figure 14.1 revised | | |
| 129 | Figure 14.4 "U0C0 Register" revised | | |
| 130 | Figure 14.5 "UCON Register" revised | | |
| 131 | 14.1 revised | | |
| 137 | Table 14.6 revised | | |

REVISION HISTORY

R8C/14 Group, R8C/15 Group Hardware

| Rev. | Date | Description | |
|------|--|-------------|---|
| | | Page | Summary |
| 1.00 | Feb 25, 2005 | 141 | Table 15.1 NOTE2 added |
| | | 145 | Figure 15.4 NOTES added |
| | | 147 | Figure 15.6 revised; The value after reset to 00h |
| | | 149 | Figure 15.8 revised |
| | | 152 | 15.2 revised; "15.2 SS Shift Register" added and 15.2 revised to 15.5.2 |
| | | 157 | Figure 15.13 NOTE2 added |
| | | 158 | Figure 15.14 revised |
| | | 160 | 15.5.4 revised |
| | | 162 | 15.6 revised |
| | | 163 | Figure 15.17 revised |
| | | 164 | 15.6.2 revised |
| | | 168 | 15.6.4 revised |
| | | 169 | Table 16.1 revised |
| | | 171-176 | Figures 16.2, 16.4 and 16.5 revised |
| | | 178 | 17.1, 17.2 and 17.3 revised; Tables 17.1, 17.2 and 17.3 added |
| | | 185 | Table 17.4 revised |
| | | | Figure 17.9 added |
| | | 188-189 | Figures 18.1 and 18.2 revised |
| | | 191 | 18.3.2 revised |
| | | 192 | Table 18.3 revised |
| | | 202 | Figure 18.12 revised |
| | | 207 | Figure 18.14 revised |
| | | 211 | Table 19.3 revised |
| | | 212 | Table 19.4 and 19.5 revised |
| | | 213 | Figure 19.2, Tables 19.6 and 19.7 revised |
| | | 214 | Tables 19.8 and 19.9 revised |
| | | 215 | Table 19.10, 19.11 revised and Table 19.12 added |
| | | 216-218 | Figures 19.4, 19.5 and 19.6 added |
| | | 219 | Table 19.13 revised |
| | | 220 | Table 19.14 revised |
| | | 221, 225 | Table 19.16 and 19.23 revised: Table title "INT2" → "INT1" |
| | | 223 | Table 19.20 NOTE revised |
| | | 224 | Table 19.21 revised |
| | | 227 | 20.1.1 and 20.1.2 revised |
| 232 | 20.4.2 revised | | |
| 233 | 20.4.3 revised | | |
| 235 | 20.6 added | | |
| 236 | 20.7 revised | | |
| 239 | 20.8.1.7 and 20.8.1.8 revised | | |
| 241 | "20. On-chip Debugger" deleted | | |
| 242 | Appendix Package Dimensions revised | | |
| 243 | Appendix Figure 2.1 revised; "USB Flash Writer" deleted and "M16C Flash Starter" NOTE3 added | | |

REVISION HISTORY

R8C/14 Group, R8C/15 Group Hardware

| Rev. | Date | Description | |
|------|---------------------------------------|-------------|--|
| | | Page | Summary |
| 2.00 | Jan 12, 2006 | 1 | 1. Overview; "20-pin plastic molded LSSOP or SDIP" → "20-pin plastic molded LSSOP" revised |
| | | 2 | Table 1.1 Performance Outline of the R8C/14 Group; Package: "20-pin plastic molded SDIP" deleted |
| | | 3 | Table 1.2 Performance Outline of the R8C/15 Group; Package: "20-pin plastic molded SDIP" deleted, Flash Memory: (Data area) → (Data flash) (Program area) → (Program ROM) revised |
| | | 4 | Figure 1.1 Block Diagram; "Peripheral Function" added, "System Clock Generation" → "System Clock Generator" revised |
| | | 5, 6 | Table 1.3 Product Information of R8C/14 Group, Table 1.4 Product Information of R8C/15 Group; revised. Figure 1.2 Part Number, Memory Size and Package of R8C/14 Group, Figure 1.3 Part Number, Memory Size and Package of R8C/15 Group; Package type: "DD : PRDP0020BA-A" deleted |
| | | 8 | Figure 1.5 PRDP0020BA-A Package Pin Assignment (top view) deleted Table 1.5 Pin Description; Timer C: "CMP0_0 to CMP0_3, CMP1_0 to CMP1_3" → "CMP0_0 to CMP0_2, CMP1_0 to CMP1_2" revised |
| | | 10 | Figure 2.1 CPU Register; "Reserved Area" → "Reserved Bit" revised |
| | | 12 | 2.8.10 Reserved Area; "Reserved Area" → "Reserved Bit" revised |
| | | 13 | Figure 3.1 Memory Map of R8C/14 Group revised |
| | | 14 | 3.2 R8C/15 Group, Figure 3.2 Memory Map of R8C/15 Group revised |
| | | 15 | Table 4.1 SFR Information(1); 0009h: "XXXXXX00b" → "00h" 000Ah: "00XXX000b" → "00h" 001Eh: "XXXXX000b" → "00h" |
| | | 17 | Table 4.3 SFR Information(3); 0085h: "Prescaler Z" → "Prescaler Z Register" 0086h: "Timer Z Secondary" → "Timer Z Secondary Register" 0087h: "Timer Z Primary" → "Timer Z Primary Register" 008Ch: "Prescaler X" → "Prescaler X Register" 008Dh: "Timer X" → "Timer X Register" 0090h, 0091h: "Timer C" → "Timer C Register" revised |
| | | 20 | Figure 5.3 Reset Sequence revised |
| | | 23 | 5.2 Power-On Reset Function; "When a capacitor is connected to ... 0.8VCC or more." added |
| | | 29 | Figure 6.5 VW1C Register revised |
| 30 | Figure 6.6 VW2C Register NOTE10 added | | |

REVISION HISTORY

R8C/14 Group, R8C/15 Group Hardware

| Rev. | Date | Description | |
|------|--------------|-------------|--|
| | | Page | Summary |
| 2.00 | Jan 12, 2006 | 32 | Table 6.2 Setting Procedure of Voltage Monitor 1 Reset Associated Bit revised |
| | | 33 | Table 6.3 Setting Procedure of Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset Associated Bit revised |
| | | 37 | Table 8.2 Bus Cycles for Access Space of the R8C/15 Group added, Table 8.3 Access Unit and Bus Operation; "SFR" → "SFR, Data flash", "ROM/RAM" → "ROM (Program ROM), RAM" revised |
| | | 38 | Table 9.1 Specification of Clock Generation Circuit NOTE2 deleted |
| | | 39 | Figure 9.1 Clock Generation Circuit revised |
| | | 40 | Figure 9.2 CM0 Register NOTE2 revised |
| | | 42 | Figure 9.4 OCD Register NOTES 3, 4 revised |
| | | 43 | Figure 9.5 HRA0 Register NOTE2 revised |
| | | 45 | 9.1 Main Clock; "After reset, ..." → "During reset and after reset, ..." revised |
| | | 46 | 9.2.1 Low-Speed On-Chip Oscillator Clock; "The application ... to accommodate the frequency range." → "The application ... for the frequency change." |
| | | 47 | 9.3.2 CPU Clock; "When changing the clock source ... the OCD2 bit." deleted |
| | | 48 | 9.4.1 Normal Operating Mode; "... into three modes" → "... into four modes" revised Table 9.2 Setting and Mode of Clock Associated Bit revised |
| | | 49 | 9.4.1.1 High-Speed Mode, 9.4.1.2 Medium-Speed Mode; "Set the CM06 bit to "1" ... on-chip oscillator mode." deleted 9.4.1.3 High-Speed, Low-Speed On-Chip Oscillator Mode; "9.4.1.3 On-Chip Oscillator Mode" → "9.4.1.3 High-Speed, Low-Speed On-Chip Oscillator Mode" revised, "Set the CM06 bit to "1" ... high-speed and medium-speed." deleted |
| | | 52 | Figure 9.8 State Transition to Stop and Wait Modes; "Figure 9.8 State Transition to Stop and Wait Modes" → "Figure 9.8 State Transition of Power Control" revised Figure 9.9 State Transition in Normal Operating Mode deleted |
| | | 53 | 9.5.1 How to Use Oscillation Stop Detection Function; "• This function cannot ... is 2 MHz or below. ..." → "• This function cannot ... is below 2 MHz. ..." revised |
| | | 54 | Figure 9.9 Procedure of Switching Clock Source From Low-Speed On-Chip Oscillator to Main Clock revised |
| | | 55 | Figure 10.1 PRCR Register "00XXX000b" → "00h" revised |
| | | 68 | Figure 11.10 Judgement Circuit of Interrupts Priority Level NOTE1 deleted |
| | | 69 | Figure 11.11 INTEN and INT0F Registers; INT0F Register "XXXXX000b" → "00h" revised |

REVISION HISTORY

R8C/14 Group, R8C/15 Group Hardware

| Rev. | Date | Description | |
|------|--------------|-------------|--|
| | | Page | Summary |
| 2.00 | Jan 12, 2006 | 76 | 11.4 Address Match Interrupt; “... , do not use an address match interrupt in a user system.” → “... , do not set an address match interrupt (the registers of AIER, RMAD0, RMAD1 and the fixed vector tables) in a user system.” revised |
| | | 77 | Figure 11.19 AIER, RMAD0 to RMAD1 Registers; AIER Register revised |
| | | 79 | Figure 12.2 OFS and WDC Registers; • Option Function Select Register NOTE1 revised, NOTE2 added • Watchdog Timer Control Register NOTE1 deleted |
| | | 83 | Table 13.1 Functional Comparison of Timers; “Decrement” → “Increment” |
| | | 84 | Figure 13.1 Block Diagram of Timer X revised |
| | | 87 | Table 13.2 Specification of Timer Mode; • “INT1/CNTR0 Signal Pin Function” → “INT10/CNTR00, INT11/CNTR01 Pin Function” revised • “• When writing ... registers (the data is transferred to the counter when the following count source is input).” → “• When writing ... registers at the following count source input and the data is transferred to the counter at the second count source input and the count re-starts at the third count source input.” revised |
| | | 88 | Table 13.3 Specification of Pulse Output Mode; • “INT1/CNTR0 Signal Pin Function” → “INT10/CNTR00 Pin Function” revised • “• When writing ... registers (the data is transferred to the counter when the following count source is input).” → “• When writing ... registers at the following count source input and the data is transferred to the counter at the second count source input and the count re-starts at the third count source input.” revised • NOTE1 added |
| | | 90, 92, 95 | Table 13.4 Specification of Event Counter Mode, Table 13.5 Specification of Pulse Width Measurement Mode, Table 13.6 Specification of Pulse Period Measurement Mode; • “INT1/CNTR0 Signal Pin Function” → “INT10/CNTR00, INT11/CNTR01 Pin Function” revised • “• When writing ... registers (the data is transferred to the counter when the following count source is input).” → “• When writing ... registers at the following count source input and the data is transferred to the counter at the second count source input and the count re-starts at the third count source input.” revised |
| | | 98 | Figure 13.11 Block Diagram of Timer Z; “Peripheral Data Bus” → “Data Bus” revised |
| | | 101 | Figure 13.14 TZOC and PUM Registers; Timer Z Output Control Register “Stops counting” → “One-shot stops”, “Starts counting” → “One-shot starts” revised |

REVISION HISTORY

R8C/14 Group, R8C/15 Group Hardware

| Rev. | Date | Description | |
|------------|---|---------------|---|
| | | Page | Summary |
| 2.00 | Jan 12, 2006 | 103 | Table 13.7 Specification of Timer Mode; “• When writing ... registers (the data is transferred to the counter when the following count source is input) while the TZWC bit is set to “0” (writing to the reload register and counter simultaneously).” → “• When writing ... registers at the following count source input and the data is transferred to the counter at the second count source input and the count re-starts at the third count source input.” revised |
| | | 108, 112 | Table 13.9 Specification of Programmable One-Shot Generation Mode, Table 13.10 Specification of Programmable Wait One-Shot Generation Mode Specifications; Count Operation; “• When a count completes, ...” → “• When a count stops, ...” revised |
| | | 112 | Table 13.10 Specification of Programmable Wait One-Shot Generation Mode Specifications; NOTE1, 2, 3 revised |
| | | 116 | Figure 13.25 Block Diagram of CMP Waveform Output Unit revised |
| | | 123 | Table 13.12 Specification of Output Compare Mode NOTE1 revised |
| | | 124 | Figure 13.31 Operating Example of Timer C in Output Compare Mode revised |
| | | 127 | Figure 14.3 U0TB, U0RB and U0BRG Registers; U0TB and U0RB Registers revised, U0BRG Register NOTE3 added |
| | | 128 | Figure 14.4 U0MR and U0C0 Registers; U0C0 Register NOTE1 added |
| | | 136 | Table 14.5 Registers to Be Used and Settings in UART Mode; U0BRG: “-” → “0 to 7” revised |
| | | 169 | Figure 16.1 Block Diagram of A/D Converter “Vref” → “Vcom” revised |
| | | 170, 173, 175 | Figure 16.2 ADCON0 and ADCON1 Registers, Figure 16.4 ADCON0 and ADCON1 Registers in One-Shot Mode, Figure 16.5 ADCON0 and ADCON1 Registers in Repeat Mode; ADCON0 Register revised |
| | | 176 to 178 | Figure 16.6 Timing Diagram of A/D Conversion revised and 16.4 A/D Conversion Cycles to 16.6 Inflow Current Bypass Circuit added |
| | | 180, 181 | Figure 17.1 Configuration of Programmable I/O Ports (1), Figure 17.2 Configuration of Programmable I/O Ports (2); NOTE1 added |
| | | 182 | Figure 17.3 Configuration of Programmable I/O Ports (3) NOTE4 added |
| | | 184 | Figure 17.5 PD1, PD3 and PD4 Registers, Figure 17.6 P1, P3 and P4 Registers; NOTE1, 2 revised |
| 185 | Figure 17.7 PUR0 and PUR1 Registers revised | | |
| 186 to 189 | 17.4 Port setting added, Table 17.4 Port P1_0/ $\overline{KI0}$ /AN8/CMP0_0 Setting to Table 17.17 Port P4_5/ $\overline{INT0}$ Setting added | | |
| 191 | Table 18.1 Flash Memory Version Performance; Program and Erase Endurance: (Program area) → (Program ROM), (Data area) → (Data flash) revised | | |

REVISION HISTORY

R8C/14 Group, R8C/15 Group Hardware

| Rev. | Date | Description | |
|------|--------------|-------------|---|
| | | Page | Summary |
| 2.00 | Jan 12, 2006 | 193 | 18.2 Memory Map; “The user ROM ... area ... Block A and B.” → “The user ROM ... area (program ROM) ... Block A and B (data flash).” revised Figure 18.1 Flash Memory Block Diagram for R8C/14 Group revised |
| | | 194 | Figure 18.2 Flash Memory Block Diagram for R8C/15 Group revised |
| | | 196 | Figure 18.4 OFS Register; NOTE1 revised, NOTE2 added |
| | | 200 | Figure 18.5 FMR0 Register; NOTE6 revised |
| | | 201 | Figure 18.6 FMR1 and FMR4 Registers; FMR4 Register NOTE2 revised |
| | | 202 | Figure 18.7 Timing on Suspend Operation added |
| | | 203 | Figure 18.8 How to Set and Exit EW0 Mode and Figure 18.9 How to Set and Exit EW1 Mode revised |
| | | 208 | Figure 18.13 Block Erase Command (When Using Erase-Suspend Function) revised |
| | | 211 | Figure 18.14 Full Status Check and Handling Procedure for Each Error revised |
| | | 212 to 213 | 18.5 Standard Serial I/O Mode revised |
| | | 214 | Figure 18.15 Pin Connections for Standard Serial I/O Mode 3; Figure title revised |
| | | 215 | Figure 18.16 Pin Process in Standard Serial I/O Mode → Figure 18.16 Pin Process in Standard Serial I/O Mode 2 revised, Figure 18.17 Pin Process in Standard Serial I/O Mode 3 added |
| | | 219 | Table 19.4 Flash Memory (Program ROM) Electrical Characteristics; • NOTES 1 to 7 added • “Topr” = “Ambient temperature” |
| | | 220 | Table 19.5 Flash Memory (Data flash Block A, Block B) Electrical Characteristics; • revised • “Topr” = “Ambient temperature” |
| | | 221 | Figure 19.2 Time delay from Suspend Request until Erase Suspend revised |
| | | 222 | Table 19.8 Reset Circuit Electrical Characteristics (When Using Voltage Monitor 1 Reset) NOTE2 revised. Table 19.12 Timing Requirements of Clock Synchronous Serial I/O (SSU) with Chip Select revised. |
| | | 223 | Table 19.10 High-speed On-Chip Oscillator Circuit Electrical Characteristics revised |
| | | 224 | Figure 19.4 I/O Timing of Clock Synchronous Serial I/O (SSU) with Chip Select (Master) revised |
| | | 225 | Figure 19.5 I/O Timing of Clock Synchronous Serial I/O (SSU) with Chip Select (Slave) revised |
| | | 227 | Table 19.13 Electrical Characteristics (1) [VCC = 5V] revised |

REVISION HISTORY

R8C/14 Group, R8C/15 Group Hardware

| Rev. | Date | Description | |
|------|--|-------------|---|
| | | Page | Summary |
| 2.00 | Jan 12, 2006 | 228 | Table 19.14 Electrical Characteristics (2) [V _{cc} = 5V] NOTE1 deleted |
| | | 229 | Table 19.18 Serial Interface; "35" → "50", "80" → "50" |
| | | 231 | Table 19.20 Electrical Characteristics (3) [V _{CC} = 3V] revised |
| | | 232 | Table 19.21 Electrical Characteristics (4) [V _{cc} = 3V] NOTE1 deleted |
| | | 233 | Table 19.25 Serial Interface; "55" → "70", "160" → "80" |
| | | 239 | 20.3.1 Oscillation Stop Detection Function; "Since ... is 2MHz or below, .." → "Since ... is below 2 MHz, .." revised 20.3.2 Oscillation Circuit Constants added |
| | | 240 | 20.4.2 Precautions on Timer X; '• ... When writing "1" (count starts) to ... writing "1" to the TXS bit.' → '• ... "0" (count stops) can be ... after the TXS bit is set to "1".' revised |
| | | 241 | 20.4.3 Precautions on Timer Z; •• "In programmable ... "0" and the timer ..." → • "In programmable ... "0" (stops counting) or setting the TZOS bit in the TZOC register to "0" (stops one-shot), the timer ..." revised •• ... When writing "1" (count starts) to ... writing "1" to the TZS bit.' → '• ... "0" (count stops) can be ... after the TZS bit is set to "1".' revised |
| | | 246 | Table 20.2 Interrupt in EW1 Mode revised |
| | | 247 | 20.8.1.9 Interrupt Request Generation During Auto-erase Operation in EW1 Mode added |
| | | 249 | 21. Precaution for On-chip Debugger (2) revised, (4) added |
| | | 250 | Appendix 1. Package Dimensions; Package "PRDP0020BA-A" deleted |
| | | 251 | Appendix Figure 2.1 Connecting Example with M16C Flash Starter (M3A-0806); • NOTE1 revised • Pulled up added |
| | | 2.10 | Jan 19, 2006 |
| 247 | 20.8.1.9 Interrupt Request Generation during Auto-erase Operation in EW1 Mode; (b) revised | | |

R8C/14 Group, R8C/15 Group Hardware Manual

Publication Data : Rev.0.10 May 17, 2004
Rev.2.10 Jan 19, 2006

Published by : Sales Strategic Planning Div.
Renesas Technology Corp.

© 2006. Renesas Technology Corp., All rights reserved. Printed in Japan

R8C/14 Group, R8C/15 Group Hardware Manual



Renesas Electronics Corporation

1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0164-0210