

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.

M16C/29 Group

Hardware Manual

RENESAS MCU

M16C FAMILY / M16C/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 Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (<http://www.renesas.com>).

Notes regarding these materials

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

General Precautions in the Handling of MPU/MCU Products

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

1. Handling of Unused Pins

Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

- The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

The state of the product is undefined at the moment when power is supplied.

- The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.

In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.

In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

Access to reserved addresses is prohibited.

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

4. Clock Signals

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

- When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

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

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

How to Use This Manual

1. Purpose and Target Readers

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

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

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

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

The following documents apply to the M16C/29 Group. Make sure to refer to the latest versions of these documents. The newest versions of the documents listed may be obtained from the Renesas Technology Web site.

| Document Type | Description | Document Title | Document No. |
|--------------------------|--|--|----------------------|
| Hardware manual | Hardware specifications (pin assignments, memory maps, peripheral function specifications, electrical characteristics, timing charts) and operation description Note: Refer to the application notes for details on using peripheral functions. | M16C/29 Group Hardware Manual | This hardware manual |
| Software manual | Description of CPU instruction set | M16C/60, M16C/20, M16C/Tiny Series Software Manual | REJ09B0137 |
| Application note | Information on using peripheral functions and application examples Sample programs Information on writing programs in assembly language and C | Available from Renesas Technology Web site. | |
| Renesas technical update | Product specifications, updates on documents, etc. | | |

2. Notation of Numbers and Symbols

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

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

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

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

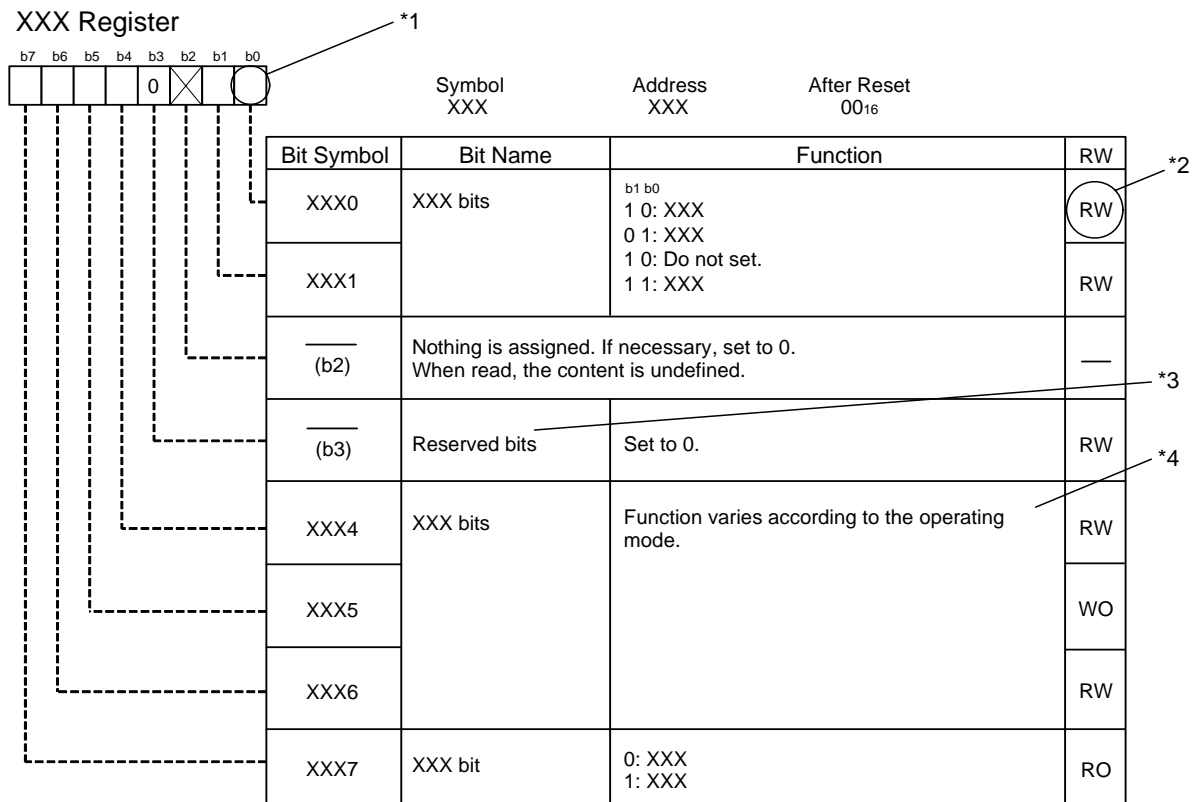
(2) Notation of Numbers

The indication “₂” is appended to numeric values given in binary format. However, nothing is appended to the values of single bits. The indication “₁₆” is appended to numeric values given in hexadecimal format. Nothing is appended to numeric values given in decimal format.

Examples Binary: 11₂
Hexadecimal: EFA0₁₆
Decimal: 1234

3. Register Notation

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



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

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

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

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

4. List of Abbreviations and Acronyms

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

All trademarks and registered trademarks are the property of their respective owners.
IEBus is a registered trademark of NEC Electronics Corporation.

Table of Contents

| | |
|--|-----|
| Quick Reference to Pages Classified by Address _____ | B-1 |
| 1. Overview _____ | 1 |
| 1.1 Features | 1 |
| 1.1.1 Applications | 1 |
| 1.1.2 Specifications | 2 |
| 1.2 Block Diagram | 4 |
| 1.3 Product List | 6 |
| 1.4 Pin Assignments | 12 |
| 1.5 Pin Description | 18 |
| 2. Central Processing Unit (CPU) _____ | 21 |
| 2.1 Data Registers (R0, R1, R2 and R3) | 21 |
| 2.2 Address Registers (A0 and A1) | 21 |
| 2.3 Frame Base Register (FB) | 22 |
| 2.4 Interrupt Table Register (INTB) | 22 |
| 2.5 Program Counter (PC) | 22 |
| 2.6 User Stack Pointer (USP) and Interrupt Stack Pointer (ISP) | 22 |
| 2.7 Static Base Register (SB) | 22 |
| 2.8 Flag Register (FLG) | 22 |
| 2.8.1 Carry Flag (C Flag) | 22 |
| 2.8.2 Debug Flag (D Flag) | 22 |
| 2.8.3 Zero Flag (Z Flag) | 22 |
| 2.8.4 Sign Flag (S Flag) | 22 |
| 2.8.5 Register Bank Select Flag (B Flag) | 22 |
| 2.8.6 Overflow Flag (O Flag) | 22 |
| 2.8.7 Interrupt Enable Flag (I Flag) | 22 |
| 2.8.8 Stack Pointer Select Flag (U Flag) | 22 |
| 2.8.9 Processor Interrupt Priority Level (IPL) | 22 |
| 2.8.10 Reserved Area | 22 |
| 3. Memory _____ | 23 |
| 4. Special Function Registers (SFRs) _____ | 24 |

| | |
|---|----|
| 5. Resets | 35 |
| 5.1 Hardware Reset | 35 |
| 5.1.1 Hardware Reset 1 | 35 |
| 5.1.2 Brown-Out Detection Reset (Hardware Reset 2) | 35 |
| 5.2 Software Reset | 36 |
| 5.3 Watchdog Timer Reset | 36 |
| 5.4 Oscillation Stop Detection Reset | 36 |
| 5.5 Voltage Detection Circuit | 38 |
| 5.5.1 Low Voltage Detection Interrupt | 41 |
| 5.5.2. Limitations on Stop Mode | 43 |
| 5.5.3. Limitations on WAIT Instruction | 43 |
| 6. Processor Mode | 44 |
| 7. Clock Generation Circuit | 47 |
| 7.1 Main Clock | 54 |
| 7.2 Sub Clock | 55 |
| 7.3 On-chip Oscillator Clock | 56 |
| 7.4 PLL Clock | 56 |
| 7.5 CPU Clock and Peripheral Function Clock | 58 |
| 7.5.1 CPU Clock | 58 |
| 7.5.2 Peripheral Function Clock(f1, f2, f8, f32, f1SIO, f2SIO, f8SIO, f32SIO, fAD, fC32, fCAN0) | 58 |
| 7.5.3 ClockOutput Function | 58 |
| 7.6 Power Control | 59 |
| 7.6.1 Normal Operation Mode | 59 |
| 7.6.2 Wait Mode | 60 |
| 7.6.3 Stop Mode | 62 |
| 7.7 System Clock Protective Function | 66 |
| 7.8 Oscillation Stop and Re-oscillation Detect Function | 66 |
| 7.8.1 Operation When CM27 bit = 0 (Oscillation Stop Detection Reset) | 67 |
| 7.8.2 Operation When CM27 bit = 1 (Oscillation Stop and Re-oscillation Detect Interrupt) | 67 |
| 7.8.3 How to Use Oscillation Stop and Re-oscillation Detect Function | 68 |
| 8. Protection | 69 |

| | |
|---|-----|
| 9. Interrupts | 70 |
| 9.1 Type of Interrupts | 70 |
| 9.1.1 Software Interrupts | 71 |
| 9.1.2 Hardware Interrupts | 72 |
| 9.2 Interrupts and Interrupt Vector | 73 |
| 9.2.1 Fixed Vector Tables | 73 |
| 9.2.2 Relocatable Vector Tables | 74 |
| 9.3 Interrupt Control | 75 |
| 9.3.1 I Flag | 78 |
| 9.3.2 IR Bit | 78 |
| 9.3.3 ILVL2 to ILVL0 Bits and IPL | 78 |
| 9.4 Interrupt Sequence | 79 |
| 9.4.1 Interrupt Response Time | 80 |
| 9.4.2 Variation of IPL when Interrupt Request is Accepted | 80 |
| 9.4.3 Saving Registers | 81 |
| 9.4.4 Returning from an Interrupt Routine | 83 |
| 9.5 Interrupt Priority | 83 |
| 9.5.1 Interrupt Priority Resolution Circuit | 83 |
| 9.6 $\overline{\text{INT}}$ Interrupt | 85 |
| 9.7 $\overline{\text{NMI}}$ Interrupt | 86 |
| 9.8 Key Input Interrupt | 86 |
| 9.9 CAN0 Wake-up Interrupt | 87 |
| 9.10 Address Match Interrupt | 87 |
| 10. Watchdog Timer | 89 |
| 10.1 Count Source Protective Mode | 90 |
| 11. DMAC | 91 |
| 11.1 Transfer Cycles | 96 |
| 11.1.1 Effect of Source and Destination Addresses | 96 |
| 11.1.2 Effect of Software Wait | 96 |
| 11.2. DMA Transfer Cycles | 98 |
| 11.3 DMA Enable | 99 |
| 11.4 DMA Request | 99 |
| 11.5 Channel Priority and DMA Transfer Timing | 100 |

| | |
|--|-----|
| 12. Timers | 101 |
| 12.1 Timer A | 103 |
| 12.1.1 Timer Mode | 106 |
| 12.1.2 Event Counter Mode | 107 |
| 12.1.3 One-shot Timer Mode | 112 |
| 12.1.4 Pulse Width Modulation (PWM) Mode | 114 |
| 12.2 Timer B | 117 |
| 12.2.1 Timer Mode | 119 |
| 12.2.2 Event Counter Mode | 120 |
| 12.2.3 Pulse Period and Pulse Width Measurement Mode | 121 |
| 12.2.4 A/D Trigger Mode | 123 |
| 12.3 Three-phase Motor Control Timer Function | 125 |
| 12.3.1 Position-Data-Retain Function | 136 |
| 12.3.2 Three-phase/Port Output Switch Function | 138 |
| 13. Timer S | 140 |
| 13.1 Base Timer | 151 |
| 13.1.1 Base Timer Reset Register(G1BTRR) | 155 |
| 13.2 Interrupt Operation | 156 |
| 13.3 DMA Support | 156 |
| 13.4 Time Measurement Function | 157 |
| 13.5 Waveform Generating Function | 161 |
| 13.5.1 Single-Phase Waveform Output Mode | 162 |
| 13.5.2 Phase-Delayed Waveform Output Mode | 164 |
| 13.5.3 Set/Reset Waveform Output (SR Waveform Output) Mode | 166 |
| 13.6 I/O Port Function Select | 168 |
| 13.6.1 INPC17 Alternate Input Pin Selection | 169 |
| 13.6.2 Digital Debounce Function for Pin P17/INT5/INPC17 | 169 |
| 14. Serial I/O | 170 |
| 14.1 UARTi (i=0 to 2) | 170 |
| 14.1.1 Clock Synchronous serial I/O Mode | 180 |
| 14.1.2 Clock Asynchronous Serial I/O (UART) Mode | 188 |
| 14.1.3 Special Mode 1 (I ² C bus mode)(UART2) | 196 |
| 14.1.4 Special Mode 2 (UART2) | 206 |
| 14.1.5 Special Mode 3 (IEBus mode)(UART2) | 210 |
| 14.1.6 Special Mode 4 (SIM Mode) (UART2) | 212 |

| | |
|--|------------|
| 14.2 SI/O3 and SI/O4 | 217 |
| 14.2.2 CLK Polarity Selection | 220 |
| 14.2.1 SI/Oi Operation Timing | 220 |
| 14.2.3 Functions for Setting an SOUTi Initial Value | 221 |
| 15. A/D Converter | 222 |
| 15.1 Operating Modes | 228 |
| 15.1.1 One-Shot Mode | 228 |
| 15.1.2 Repeat mode | 230 |
| 15.1.3 Single Sweep Mode | 232 |
| 15.1.4 Repeat Sweep Mode 0 | 234 |
| 15.1.5 Repeat Sweep Mode 1 | 236 |
| 15.1.6 Simultaneous Sample Sweep Mode | 238 |
| 15.1.7 Delayed Trigger Mode 0 | 241 |
| 15.1.8 Delayed Trigger Mode 1 | 247 |
| 15.2 Resolution Select Function | 253 |
| 15.3 Sample and Hold | 253 |
| 15.4 Power Consumption Reducing Function | 253 |
| 15.5 Output Impedance of Sensor under A/D Conversion | 254 |
| 16. Multi-master I²C bus Interface | 255 |
| 16.1 I ² C0 Data Shift Register (S00 register) | 264 |
| 16.2 I ² C0 Address Register (S0D0 register) | 264 |
| 16.3 I ² C0 Clock Control Register (S20 register) | 265 |
| 16.3.1 Bits 0 to 4: SCL Frequency Control Bits (CCR0–CCR4) | 265 |
| 16.3.2 Bit 5: SCL Mode Specification Bit (FAST MODE) | 265 |
| 16.3.3 Bit 6: ACK Bit (ACKBIT) | 265 |
| 16.3.4 Bit 7: ACK Clock Bit (ACK-CLK) | 265 |
| 16.4 I ² C0 Control Register 0 (S1D0) | 267 |
| 16.4.1 Bits 0 to 2: Bit Counter (BC0–BC2) | 267 |
| 16.4.2 Bit 3: I ² C Interface Enable Bit (ES0) | 267 |
| 16.4.3 Bit 4: Data Format Select Bit (ALS) | 267 |
| 16.4.4 Bit 6: I ² C bus Interface Reset Bit (IHR) | 267 |
| 16.4.5 Bit 7: I ² C bus Interface Pin Input Level Select Bit (TISS) | 268 |
| 16.5 I ² C0 Status Register (S10 register) | 269 |
| 16.5.1 Bit 0: Last Receive Bit (LRB) | 269 |
| 16.5.2 Bit 1: General Call Detection Flag (ADR0) | 269 |

| | | |
|------------|---|------------|
| 16.5.3 | Bit 2: Slave Address Comparison Flag (AAS) | 269 |
| 16.5.4 | Bit 3: Arbitration Lost Detection Flag (AL) | 269 |
| 16.5.5 | Bit 4: I ² C bus Interface Interrupt Request Bit (PIN) | 270 |
| 16.5.6 | Bit 5: Bus Busy Flag (BB) | 270 |
| 16.5.7 | Bit 6: Communication Mode Select Bit (Transfer Direction Select Bit: TRX) | 271 |
| 16.5.8 | Bit 7: Communication mode select bit (master/slave select bit: MST) | 271 |
| 16.6 | I ² C0 Control Register 1 (S3D0 register) | 272 |
| 16.6.1 | Bit 0 : Interrupt Enable Bit by STOP Condition (SIM) | 272 |
| 16.6.2 | Bit 1: Interrupt Enable Bit at the Completion of Data Receive (WIT) | 272 |
| 16.6.3 | Bits 2,3 : Port Function Select Bits PED, PEC | 273 |
| 16.6.4 | Bits 4,5 : SDA/SCL Logic Output Value Monitor Bits SDAM/SCLM | 274 |
| 16.6.5 | Bits 6,7 : I ² C System Clock Select Bits ICK0, ICK1 | 274 |
| 16.6.6 | Address Receive in STOP/WAIT Mode | 274 |
| 16.7 | I ² C0 Control Register 2 (S4D0 Register) | 275 |
| 16.7.1 | Bit0: Time-Out Detection Function Enable Bit (TOE) | 276 |
| 16.7.2 | Bit1: Time-Out Detection Flag (TOF) | 276 |
| 16.7.3 | Bit2: Time-Out Detection Period Select Bit (TOSEL) | 276 |
| 16.7.4 | Bits 3,4,5: I ² C System Clock Select Bits (ICK2-4) | 276 |
| 16.7.5 | Bit7: STOP Condition Detection Interrupt Request Bit (SCPIN) | 276 |
| 16.8 | I ² C0 START/STOP Condition Control Register (S2D0 Register) | 277 |
| 16.8.1 | Bit0-Bit4: START/STOP Condition Setting Bits (SSC0-SSC4) | 277 |
| 16.8.2 | Bit5: SCL/SDA Interrupt Pin Polarity Select Bit (SIP) | 277 |
| 16.8.3 | Bit6 : SCL/SDA Interrupt Pin Select Bit (SIS) | 277 |
| 16.8.4 | Bit7: START/STOP Condition Generation Select Bit (STSPSEL) | 277 |
| 16.9 | START Condition Generation Method | 278 |
| 16.10 | START Condition Duplicate Protect Function | 279 |
| 16.11 | STOP Condition Generation Method | 279 |
| 16.12 | START/STOP Condition Detect Operation | 281 |
| 16.13 | Address Data Communication | 282 |
| 16.13.1 | Example of Master Transmit | 282 |
| 16.13.2 | Example of Slave Receive | 283 |
| 16.14 | Precautions | 284 |
| 17. | CAN Module | 287 |
| 17.1 | CAN Module-Related Registers | 288 |
| 17.1.1 | CAN0 Message Box | 289 |
| 17.1.2 | Acceptance Mask Registers | 291 |
| 17.1.3 | CAN SFR Registers | 292 |

| | |
|---|------------|
| 17.2 Operating Modes | 300 |
| 17.2.1 CAN Reset/Initialization Mode | 300 |
| 17.2.2 CAN Operating Mode | 301 |
| 17.2.3 CAN Sleep Mode | 301 |
| 17.2.4 CAN Interface Sleep Mode | 302 |
| 17.2.5 Bus Off State | 302 |
| 17.3 Configuration of the CAN Module System Clock | 303 |
| 17.3.1 Bit Timing Configuration | 303 |
| 17.3.2 Bit-rate | 304 |
| 17.4 Acceptance Filtering Function and Masking Function | 305 |
| 17.5 Acceptance Filter Support Unit (ASU) | 306 |
| 17.6 BasicCAN Mode | 307 |
| 17.7 Return from Bus off Function | 308 |
| 17.8 Time Stamp Counter and Time Stamp Function | 308 |
| 17.9 Listen-Only Mode | 308 |
| 17.10 Reception and Transmission | 309 |
| 17.10.1 Reception | 310 |
| 17.10.2 Transmission | 311 |
| 17.11 CAN Interrupts | 312 |
| 18. CRC Calculation Circuit _____ | 313 |
| 18.1 CRC Snoop | 313 |
| 19. Programmable I/O Ports _____ | 316 |
| 19.1 Port Pi Direction Register (PDi Register, i = 0 to 3, 6 to 10) | 316 |
| 19.2 Port Pi Register (Pi Register, i = 0 to 3, 6 to 10) | 316 |
| 19.3 Pull-up Control Register 0 to 2 (PUR0 to PUR2 Registers) | 316 |
| 19.4 Port Control Register (PCR Register) | 316 |
| 19.5 Pin Assignment Control Register (PACR) | 317 |
| 19.6 Digital Debounce Function | 317 |
| 20. Flash Memory Version _____ | 330 |
| 20.1 Flash Memory Performance | 330 |
| 20.1.1 Boot Mode | 331 |
| 20.2 Memory Map | 332 |
| 20.3 Functions To Prevent Flash Memory from Rewriting | 335 |
| 20.3.1 ROM Code Protect Function | 335 |
| 20.3.2 ID Code Check Function | 335 |

| | |
|--|-----|
| 20.4 CPU Rewrite Mode | 337 |
| 20.4.1 EW Mode 0 | 338 |
| 20.4.2 EW Mode 1 | 338 |
| 20.5 Register Description | 339 |
| 20.5.1 Flash Memory Control Register 0 (FMR0) | 339 |
| 20.5.2 Flash Memory Control Register 1 (FMR1) | 340 |
| 20.5.3 Flash Memory Control Register 4 (FMR4) | 340 |
| 20.6 Precautions in CPU Rewrite Mode | 345 |
| 20.6.1 Operation Speed | 345 |
| 20.6.2 Prohibited Instructions | 345 |
| 20.6.3 Interrupts | 345 |
| 20.6.4 How to Access | 345 |
| 20.6.5 Writing in the User ROM Area | 345 |
| 20.6.6 DMA Transfer | 346 |
| 20.6.7 Writing Command and Data | 346 |
| 20.6.8 Wait Mode | 346 |
| 20.6.9 Stop Mode | 346 |
| 20.6.10 Low Power Consumption Mode and On-Chip Oscillator-Low Power Consumption Mode ... | 346 |
| 20.7 Software Commands | 347 |
| 20.7.1 Read Array Command (FF16)..... | 347 |
| 20.7.2 Read Status Register Command (7016)..... | 347 |
| 20.7.3 Clear Status Register Command (5016)..... | 347 |
| 20.7.4 Program Command (4016) | 348 |
| 20.7.5 Block Erase | 349 |
| 20.8 Status Register | 351 |
| 20.8.1 Sequence Status (SR7 and FMR00 Bits)..... | 351 |
| 20.8.2 Erase Status (SR5 and FMR07 Bits) | 351 |
| 20.8.3 Program Status (SR4 and FMR06 Bits) | 351 |
| 20.8.4 Full Status Check | 352 |
| 20.9 Standard Serial I/O Mode | 354 |
| 20.9.1 ID Code Check Function | 354 |
| 20.9.2 Example of Circuit Application in Standard Serial I/O Mode | 358 |
| 20.10 Parallel I/O Mode | 360 |
| 20.10.1 ROM Code Protect Function | 360 |
| 20.11 CAN I/O Mode | 361 |
| 20.11.1 ID code check function | 361 |
| 20.11.2 Example of Circuit Application in CAN I/O Mode | 365 |

| | |
|---|-----|
| 21. Electrical Characteristics | 366 |
| 21.1 Normal version | 366 |
| 21.2 T version | 387 |
| 21.3 V Version | 408 |
| 22. Usage Notes | 421 |
| 22.1 SFRs | 421 |
| 22.1.1 For 80-Pin Package | 421 |
| 22.1.2 For 64-Pin Package | 421 |
| 22.1.3 Register Setting | 421 |
| 22.2 Clock Generation Circuit | 422 |
| 22.2.1 PLL Frequency Synthesizer | 422 |
| 22.2.2 Power Control | 423 |
| 22.3 Protection | 425 |
| 22.4 Interrupts | 426 |
| 22.4.1 Reading Address 00000 ₁₆ | 426 |
| 22.4.2 Setting the SP | 426 |
| 22.4.3 $\overline{\text{NMI}}$ Interrupt | 426 |
| 22.4.4 Changing the Interrupt Generate Factor | 426 |
| 22.4.5 $\overline{\text{INT}}$ Interrupt | 427 |
| 22.4.6 Rewrite the Interrupt Control Register | 428 |
| 22.4.7 Watchdog Timer Interrupt | 428 |
| 22.5 DMAC | 429 |
| 22.5.1 Write to DMAE Bit in DMiCON Register | 429 |
| 22.6 Timers | 430 |
| 22.6.1 Timer A | 430 |
| 22.6.2 Timer B | 433 |
| 22.6.3 Three-phase Motor Control Timer Function | 434 |
| 22.7 Timer S | 435 |
| 22.7.1 Rewrite the G1IR Register | 435 |
| 22.7.2 Rewrite the ICOCiIC Register | 436 |
| 22.7.3 Waveform Generating Function | 436 |
| 22.7.4 IC/OC Base Timer Interrupt | 436 |
| 22.8 Serial I/O | 437 |
| 22.8.1 Clock-Synchronous Serial I/O | 437 |
| 22.8.2 UART Mode | 438 |
| 22.8.3 SI/O3, SI/O4 | 438 |

| | |
|--|-----|
| 22.9 A/D Converter | 439 |
| 22.10 Multi-Master I ² C bus Interface | 441 |
| 22.10.1 Writing to the S00 Register | 441 |
| 22.10.2 AL Flag | 441 |
| 22.11 CAN Module | 442 |
| 22.11.1 Reading C0STR Register | 442 |
| 22.11.2 CAN Transceiver in Boot Mode | 444 |
| 22.12 Programmable I/O Ports | 445 |
| 22.13 Electric Characteristic Differences Between Mask ROM | 446 |
| 22.14 Mask ROM Version | 447 |
| 22.14.1 Internal ROM Area | 447 |
| 22.14.2 Reserved Bit | 447 |
| 22.15 Flash Memory Version | 448 |
| 22.15.1 Functions to Inhibit Rewriting Flash Memory Rewrite | 448 |
| 22.15.2 Stop Mode | 448 |
| 22.15.3 Wait Mode | 448 |
| 22.15.4 Low PowerDissipation Mode, On-Chip Oscillator Low Power Dissipation Mode .. | 448 |
| 22.15.5 Writing Command and Data | 448 |
| 22.15.6 Program Command | 448 |
| 22.15.7 Operation Speed | 448 |
| 22.15.8 Instructions Inhibited Against Use | 448 |
| 22.15.9 Interrupts | 449 |
| 22.15.10 How to Access | 449 |
| 22.15.11 Writing in the User ROM Area | 449 |
| 22.15.12 DMA Transfer | 449 |
| 22.15.13 Regarding Programming/Erase Times and Execution Time | 449 |
| 22.15.14 Definition of Programming/Erase Times | 450 |
| 22.15.15 Flash Memory Version Electrical Characteristics 10,000 E/W cycle products (Normal: U7, U9; T-ver./V-ver.: U7) | 450 |
| 22.15.16 Boot Mode | 450 |
| 22.16 Noise | 451 |
| 22.17 Instruction for a Device Use | 452 |

| | |
|--|-----|
| Appendix 1. Package Dimensions _____ | 453 |
| Appendix 2. Functional Comparison _____ | 454 |
| Appendix 2.1 Difference between M16C/28 Group and M16C/29 Group (Normal-ver.) | 454 |
| Appendix 2.2 Difference between M16C/28 and M16C/29 Group (T-ver./V-ver.) | 455 |
| Register Index _____ | 456 |

Quick Reference to Pages Classified by Address

| Address | Register | Symbol | Page |
|--------------------|--|--------|------|
| 0000 ₁₆ | | | |
| 0001 ₁₆ | | | |
| 0002 ₁₆ | | | |
| 0003 ₁₆ | | | |
| 0004 ₁₆ | Processor mode register 0 | PM0 | 44 |
| 0005 ₁₆ | Processor mode register 1 | PM1 | 44 |
| 0006 ₁₆ | System clock control register 0 | CM0 | 49 |
| 0007 ₁₆ | System clock control register 1 | CM1 | 50 |
| 0008 ₁₆ | | | |
| 0009 ₁₆ | Address match interrupt enable register | AIER | 88 |
| 000A ₁₆ | Protect register | PRCR | 69 |
| 000B ₁₆ | | | |
| 000C ₁₆ | Oscillation stop detection register | CM2 | 51 |
| 000D ₁₆ | | | |
| 000E ₁₆ | Watchdog timer start register | WDTS | 90 |
| 000F ₁₆ | Watchdog timer control register | WDC | 90 |
| 0010 ₁₆ | | | |
| 0011 ₁₆ | Address match interrupt register 0 | RMAD0 | 88 |
| 0012 ₁₆ | | | |
| 0013 ₁₆ | | | |
| 0014 ₁₆ | | | |
| 0015 ₁₆ | Address match interrupt register 1 | RMAD1 | 88 |
| 0016 ₁₆ | | | |
| 0017 ₁₆ | | | |
| 0018 ₁₆ | | | |
| 0019 ₁₆ | Voltage detection register 1 | VCR1 | 41 |
| 001A ₁₆ | Voltage detection register 2 | VCR2 | 41 |
| 001B ₁₆ | | | |
| 001C ₁₆ | PLL control register 0 | PLC0 | 53 |
| 001D ₁₆ | | | |
| 001E ₁₆ | Processor mode register 2 | PM2 | 52 |
| 001F ₁₆ | Low voltage detection interrupt register | D4INT | 42 |
| 0020 ₁₆ | | | |
| 0021 ₁₆ | DMA0 source pointer | SAR0 | 95 |
| 0022 ₁₆ | | | |
| 0023 ₁₆ | | | |
| 0024 ₁₆ | | | |
| 0025 ₁₆ | DMA0 destination pointer | DAR0 | 95 |
| 0026 ₁₆ | | | |
| 0027 ₁₆ | | | |
| 0028 ₁₆ | DMA0 transfer counter | TCR0 | 95 |
| 0029 ₁₆ | | | |
| 002A ₁₆ | | | |
| 002B ₁₆ | | | |
| 002C ₁₆ | DMA0 control register | DM0CON | 94 |
| 002D ₁₆ | | | |
| 002E ₁₆ | | | |
| 002F ₁₆ | | | |
| 0030 ₁₆ | | | |
| 0031 ₁₆ | DMA1 source pointer | SAR1 | 95 |
| 0032 ₁₆ | | | |
| 0033 ₁₆ | | | |
| 0034 ₁₆ | | | |
| 0035 ₁₆ | DMA1 destination pointer | DAR1 | 95 |
| 0036 ₁₆ | | | |
| 0037 ₁₆ | | | |
| 0038 ₁₆ | DMA1 transfer counter | TCR1 | 95 |
| 0039 ₁₆ | | | |
| 003A ₁₆ | | | |
| 003B ₁₆ | | | |
| 003C ₁₆ | DMA1 control register | DM1CON | 94 |
| 003D ₁₆ | | | |
| 003E ₁₆ | | | |
| 003F ₁₆ | | | |

Note: The blank areas are reserved and cannot be accessed by users.

| Address | Register | Symbol | Page |
|--------------------|--|------------------|----------|
| 0040 ₁₆ | | | |
| 0041 ₁₆ | CAN0 wakeup interrupt control register | C01WKIC | 76 |
| 0042 ₁₆ | CAN0 successful reception interrupt control register | C0RECIC | 76 |
| 0043 ₁₆ | CAN0 successful transmission interrupt control register | C0TRMIC | 76 |
| 0044 ₁₆ | INT3 interrupt control register | INT3IC | 76 |
| 0045 ₁₆ | IC/OC 0 interrupt control register | ICOC0IC | 76 |
| 0046 ₁₆ | IC/OC 1 interrupt control register, I ² C bus interface interrupt control register | ICOC1IC IICIC | 76 76 |
| 0047 ₁₆ | IC/OC base timer interrupt control register, SCLSDA interrupt control register | BTIC SCLDAIC | 76 76 |
| 0048 ₁₆ | SI/O4 interrupt control register, INT5 interrupt control register | S4IC INT5IC | 76 76 |
| 0049 ₁₆ | SI/O3 interrupt control register, INT4 interrupt control register | S3IC INT4IC | 76 76 |
| 004A ₁₆ | UART2 Bus collision detection interrupt control register | BCNIC | 76 |
| 004B ₁₆ | DMA0 interrupt control register | DM0IC | 76 |
| 004C ₁₆ | DMA1 interrupt control register | DM1IC | 76 |
| 004D ₁₆ | CAN0 error interrupt control register | C0ERRIC | 76 |
| 004E ₁₆ | A/D conversion interrupt control register Key input interrupt control register | ADIC KUPIC | 76 76 |
| 004F ₁₆ | UART2 transmit interrupt control register | S2TIC | 76 |
| 0050 ₁₆ | UART2 receive interrupt control register | S2RIC | 76 |
| 0051 ₁₆ | UART0 transmit interrupt control register | S0TIC | 76 |
| 0052 ₁₆ | UART0 receive interrupt control register | S0RIC | 76 |
| 0053 ₁₆ | UART1 transmit interrupt control register | S1TIC | 76 |
| 0054 ₁₆ | UART1 receive interrupt control register | S1RIC | 76 |
| 0055 ₁₆ | Timer A0 interrupt control register | TA0IC | 76 |
| 0056 ₁₆ | Timer A1 interrupt control register | TA1IC | 76 |
| 0057 ₁₆ | Timer A2 interrupt control register | TA2IC | 76 |
| 0058 ₁₆ | Timer A3 interrupt control register | TA3IC | 76 |
| 0059 ₁₆ | Timer A4 interrupt control register | TA4IC | 76 |
| 005A ₁₆ | Timer B0 interrupt control register | TB0IC | 76 |
| 005B ₁₆ | Timer B1 interrupt control register | TB1IC | 76 |
| 005C ₁₆ | Timer B2 interrupt control register | TB2IC | 76 |
| 005D ₁₆ | INT0 interrupt control register | INT0IC | 76 |
| 005E ₁₆ | INT1 interrupt control register | INT1IC | 76 |
| 005F ₁₆ | INT2 interrupt control register | INT2IC | 76 |
| 0060 ₁₆ | | | |
| 0061 ₁₆ | | | |
| 0062 ₁₆ | CAN0 message box 0: Identifier/DLC | | 289 |
| 0063 ₁₆ | | | |
| 0064 ₁₆ | | | |
| 0065 ₁₆ | | | |
| 0066 ₁₆ | | | |
| 0067 ₁₆ | CAN 0 message box 0: Data field | | 289 |
| 0068 ₁₆ | | | |
| 0069 ₁₆ | | | |
| 006A ₁₆ | | | |
| 006B ₁₆ | | | |
| 006C ₁₆ | | | |
| 006D ₁₆ | | | |
| 006E ₁₆ | CAN0 message box 0: Time stamp | | 289 |
| 006F ₁₆ | | | |
| 0070 ₁₆ | | | |
| 0071 ₁₆ | CAN0 message box 1: Identifier/DLC | | 289 |
| 0072 ₁₆ | | | |
| 0073 ₁₆ | | | |
| 0074 ₁₆ | | | |
| 0075 ₁₆ | | | |
| 0076 ₁₆ | CAN 0 message box 1: Data field | | 289 |
| 0077 ₁₆ | | | |
| 0078 ₁₆ | | | |
| 0079 ₁₆ | | | |
| 007A ₁₆ | | | |
| 007B ₁₆ | | | |
| 007C ₁₆ | | | |
| 007D ₁₆ | | | |
| 007E ₁₆ | CAN0 message box 1: Time stamp | | 289 |
| 007F ₁₆ | | | |

Quick Reference to Pages Classified by Address

| Address | Register | Symbol | Page | Address | Register | Symbol | Page |
|--|------------------------------------|--------|------|--|------------------------------------|--------|------|
| 0080 ₁₆ 0081 ₁₆ 0082 ₁₆ 0083 ₁₆ 0084 ₁₆ 0085 ₁₆ | CAN0 message box 2: Identifier/DLC | | 289 | 00C0 ₁₆ 00C1 ₁₆ 00C2 ₁₆ 00C3 ₁₆ 00C4 ₁₆ 00C5 ₁₆ | CAN0 message box 6: Identifier/DLC | | 289 |
| 0086 ₁₆ 0087 ₁₆ 0088 ₁₆ 0089 ₁₆ 008A ₁₆ 008B ₁₆ 008C ₁₆ 008D ₁₆ | CAN0 message box 2: Data field | | 289 | 00C6 ₁₆ 00C7 ₁₆ 00C8 ₁₆ 00C9 ₁₆ 00CA ₁₆ 00CB ₁₆ 00CC ₁₆ 00CD ₁₆ | CAN0 message box 6: Data field | | 289 |
| 008E ₁₆ 008F ₁₆ | CAN0 message box 2: time stamp | | 289 | 00CE ₁₆ 00CF ₁₆ | CAN0 message box 6: time stamp | | 289 |
| 0090 ₁₆ 0091 ₁₆ 0092 ₁₆ 0093 ₁₆ 0094 ₁₆ 0095 ₁₆ | CAN0 message box 3: Identifier/DLC | | 289 | 00D0 ₁₆ 00D1 ₁₆ 00D2 ₁₆ 00D3 ₁₆ 00D4 ₁₆ 00D5 ₁₆ | CAN0 message box 7: Identifier/DLC | | 289 |
| 0096 ₁₆ 0097 ₁₆ 0098 ₁₆ 0099 ₁₆ 009A ₁₆ 009B ₁₆ 009C ₁₆ 009D ₁₆ | CAN0 message box 3: Data field | | 289 | 00D6 ₁₆ 00D7 ₁₆ 00D8 ₁₆ 00D9 ₁₆ 00DA ₁₆ 00DB ₁₆ 00DC ₁₆ 00DD ₁₆ | CAN0 message box 7: Data field | | 289 |
| 009E ₁₆ 009F ₁₆ | CAN0 message box 3: time stamp | | 289 | 00DE ₁₆ 00DF ₁₆ | CAN0 message box 7: time stamp | | 289 |
| 00A0 ₁₆ 00A1 ₁₆ 00A2 ₁₆ 00A3 ₁₆ 00A4 ₁₆ 00A5 ₁₆ | CAN0 message box 4: Identifier/DLC | | 289 | 00E0 ₁₆ 00E1 ₁₆ 00E2 ₁₆ 00E3 ₁₆ 00E4 ₁₆ 00E5 ₁₆ | CAN0 message box 8: Identifier/DLC | | 289 |
| 00A6 ₁₆ 00A7 ₁₆ 00A8 ₁₆ 00A9 ₁₆ 00AA ₁₆ 00AB ₁₆ 00AC ₁₆ 00AD ₁₆ | CAN0 message box 4: Data field | | 289 | 00E6 ₁₆ 00E7 ₁₆ 00E8 ₁₆ 00E9 ₁₆ 00EA ₁₆ 00EB ₁₆ 00EC ₁₆ 00ED ₁₆ | CAN0 message box 8: Data field | | 289 |
| 00AE ₁₆ 00AF ₁₆ | CAN0 message box 4: time stamp | | 289 | 00EE ₁₆ 00EF ₁₆ | CAN0 message box 8: time stamp | | 289 |
| 00B0 ₁₆ 00B1 ₁₆ 00B2 ₁₆ 00B3 ₁₆ 00B4 ₁₆ 00B5 ₁₆ | CAN0 message box 5: Identifier/DLC | | 289 | 00F0 ₁₆ 00F1 ₁₆ 00F2 ₁₆ 00F3 ₁₆ 00F4 ₁₆ 00F5 ₁₆ | CAN0 message box 9: Identifier/DLC | | 289 |
| 00B6 ₁₆ 00B7 ₁₆ 00B8 ₁₆ 00B9 ₁₆ 00BA ₁₆ 00BB ₁₆ 00BC ₁₆ 00BD ₁₆ | CAN0 message box 5: Data field | | 289 | 00F6 ₁₆ 00F7 ₁₆ 00F8 ₁₆ 00F9 ₁₆ 00FA ₁₆ 00FB ₁₆ 00FC ₁₆ 00FD ₁₆ | CAN0 message box 9: Data field | | 289 |
| 00BE ₁₆ 00BF ₁₆ | CAN0 message box 5: time stamp | | 289 | 00FE ₁₆ 00FF ₁₆ | CAN0 message box 9: time stamp | | 289 |

Note: The blank areas are reserved and cannot be accessed by users.

Quick Reference to Pages Classified by Address

| Address | Register | Symbol | Page | Address | Register | Symbol | Page |
|--|-------------------------------------|--------|------|--|-------------------------------------|--------|------|
| 0100 ₁₆ 0101 ₁₆ 0102 ₁₆ 0103 ₁₆ 0104 ₁₆ 0105 ₁₆ | CAN0 message box 10: Identifier/DLC | | 289 | 0140 ₁₆ 0141 ₁₆ 0142 ₁₆ 0143 ₁₆ 0144 ₁₆ 0145 ₁₆ | CAN0 message box 14: Identifier/DLC | | 289 |
| 0106 ₁₆ 0107 ₁₆ 0108 ₁₆ 0109 ₁₆ 010A ₁₆ 010B ₁₆ 010C ₁₆ 010D ₁₆ | CAN0 message box 10: Data field | | 289 | 0146 ₁₆ 0147 ₁₆ 0148 ₁₆ 0149 ₁₆ 014A ₁₆ 014B ₁₆ 014C ₁₆ 014D ₁₆ | CAN0 message box 14: Data field | | 289 |
| 010E ₁₆ 010F ₁₆ | CAN0 message box 10: time stamp | | 289 | 014E ₁₆ 014F ₁₆ | CAN0 message box 14: time stamp | | 289 |
| 0110 ₁₆ 0111 ₁₆ 0112 ₁₆ 0113 ₁₆ 0114 ₁₆ 0115 ₁₆ | CAN0 message box 11: Identifier/DLC | | 289 | 0150 ₁₆ 0151 ₁₆ 0152 ₁₆ 0153 ₁₆ 0154 ₁₆ 0155 ₁₆ | CAN0 message box 15: Identifier/DLC | | 289 |
| 0116 ₁₆ 0117 ₁₆ 0118 ₁₆ 0119 ₁₆ 011A ₁₆ 011B ₁₆ 011C ₁₆ 011D ₁₆ | CAN0 message box 11: Data field | | 289 | 0156 ₁₆ 0157 ₁₆ 0158 ₁₆ 0159 ₁₆ 015A ₁₆ 015B ₁₆ 015C ₁₆ 015D ₁₆ | CAN0 message box 15: Data field | | 289 |
| 011E ₁₆ 011F ₁₆ | CAN0 message box 11: time stamp | | 289 | 015E ₁₆ 015F ₁₆ | CAN0 message box 15: time stamp | | 289 |
| 0120 ₁₆ 0121 ₁₆ 0122 ₁₆ 0123 ₁₆ 0124 ₁₆ 0125 ₁₆ | CAN0 message box 12: Identifier/DLC | | 289 | 0160 ₁₆ 0161 ₁₆ 0162 ₁₆ 0163 ₁₆ 0164 ₁₆ 0165 ₁₆ | CAN0 global mask register | COGMR | 291 |
| 0126 ₁₆ 0127 ₁₆ 0128 ₁₆ 0129 ₁₆ 012A ₁₆ 012B ₁₆ 012C ₁₆ 012D ₁₆ | CAN0 message box 12: Data field | | 289 | 0166 ₁₆ 0167 ₁₆ 0168 ₁₆ 0169 ₁₆ 016A ₁₆ 016B ₁₆ | CAN0 local mask A register | COLMAR | 291 |
| 012E ₁₆ 012F ₁₆ | CAN0 message box 12: time stamp | | 289 | 016C ₁₆ 016D ₁₆ 016E ₁₆ 016F ₁₆ 0170 ₁₆ 0171 ₁₆ | CAN0 local mask B register | COLMBR | 291 |
| 0130 ₁₆ 0131 ₁₆ 0132 ₁₆ 0133 ₁₆ 0134 ₁₆ 0135 ₁₆ | CAN0 message box 13: Identifier/DLC | | 289 | 0172 ₁₆ 0173 ₁₆ 0174 ₁₆ 0175 ₁₆ | | | |
| 0136 ₁₆ 0137 ₁₆ 0138 ₁₆ 0139 ₁₆ 013A ₁₆ 013B ₁₆ 013C ₁₆ 013D ₁₆ | CAN0 message box 13: Data field | | 289 | 0176 ₁₆ 0177 ₁₆ 0178 ₁₆ 0179 ₁₆ 017A ₁₆ 017B ₁₆ 017C ₁₆ 017D ₁₆ | | | |
| 013E ₁₆ 013F ₁₆ | CAN0 message box 13: time stamp | | 289 | 017E ₁₆ 017F ₁₆ | | | |

Note: The blank areas are reserved and cannot be accessed by users.

Quick Reference to Pages Classified by Address

| Address | Register | Symbol | Page |
|--------------------|--|----------|------|
| 0180 ₁₆ | | | |
| 0181 ₁₆ | | | |
| 0182 ₁₆ | | | |
| 0183 ₁₆ | | | |
| 0184 ₁₆ | | | |
| 0185 ₁₆ | | | |
| 0186 ₁₆ | | | |
| | | | |
| 01B0 ₁₆ | | | |
| 01B1 ₁₆ | | | |
| 01B2 ₁₆ | | | |
| 01B3 ₁₆ | Flash memory control register 4 (Note 2) | FMR4 | 342 |
| 01B4 ₁₆ | | | |
| 01B5 ₁₆ | Flash memory control register 1 (Note 2) | FMR1 | 341 |
| 01B6 ₁₆ | | | |
| 01B7 ₁₆ | Flash memory control register 0 (Note 2) | FMR0 | 341 |
| 01B8 ₁₆ | | | |
| 01B9 ₁₆ | | | |
| 01BA ₁₆ | | | |
| 01BB ₁₆ | | | |
| 01BC ₁₆ | | | |
| | | | |
| 0200 ₁₆ | CAN0 message control register 0 | COMCTL0 | 292 |
| 0201 ₁₆ | CAN0 message control register 1 | COMCTL1 | 292 |
| 0202 ₁₆ | CAN0 message control register 2 | COMCTL2 | 292 |
| 0203 ₁₆ | CAN0 message control register 3 | COMCTL3 | 292 |
| 0204 ₁₆ | CAN0 message control register 4 | COMCTL4 | 292 |
| 0205 ₁₆ | CAN0 message control register 5 | COMCTL5 | 292 |
| 0206 ₁₆ | CAN0 message control register 6 | COMCTL6 | 292 |
| 0207 ₁₆ | CAN0 message control register 7 | COMCTL7 | 292 |
| 0208 ₁₆ | CAN0 message control register 8 | COMCTL8 | 292 |
| 0209 ₁₆ | CAN0 message control register 9 | COMCTL9 | 292 |
| 020A ₁₆ | CAN0 message control register 10 | COMCTL10 | 292 |
| 020B ₁₆ | CAN0 message control register 11 | COMCTL11 | 292 |
| 020C ₁₆ | CAN0 message control register 12 | COMCTL12 | 292 |
| 020D ₁₆ | CAN0 message control register 13 | COMCTL13 | 292 |
| 020E ₁₆ | CAN0 message control register 14 | COMCTL14 | 292 |
| 020F ₁₆ | CAN0 message control register 15 | COMCTL15 | 292 |
| 0210 ₁₆ | | | |
| 0211 ₁₆ | CAN0 control register | COCTRL | 293 |
| 0212 ₁₆ | | | |
| 0213 ₁₆ | CAN0 status register | COSTR | 294 |
| 0214 ₁₆ | | | |
| 0215 ₁₆ | CAN0 slot status register | COSSTR | 295 |
| 0216 ₁₆ | | | |
| 0217 ₁₆ | CAN 0 interrupt control register | COICR | 296 |
| 0218 ₁₆ | | | |
| 0219 ₁₆ | CAN0 extended ID register | COIDR | 296 |
| 021A ₁₆ | | | |
| 021B ₁₆ | CAN0 configuration register | COCONR | 297 |
| 021C ₁₆ | CAN0 receive error count register | CORECR | 298 |
| 021D ₁₆ | CAN0 transmit error count register | COTECR | 298 |
| 021E ₁₆ | | | |
| 021F ₁₆ | CAN0 time stamp register | COTSR | 299 |
| 0210 ₁₆ | | | |
| | | | |
| 02FE ₁₆ | | | |
| 02FF ₁₆ | | | |

Note 1: The blank areas are reserved and cannot be accessed by users.

Note 2: This register is included in the flash memory version.

| Address | Register | Symbol | Page | | |
|--------------------|---|--------|----------|--|--|
| 0240 ₁₆ | | | | | |
| 0241 ₁₆ | | | | | |
| 0242 ₁₆ | CAN0 acceptance filter support register | COAFS | 299 | | |
| 0243 ₁₆ | | | | | |
| 0244 ₁₆ | | | | | |
| 0245 ₁₆ | | | | | |
| 0246 ₁₆ | | | | | |
| 0247 ₁₆ | | | | | |
| 0248 ₁₆ | | | | | |
| 0249 ₁₆ | | | | | |
| 024A ₁₆ | | | | | |
| 024C ₁₆ | | | | | |
| 024D ₁₆ | | | | | |
| 024E ₁₆ | | | | | |
| 024F ₁₆ | | | | | |
| 0250 ₁₆ | | | | | |
| 0251 ₁₆ | | | | | |
| 0252 ₁₆ | | | | | |
| 0253 ₁₆ | | | | | |
| 0254 ₁₆ | | | | | |
| 0255 ₁₆ | | | | | |
| 0256 ₁₆ | | | | | |
| 0257 ₁₆ | | | | | |
| 0258 ₁₆ | | | | | |
| 0259 ₁₆ | | | | | |
| 025A ₁₆ | Three-phase protect control register | TPRC | 139 | | |
| 025B ₁₆ | | | | | |
| 025C ₁₆ | On-chip oscillator control register | ROCR | 50 | | |
| 025D ₁₆ | Pin assignment control register | PACR | 177, 326 | | |
| 025E ₁₆ | Peripheral clock select register | PCLKR | 52 | | |
| 025F ₁₆ | CAN0 clock select register | CCLKR | 53 | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| 02E0 ₁₆ | I ² C0 data shift register | S00 | 258 | | |
| 02E1 ₁₆ | | | | | |
| 02E2 ₁₆ | I ² C0 address register | S0D0 | 257 | | |
| 02E3 ₁₆ | I ² C0 control register 0 | S1D0 | 259 | | |
| 02E4 ₁₆ | I ² C0 clock control register | S20 | 258 | | |
| 02E5 ₁₆ | I ² C0 start/stop condition control register | S2D0 | 263 | | |
| 02E6 ₁₆ | I ² C0 control register 1 | S3D0 | 261 | | |
| 02E7 ₁₆ | I ² C0 control register 2 | S4D0 | 262 | | |
| 02E8 ₁₆ | I ² C0 status register | S10 | 260 | | |
| 02E9 ₁₆ | | | | | |
| 02EA ₁₆ | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| 02FE ₁₆ | | | | | |
| 02FF ₁₆ | | | | | |

Quick Reference to Pages Classified by Address

| Address | Register | Symbol | Page |
|--|-------------------------------|--------------|----------|
| 0300 ₁₆ 0301 ₁₆ | TM, WG register 0 | G1TM0, G1PO0 | 146, 147 |
| 0302 ₁₆ 0303 ₁₆ | TM, WG register 1 | G1TM1, G1PO1 | 146, 147 |
| 0304 ₁₆ 0305 ₁₆ | TM, WG register 2 | G1TM2, G1PO2 | 146, 147 |
| 0306 ₁₆ 0307 ₁₆ | TM, WG register 3 | G1TM3, G1PO3 | 146, 147 |
| 0308 ₁₆ 0309 ₁₆ | TM, WG register 4 | G1TM4, G1PO4 | 146, 147 |
| 030A ₁₆ 030B ₁₆ | TM, WG register 5 | G1TM5, G1PO5 | 146, 147 |
| 030C ₁₆ 030D ₁₆ | TM, WG register 6 | G1TM6, G1PO6 | 146, 147 |
| 030E ₁₆ 030F ₁₆ | TM, WG register 7 | G1TM7, G1PO7 | 146, 147 |
| 0310 ₁₆ | WG control register 0 | G1POCR0 | 146 |
| 0311 ₁₆ | WG control register 1 | G1POCR1 | 146 |
| 0312 ₁₆ | WG control register 2 | G1POCR2 | 146 |
| 0313 ₁₆ | WG control register 3 | G1POCR3 | 146 |
| 0314 ₁₆ | WG control register 4 | G1POCR4 | 146 |
| 0315 ₁₆ | WG control register 5 | G1POCR5 | 146 |
| 0316 ₁₆ | WG control register 6 | G1POCR6 | 146 |
| 0317 ₁₆ | WG control register 7 | G1POCR7 | 146 |
| 0318 ₁₆ | TM control register 0 | G1TMCR0 | 145 |
| 0319 ₁₆ | TM control register 1 | G1TMCR1 | 145 |
| 031A ₁₆ | TM control register 2 | G1TMCR2 | 145 |
| 031B ₁₆ | TM control register 3 | G1TMCR3 | 145 |
| 031C ₁₆ | TM control register 4 | G1TMCR4 | 145 |
| 031D ₁₆ | TM control register 5 | G1TMCR5 | 145 |
| 031E ₁₆ | TM control register 6 | G1TMCR6 | 145 |
| 031F ₁₆ | TM control register 7 | G1TMCR7 | 145 |
| 0320 ₁₆ 0321 ₁₆ | Base timer register | G1BT | 142 |
| 0322 ₁₆ | Base timer control register 0 | G1BCR0 | 142 |
| 0323 ₁₆ | Base timer control register 1 | G1BCR1 | 143 |
| 0324 ₁₆ | TM prescale register 6 | G1TPR6 | 145 |
| 0325 ₁₆ | TM prescale register 7 | G1TPR7 | 145 |
| 0326 ₁₆ | Function enable register | G1FE | 148 |
| 0327 ₁₆ | Function select register | G1FS | 148 |
| 0328 ₁₆ 0329 ₁₆ | Base timer reset register | G1BTRR | 144 |
| 032A ₁₆ | Divider register | G1DV | 143 |
| 032B ₁₆ | | | |
| 032C ₁₆ | | | |
| 032D ₁₆ | | | |
| 032E ₁₆ | | | |
| 032F ₁₆ | | | |
| 0330 ₁₆ | Interrupt request register | G1IR | 149 |
| 0331 ₁₆ | Interrupt enable register 0 | G1IE0 | 150 |
| 0332 ₁₆ | Interrupt enable register 1 | G1IE1 | 150 |
| 0333 ₁₆ | | | |
| 0334 ₁₆ | | | |
| 0335 ₁₆ | | | |
| 0336 ₁₆ | | | |
| 0337 ₁₆ | | | |
| 0338 ₁₆ | | | |
| 0339 ₁₆ | | | |
| 033A ₁₆ | | | |
| 033B ₁₆ | | | |
| 033C ₁₆ | | | |
| 033D ₁₆ | | | |
| 033E ₁₆ | NMI digital debounce register | NDDR | 327 |
| 033F ₁₆ | P17 digital debounce register | P17DDR | 327 |

| Address | Register | Symbol | Page |
|--|---|--------|--------|
| 0340 ₁₆ | | | |
| 0341 ₁₆ | | | |
| 0342 ₁₆ 0343 ₁₆ | Timer A1-1 register | TA11 | 130 |
| 0344 ₁₆ 0345 ₁₆ | Timer A2-1 register | TA21 | 130 |
| 0346 ₁₆ 0347 ₁₆ | Timer A4-1 register | TA41 | 130 |
| 0348 ₁₆ | Three-phase PWM control register 0 | INVC0 | 127 |
| 0349 ₁₆ | Three-phase PWM control register 1 | INVC1 | 128 |
| 034A ₁₆ | Three-phase output buffer register 0 | IDB0 | 129 |
| 034B ₁₆ | Three-phase output buffer register 1 | IDB1 | 129 |
| 034C ₁₆ | Dead time timer | DTT | 129 |
| 034D ₁₆ | Timer B2 interrupt occurrence frequency set counter | ICTB2 | 129 |
| 034E ₁₆ | Position data retain function control register | PDRF | 137 |
| 034F ₁₆ | | | |
| 0350 ₁₆ | | | |
| 0351 ₁₆ | | | |
| 0352 ₁₆ | | | |
| 0353 ₁₆ | | | |
| 0354 ₁₆ | | | |
| 0355 ₁₆ | | | |
| 0356 ₁₆ | | | |
| 0357 ₁₆ | | | |
| 0358 ₁₆ | Port function control register | PFCR | 139 |
| 0359 ₁₆ | | | |
| 035A ₁₆ | | | |
| 035B ₁₆ | | | |
| 035C ₁₆ | | | |
| 035D ₁₆ | | | |
| 035E ₁₆ | Interrupt request cause select register 2 | IFSR2A | 77 |
| 035F ₁₆ | Interrupt request cause select register | IFSR | 77, 85 |
| 0360 ₁₆ | SI/O3 transmit/receive register | S3TRR | 218 |
| 0361 ₁₆ | | | |
| 0362 ₁₆ | SI/O3 control register | S3C | 218 |
| 0363 ₁₆ | SI/O3 bit rate generator | S3BRG | 218 |
| 0364 ₁₆ | SI/O4 transmit/receive register | S4TRR | 218 |
| 0365 ₁₆ | | | |
| 0366 ₁₆ | SI/O4 control register | S4C | 218 |
| 0367 ₁₆ | SI/O4 bit rate generator | S4BRG | 218 |
| 0368 ₁₆ | | | |
| 0369 ₁₆ | | | |
| 036A ₁₆ | | | |
| 036B ₁₆ | | | |
| 036C ₁₆ | | | |
| 036D ₁₆ | | | |
| 036E ₁₆ | | | |
| 036F ₁₆ | | | |
| 0370 ₁₆ | | | |
| 0371 ₁₆ | | | |
| 0372 ₁₆ | | | |
| 0373 ₁₆ | | | |
| 0374 ₁₆ | UART2 special mode register 4 | U2SMR4 | 179 |
| 0375 ₁₆ | UART2 special mode register 3 | U2SMR3 | 179 |
| 0376 ₁₆ | UART2 special mode register 2 | U2SMR2 | 178 |
| 0377 ₁₆ | UART2 special mode register | U2SMR | 178 |
| 0378 ₁₆ | UART2 transmit/receive mode register | U2MR | 175 |
| 0379 ₁₆ | UART2 bit rate generator | U2BRG | 174 |
| 037A ₁₆ 037B ₁₆ | UART2 transmit buffer register | U2TB | 174 |
| 037C ₁₆ | UART2 transmit/receive control register 0 | U2C0 | 176 |
| 037D ₁₆ | UART2 transmit/receive control register 1 | U2C1 | 177 |
| 037E ₁₆ 037F ₁₆ | UART2 receive buffer register | U2RB | 174 |

Note : The blank areas are reserved and cannot be accessed by users.

Quick Reference to Pages Classified by Address

| Address | Register | Symbol | Page |
|--------------------|---|--------|---------------|
| 0380 ₁₆ | Count start flag | TABSR | 104, 118, 132 |
| 0381 ₁₆ | Clock prescaler reset flag | CPSRF | 105, 118 |
| 0382 ₁₆ | One-shot start flag | ONSF | 105 |
| 0383 ₁₆ | Trigger select register | TRGSR | 105, 132 |
| 0384 ₁₆ | Up-down flag | UDF | 104 |
| 0385 ₁₆ | | | |
| 0386 ₁₆ | Timer A0 register | TA0 | 104 |
| 0387 ₁₆ | | | |
| 0388 ₁₆ | Timer A1 register | TA1 | 104 |
| 0389 ₁₆ | | | |
| 038A ₁₆ | Timer A2 register | TA2 | 104 |
| 038B ₁₆ | | | |
| 038C ₁₆ | Timer A3 register | TA3 | 104 |
| 038D ₁₆ | | | |
| 038E ₁₆ | Timer A4 register | TA4 | 104 |
| 038F ₁₆ | | | |
| 0390 ₁₆ | Timer B0 register | TB0 | 118 |
| 0391 ₁₆ | | | |
| 0392 ₁₆ | Timer B1 register | TB1 | 118 |
| 0393 ₁₆ | | | |
| 0394 ₁₆ | Timer B2 register | TB2 | 118 |
| 0395 ₁₆ | | | |
| 0396 ₁₆ | Timer A0 mode register | TA0MR | 103 |
| 0397 ₁₆ | Timer A1 mode register | TA1MR | 133 |
| 0398 ₁₆ | Timer A2 mode register | TA2MR | 133 |
| 0399 ₁₆ | Timer A3 mode register | TA3MR | 103 |
| 039A ₁₆ | Timer A4 mode register | TA4MR | 133 |
| 039B ₁₆ | Timer B0 mode register | TB0MR | 117 |
| 039C ₁₆ | Timer B1 mode register | TB1MR | 117 |
| 039D ₁₆ | Timer B2 mode register | TB2MR | 133 |
| 039E ₁₆ | Timer B2 special mode register | TB2SC | 131 |
| 039F ₁₆ | | | |
| 03A0 ₁₆ | UART0 transmit/receive mode register | U0MR | 175 |
| 03A1 ₁₆ | UART0 bit rate generator | U0BRG | 174 |
| 03A2 ₁₆ | UART0 transmit buffer register | U0TB | 174 |
| 03A3 ₁₆ | | | |
| 03A4 ₁₆ | UART0 transmit/receive control register 0 | U0C0 | 176 |
| 03A5 ₁₆ | UART0 transmit/receive control register 1 | U0C1 | 177 |
| 03A6 ₁₆ | UART0 receive buffer register | U0RB | 174 |
| 03A7 ₁₆ | | | |
| 03A8 ₁₆ | UART1 transmit/receive mode register | U1MR | 175 |
| 03A9 ₁₆ | UART1 bit rate generator | U1BRG | 174 |
| 03AA ₁₆ | UART1 transmit buffer register | U1TB | 174 |
| 03AB ₁₆ | | | |
| 03AC ₁₆ | UART1 transmit/receive control register 0 | U1C0 | 176 |
| 03AD ₁₆ | UART1 transmit/receive control register 1 | U1C1 | 177 |
| 03AE ₁₆ | UART1 receive buffer register | U1RB | 174 |
| 03AF ₁₆ | | | |
| 03B0 ₁₆ | UART transmit/receive control register 2 | UCON | 176 |
| 03B1 ₁₆ | | | |
| 03B2 ₁₆ | | | |
| 03B3 ₁₆ | | | |
| 03B4 ₁₆ | CRC snoop address register | CRCSAR | 314 |
| 03B5 ₁₆ | | | |
| 03B6 ₁₆ | CRC mode register | CRCMR | 314 |
| 03B7 ₁₆ | | | |
| 03B8 ₁₆ | DMA0 request cause select register | DM0SL | 93 |
| 03B9 ₁₆ | | | |
| 03BA ₁₆ | DMA1 request cause select register | DM1SL | 94 |
| 03BB ₁₆ | | | |
| 03BC ₁₆ | CRC data register | CRCD | 314 |
| 03BD ₁₆ | | | |
| 03BE ₁₆ | CRC input register | CRCIN | 314 |
| 03BF ₁₆ | | | |

Note : The blank areas are reserved and cannot be accessed by users.

| Address | Register | Symbol | Page |
|--------------------|-------------------------------|----------|------|
| 03C0 ₁₆ | A/D register 0 | AD0 | 226 |
| 03C1 ₁₆ | | | |
| 03C2 ₁₆ | A/D register 1 | AD1 | 226 |
| 03C3 ₁₆ | | | |
| 03C4 ₁₆ | A/D register 2 | AD2 | 226 |
| 03C5 ₁₆ | | | |
| 03C6 ₁₆ | A/D register 3 | AD3 | 226 |
| 03C7 ₁₆ | | | |
| 03C8 ₁₆ | A/D register 4 | AD4 | 226 |
| 03C9 ₁₆ | | | |
| 03CA ₁₆ | A/D register 5 | AD5 | 226 |
| 03CB ₁₆ | | | |
| 03CC ₁₆ | A/D register 6 | AD6 | 226 |
| 03CD ₁₆ | | | |
| 03CE ₁₆ | A/D register 7 | AD7 | 226 |
| 03CF ₁₆ | | | |
| 03D0 ₁₆ | | | |
| 03D1 ₁₆ | | | |
| 03D2 ₁₆ | A/D trigger control register | ADTRGCON | 225 |
| 03D3 ₁₆ | A/D convert status register 0 | ADSTAT0 | 226 |
| 03D4 ₁₆ | A/D control register 2 | ADCON2 | 224 |
| 03D5 ₁₆ | | | |
| 03D6 ₁₆ | A/D control register 0 | ADCON0 | 224 |
| 03D7 ₁₆ | A/D control register 1 | ADCON1 | 224 |
| 03D8 ₁₆ | | | |
| 03D9 ₁₆ | | | |
| 03DA ₁₆ | | | |
| 03DB ₁₆ | | | |
| 03DC ₁₆ | | | |
| 03DD ₁₆ | | | |
| 03DE ₁₆ | | | |
| 03DF ₁₆ | | | |
| 03E0 ₁₆ | Port P0 register | P0 | 324 |
| 03E1 ₁₆ | Port P1 register | P1 | 324 |
| 03E2 ₁₆ | Port P0 direction register | PD0 | 323 |
| 03E3 ₁₆ | Port P1 direction register | PD1 | 323 |
| 03E4 ₁₆ | Port P2 register | P2 | 324 |
| 03E5 ₁₆ | Port P3 register | P3 | 324 |
| 03E6 ₁₆ | Port P2 direction register | PD2 | 323 |
| 03E7 ₁₆ | Port P3 direction register | PD3 | 323 |
| 03E8 ₁₆ | | | |
| 03E9 ₁₆ | | | |
| 03EA ₁₆ | | | |
| 03EB ₁₆ | | | |
| 03EC ₁₆ | Port P6 register | P6 | 324 |
| 03ED ₁₆ | Port P7 register | P7 | 324 |
| 03EE ₁₆ | Port P6 direction register | PD6 | 323 |
| 03EF ₁₆ | Port P7 direction register | PD7 | 323 |
| 03F0 ₁₆ | Port P8 register | P8 | 324 |
| 03F1 ₁₆ | Port P9 register | P9 | 324 |
| 03F2 ₁₆ | Port P8 direction register | PD8 | 323 |
| 03F3 ₁₆ | Port P9 direction register | PD9 | 323 |
| 03F4 ₁₆ | Port P10 register | P10 | 324 |
| 03F5 ₁₆ | | | |
| 03F6 ₁₆ | Port P10 direction register | PD10 | 323 |
| 03F7 ₁₆ | | | |
| 03F8 ₁₆ | | | |
| 03F9 ₁₆ | | | |
| 03FA ₁₆ | | | |
| 03FB ₁₆ | | | |
| 03FC ₁₆ | Pull-up control register 0 | PUR0 | 325 |
| 03FD ₁₆ | Pull-up control register 1 | PUR1 | 325 |
| 03FE ₁₆ | Pull-up control register 2 | PUR2 | 325 |
| 03FF ₁₆ | Port control register | PCR | 326 |

M16C/29 Group

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

1. Overview

1.1 Features

The M16C/29 Group of single-chip control MCU incorporates the M16C/60 series CPU core, employing the high-performance silicon gate CMOS technology and sophisticated instructions for a high level of efficiency. The M16C/29 Group is housed in 64-pin and 80-pin plastic molded LQFP packages. These single-chip MCUs operate using sophisticated instructions featuring a high level of instruction efficiency. This MCU is capable of executing instructions at high speed and it has one CAN module, makes it suitable for control of cars and LAN system of FA. In addition, the CPU core boasts a multiplier and DMAC for high-speed processing to make adequate for office automation, communication devices, and other high-speed processing applications.

1.1.1 Applications

Automotive body, car audio, LAN system of FA, etc.

1.1.2 Specifications

Table 1.1 lists performance overview of M16C/29 Group 80-pin package.

Table 1.2 lists performance overview of M16C/29 Group 64-pin package.

Table 1.1 Performance Overview of M16C/29 Group (T-ver./V-ver.) (80-Pin Package)

| Item | | Performance | | | |
|---|---|---|---|----------------------------------|---|
| CPU | Number of basic instructions | 91 instructions | | | |
| | Shortest instruction execution time | 50 ns (f(BCLK) = 20MHz, V _{CC} = 3.0 to 5.5 V) (Normal-ver./T-ver.) | | | |
| | | 100 ns(f(BCLK) = 10MHz, V _{CC} = 2.7 to 5.5 V) (Normal-ver.) | | | |
| | | 50 ns (f(BCLK) = 20MHz, V _{CC} = 4.2 to 5.5 V, -40 to 105°C) (V-ver.) | | | |
| | | 62.5 ns (f(BCLK) = 16MHz, V _{CC} = 4.2 to 5.5 V, -40 to 125°C) (V-ver.) | | | |
| | Operation mode | Single chip mode | | | |
| Address space | 1 Mbyte | | | | |
| Memory capacity | ROM/RAM: See Tables 1.3 to 1.5 | | | | |
| Peripheral Function | Port | Input/Output: 71 lines | | | |
| | Multifunction timer | TimerA:16 bits x 5 channels, TimerB:16 bits x 3 channels Three-phase Motor Control Timer TimerS (Input Capture/Output Compare): 16 bit base timer x 1 channel (Input/Output x 8 channels) | | | |
| | Serial I/O | 2 channels (UART, clock synchronous serial I/O) 1 channel (UART, clock synchronous serial I/O, I ² C bus, or IEBus ⁽¹⁾) 2 channels (Clock synchronous serial I/O) 1 channel (Multi-master I ² C bus) | | | |
| | A/D converter | 10 bits x 27 channels | | | |
| | DMAC | 2 channels | | | |
| | CRC calculation circuit | 2 polynomial (CRC-CCITT and CRC-16) with MSB/LSB selectable | | | |
| | CAN module | 1 channel, supporting CAN 2.0B specification | | | |
| | Watchdog timer | 15 bits x 1 channel (with prescaler) | | | |
| | Interrupt | 29 internal and 8 external sources, 4 software sources, interrupt priority level: 7 | | | |
| | Clock generation circuit | 4 circuits <ul style="list-style-type: none"> • Main clock • Sub-clock • On-chip oscillator(main-clock oscillation stop detect function) • PLL frequency synthesizer | (These circuits contain a built-in feedback resistor) | | |
| | | | | Oscillation stop detect Function | Main clock oscillation stop, re-oscillation detect function |
| | | | | Voltage detection circuit | Available (Normal-ver.) / Not available (T-ver., V-ver.) |
| | | | | Electrical Characteristics | Power supply voltage |
| | V _{CC} = 2.7 to 5.5 V (f(BCLK) = 10 MHz) | | | | |
| V _{CC} = 3.0 to 5.5 V (T-ver.) | | | | | |
| V _{CC} = 4.2 to 5.5 V (V-ver.) | | | | | |
| Power consumption | 18 mA (V _{CC} = 5 V, f(BCLK) = 20 MHz) | | | | |
| | 25 μA (f(X _{CIN}) = 32 kHz on RAM) | | | | |
| | 3 μA (V _{CC} = 5 V, f(X _{CIN}) = 32 kHz, in wait mode) | | | | |
| | 0.8 μA (V _{CC} = 5 V, in stop mode) | | | | |
| Flash memory | Program/erase supply voltage | 2.7 to 5.5 V (Normal-ver.), 3.0 to 5.5V (T-ver.), 4.2 to 5.5 V (V-ver.) | | | |
| | Program and erase endurance | 100 times (all space) or 1,000 times (blocks 0 to 5)/ 10,000 times (blocks A and B ⁽²⁾) | | | |
| Operating ambient temperature | | -20 to 85°C/-40 to 85°C ⁽²⁾ (Normal-ver.) | | | |
| | | -40 to 85°C (T-ver.), -40 to 125°C (V-ver.) | | | |
| Package | | 80-pin plastic mold LQFP | | | |

NOTES:

1. IEBus is a trademark of NEC Electronics Corporation.
2. Refer to **Table 1.6 to Table 1.8 Product code**.

Table 1.2 Performance Overview of M16C/29 Group (64-Pin Package)

| Item | | Performance |
|-------------------------------|-------------------------------------|---|
| CPU | Number of basic instructions | 91 instructions |
| | Shortest instruction execution time | 50 ns (f(BCLK) = 20MHz, V _{CC} = 3.0 to 5.5 V) (Normal-ver./T-ver.) 100 ns(f(BCLK) = 10MHz, V _{CC} = 2.7 to 5.5 V) (Normal-ver.) 50 ns (f(BCLK) = 20MHz, V _{CC} = 4.2 to 5.5 V, -40 to 105°C) (V-ver.) 62.5 ns (f(BCLK) = 16MHz, V _{CC} = 4.2 to 5.5 V, -40 to 125°C) (V-ver.) |
| | Operation mode | Single chip mode |
| | Address space | 1 Mbytes |
| | Memory capacity | ROM/RAM: See Tables 1.3 to 1.5 |
| Peripheral function | Port | Input/Output: 55 lines |
| | Multifunction timer | TimerA:16 bits x 5 channels, TimerB:16 bits x 3 channels Three-phase Motor Control Timer TimerS (Input Capture/Output Compare): 16bit base timer x 1 channel (Input/Output x 8 channels) |
| | Serial I/O | 2 channels (UART, clock synchronous serial I/O) 1 channel (UART, clock synchronous serial I/O, I ² C bus, or IEBus ⁽¹⁾) 1 channel (Clock synchronous serial I/O) 1 channel (Multi-master I ² C bus) |
| | A/D converter | 10 bits x 16 channels |
| | DMAC | 2 channels |
| | CRC calculation circuit | 2 polynomial (CRC-CCITT and CRC-16) with MSB/LSB selectable |
| | CAN module | 1 channel, supporting CAN 2.0B specification |
| | Watchdog timer | 15 bits x 1 channel (with prescaler) |
| | Interrupt | 28 internal and 8 external sources, 4 software sources, interrupt priority level: 7 |
| | Clock generation circuit | 4 circuits • Main clock } (These circuits contain a built-in feedback resistor) • Sub-clock } • On-chip oscillator(main-clock oscillation stop detect function) • PLL frequency synthesizer |
| | Oscillation stop detect function | Main clock oscillation stop, re-oscillation detect function |
| | Voltage detection circuit | Available (Normal-ver.) / Not available (T-ver., V-ver.) |
| | Electrical Characteristics | Power supply voltage |
| Power consumption | | 18 mA (V _{CC} = 5 V, f(BCLK) = 20 MHz) 25 μA (f(X _{CIN}) = 32 kHz on RAM) 3 μA (V _{CC} = 5 V, f(X _{CIN}) = 32 kHz, in wait mode) 0.8 μA (V _{CC} = 5 V, in stop mode) |
| Flash memory | Program/erase supply voltage | 2.7 to 5.5 V (Normal-ver.), 3.0 to 5.5V (T-ver.), 4.2 to 5.5 V (V-ver.) |
| | Program and erase endurance | 100 times (all space) or 1,000 times (blocks 0 to 5)/ 10,000 times (blocks A and B ⁽²⁾) |
| Operating ambient temperature | | -20 to 85°C/-40 to 85°C ⁽²⁾ (Normal-ver.) -40 to 85°C (T-ver.), -40 to 125°C (V-ver.) |
| Package | | 64-pin plastic mold LQFP |

NOTES:

- IEBus is a trademark of NEC Electronics Corporation.
- Refer to **Table 1.6** to **Table 1.8 Product code**.

1.2 Block Diagram

Figure 1.1 is a block diagram of the M16C/29 Group, 80-pin package.

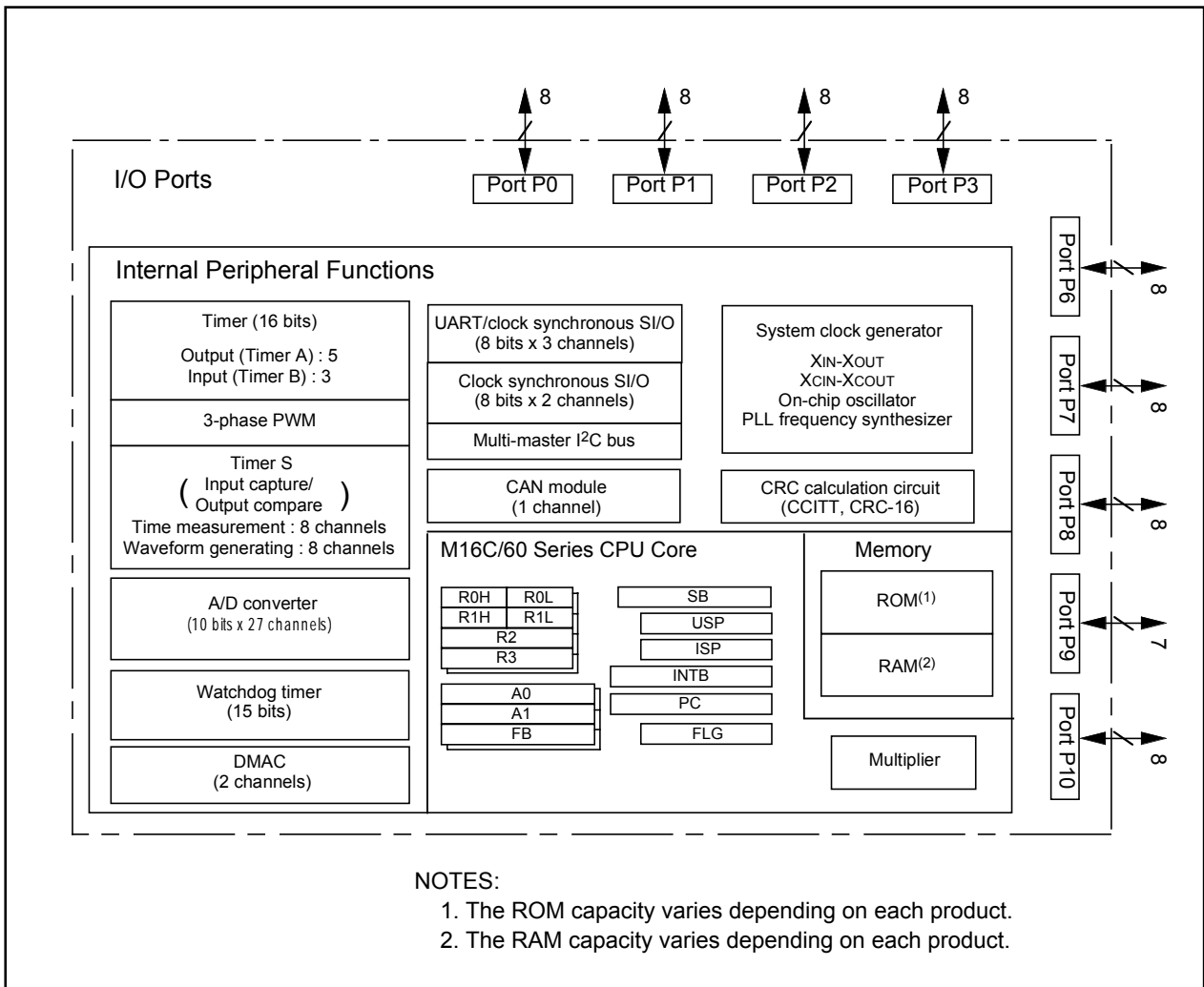


Figure 1.1 M16C/29 Group, 80-Pin Block Diagram

Figure 1.2 is a block diagram of the M16C/29 Group, 64-pin package.

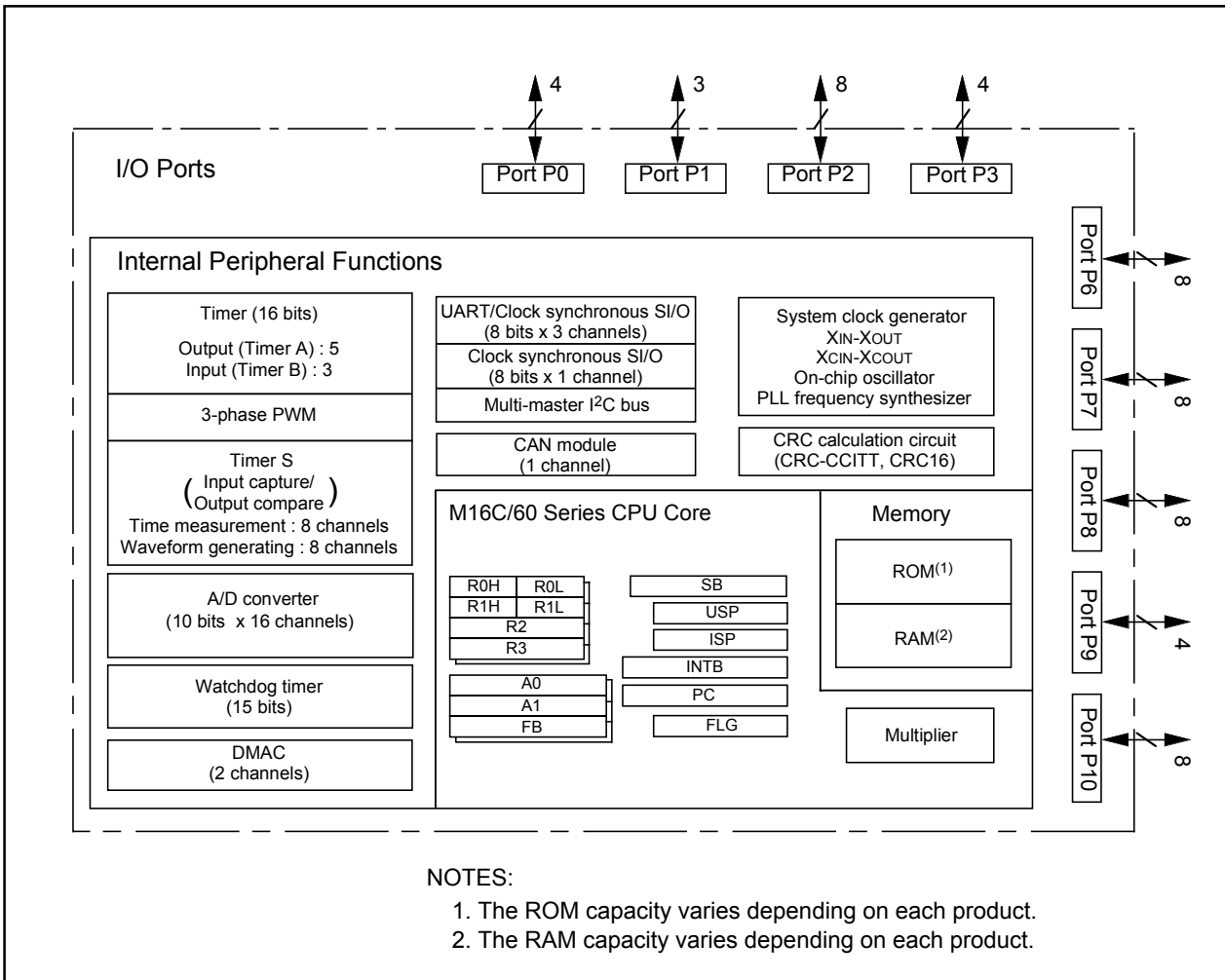


Figure 1.2 M16C/29 Group, 64-Pin Block Diagram

1.3 Product List

Tables 1.3 to 1.5 list the M16C/29 Group products and Figure 1.3 shows the type numbers, memory sizes and packages. Tables 1.6 to 1.8 list the product code of flash memory version for M16C/29 Group. Figure 1.4 to Figure 1.6 show the marking diagram of flash memory version for M16C/29 Group.

Table 1.3 Product List (1) -Normal Version

As of March, 2007

| Type Number | ROM Capacity | RAM Capacity | Package Type | Remarks | Product Code |
|----------------|--------------|--------------|------------------------|--------------|----------------|
| M30290FAHP | 96 K + 4 K | 8 K | PLQP0080KB-A (80P6Q-A) | Flash Memory | U3, U5, U7, U9 |
| M30290FCHP | 128 K + 4 K | 12 K | | | |
| M30291FAHP | 96 K + 4 K | 8 K | PLQP0064KB-A (64P6Q-A) | | |
| M30291FCHP | 128 K + 4 K | 12 K | | | |
| M30290M8-XXXHP | 64 K | 4 K | PLQP0080KB-A (80P6Q-A) | Mask ROM | U3, U5 |
| M30290MA-XXXHP | 96 K | 8 K | | | |
| M30290MC-XXXHP | 128 K | 12 K | | | |
| M30291M8-XXXHP | 64 K | 4 K | PLQP0064KB-A (64P6Q-A) | | |
| M30291MA-XXXHP | 96 K | 8 K | | | |
| M30291MC-XXXHP | 128 K | 12 K | | | |

Table 1.4 Product List (2) -T Version

As of March, 2007

| Type Number | ROM Capacity | RAM Capacity | Package Type | Remarks | Product Code |
|-----------------|--------------|--------------|------------------------|--------------|----------------|
| M30290FATHP | 96 K + 4 K | 8 K | PLQP0080KB-A (80P6Q-A) | Flash Memory | U3, U5, U7, U9 |
| M30290FCTHP | 128 K + 4 K | 12 K | | | |
| M30291FATHP | 96 K + 4 K | 8 K | PLQP0064KB-A (64P6Q-A) | | |
| M30291FCTHP | 128 K + 4 K | 12 K | | | |
| M30290M8T-XXXHP | 64 K | 4 K | PLQP0080KB-A (80P6Q-A) | Mask ROM | U0 |
| M30290MAT-XXXHP | 96 K | 8 K | | | |
| M30290MCT-XXXHP | 128 K | 12 K | | | |
| M30291M8T-XXXHP | 64 K | 4 K | PLQP0064KB-A (64P6Q-A) | | |
| M30291MAT-XXXHP | 96 K | 8 K | | | |
| M30291MCT-XXXHP | 128 K | 12 K | | | |

Table 1.5 Product List (3) -V Version **As of March, 2007**

| Type Number | ROM Capacity | RAM Capacity | Package Type | Remarks | Product Code |
|-----------------|--------------|--------------|------------------------|--------------|----------------|
| M30290FAVHP | 96 K + 4 K | 8 K | PLQP0080KB-A (80P6Q-A) | Flash Memory | U3, U5, U7, U9 |
| M30290FCVHP | 128 K + 4 K | 12 K | | | |
| M30291FAVHP | 96 K + 4 K | 8 K | PLQP0064KB-A (64P6Q-A) | | |
| M30291FCVHP | 128 K + 4 K | 12 K | | | |
| M30290M8V-XXXHP | 64 K | 4 K | PLQP0080KB-A (80P6Q-A) | Mask ROM | U0 |
| M30290MAV-XXXHP | 96 K | 8 K | | | |
| M30290MCV-XXXHP | 128 K | 12 K | | | |
| M30291M8V-XXXHP | 64 K | 4 K | PLQP0064KB-A (64P6Q-A) | | |
| M30291MAV-XXXHP | 96 K | 8 K | | | |
| M30291MCV-XXXHP | 128 K | 12 K | | | |

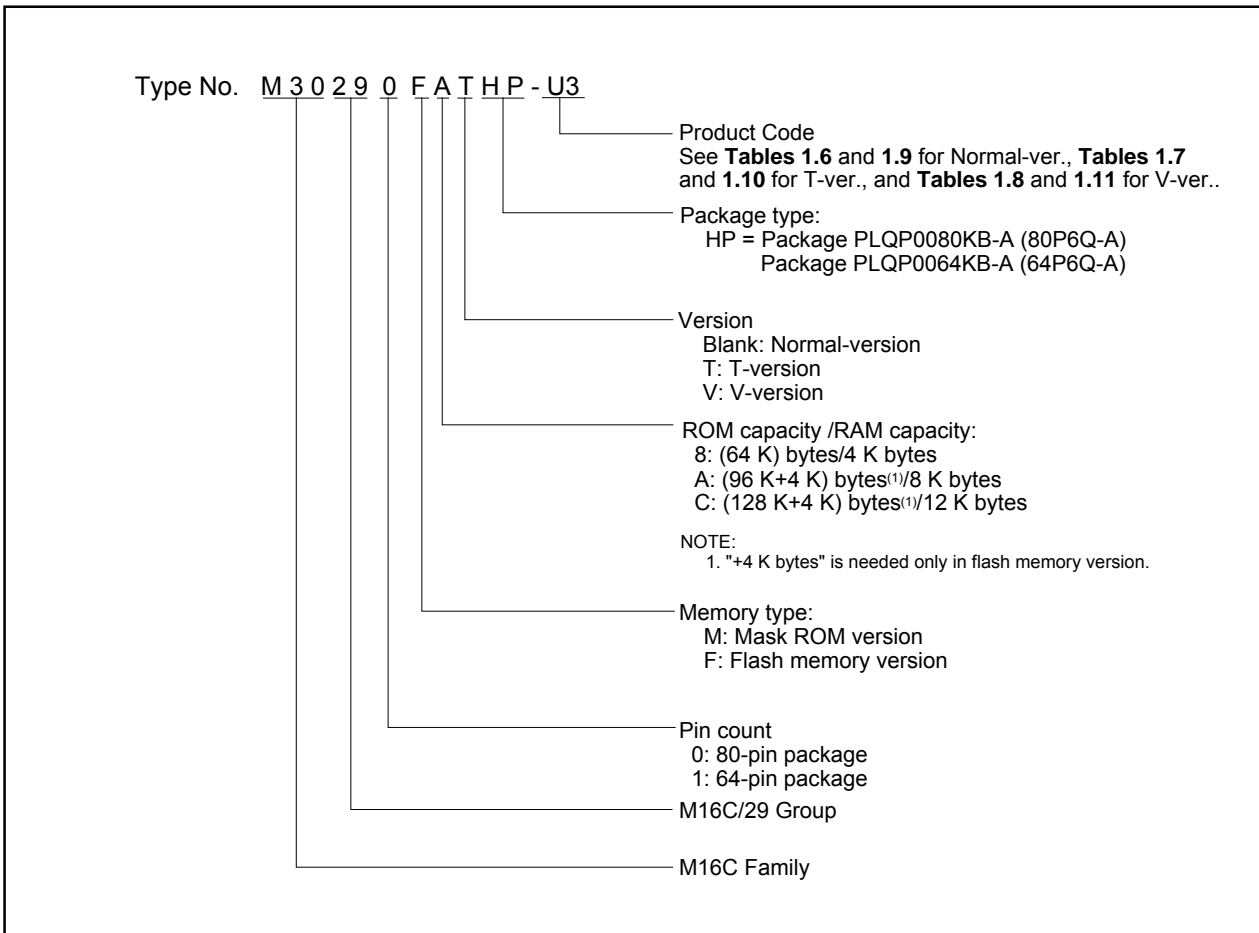


Figure 1.3 Type No., Memory Size, and Package

Table 1.6 Product Codes of Flash Memory Version -M16C/29 Group, Normal-ver.

| Product Code | Package | Internal ROM (User Program Space: Blocks 0 to 5) | | Internal ROM (Data Space: Blocks A and B) | | Operating Ambient Temperature |
|--------------|-----------|---|-------------------|--|-------------------|-------------------------------|
| | | Program and Erase Endurance | Temperature Range | Program and Erase Endurance | Temperature Range | |
| U3 | Lead-free | 100 | 0 to 60°C | 100 | 0 to 60°C | -40 to 85°C |
| U5 | | | | | | -20 to 85°C |
| U7 | | 1,000 | | 10,000 | -40 to 85°C | |
| U9 | | | | | -20 to 85°C | |

Table 1.7 Product Codes of Flash Memory Version -M16C/29 Group, T-ver.

| Product Code | Package | Internal ROM (User Program Space: Blocks 0 to 5) | | Internal ROM (Data Space: Blocks A and B) | | Operating Ambient Temperature |
|--------------|-----------|---|-------------------|--|-------------------|-------------------------------|
| | | Program and Erase Endurance | Temperature Range | Program and Erase Endurance | Temperature Range | |
| U3 | Lead-free | 100 | 0 to 60°C | 100 | -40 to 85°C | -40 to 85°C |
| U7 | | 1,000 | | 10,000 | | |

Table 1.8 Product Codes of Flash Memory Version -M16C/29 Group, V-ver.

| Product Code | Package | Internal ROM (User Program Space: Blocks 0 to 5) | | Internal ROM (Data Space: Blocks A and B) | | Operating Ambient Temperature |
|--------------|-----------|---|-------------------|--|-------------------|-------------------------------|
| | | Program and Erase Endurance | Temperature Range | Program and Erase Endurance | Temperature Range | |
| U3 | Lead-free | 100 | 0 to 60°C | 100 | -40 to 125°C | -40 to 125°C |
| U7 | | 1,000 | | 10,000 | | |

Table 1.9 Product Codes of Mask ROM Version -M16C/29 Group, Normal-ver.

| Product Code | Package | Operating Ambient Temperature |
|--------------|-----------|-------------------------------|
| U3 | Lead-free | -40 to 85°C |
| U5 | | -20 to 85°C |

Table 1.10 Product Code of Mask ROM Version -M16C/29 Group, T-ver.

| Product Code | Package | Operating Ambient Temperature |
|--------------|-----------|-------------------------------|
| U0 | Lead-free | -40 to 85°C |

Table 1.11 Product Code of Mask ROM Version -M16C/29 Group, V-ver.

| Product Code | Package | Operating Ambient Temperature |
|--------------|-----------|-------------------------------|
| U0 | Lead-free | -40 to 125°C |

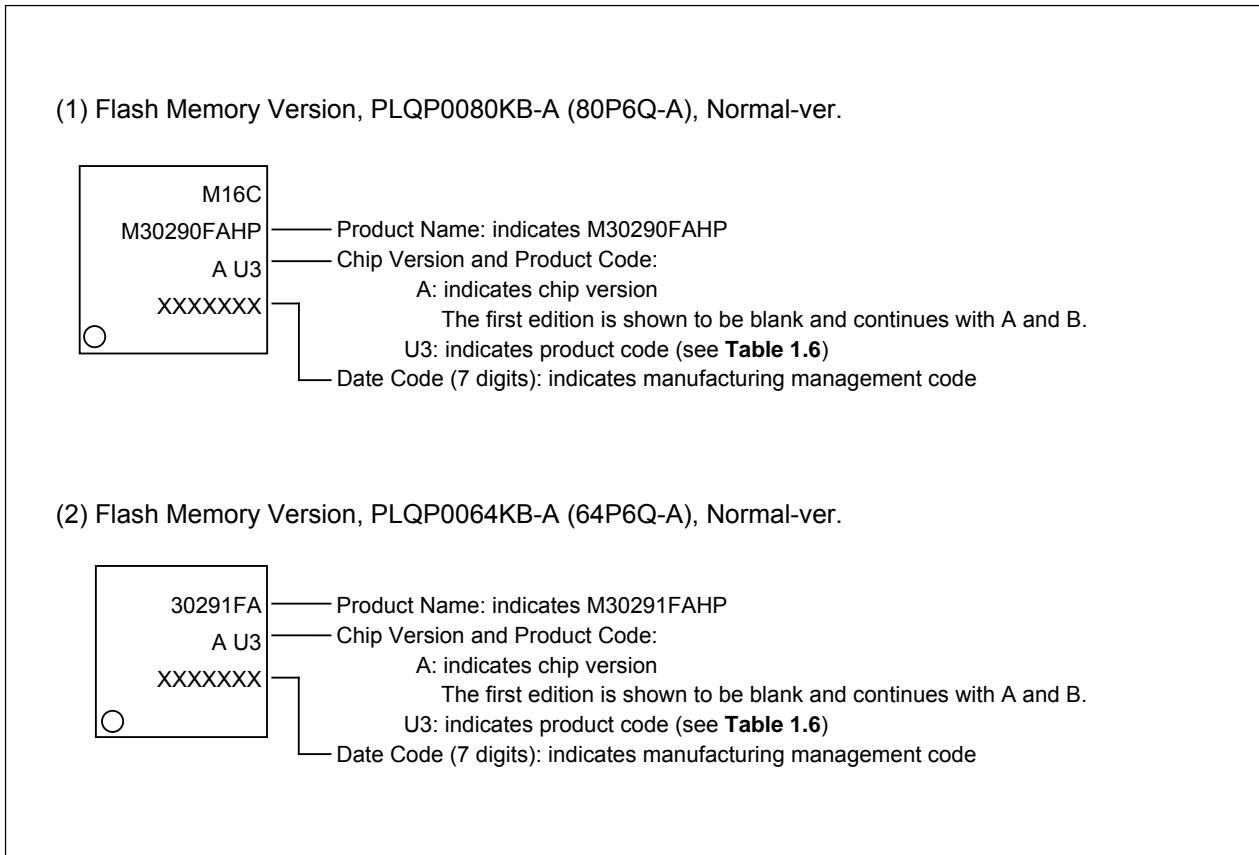


Figure 1.4 Marking Diagrams of Flash Memory Version - M16C/29 Group Normal-ver. (Top View)

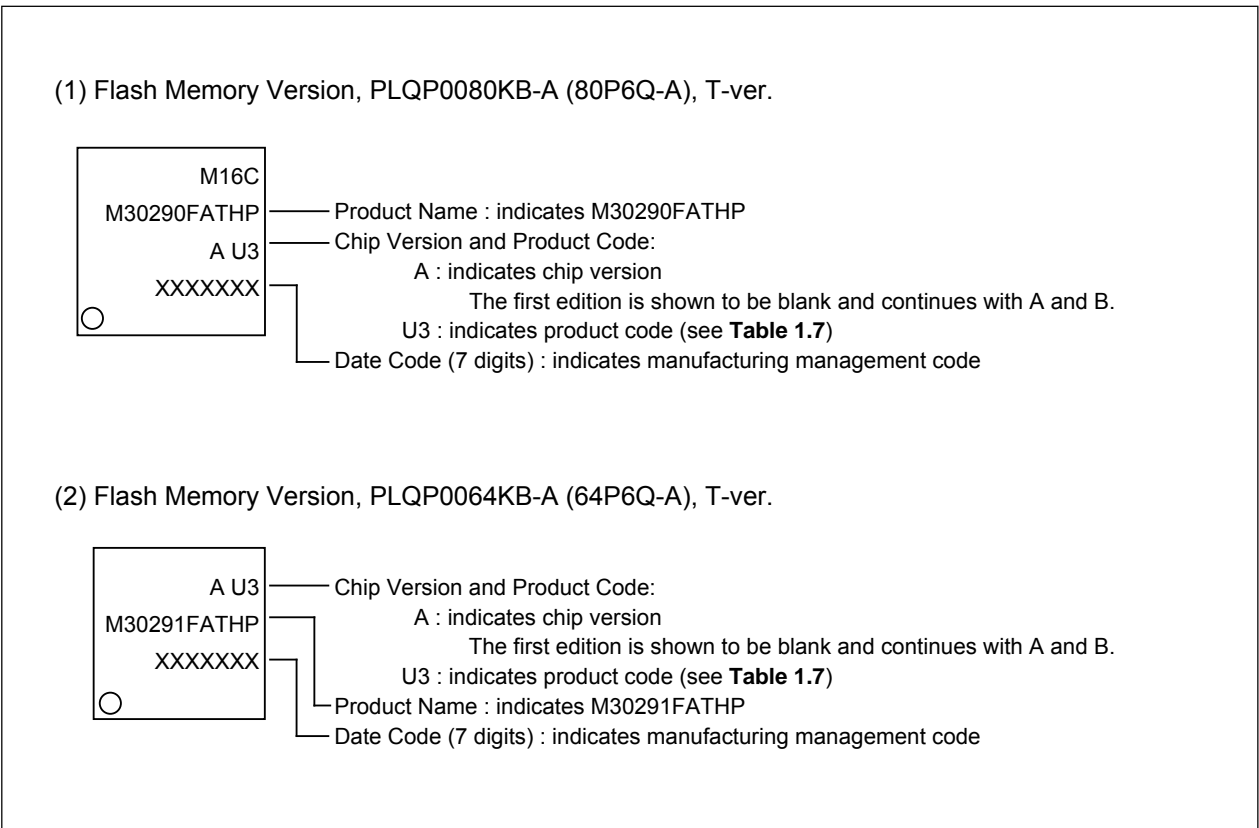


Figure 1.5 Marking Diagrams of Flash Memory Version - M16C/29 Group T-ver. (Top View)

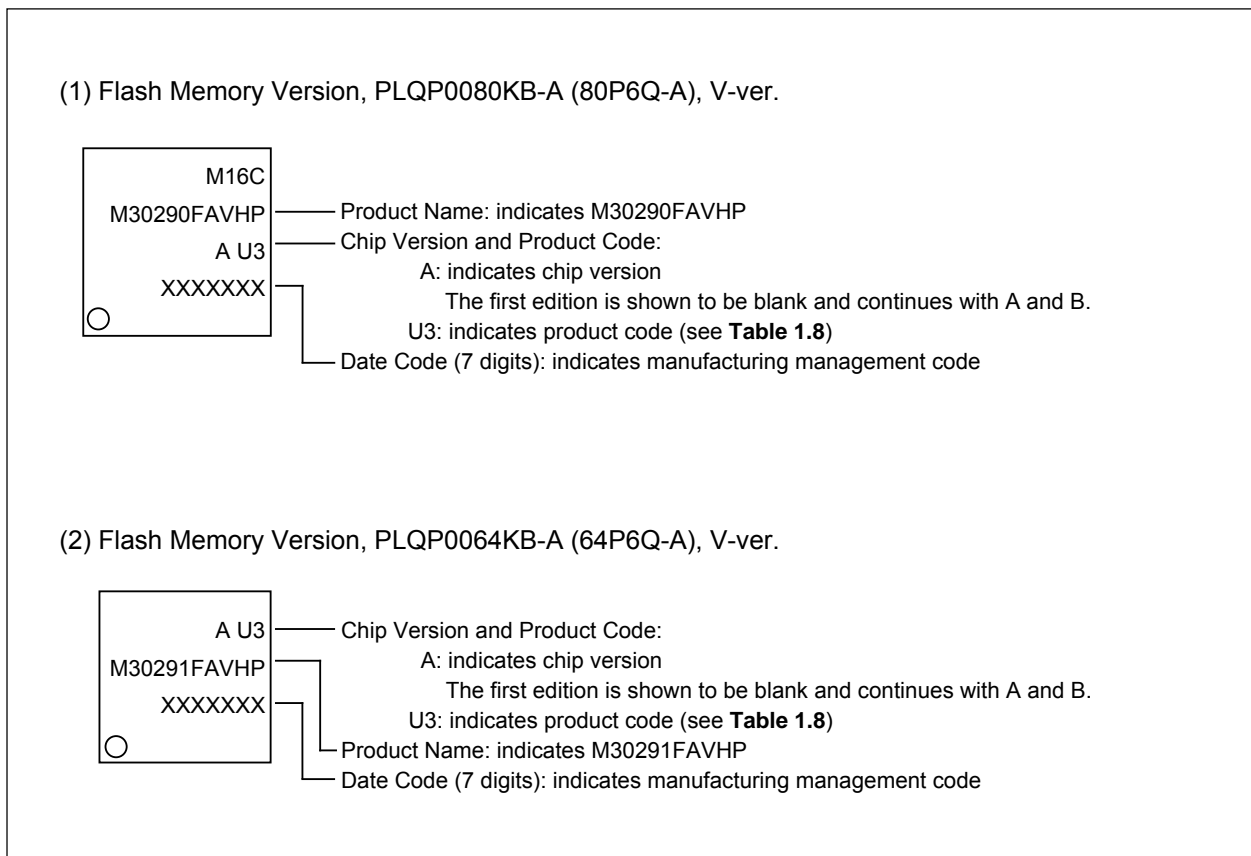


Figure 1.6 Marking Diagrams of Flash Memory Version - M16C/29 Group V-ver. (Top View)

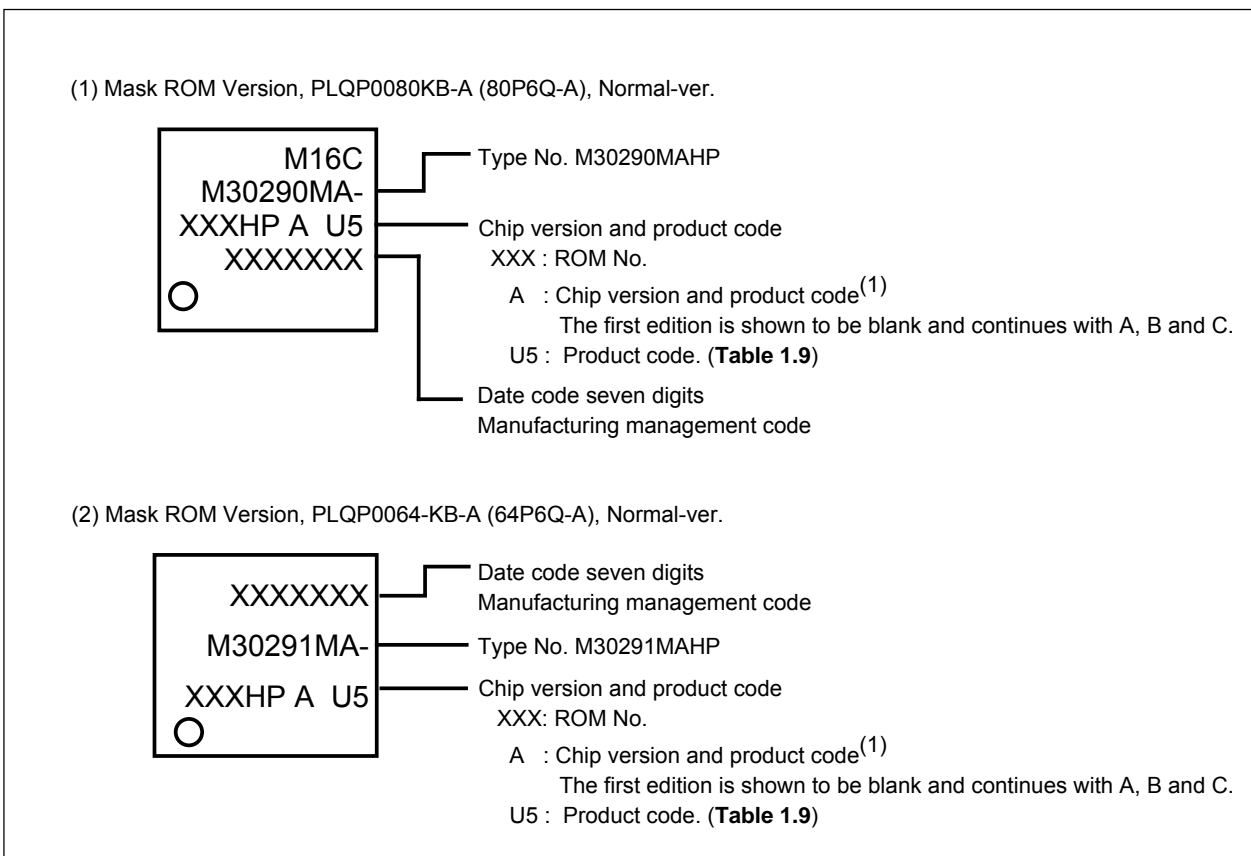


Figure 1.7 Marking Diagrams of Mask ROM Version - M16C/29 Group Normal-ver. (Top View)

1.4 Pin Assignments

Figures 1.7 and 1.8 show the pin assignments (top view).

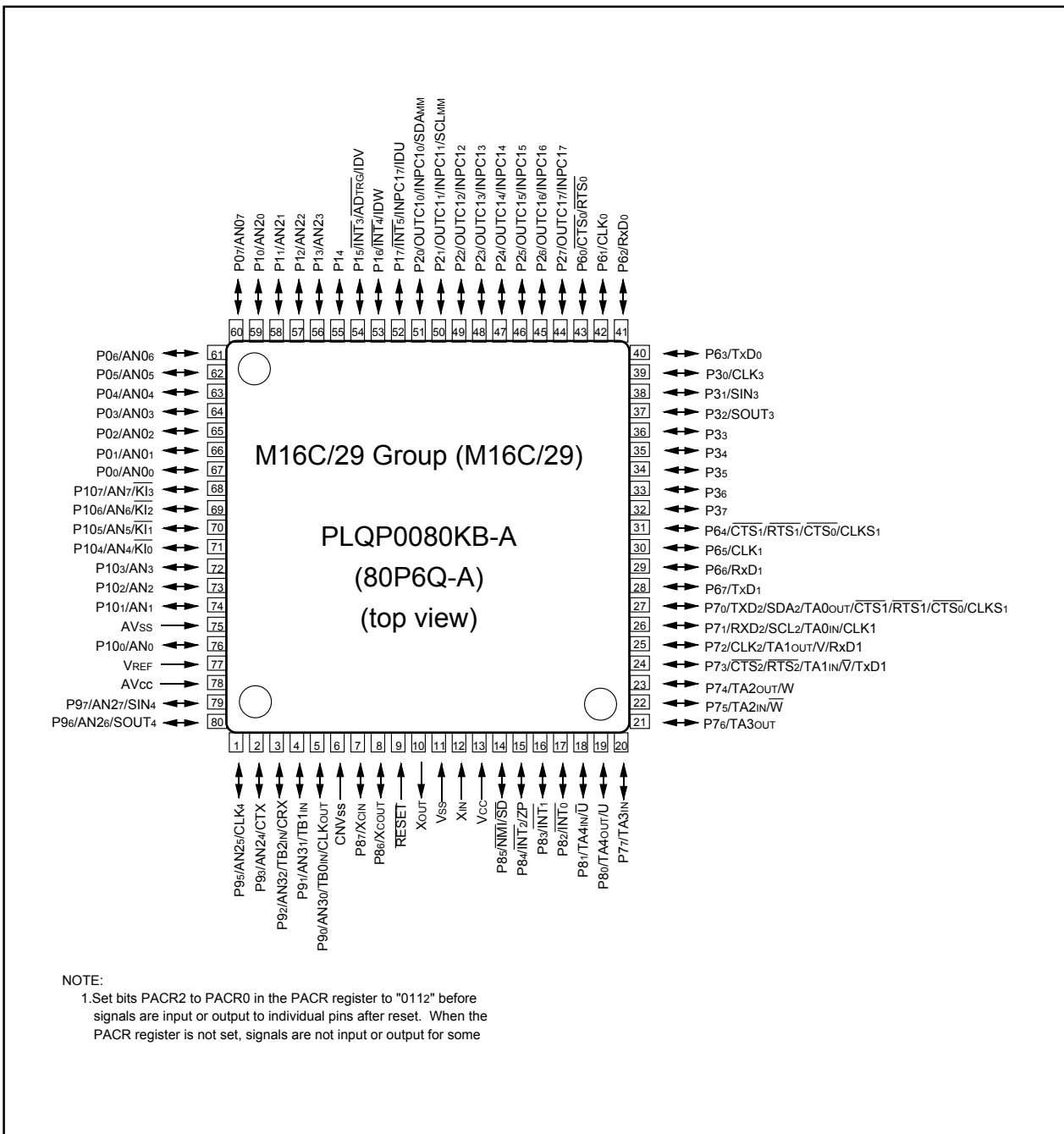


Figure 1.8 Pin Assignment (Top View) of 80-Pin Package

Table 1.12 Pin Characteristics for 80-Pin Package

| Pin No. | Control Pin | Port | Interrupt Pin | Timer Pin | Timer S Pin | UART/CAN Pin | Multi-master I ² C bus Pin | Analog Pin |
|---------|-------------|------|------------------|------------|-------------|--|---------------------------------------|------------|
| 1 | | P95 | | | | CLK4 | | AN25 |
| 2 | | P93 | | | | CTX | | AN24 |
| 3 | | P92 | | TB2IN | | CRX | | AN32 |
| 4 | | P91 | | TB1IN | | | | AN31 |
| 5 | CLKOUT | P90 | | TB0IN | | | | AN30 |
| 6 | CNVss | | | | | | | |
| 7 | XCIN | P87 | | | | | | |
| 8 | XCOUT | P86 | | | | | | |
| 9 | RESET | | | | | | | |
| 10 | XOUT | | | | | | | |
| 11 | Vss | | | | | | | |
| 12 | XIN | | | | | | | |
| 13 | Vcc | | | | | | | |
| 14 | | P85 | NMI | SD | | | | |
| 15 | | P84 | INT ₂ | ZP | | | | |
| 16 | | P83 | INT ₁ | | | | | |
| 17 | | P82 | INT ₀ | | | | | |
| 18 | | P81 | | TA4IN / U | | | | |
| 19 | | P80 | | TA4OUT / U | | | | |
| 20 | | P77 | | TA3IN | | | | |
| 21 | | P76 | | TA3OUT | | | | |
| 22 | | P75 | | TA2IN / W | | | | |
| 23 | | P74 | | TA2OUT / W | | | | |
| 24 | | P73 | | TA1IN / V | | CTS ₂ / RTS ₂ / TxD ₁ | | |
| 25 | | P72 | | TA1OUT / V | | CLK ₂ / RxD ₁ | | |
| 26 | | P71 | | TA0IN | | RxD ₂ / SCL ₂ / CLK ₁ | | |
| 27 | | P70 | | TA0OUT | | TxD ₂ / SDA ₂ / RTS ₁ / CTS ₁ / CTS ₀ / CLKS ₁ | | |
| 28 | | P67 | | | | TxD ₁ | | |
| 29 | | P66 | | | | RxD ₁ | | |
| 30 | | P65 | | | | CLK ₁ | | |
| 31 | | P64 | | | | RTS ₁ / CTS ₁ / CTS ₀ / CLKS ₁ | | |
| 32 | | P37 | | | | | | |
| 33 | | P36 | | | | | | |
| 34 | | P35 | | | | | | |
| 35 | | P34 | | | | | | |
| 36 | | P33 | | | | | | |
| 37 | | P32 | | | | SOUT ₃ | | |
| 38 | | P31 | | | | SIN ₃ | | |
| 39 | | P30 | | | | CLK ₃ | | |
| 40 | | P63 | | | | TxD ₀ | | |

Table 1.12 Pin Characteristics for 80-Pin Package (continued)

| Pin No. | Control Pin | Port | Interrupt Pin | Timer Pin | Timer S Pin | UART/CAN Pin | Multi-master I ² C bus Pin | Analog Pin |
|---------|-------------|------|---------------------------|-----------|-----------------|--------------|---------------------------------------|---------------------------|
| 41 | | P62 | | | | RxD0 | | |
| 42 | | P61 | | | | CLK0 | | |
| 43 | | P60 | | | | RTS0 / CTS0 | | |
| 44 | | P27 | | | OUTC17 / INPC17 | | | |
| 45 | | P26 | | | OUTC16 / INPC16 | | | |
| 46 | | P25 | | | OUTC15 / INPC15 | | | |
| 47 | | P24 | | | OUTC14 / INPC14 | | | |
| 48 | | P23 | | | OUTC13 / INPC13 | | | |
| 49 | | P22 | | | OUTC12 / INPC12 | | | |
| 50 | | P21 | | | OUTC11 / INPC11 | | SCLMM | |
| 51 | | P20 | | | OUTC10 / INPC10 | | SDAMM | |
| 52 | | P17 | $\overline{\text{INT}}_5$ | IDU | INPC17 | | | |
| 53 | | P16 | $\overline{\text{INT}}_4$ | IDW | | | | |
| 54 | | P15 | $\overline{\text{INT}}_3$ | IDV | | | | $\overline{\text{ADTRG}}$ |
| 55 | | P14 | | | | | | |
| 56 | | P13 | | | | | | AN23 |
| 57 | | P12 | | | | | | AN22 |
| 58 | | P11 | | | | | | AN21 |
| 59 | | P10 | | | | | | AN20 |
| 60 | | P07 | | | | | | AN07 |
| 61 | | P06 | | | | | | AN06 |
| 62 | | P05 | | | | | | AN05 |
| 63 | | P04 | | | | | | AN04 |
| 64 | | P03 | | | | | | AN03 |
| 65 | | P02 | | | | | | AN02 |
| 66 | | P01 | | | | | | AN01 |
| 67 | | P00 | | | | | | AN00 |
| 68 | | P107 | $\overline{\text{K}}_3$ | | | | | AN7 |
| 69 | | P106 | $\overline{\text{K}}_2$ | | | | | AN6 |
| 70 | | P105 | $\overline{\text{K}}_1$ | | | | | AN5 |
| 71 | | P104 | $\overline{\text{K}}_0$ | | | | | AN4 |
| 72 | | P103 | | | | | | AN3 |
| 73 | | P102 | | | | | | AN2 |
| 74 | | P101 | | | | | | AN1 |
| 75 | AVss | | | | | | | |
| 76 | | P100 | | | | | | AN0 |
| 77 | VREF | | | | | | | |
| 78 | AVcc | | | | | | | |
| 79 | | P97 | | | | SIN4 | | AN27 |
| 80 | | P96 | | | | SOUT4 | | AN26 |

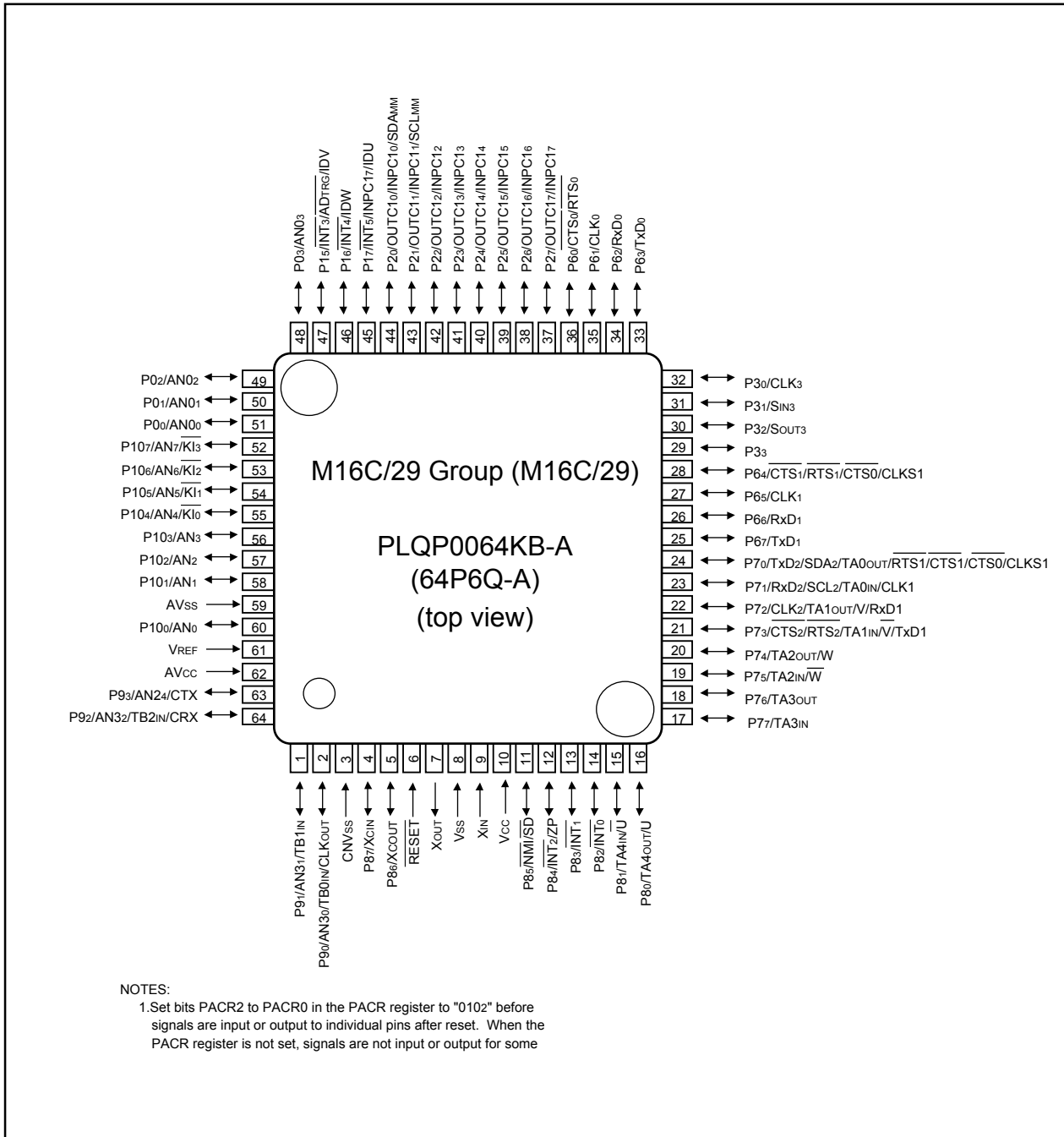


Figure 1.9 Pin Assignment (Top View) of 64-Pin Package

Table 1.13 Pin Characteristics for 64-Pin Package

| Pin No. | Control Pin | Port | Interrupt Pin | Timer Pin | Timer S Pin | UART/CAN Pin | Multi-master I ² C bus Pin | Analog Pin |
|---------|-------------|------|---------------|-------------------|-----------------|--|---------------------------------------|------------|
| 1 | | P91 | | TB1IN | | | | AN31 |
| 2 | CLKOUT | P90 | | TB0IN | | | | AN30 |
| 3 | CNVss | | | | | | | |
| 4 | XCIN | P87 | | | | | | |
| 5 | XCOUT | P86 | | | | | | |
| 6 | RESET | | | | | | | |
| 7 | XOUT | | | | | | | |
| 8 | Vss | | | | | | | |
| 9 | XIN | | | | | | | |
| 10 | Vcc | | | | | | | |
| 11 | | P85 | NMI | SD | | | | |
| 12 | | P84 | INT2 | ZP | | | | |
| 13 | | P83 | INT1 | | | | | |
| 14 | | P82 | INT0 | | | | | |
| 15 | | P81 | | TA4IN / \bar{U} | | | | |
| 16 | | P80 | | TA4OUT / U | | | | |
| 17 | | P77 | | TA3IN | | | | |
| 18 | | P76 | | TA3OUT | | | | |
| 19 | | P75 | | TA2IN / \bar{W} | | | | |
| 20 | | P74 | | TA2OUT / W | | | | |
| 21 | | P73 | | TA1IN / \bar{V} | | CTS2 / RTS2 / TxD1 | | |
| 22 | | P72 | | TA1OUT / V | | CLK2 / RxD1 | | |
| 23 | | P71 | | TA0IN | | RxD2 / SCL2 / CLK1 | | |
| 24 | | P70 | | TA0OUT | | TxD2 / SDA2 / RTS1 / CTS1 / CTS0 / CLKS1 | | |
| 25 | | P67 | | | | TxD1 | | |
| 26 | | P66 | | | | RxD1 | | |
| 27 | | P65 | | | | CLK1 | | |
| 28 | | P64 | | | | RTS1 / CTS1 / CTS0 / CLKS1 | | |
| 29 | | P33 | | | | | | |
| 30 | | P32 | | | | SOUT3 | | |
| 31 | | P31 | | | | SIN3 | | |
| 32 | | P30 | | | | CLK3 | | |
| 33 | | P63 | | | | TxD0 | | |
| 34 | | P62 | | | | RxD0 | | |
| 35 | | P61 | | | | CLK0 | | |
| 36 | | P60 | | | | RTS0 / CTS0 | | |
| 37 | | P27 | | | OUTC17 / INPC17 | | | |
| 38 | | P26 | | | OUTC16 / INPC16 | | | |
| 39 | | P25 | | | OUTC15 / INPC15 | | | |
| 40 | | P24 | | | OUTC14 / INPC14 | | | |

Table 1.13 Pin Characteristics for 64-Pin Package (continued)

| Pin No. | Control Pin | Port | Interrupt Pin | Timer Pin | Timer S Pin | UART/CAN Pin | Multi-master I ² C bus Pin | Analog Pin |
|---------|-------------|------|---------------------------|-----------|-----------------|--------------|---------------------------------------|---------------------------|
| 41 | | P23 | | | OUTC13 / INPC13 | | | |
| 42 | | P22 | | | OUTC12 / INPC12 | | | |
| 43 | | P21 | | | OUTC11 / INPC11 | | SCLMM | |
| 44 | | P20 | | | OUTC10 / INPC10 | | SDAMM | |
| 45 | | P17 | $\overline{\text{INT}}_5$ | IDU | INPC17 | | | |
| 46 | | P16 | $\overline{\text{INT}}_4$ | IDW | | | | |
| 47 | | P15 | $\overline{\text{INT}}_3$ | IDV | | | | $\overline{\text{ADTRG}}$ |
| 48 | | P03 | | | | | | AN03 |
| 49 | | P02 | | | | | | AN02 |
| 50 | | P01 | | | | | | AN01 |
| 51 | | P00 | | | | | | AN00 |
| 52 | | P107 | $\overline{\text{KI}}_3$ | | | | | AN7 |
| 53 | | P106 | $\overline{\text{KI}}_2$ | | | | | AN6 |
| 54 | | P105 | $\overline{\text{KI}}_1$ | | | | | AN5 |
| 55 | | P104 | $\overline{\text{KI}}_0$ | | | | | AN4 |
| 56 | | P103 | | | | | | AN3 |
| 57 | | P102 | | | | | | AN2 |
| 58 | | P101 | | | | | | AN1 |
| 59 | AVss | | | | | | | |
| 60 | | P100 | | | | | | AN0 |
| 61 | VREF | | | | | | | |
| 62 | AVcc | | | | | | | |
| 63 | | P93 | | | | CTX | | AN24 |
| 64 | | P92 | | TB2IN | | CRX | | AN32 |

1.5 Pin Description

Table 1.14 Pin Description (64-pin and 80-pin packages)

| Classification | Symbol | I/O Type | Function |
|--|--|----------|--|
| Power supply | Vcc, Vss | I | Apply 0V to the Vss pin. Apply following voltage to the Vcc pin. 2.7 to 5.5 V (Normal), 3.0 to 5.5 V (T-ver.), 4.2 to 5.5 V (V-ver.) |
| Analog power supply | AVcc AVss | I | Supplies power to the A/D converter. Connect the AVcc pin to Vcc and the AVss pin to Vss |
| Reset input | RESET | I | The microcomputer is in a reset state when "L" is applied to the RESET pin |
| CNVss | CNVss | I | Connect the CNVss pin to Vss |
| Main clock input | XIN | I | I/O pins for the main clock oscillation circuit. Connect a ceramic resonator or crystal oscillator between XIN and XOUT. To apply external clock, apply it to XIN and leave XOUT open. If XIN is not used (for external oscillator or external clock) connect XIN pin to Vcc and leave XOUT open |
| Main clock output | XOUT | O | |
| Sub clock input | XCIN | I | I/O pins for the sub clock oscillation circuit. Connect a crystal oscillator between XCIN and XCOUNT |
| Sub clock output | XCOUT | O | |
| Clock output | CLKOUT | O | Outputs the clock having the same frequency as f1, f8, f32, or fc |
| INT interrupt input | INT0 to INT5 | I | Input pins for the INT interrupt. INT2 can be used for Timer A Z-phase function |
| NMI interrupt input | NMI | I | Input pin for the NMI interrupt. NMI cannot be used as I/O port while the three-phase motor control is enabled. Apply a stable "H" to NMI after setting its direction register to "0" when the three-phase motor control is enabled |
| Key input interrupt | KI0 to KI3 | I | Input pins for the key input interrupt |
| Timer A | TA0OUT to TA4OUT | I/O | I/O pins for the timer A0 to A4 |
| | TA0IN to TA4IN | I | Input pins for the timer A0 to A4 |
| | ZP | I | Input pin for Z-phase |
| Timer B | TB0IN to TB2IN | I | Input pins for the timer B0 to B2 |
| Three-phase motor control timer output | U, \bar{U} , V, \bar{V} , W, \bar{W} | O | Output pins for the three-phase motor control timer |
| | IDU, IDW, IDV, \bar{S} D | I/O | Input and output pins for the three-phase motor control timer |
| Serial I/O | CTS0 to CTS2 | I | Input pins for data transmission control |
| | RTS0 to RTS2 | O | Output pins for data reception control |
| | CLK0 to CLK3 | I/O | Inputs and outputs the transfer clock |
| | RxD0 to RxD2 | I | Inputs serial data |
| | SIN3 | I | Inputs serial data |
| | TxD0 to TxD2 | O | Outputs serial data |
| | SOUT3 | O | Outputs serial data |
| I ² C bus Mode | SDA2 | I/O | Inputs and outputs serial data |
| | SCL2 | | Inputs and outputs the transfer clock |
| Multi-master I ² C bus | SDAMM | I/O | Inputs and outputs serial data |
| | SCLMM | | Inputs and outputs the transfer clock |
| Reference voltage input | VREF | I | Applies reference voltage to the A/D converter |
| A/D converter | AN0 to AN7 AN00 to AN03 AN24 AN30 to AN32 | I | Analog input pins for the A/D converter |
| | ADTRG | I | Input pin for an external A/D trigger |

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

Table 1.14 Pin Description (64-pin and 80-pin packages) (Continued)

| Classification | Symbol | I/O Type | Function |
|----------------|--|----------|---|
| Timer S | INPC10 to INPC17 | I | Input pins for the time measurement function |
| | OUTC10 to OUTC17 | O | Output pins for the waveform generating function |
| CAN | CRX | I | Input pin for the CAN communication function |
| | CTX | O | Output pin for the CAN communication function |
| I/O Ports | P00 to P03 P15 to P17 P20 to P27 P30 to P33 P60 to P67 P70 to P77 P80 to P87 P90 to P93 P100 to P107 | I/O | CMOS I/O ports which have a direction register determines an individual pin is used as an input port or an output port. A pull-up resistor is selectable for every 4 input ports. |

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

Table 1.14 Pin Description (80-pin packages only) (Continued)

| Classification | Symbol | I/O Type | Function |
|----------------|--|----------|---|
| Serial I/O | CLK4 | I/O | Inputs and outputs the transfer clock |
| | SIN4 | I | Inputs serial data |
| | SOUT4 | O | Outputs serial data |
| A/D Converter | AN04 to AN07 AN20 to AN23 AN25 to AN27 | I | Analog input pins for the A/D converter |
| I/O Ports | P04 to P07 P10 to P14 P34 to P37 P95 to P97 | I/O | CMOS I/O ports which have a direction register determines an individual pin is used as an input port or an output port. A pull-up resistor is selectable for every 4 input ports. |

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

2. Central Processing Unit (CPU)

Figure 2.1 shows the CPU registers. The CPU has 13 registers. Of these, R0, R1, R2, R3, A0, A1 and FB comprise a register bank. There are two register banks.

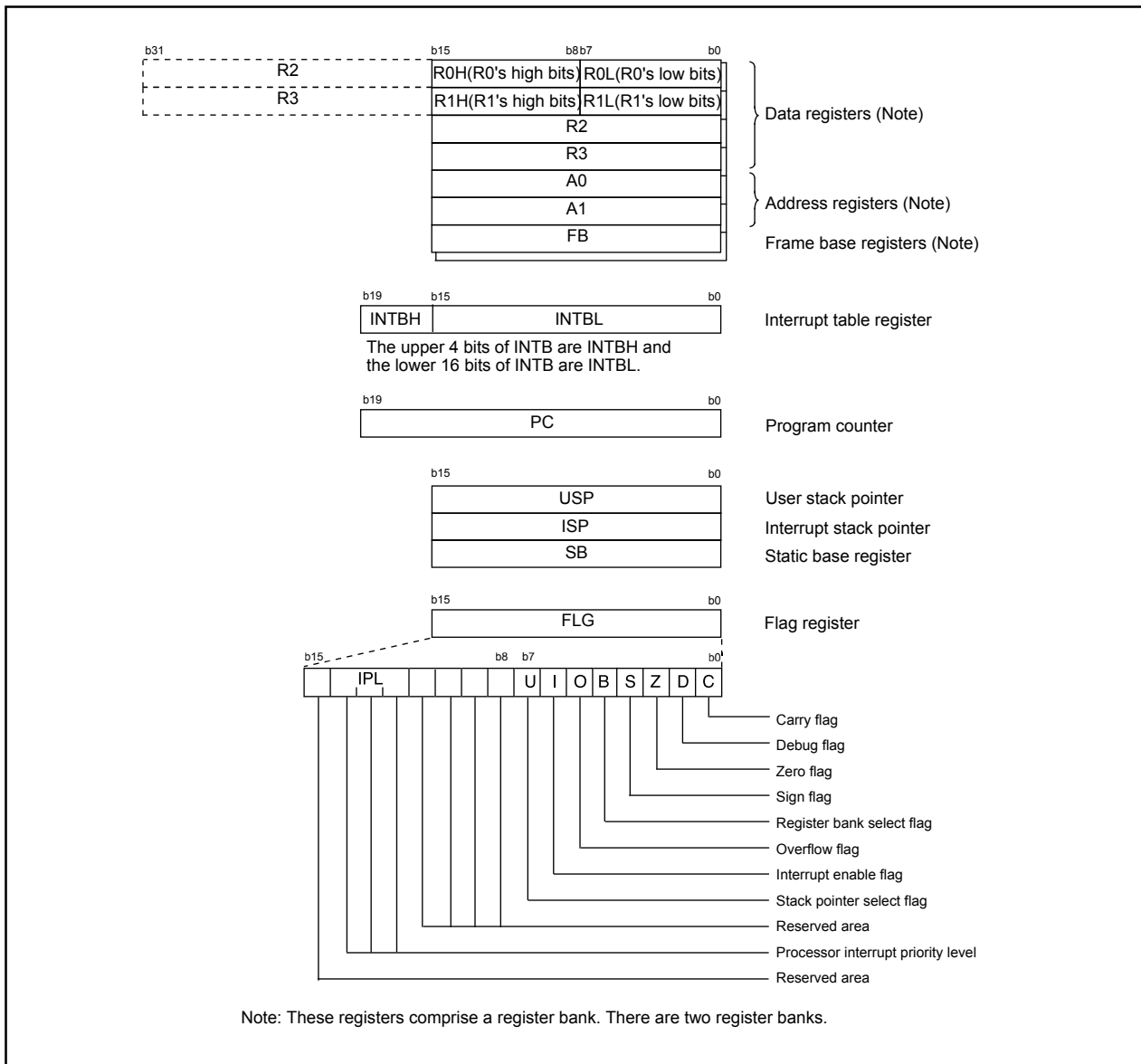


Figure 2.1. Central Processing Unit Register

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

The R0 register consists of 16 bits, and is used mainly for transfers and arithmetic/logic operations. R1 to R3 are the same as R0.

The R0 register can be separated between high (R0H) and low (R0L) for use as two 8-bit data registers. R1H and R1L are the same as R0H and R0L. Conversely, R2 and R0 can be combined for use as a 32-bit data register (R2R0). R3R1 is the same as R2R0.

2.2 Address Registers (A0 and A1)

The register A0 consists of 16 bits, and is used for address register indirect addressing and address register relative addressing. They also are used for transfers and arithmetic/logic operations. A1 is the same as A0.

In some instructions, registers A1 and A0 can be combined for use as a 32-bit address register (A1A0).

2.3 Frame Base Register (FB)

FB is configured with 16 bits, and is used for FB relative addressing.

2.4 Interrupt Table Register (INTB)

INTB is configured with 20 bits, indicating the start address of an interrupt vector table.

2.5 Program Counter (PC)

PC is configured with 20 bits, indicating the address of an instruction to be executed.

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

Stack pointer (SP) comes in two types: USP and ISP, each configured with 16 bits.

Your desired type of stack pointer (USP or ISP) can be selected by the U flag of FLG.

2.7 Static Base Register (SB)

SB is configured with 16 bits, and is used for SB relative addressing.

2.8 Flag Register (FLG)

FLG consists of 11 bits, indicating the CPU status.

2.8.1 Carry Flag (C Flag)

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

2.8.2 Debug Flag (D Flag)

The D flag is used exclusively for debugging purpose. During normal use, it must be set to 0.

2.8.3 Zero Flag (Z Flag)

This flag is set to 1 when an arithmetic operation resulted in 0; otherwise, it is 0.

2.8.4 Sign Flag (S Flag)

This flag is set to 1 when an arithmetic operation resulted in a negative value; otherwise, it is 0.

2.8.5 Register Bank Select Flag (B Flag)

Register bank 0 is selected when this flag is 0 ; register bank 1 is selected when this flag is 1.

2.8.6 Overflow Flag (O Flag)

This flag is set to 1 when the operation resulted in an overflow; otherwise, it is 0.

2.8.7 Interrupt Enable Flag (I Flag)

This flag enables a maskable interrupt.

Maskable interrupts are disabled when the I flag is 0, and are enabled when the I flag is 1.

The I flag is cleared to 0 when the interrupt request is accepted.

2.8.8 Stack Pointer Select Flag (U Flag)

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

The U flag is cleared to 0 when a hardware interrupt request is accepted or an INT instruction for software interrupt Nos. 0 to 31 is executed.

2.8.9 Processor Interrupt Priority Level (IPL)

IPL is configured with three bits, for specification of up to eight processor interrupt priority levels from level 0 to level 7.

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

2.8.10 Reserved Area

When write to this bit, write 0. When read, its content is undefined.

3. Memory

Figure 3.1 is a memory map of the M16C/29 Group. M16C/29 Group provides 1-Mbyte address space from addresses 00000₁₆ to FFFFF₁₆. The internal ROM is allocated lower addresses beginning with address FFFFF₁₆. For example, 64-Kbytes internal ROM is allocated addresses F0000₁₆ to FFFFF₁₆.

Two 2-Kbyte internal ROM areas, block A and block B, are available in the flash memory version. The blocks are allocated addresses F000₁₆ to FFFF₁₆.

The fixed interrupt vector tables are allocated addresses FFFDC₁₆ to FFFFF₁₆. It stores the starting address of each interrupt routine. See the section on interrupts for details.

The internal RAM is allocated higher addresses beginning with address 00400₁₆. For example, 4-Kbytes internal RAM is allocated addresses 00400₁₆ to 013FF₁₆. Besides storing data, it becomes stacks when the subroutines is called or an interrupt is acknowledged.

SFR, consisting of control registers for peripheral functions such as I/O port, A/D converter, serial I/O, timers is allocated addresses 00000₁₆ to 003FF₁₆. All blank spaces within SFR are reserved and cannot be accessed by users.

The special page vector table is allocated to the addresses FFE00₁₆ to FFFDB₁₆. This vector is used by the JMPS or JSRS instruction. For details, refer to the *M16C/60 and M16C/20 Series Software Manual*.

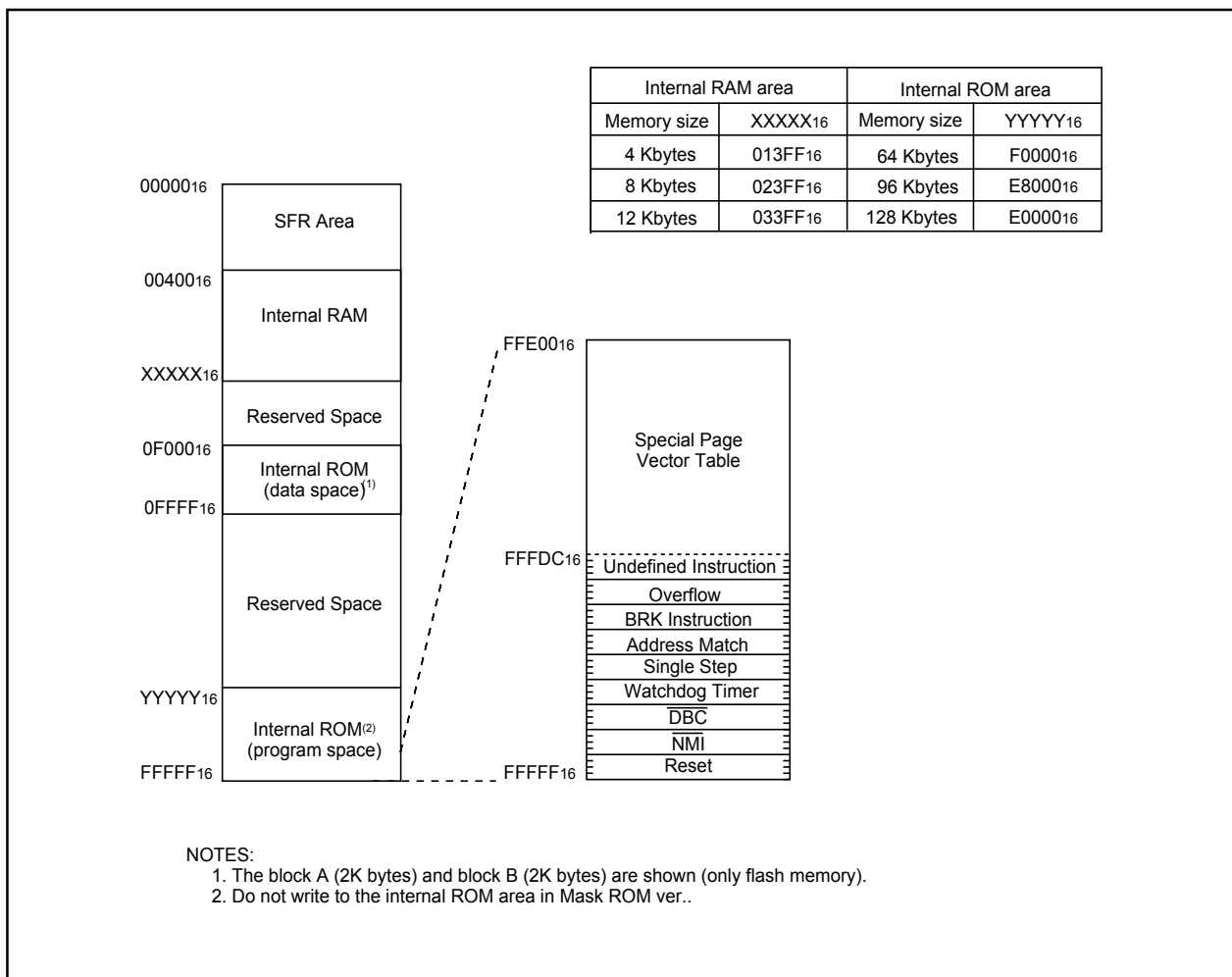


Figure 3.1 Memory Map

4. Special Function Registers (SFRs)

SFRs (Special Function Registers) are the control registers of peripheral functions. **Table 4.1** to **4.11** list the SFR address map.

Table 4.1 SFR Information (1)

| Address | Register | Symbol | After reset |
|--------------------|---|--------|-----------------------|
| 0000 ₁₆ | | | |
| 0001 ₁₆ | | | |
| 0002 ₁₆ | | | |
| 0003 ₁₆ | | | |
| 0004 ₁₆ | Processor mode register 0 | PM0 | 00 ₁₆ |
| 0005 ₁₆ | Processor mode register 1 | PM1 | 00001000 ₂ |
| 0006 ₁₆ | System clock control register 0 | CM0 | 01001000 ₂ |
| 0007 ₁₆ | System clock control register 1 | CM1 | 00100000 ₂ |
| 0008 ₁₆ | | | |
| 0009 ₁₆ | Address match interrupt enable register | AIER | XXXXXX00 ₂ |
| 000A ₁₆ | Protect register | PRCR | XX000000 ₂ |
| 000B ₁₆ | | | |
| 000C ₁₆ | Oscillation stop detection register (Note 2) | CM2 | 0X000010 ₂ |
| 000D ₁₆ | | | |
| 000E ₁₆ | Watchdog timer start register | WDTS | XX ₁₆ |
| 000F ₁₆ | Watchdog timer control register | WDC | 00XXXXXX ₂ |
| 0010 ₁₆ | Address match interrupt register 0 | RMAD0 | 00 ₁₆ |
| 0011 ₁₆ | | | 00 ₁₆ |
| 0012 ₁₆ | | | X0 ₁₆ |
| 0013 ₁₆ | | | |
| 0014 ₁₆ | Address match interrupt register 1 | RMAD1 | 00 ₁₆ |
| 0015 ₁₆ | | | 00 ₁₆ |
| 0016 ₁₆ | | | X0 ₁₆ |
| 0017 ₁₆ | | | |
| 0018 ₁₆ | | | |
| 0019 ₁₆ | Voltage detection register 1 (Note 3,4) | VCR1 | 00001000 ₂ |
| 001A ₁₆ | Voltage detection register 2 (Note 3,4) | VCR2 | 00 ₁₆ |
| 001B ₁₆ | | | |
| 001C ₁₆ | PLL control register 0 | PLC0 | 0001X010 ₂ |
| 001D ₁₆ | | | |
| 001E ₁₆ | Processor mode register 2 | PM2 | XXX00000 ₂ |
| 001F ₁₆ | Low voltage detection interrupt register (Note 4) | D4INT | 00 ₁₆ |
| 0020 ₁₆ | DMA0 source pointer | SAR0 | XX ₁₆ |
| 0021 ₁₆ | | | XX ₁₆ |
| 0022 ₁₆ | | | XX ₁₆ |
| 0023 ₁₆ | | | |
| 0024 ₁₆ | DMA0 destination pointer | DAR0 | XX ₁₆ |
| 0025 ₁₆ | | | XX ₁₆ |
| 0026 ₁₆ | | | XX ₁₆ |
| 0027 ₁₆ | | | |
| 0028 ₁₆ | DMA0 transfer counter | TCR0 | XX ₁₆ |
| 0029 ₁₆ | | | XX ₁₆ |
| 002A ₁₆ | | | |
| 002B ₁₆ | | | |
| 002C ₁₆ | DMA0 control register | DM0CON | 00000X00 ₂ |
| 002D ₁₆ | | | |
| 002E ₁₆ | | | |
| 002F ₁₆ | | | |
| 0030 ₁₆ | DMA1 source pointer | SAR1 | XX ₁₆ |
| 0031 ₁₆ | | | XX ₁₆ |
| 0032 ₁₆ | | | XX ₁₆ |
| 0033 ₁₆ | | | |
| 0034 ₁₆ | DMA1 destination pointer | DAR1 | XX ₁₆ |
| 0035 ₁₆ | | | XX ₁₆ |
| 0036 ₁₆ | | | XX ₁₆ |
| 0037 ₁₆ | | | |
| 0038 ₁₆ | DMA1 transfer counter | TCR1 | XX ₁₆ |
| 0039 ₁₆ | | | XX ₁₆ |
| 003A ₁₆ | | | |
| 003B ₁₆ | | | |
| 003C ₁₆ | DMA1 control register | DM1CON | 00000X00 ₂ |
| 003D ₁₆ | | | |
| 003E ₁₆ | | | |
| 003F ₁₆ | | | |

Note 1: The blank areas are reserved and cannot be used by users.

Note 2: Bits CM20, CM21, and CM27 do not change at oscillation stop detection reset.

Note 3: This register does not change at software reset, watchdog timer reset and oscillation stop detection reset.

Note 4: This register can not use for T-ver. and V-ver.

X : Undefined

Table 4.2 SFR Information (2)

| Address | Register | Symbol | After reset |
|--------------------|--|------------------|-----------------------|
| 0040 ₁₆ | | | |
| 0041 ₁₆ | CAN0 wakeup interrupt control register | C01WKIC | XXXXX000 ₂ |
| 0042 ₁₆ | CAN0 successful reception interrupt control register | C0RECIC | XXXXX000 ₂ |
| 0043 ₁₆ | CAN0 successful transmission interrupt control register | C0TRMIC | XXXXX000 ₂ |
| 0044 ₁₆ | INT3 interrupt control register | INT3IC | XX00X000 ₂ |
| 0045 ₁₆ | ICOC 0 interrupt control register | ICOC0IC | XXXXX000 ₂ |
| 0046 ₁₆ | ICOC 1 interrupt control register, I ² C bus interface interrupt control register 1 | ICOC1IC, IICIC | XXXXX000 ₂ |
| 0047 ₁₆ | ICOC base timer interrupt control register, SCL/SbA interrupt control register 2 | BTIC, SCLDAIC | XXXXX000 ₂ |
| 0048 ₁₆ | SI/O4 interrupt control register, INT5 interrupt control register | S4IC, INT5IC | XX00X000 ₂ |
| 0049 ₁₆ | SI/O3 interrupt control register, INT4 interrupt control register | S3IC, INT4IC | XX00X000 ₂ |
| 004A ₁₆ | UART2 Bus collision detection interrupt control register | BCNIC | XXXXX000 ₂ |
| 004B ₁₆ | DMA0 interrupt control register | DM0IC | XXXXX000 ₂ |
| 004C ₁₆ | DMA1 interrupt control register | DM1IC | XXXXX000 ₂ |
| 004D ₁₆ | CAN0 error interrupt control register | C01ERRIC | XXXXX000 ₂ |
| 004E ₁₆ | A/D conversion interrupt control register, Key input interrupt control register (Note 2) | ADIC, KUPIC | XXXXX000 ₂ |
| 004F ₁₆ | UART2 transmit interrupt control register | S2TIC | XXXXX000 ₂ |
| 0050 ₁₆ | UART2 receive interrupt control register | S2RIC | XXXXX000 ₂ |
| 0051 ₁₆ | UART0 transmit interrupt control register | S0TIC | XXXXX000 ₂ |
| 0052 ₁₆ | UART0 receive interrupt control register | S0RIC | XXXXX000 ₂ |
| 0053 ₁₆ | UART1 transmit interrupt control register | S1TIC | XXXXX000 ₂ |
| 0054 ₁₆ | UART1 receive interrupt control register | S1RIC | XXXXX000 ₂ |
| 0055 ₁₆ | TimerA0 interrupt control register | TA0IC | XXXXX000 ₂ |
| 0056 ₁₆ | TimerA1 interrupt control register | TA1IC | XXXXX000 ₂ |
| 0057 ₁₆ | TimerA2 interrupt control register | TA2IC | XXXXX000 ₂ |
| 0058 ₁₆ | TimerA3 interrupt control register | TA3IC | XXXXX000 ₂ |
| 0059 ₁₆ | TimerA4 interrupt control register | TA4IC | XXXXX000 ₂ |
| 005A ₁₆ | TimerB0 interrupt control register | TB0IC | XXXXX000 ₂ |
| 005B ₁₆ | TimerB1 interrupt control register | TB1IC | XXXXX000 ₂ |
| 005C ₁₆ | TimerB2 interrupt control register | TB2IC | XXXXX000 ₂ |
| 005D ₁₆ | INT0 interrupt control register | INT0IC | XX00X000 ₂ |
| 005E ₁₆ | INT1 interrupt control register | INT1IC | XX00X000 ₂ |
| 005F ₁₆ | INT2 interrupt control register | INT2IC | XX00X000 ₂ |
| 0060 ₁₆ | CAN0 message box 0: Identifier/DLC | | XX ₁₆ |
| 0061 ₁₆ | | | XX ₁₆ |
| 0062 ₁₆ | | | XX ₁₆ |
| 0063 ₁₆ | | | XX ₁₆ |
| 0064 ₁₆ | | | XX ₁₆ |
| 0065 ₁₆ | | | XX ₁₆ |
| 0066 ₁₆ | CAN0 message box 0 : Data field | | XX ₁₆ |
| 0067 ₁₆ | | | XX ₁₆ |
| 0068 ₁₆ | | | XX ₁₆ |
| 0069 ₁₆ | | | XX ₁₆ |
| 006A ₁₆ | | | XX ₁₆ |
| 006B ₁₆ | | | XX ₁₆ |
| 006C ₁₆ | | XX ₁₆ | |
| 006D ₁₆ | | XX ₁₆ | |
| 006E ₁₆ | CAN0 message box 0 : Time stamp | | XX ₁₆ |
| 006F ₁₆ | | | XX ₁₆ |
| 0070 ₁₆ | CAN0 message box 1 : Identifier/DLC | | XX ₁₆ |
| 0071 ₁₆ | | | XX ₁₆ |
| 0072 ₁₆ | | | XX ₁₆ |
| 0073 ₁₆ | | | XX ₁₆ |
| 0074 ₁₆ | | | XX ₁₆ |
| 0075 ₁₆ | | | XX ₁₆ |
| 0076 ₁₆ | CAN0 message box 1 : Data field | | XX ₁₆ |
| 0077 ₁₆ | | | XX ₁₆ |
| 0078 ₁₆ | | | XX ₁₆ |
| 0079 ₁₆ | | | XX ₁₆ |
| 007A ₁₆ | | | XX ₁₆ |
| 007B ₁₆ | | | XX ₁₆ |
| 007C ₁₆ | | XX ₁₆ | |
| 007D ₁₆ | | XX ₁₆ | |
| 007E ₁₆ | CAN0 message box 1 : Time stamp | | XX ₁₆ |
| 007F ₁₆ | | | XX ₁₆ |

Note 1: The blank areas are reserved and cannot be used by users.

Note 2: A/D conversion interrupt control register is effective when the bit1 (Interrupt source select register (address 35Eh IFSR2A) is set to "0". Key input interrupt control register is effective when the bit1 is set to "1".

X : Undefined

Table 4.3 SFR Information (3)

| Address | Register | Symbol | After reset |
|--|-------------------------------------|--------|--|
| 0080 ₁₆ 0081 ₁₆ 0082 ₁₆ 0083 ₁₆ 0084 ₁₆ 0085 ₁₆ | CAN0 message box 2: Identifier/DLC | | XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ |
| 0086 ₁₆ 0087 ₁₆ 0088 ₁₆ 0089 ₁₆ 008A ₁₆ 008B ₁₆ 008C ₁₆ 008D ₁₆ | CAN0 message box 2 : Data field | | XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ |
| 008E ₁₆ 008F ₁₆ | CAN0 message box 2 : Time stamp | | XX ₁₆ XX ₁₆ |
| 0090 ₁₆ 0091 ₁₆ 0092 ₁₆ 0093 ₁₆ 0094 ₁₆ 0095 ₁₆ | CAN0 message box 3 : Identifier/DLC | | XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ |
| 0096 ₁₆ 0097 ₁₆ 0098 ₁₆ 0099 ₁₆ 009A ₁₆ 009B ₁₆ 009C ₁₆ 009D ₁₆ | CAN0 message box 3 : Data field | | XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ |
| 009E ₁₆ 009F ₁₆ | CAN0 message box 3 : Time stamp | | XX ₁₆ XX ₁₆ |
| 00A0 ₁₆ 00A1 ₁₆ 00A2 ₁₆ 00A3 ₁₆ 00A4 ₁₆ 00A5 ₁₆ | CAN0 message box 4: Identifier/DLC | | XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ |
| 00A6 ₁₆ 00A7 ₁₆ 00A8 ₁₆ 00A9 ₁₆ 00AA ₁₆ 00AB ₁₆ 00AC ₁₆ 00AD ₁₆ | CAN0 message box 4 : Data field | | XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ |
| 00AE ₁₆ 00AF ₁₆ | CAN0 message box 4 : Time stamp | | XX ₁₆ XX ₁₆ |
| 00B0 ₁₆ 00B1 ₁₆ 00B2 ₁₆ 00B3 ₁₆ 00B4 ₁₆ 00B5 ₁₆ | CAN0 message box 5 : Identifier/DLC | | XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ |
| 00B6 ₁₆ 00B7 ₁₆ 00B8 ₁₆ 00B9 ₁₆ 00BA ₁₆ 00BB ₁₆ 00BC ₁₆ 00BD ₁₆ | CAN0 message box 5 : Data field | | XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ |
| 00BE ₁₆ 00BF ₁₆ | CAN0 message box 5 : Time stamp | | XX ₁₆ XX ₁₆ |

Note 1: The blank areas are reserved and cannot be used by users.

X : Undefined

Table 4.4 SFR Information (4)

| Address | Register | Symbol | After reset |
|--|-------------------------------------|--------|--|
| 00C0 ₁₆ 00C1 ₁₆ 00C2 ₁₆ 00C3 ₁₆ 00C4 ₁₆ 00C5 ₁₆ | CAN0 message box 6: Identifier/DLC | | XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ |
| 00C6 ₁₆ 00C7 ₁₆ 00C8 ₁₆ 00C9 ₁₆ 00CA ₁₆ 00CB ₁₆ 00CC ₁₆ 00CD ₁₆ | CAN0 message box 6 : Data field | | XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ |
| 00CE ₁₆ 00CF ₁₆ | CAN0 message box 6 : Time stamp | | XX ₁₆ XX ₁₆ |
| 00D0 ₁₆ 00D1 ₁₆ 00D2 ₁₆ 00D3 ₁₆ 00D4 ₁₆ 00D5 ₁₆ | CAN0 message box 7 : Identifier/DLC | | XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ |
| 00D6 ₁₆ 00D7 ₁₆ 00D8 ₁₆ 00D9 ₁₆ 00DA ₁₆ 00DB ₁₆ 00DC ₁₆ 00DD ₁₆ | CAN0 message box 7 : Data field | | XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ |
| 00DE ₁₆ 00DF ₁₆ | CAN0 message box 7 : Time stamp | | XX ₁₆ XX ₁₆ |
| 00E0 ₁₆ 00E1 ₁₆ | CAN0 message box 8: Identifier/DLC | | XX ₁₆ XX ₁₆ |
| 00E2 ₁₆ 00E3 ₁₆ 00E4 ₁₆ 00E5 ₁₆ | | | XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ |
| 00E6 ₁₆ 00E7 ₁₆ 00E8 ₁₆ 00E9 ₁₆ 00EA ₁₆ 00EB ₁₆ 00EC ₁₆ 00ED ₁₆ | CAN0 message box 8: Data field | | XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ |
| 00EE ₁₆ 00EF ₁₆ | CAN0 message box 8 : Time stamp | | XX ₁₆ XX ₁₆ |
| 00F0 ₁₆ 00F1 ₁₆ 00F2 ₁₆ 00F3 ₁₆ 00F4 ₁₆ 00F5 ₁₆ | CAN0 message box 9 : Identifier/DLC | | XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ |
| 00F6 ₁₆ 00F7 ₁₆ 00F8 ₁₆ 00F9 ₁₆ 00FA ₁₆ 00FB ₁₆ 00FC ₁₆ 00FD ₁₆ | CAN0 message box 9 : Data field | | XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ |
| 00FE ₁₆ 00FF ₁₆ | CAN0 message box 9 : Time stamp | | XX ₁₆ XX ₁₆ |

Note 1: The blank areas are reserved and cannot be used by users.

X : Undefined

Table 4.5 SFR Information (5)

| Address | Register | Symbol | After reset |
|--|--------------------------------------|--------|--|
| 0100 ₁₆ 0101 ₁₆ 0102 ₁₆ 0103 ₁₆ 0104 ₁₆ 0105 ₁₆ | CAN0 message box 10: Identifier/DLC | | XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ |
| 0106 ₁₆ 0107 ₁₆ 0108 ₁₆ 0109 ₁₆ 010A ₁₆ 010B ₁₆ 010C ₁₆ 010D ₁₆ | CAN0 message box 10 : Data field | | XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ |
| 010E ₁₆ 010F ₁₆ | CAN0 message box 10 : Time stamp | | XX ₁₆ XX ₁₆ |
| 0110 ₁₆ 0111 ₁₆ 0112 ₁₆ 0113 ₁₆ 0114 ₁₆ 0115 ₁₆ | CAN0 message box 11 : Identifier/DLC | | XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ |
| 0116 ₁₆ 0117 ₁₆ 0118 ₁₆ 0119 ₁₆ 011A ₁₆ 011B ₁₆ 011C ₁₆ 011D ₁₆ | CAN0 message box 11 : Data field | | XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ |
| 011E ₁₆ 011F ₁₆ | CAN0 message box 11 : Time stamp | | XX ₁₆ XX ₁₆ |
| 0120 ₁₆ 0121 ₁₆ 0122 ₁₆ 0123 ₁₆ 0124 ₁₆ 0125 ₁₆ | CAN0 message box 12: Identifier/DLC | | XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ |
| 0126 ₁₆ 0127 ₁₆ 0128 ₁₆ 0129 ₁₆ 012A ₁₆ 012B ₁₆ 012C ₁₆ 012D ₁₆ | CAN0 message box 12: Data field | | XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ |
| 012E ₁₆ 012F ₁₆ | CAN0 message box 12 : Time stamp | | XX ₁₆ XX ₁₆ |
| 0130 ₁₆ 0131 ₁₆ 0132 ₁₆ 0133 ₁₆ 0134 ₁₆ 0135 ₁₆ | CAN0 message box 13 : Identifier/DLC | | XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ |
| 0136 ₁₆ 0137 ₁₆ 0138 ₁₆ 0139 ₁₆ 013A ₁₆ 013B ₁₆ 013C ₁₆ 013D ₁₆ | CAN0 message box 13 : Data field | | XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ |
| 013E ₁₆ 013F ₁₆ | CAN0 message box 13 : Time stamp | | XX ₁₆ XX ₁₆ |

Note 1: The blank areas are reserved and cannot be used by users.

X : Undefined

Table 4.6 SFR Information (6)

| Address | Register | Symbol | After reset |
|--|--|--------|--|
| 0140 ₁₆ 0141 ₁₆ 0142 ₁₆ 0143 ₁₆ 0144 ₁₆ 0145 ₁₆ | CAN0 message box 14: Identifier/DLC | | XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ |
| 0146 ₁₆ 0147 ₁₆ 0148 ₁₆ 0149 ₁₆ 014A ₁₆ 014B ₁₆ 014C ₁₆ 014D ₁₆ | CAN0 message box 14 : Data field | | XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ |
| 014E ₁₆ 014F ₁₆ | CAN0 message box 14 : Time stamp | | XX ₁₆ XX ₁₆ |
| 0150 ₁₆ 0151 ₁₆ 0152 ₁₆ 0153 ₁₆ 0154 ₁₆ 0155 ₁₆ | CAN0 message box 15 : Identifier/DLC | | XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ |
| 0156 ₁₆ 0157 ₁₆ 0158 ₁₆ 0159 ₁₆ 015A ₁₆ 015B ₁₆ 015C ₁₆ 015D ₁₆ | CAN0 message box 15 : Data field | | XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ |
| 015E ₁₆ 015F ₁₆ | CAN0 message box 15 : Time stamp | | XX ₁₆ XX ₁₆ |
| 0160 ₁₆ 0161 ₁₆ 0162 ₁₆ 0163 ₁₆ 0164 ₁₆ 0165 ₁₆ | CAN0 global mask register | COGMR | XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ |
| 0166 ₁₆ 0167 ₁₆ 0168 ₁₆ 0169 ₁₆ 016A ₁₆ 016B ₁₆ | CAN0 local mask A register | COLMAR | XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ |
| 016C ₁₆ 016D ₁₆ 016E ₁₆ 016F ₁₆ 0170 ₁₆ 0171 ₁₆ | CAN0 local mask B register | COLMBR | XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ XX ₁₆ |
| ≈ | | | ≈ |
| 01B3 ₁₆ 01B4 ₁₆ | Flash memory control register 4 (Note 2) | FMR4 | 0100000X ₂ |
| 01B5 ₁₆ 01B6 ₁₆ | Flash memory control register 1 (Note 2) | FMR1 | 000XXX0X ₂ |
| 01B7 ₁₆ | Flash memory control register 0 (Note 2) | FMR0 | 01 ₁₆ |
| ≈ | | | ≈ |
| 01FD ₁₆ 01FE ₁₆ 01FF ₁₆ | | | |

Note 1: The blank areas are reserved and cannot be used by users.

Note 2: This register is included in the flash memory version.

X : Undefined

Table 4.7 SFR Information (7)

| Address | Register | Symbol | After reset |
|--------------------|---|----------|-----------------------|
| 0200 ₁₆ | CAN0 message control register 0 | C0MCTL0 | 00 ₁₆ |
| 0201 ₁₆ | CAN0 message control register 1 | C0MCTL1 | 00 ₁₆ |
| 0202 ₁₆ | CAN0 message control register 2 | C0MCTL2 | 00 ₁₆ |
| 0203 ₁₆ | CAN0 message control register 3 | C0MCTL3 | 00 ₁₆ |
| 0204 ₁₆ | CAN0 message control register 4 | C0MCTL4 | 00 ₁₆ |
| 0205 ₁₆ | CAN0 message control register 5 | C0MCTL5 | 00 ₁₆ |
| 0206 ₁₆ | CAN0 message control register 6 | C0MCTL6 | 00 ₁₆ |
| 0207 ₁₆ | CAN0 message control register 7 | C0MCTL7 | 00 ₁₆ |
| 0208 ₁₆ | CAN0 message control register 8 | C0MCTL8 | 00 ₁₆ |
| 0209 ₁₆ | CAN0 message control register 9 | C0MCTL9 | 00 ₁₆ |
| 020A ₁₆ | CAN0 message control register 10 | C0MCTL10 | 00 ₁₆ |
| 020B ₁₆ | CAN0 message control register 11 | C0MCTL11 | 00 ₁₆ |
| 020C ₁₆ | CAN0 message control register 12 | C0MCTL12 | 00 ₁₆ |
| 020D ₁₆ | CAN0 message control register 13 | C0MCTL13 | 00 ₁₆ |
| 020E ₁₆ | CAN0 message control register 14 | C0MCTL14 | 00 ₁₆ |
| 020F ₁₆ | CAN0 message control register 15 | C0MCTL15 | 00 ₁₆ |
| 0210 ₁₆ | CAN0 control register | C0CTLR | X000000 ₁₂ |
| 0211 ₁₆ | | | XX0X0000 ₂ |
| 0212 ₁₆ | CAN0 status register | C0STR | 00 ₁₆ |
| 0213 ₁₆ | | | X000000 ₁₂ |
| 0214 ₁₆ | CAN0 slot status register | C0SSTR | 00 ₁₆ |
| 0215 ₁₆ | | | 00 ₁₆ |
| 0216 ₁₆ | CAN0 interrupt control register | C0ICR | 00 ₁₆ |
| 0217 ₁₆ | | | 00 ₁₆ |
| 0218 ₁₆ | CAN0 extended ID register | C0IDR | 00 ₁₆ |
| 0219 ₁₆ | | | 00 ₁₆ |
| 021A ₁₆ | CAN0 configuration register | C0CONR | XX ₁₆ |
| 021B ₁₆ | | | XX ₁₆ |
| 021C ₁₆ | CAN0 receive error count register | C0RECR | 00 ₁₆ |
| 021D ₁₆ | CAN0 transmit error count register | C0TECR | 00 ₁₆ |
| 021E ₁₆ | CAN0 time stamp register | C0TSR | 00 ₁₆ |
| 021F ₁₆ | | | 00 ₁₆ |
| 0242 ₁₆ | CAN0 acceptance filter support register | C0AFS | XX ₁₆ |
| 0243 ₁₆ | | | XX ₁₆ |
| 025A ₁₆ | Three-phase protect control register | TPRC | 00 ₁₆ |
| 025B ₁₆ | | | |
| 025C ₁₆ | On-chip oscillator control register | ROCR | 0000010 ₁₂ |
| 025D ₁₆ | Pin assignment control register | PACR | 00 ₁₆ |
| 025E ₁₆ | Peripheral clock select register | PCLKR | 0000001 ₁₂ |
| 025F ₁₆ | CAN0 clock select register | CCLKR | 00 ₁₆ |
| 02E0 ₁₆ | I ² C0 data-shift register | S00 | XX ₁₆ |
| 02E1 ₁₆ | | | |
| 02E2 ₁₆ | I ² C0 address register | S0D0 | 00 ₁₆ |
| 02E3 ₁₆ | I ² C0 control register 0 | S1D0 | 00 ₁₆ |
| 02E4 ₁₆ | I ² C0 clock control register | S20 | 00 ₁₆ |
| 02E5 ₁₆ | I ² C0 start/stop condition control register | S2D0 | 00011010 ₂ |
| 02E6 ₁₆ | I ² C0 control register 1 | S3D0 | 00110000 ₂ |
| 02E7 ₁₆ | I ² C0 control register 2 | S4D0 | 00 ₁₆ |
| 02E8 ₁₆ | I ² C0 status register | S10 | 0001000X ₂ |
| 02FD ₁₆ | | | |
| 02FE ₁₆ | | | |
| 02FF ₁₆ | | | |

Note 1: The blank areas are reserved and cannot be used by users.

X : Undefined

Table 4.8 SFR Information (8)

| Address | Register | Symbol | After reset |
|--|---|-------------|--------------------------------------|
| 0300 ₁₆ 0301 ₁₆ | Time measurement, Pulse generation register 0 | G1TM0,G1PO0 | XX ₁₆ XX ₁₆ |
| 0302 ₁₆ 0303 ₁₆ | Time measurement, Pulse generation register 1 | G1TM1,G1PO1 | XX ₁₆ XX ₁₆ |
| 0304 ₁₆ 0305 ₁₆ | Time measurement, Pulse generation register 2 | G1TM2,G1PO2 | XX ₁₆ XX ₁₆ |
| 0306 ₁₆ 0307 ₁₆ | Time measurement, Pulse generation register 3 | G1TM3,G1PO3 | XX ₁₆ XX ₁₆ |
| 0308 ₁₆ 0309 ₁₆ | Time measurement, Pulse generation register 4 | G1TM4,G1PO4 | XX ₁₆ XX ₁₆ |
| 030A ₁₆ 030B ₁₆ | Time measurement, Pulse generation register 5 | G1TM5,G1PO5 | XX ₁₆ XX ₁₆ |
| 030C ₁₆ 030D ₁₆ | Time measurement, Pulse generation register 6 | G1TM6,G1PO6 | XX ₁₆ XX ₁₆ |
| 030E ₁₆ 030F ₁₆ | Time measurement, Pulse generation register 7 | G1TM7,G1PO7 | XX ₁₆ XX ₁₆ |
| 0310 ₁₆ | Pulse generation control register 0 | G1POCR0 | 0X00XX00 ₂ |
| 0311 ₁₆ | Pulse generation control register 1 | G1POCR1 | 0X00XX00 ₂ |
| 0312 ₁₆ | Pulse generation control register 2 | G1POCR2 | 0X00XX00 ₂ |
| 0313 ₁₆ | Pulse generation control register 3 | G1POCR3 | 0X00XX00 ₂ |
| 0314 ₁₆ | Pulse generation control register 4 | G1POCR4 | 0X00XX00 ₂ |
| 0315 ₁₆ | Pulse generation control register 5 | G1POCR5 | 0X00XX00 ₂ |
| 0316 ₁₆ | Pulse generation control register 6 | G1POCR6 | 0X00XX00 ₂ |
| 0317 ₁₆ | Pulse generation control register 7 | G1POCR7 | 0X00XX00 ₂ |
| 0318 ₁₆ | Time measurement control register 0 | G1TMCR0 | 00 ₁₆ |
| 0319 ₁₆ | Time measurement control register 1 | G1TMCR1 | 00 ₁₆ |
| 031A ₁₆ | Time measurement control register 2 | G1TMCR2 | 00 ₁₆ |
| 031B ₁₆ | Time measurement control register 3 | G1TMCR3 | 00 ₁₆ |
| 031C ₁₆ | Time measurement control register 4 | G1TMCR4 | 00 ₁₆ |
| 031D ₁₆ | Time measurement control register 5 | G1TMCR5 | 00 ₁₆ |
| 031E ₁₆ | Time measurement control register 6 | G1TMCR6 | 00 ₁₆ |
| 031F ₁₆ | Time measurement control register 7 | G1TMCR7 | 00 ₁₆ |
| 0320 ₁₆ 0321 ₁₆ | Base timer register | G1BT | XX ₁₆ XX ₁₆ |
| 0322 ₁₆ | Base timer control register 0 | G1BCR0 | 00 ₁₆ |
| 0323 ₁₆ | Base timer control register 1 | G1BCR1 | 00 ₁₆ |
| 0324 ₁₆ | Time measurement prescale register 6 | G1TPR6 | 00 ₁₆ |
| 0325 ₁₆ | Time measurement prescale register 7 | G1TPR7 | 00 ₁₆ |
| 0326 ₁₆ | Function enable register | G1FE | 00 ₁₆ |
| 0327 ₁₆ | Function select register | G1FS | 00 ₁₆ |
| 0328 ₁₆ 0329 ₁₆ | Base timer reset register | G1BTRR | XX ₁₆ XX ₁₆ |
| 032A ₁₆ 032B ₁₆ 032C ₁₆ 032D ₁₆ 032E ₁₆ 032F ₁₆ | Count source division register | G1DV | 00 ₁₆ |
| 0330 ₁₆ | Interrupt request register | G1IR | XX ₁₆ |
| 0331 ₁₆ | Interrupt enable register 0 | G1IE0 | 00 ₁₆ |
| 0332 ₁₆ | Interrupt enable register 1 | G1IE1 | 00 ₁₆ |
| 0333 ₁₆ 0334 ₁₆ 0335 ₁₆ 0336 ₁₆ 0337 ₁₆ 0338 ₁₆ 0339 ₁₆ 033A ₁₆ 033B ₁₆ 033C ₁₆ 033D ₁₆ | | | |
| 033E ₁₆ | NMI digital debounce register | NDDR | FF ₁₆ |
| 033F ₁₆ | Port P17 digital debounce register | P17DDR | FF ₁₆ |

Note 1: The blank areas are reserved and cannot be used by users.

X : Undefined

Table 4.9 SFR Information (9)

| Address | Register | Symbol | After reset |
|--------------------|---|--------|-----------------------|
| 0340 ₁₆ | | | |
| 0341 ₁₆ | | | |
| 0342 ₁₆ | Timer A1-1 register | TA11 | XX ₁₆ |
| 0343 ₁₆ | | | XX ₁₆ |
| 0344 ₁₆ | Timer A2-1 register | TA21 | XX ₁₆ |
| 0345 ₁₆ | | | XX ₁₆ |
| 0346 ₁₆ | Timer A4-1 register | TA41 | XX ₁₆ |
| 0347 ₁₆ | | | XX ₁₆ |
| 0348 ₁₆ | Three phase PWM control register 0 | INVC0 | 00 ₁₆ |
| 0349 ₁₆ | Three phase PWM control register 1 | INVC1 | 00 ₁₆ |
| 034A ₁₆ | Three phase output buffer register 0 | IDB0 | 00 ₁₆ |
| 034B ₁₆ | Three phase output buffer register 1 | IDB1 | 00 ₁₆ |
| 034C ₁₆ | Dead time timer | DTT | XX ₁₆ |
| 034D ₁₆ | Timer B2 Interrupt occurrence frequency set counter | ICTB2 | XX ₁₆ |
| 034E ₁₆ | Position - data - retain function control register | PDRF | XXXX0000 ₂ |
| 034F ₁₆ | | | |
| 0350 ₁₆ | | | |
| 0351 ₁₆ | | | |
| 0352 ₁₆ | | | |
| 0353 ₁₆ | | | |
| 0354 ₁₆ | | | |
| 0355 ₁₆ | | | |
| 0356 ₁₆ | | | |
| 0357 ₁₆ | | | |
| 0358 ₁₆ | Port function control register | PFCR | 00111111 ₂ |
| 0359 ₁₆ | | | |
| 035A ₁₆ | | | |
| 035B ₁₆ | | | |
| 035C ₁₆ | | | |
| 035D ₁₆ | | | |
| 035E ₁₆ | Interrupt cause select register 2 ⁽²⁾ | IFSR2A | 00XXX000 ₂ |
| 035F ₁₆ | Interrupt cause select register | IFSR | 00 ₁₆ |
| 0360 ₁₆ | SI/O3 transmit/receive register | S3TRR | XX ₁₆ |
| 0361 ₁₆ | | | |
| 0362 ₁₆ | SI/O3 control register | S3C | 01000000 ₂ |
| 0363 ₁₆ | SI/O3 bit rate register | S3BRG | XX ₁₆ |
| 0364 ₁₆ | SI/O4 transmit/receive register | S4TRR | XX ₁₆ |
| 0365 ₁₆ | | | |
| 0366 ₁₆ | SI/O4 control register | S4C | 01000000 ₂ |
| 0367 ₁₆ | SI/O4 bit rate register | S4BRG | XX ₁₆ |
| 0368 ₁₆ | | | |
| 0369 ₁₆ | | | |
| 036A ₁₆ | | | |
| 036B ₁₆ | | | |
| 036C ₁₆ | | | |
| 036D ₁₆ | | | |
| 036E ₁₆ | | | |
| 036F ₁₆ | | | |
| 0370 ₁₆ | | | |
| 0371 ₁₆ | | | |
| 0372 ₁₆ | | | |
| 0373 ₁₆ | | | |
| 0374 ₁₆ | UART2 special mode register 4 | U2SMR4 | 00 ₁₆ |
| 0375 ₁₆ | UART2 special mode register 3 | U2SMR3 | 000X0X0X ₂ |
| 0376 ₁₆ | UART2 special mode register 2 | U2SMR2 | X0000000 ₂ |
| 0377 ₁₆ | UART2 special mode register | U2SMR | X0000000 ₂ |
| 0378 ₁₆ | UART2 transmit/receive mode register | U2MR | 00 ₁₆ |
| 0379 ₁₆ | UART2 bit rate register | U2BRG | XX ₁₆ |
| 037A ₁₆ | UART2 transmit buffer register | U2TB | XX ₁₆ |
| 037B ₁₆ | | | XX ₁₆ |
| 037C ₁₆ | UART2 transmit/receive control register 0 | U2C0 | 00001000 ₂ |
| 037D ₁₆ | UART2 transmit/receive control register 1 | U2C1 | 00000010 ₂ |
| 037E ₁₆ | UART2 receive buffer register | U2RB | XX ₁₆ |
| 037F ₁₆ | | | XX ₁₆ |

Note 1: The blank areas are reserved and cannot be used by users.

Note 2: Write 0 to the bit 0 after reset.

X : Undefined

Table 4.10 SFR Information (10)

| Address | Register | Symbol | After reset |
|--|---|--------|---|
| 0380 ₁₆ | Count start flag | TABSR | 00 ₁₆ |
| 0381 ₁₆ | Clock prescaler reset flag | CPSRF | 0XXXXXX ₂ |
| 0382 ₁₆ | One-shot start flag | ONSF | 00 ₁₆ |
| 0383 ₁₆ | Trigger select register | TRGSR | 00 ₁₆ |
| 0384 ₁₆ | Up-downm flag | UDF | 00 ₁₆ |
| 0385 ₁₆ | | | |
| 0386 ₁₆ 0387 ₁₆ | Timer A0 register | TA0 | XX ₁₆ XX ₁₆ |
| 0388 ₁₆ 0389 ₁₆ | Timer A1 register | TA1 | XX ₁₆ XX ₁₆ |
| 038A ₁₆ 038B ₁₆ | Timer A2 register | TA2 | XX ₁₆ XX ₁₆ |
| 038C ₁₆ 038D ₁₆ | Timer A3 register | TA3 | XX ₁₆ XX ₁₆ |
| 038E ₁₆ 038F ₁₆ | Timer A4 register | TA4 | XX ₁₆ XX ₁₆ |
| 0390 ₁₆ 0391 ₁₆ | Timer B0 register | TB0 | XX ₁₆ XX ₁₆ |
| 0392 ₁₆ 0393 ₁₆ | Timer B1 register | TB1 | XX ₁₆ XX ₁₆ |
| 0394 ₁₆ 0395 ₁₆ | Timer B2 register | TB2 | XX ₁₆ XX ₁₆ |
| 0396 ₁₆ | Timer A0 mode register | TA0MR | 00 ₁₆ |
| 0397 ₁₆ | Timer A1 mode register | TA1MR | 00 ₁₆ |
| 0398 ₁₆ | Timer A2 mode register | TA2MR | 00 ₁₆ |
| 0399 ₁₆ | Timer A3 mode register | TA3MR | 00 ₁₆ |
| 039A ₁₆ | Timer A4 mode register | TA4MR | 00 ₁₆ |
| 039B ₁₆ | Timer B0 mode register | TB0MR | 00XX0000 ₂ |
| 039C ₁₆ | Timer B1 mode register | TB1MR | 00XX0000 ₂ |
| 039D ₁₆ | Timer B2 mode register | TB2MR | 00XX0000 ₂ |
| 039E ₁₆ | Timer B2 special mode register | TB2SC | X0000000 ₂ |
| 039F ₁₆ | | | |
| 03A0 ₁₆ | UART0 transmit/receive mode register | U0MR | 00 ₁₆ |
| 03A1 ₁₆ | UART0 bit rate register | U0BRG | XX ₁₆ |
| 03A2 ₁₆ 03A3 ₁₆ | UART0 transmit buffer register | U0TB | XX ₁₆ XX ₁₆ |
| 03A4 ₁₆ | UART0 transmit/receive control register 0 | U0C0 | 00001000 ₂ |
| 03A5 ₁₆ | UART0 transmit/receive control register 1 | U0C1 | 00000010 ₂ |
| 03A6 ₁₆ 03A7 ₁₆ | UART0 receive buffer register | U0RB | XX ₁₆ XX ₁₆ |
| 03A8 ₁₆ | UART1 transmit/receive mode register | U1MR | 00 ₁₆ |
| 03A9 ₁₆ | UART1 bit rate register | U1BRG | XX ₁₆ |
| 03AA ₁₆ 03AB ₁₆ | UART1 transmit buffer register | U1TB | XX ₁₆ XX ₁₆ |
| 03AC ₁₆ | UART1 transmit/receive control register 0 | U1C0 | 00001000 ₂ |
| 03AD ₁₆ | UART1 transmit/receive control register 1 | U1C1 | 00000010 ₂ |
| 03AE ₁₆ 03AF ₁₆ | UART1 receive buffer register | U1RB | XX ₁₆ XX ₁₆ |
| 03B0 ₁₆ | UART transmit/receive control register 2 | UCON | X0000000 ₂ |
| 03B1 ₁₆ | | | |
| 03B2 ₁₆ | | | |
| 03B3 ₁₆ | | | |
| 03B4 ₁₆ 03B5 ₁₆ | CRC snoop address register | CRCSAR | XX ₁₆ 00XXXXXX ₂ |
| 03B6 ₁₆ 03B7 ₁₆ | CRC mode register | CRCMR | 0XXXXXX0 ₂ |
| 03B8 ₁₆ | DMA0 request cause select register | DM0SL | 00 ₁₆ |
| 03B9 ₁₆ | | | |
| 03BA ₁₆ 03BB ₁₆ | DMA1 request cause select register | DM1SL | 00 ₁₆ |
| 03BC ₁₆ 03BD ₁₆ | CRC data register | CRCD | XX ₁₆ XX ₁₆ |
| 03BE ₁₆ 03BF ₁₆ | CRC input register | CRCIN | XX ₁₆ |

Note 1: The blank areas are reserved and cannot be used by users.

X : Undefined

Table 4.11 SFR Information (11)

| Address | Register | Symbol | After reset |
|--|------------------------------|----------|--------------------------------------|
| 03C0 ₁₆ 03C1 ₁₆ | A/D register 0 | AD0 | XX ₁₆ XX ₁₆ |
| 03C2 ₁₆ 03C3 ₁₆ | A/D register 1 | AD1 | XX ₁₆ XX ₁₆ |
| 03C4 ₁₆ 03C5 ₁₆ | A/D register 2 | AD2 | XX ₁₆ XX ₁₆ |
| 03C6 ₁₆ 03C7 ₁₆ | A/D register 3 | AD3 | XX ₁₆ XX ₁₆ |
| 03C8 ₁₆ 03C9 ₁₆ | A/D register 4 | AD4 | XX ₁₆ XX ₁₆ |
| 03CA ₁₆ 03CB ₁₆ | A/D register 5 | AD5 | XX ₁₆ XX ₁₆ |
| 03CC ₁₆ 03CD ₁₆ | A/D register 6 | AD6 | XX ₁₆ XX ₁₆ |
| 03CE ₁₆ 03CF ₁₆ | A/D register 7 | AD7 | XX ₁₆ XX ₁₆ |
| 03D0 ₁₆ | | | |
| 03D1 ₁₆ | | | |
| 03D2 ₁₆ | A/D trigger control register | ADTRGCON | XXXX0000 ₂ |
| 03D3 ₁₆ | A/D status register 0 | ADSTAT0 | 0000X00 ₂ |
| 03D4 ₁₆ 03D5 ₁₆ | A/D control register 2 | ADCON2 | 00 ₁₆ |
| 03D6 ₁₆ | A/D control register 0 | ADCON0 | 0000XXX ₂ |
| 03D7 ₁₆ | A/D control register 1 | ADCON1 | 00 ₁₆ |
| 03D8 ₁₆ | | | |
| 03D9 ₁₆ | | | |
| 03DA ₁₆ | | | |
| 03DB ₁₆ | | | |
| 03DC ₁₆ | | | |
| 03DD ₁₆ | | | |
| 03DE ₁₆ | | | |
| 03DF ₁₆ | | | |
| 03E0 ₁₆ | Port P0 register | P0 | XX ₁₆ |
| 03E1 ₁₆ | Port P1 register | P1 | XX ₁₆ |
| 03E2 ₁₆ | Port P0 direction register | PD0 | 00 ₁₆ |
| 03E3 ₁₆ | Port P1 direction register | PD1 | 00 ₁₆ |
| 03E4 ₁₆ | Port P2 register | P2 | XX ₁₆ |
| 03E5 ₁₆ | Port P3 register | P3 | XX ₁₆ |
| 03E6 ₁₆ | Port P2 direction register | PD2 | 00 ₁₆ |
| 03E7 ₁₆ | Port P3 direction register | PD3 | 00 ₁₆ |
| 03E8 ₁₆ | | | |
| 03E9 ₁₆ | | | |
| 03EA ₁₆ | | | |
| 03EB ₁₆ | | | |
| 03EC ₁₆ | Port P6 register | P6 | XX ₁₆ |
| 03ED ₁₆ | Port P7 register | P7 | XX ₁₆ |
| 03EE ₁₆ | Port P6 direction register | PD6 | 00 ₁₆ |
| 03EF ₁₆ | Port P7 direction register | PD7 | 00 ₁₆ |
| 03F0 ₁₆ | Port P8 register | P8 | XX ₁₆ |
| 03F1 ₁₆ | Port P9 register | P9 | XX ₁₆ |
| 03F2 ₁₆ | Port P8 direction register | PD8 | 00 ₁₆ |
| 03F3 ₁₆ | Port P9 direction register | PD9 | 000X0000 ₂ |
| 03F4 ₁₆ 03F5 ₁₆ | Port P10 register | P10 | XX ₁₆ |
| 03F6 ₁₆ | Port P10 direction register | PD10 | 00 ₁₆ |
| 03F7 ₁₆ | | | |
| 03F8 ₁₆ | | | |
| 03F9 ₁₆ | | | |
| 03FA ₁₆ | | | |
| 03FB ₁₆ | | | |
| 03FC ₁₆ | Pull-up control register 0 | PUR0 | 00 ₁₆ |
| 03FD ₁₆ | Pull-up control register 1 | PUR1 | 00 ₁₆ |
| 03FE ₁₆ | Pull-up control register 2 | PUR2 | 00 ₁₆ |
| 03FF ₁₆ | Port control register | PCR | 00 ₁₆ |

Note 1: The blank areas are reserved and cannot be used by users.

X : Undefined

5. Resets

Hardware reset 1, brown-out detection reset (hardware reset 2), software reset, watchdog timer reset, and oscillation stop detection reset are implemented to reset the MCU.

5.1 Hardware Reset

Hardware reset 1 and brown-out detection reset are available as the hardware reset.

5.1.1 Hardware Reset 1

Pins, CPU, and SFRs are reset by using the $\overline{\text{RESET}}$ pin. When a low-level (“L”) signal is applied to the $\overline{\text{RESET}}$ pin while the supply voltage meets the recommended operating condition, pins, CPU, and SFRs are reset (see **Table 5.1** Pin Status When $\overline{\text{RESET}}$ Pin Level is “L”). The oscillation circuit is also reset and the on-chip oscillator starts oscillating as the CPU clock. CPU and SFRs are reset when the signal applied to the $\overline{\text{RESET}}$ pin changes from “L” to high (“H”). The MCU executes a program beginning with the address indicated by the reset vector. The internal RAM is not reset. When an “L” signal is applied to the $\overline{\text{RESET}}$ pin while writing data to the internal RAM, the content of internal RAM is undefined.

Figure 5.1 shows an example of the reset circuit. **Figure 5.2** shows a reset sequence. **Table 5.1** shows status of the other pins while the $\overline{\text{RESET}}$ pin is held “L”. **Figure 5.3** shows CPU register states after reset. Refer to **4. Special Function Register (SFR)** about SFR states after reset.

1. Reset on a stable supply voltage

- (1) Apply an “L” signal to the $\overline{\text{RESET}}$ pin
- (2) Wait $t_d(\text{ROC})$ or more
- (3) Apply an “H” signal to the $\overline{\text{RESET}}$ pin

2. Power-on reset

- (1) Apply an “L” signal to the $\overline{\text{RESET}}$ pin
- (2) Increase the supply voltage until it meets the recommended performance condition
- (3) Wait for $t_d(\text{P-R})$ or more to allow the internal power supply to stabilize
- (4) Wait $t_d(\text{ROC})$ or more
- (5) Apply an “H” signal to the $\overline{\text{RESET}}$ pin

5.1.2 Brown-Out Detection Reset (Hardware Reset 2)

Note

Brown-out detection reset in the M16C/29 Group, T-ver. and V-ver. cannot be used.

Pins, CPU, and SFR are reset by using the on-chip voltage detection circuit, which monitors the voltage applied to Vcc pin.

When the VC26 bit in the VCR2 register is set to 1 (reset level detection circuit enabled), pins, CPU, and SFR are reset as soon as the voltage applied to the Vcc pin drops to Vdet3 or below.

Then, pins, CPU, and SFR are reset as soon as the voltage applied to the Vcc pin reaches Vdet3r or above. The MCU executes the program in an address determined by the reset vector.

The MCU executes the program after detecting Vdet3r and waiting $t_d(\text{S-R})$ ms. The same pins and registers are reset by the hardware reset 1 and brown-out detection reset, and are also placed in the same reset state.

The MCU cannot exit stop mode by brown-out detection reset.

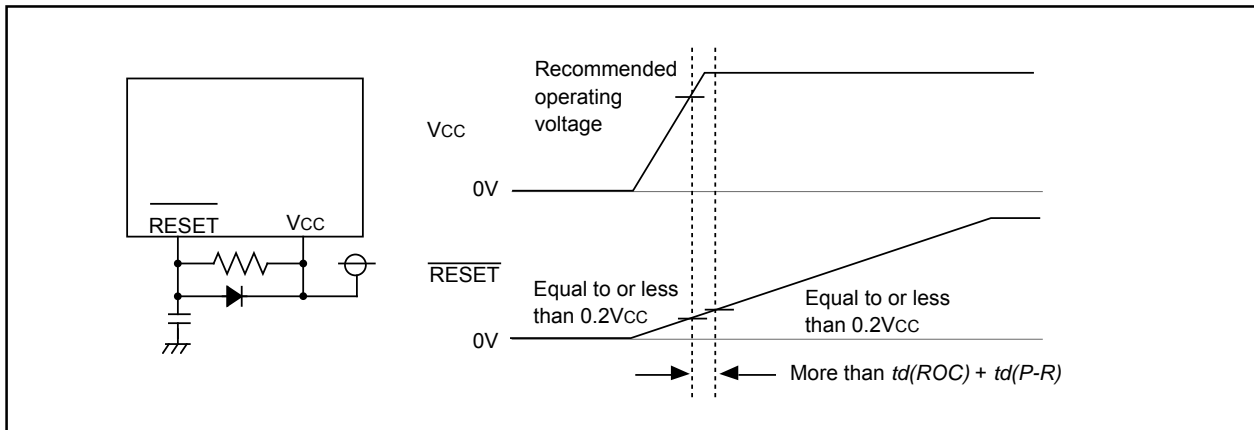


Figure 5.1 Example Reset Circuit

5.2 Software Reset

The MCU resets its pins, CPU, and SFRs when the PM03 bit in the PM0 register is set to 1 (reset) and the MCU executes a program in an address indicated by the reset vector. Then the on-chip oscillator is selected as the CPU clock.

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

5.3 Watchdog Timer Reset

The MCU resets its pins, CPU, and SFRs when the PM12 bit in the PM1 register is set to 1 (watchdog timer reset) and the watchdog timer underflows. The MCU executes a program in an address indicated by the reset vector. Then the on-chip oscillator is selected as the CPU clock.

The watchdog timer reset does not reset some portions of the SFRs. Refer to **4. Special Function Registers (SFRs)** for details.

5.4 Oscillation Stop Detection Reset

The MCU resets its pins, CPU, and SFRs and stops if the main clock stop is detected when the CM20 bit in the CM2 register is set to 1 (oscillation stop, re-oscillation detection function enabled) and the CM27 bit in the CM2 register is 0 (reset at oscillation stop detection). Refer to the section **7.8 oscillation stop, re-oscillation detection function** for details.

The oscillation stop detection reset does not reset some portions of the SFRs. Refer to **4. Special Function Registers (SFRs)**.

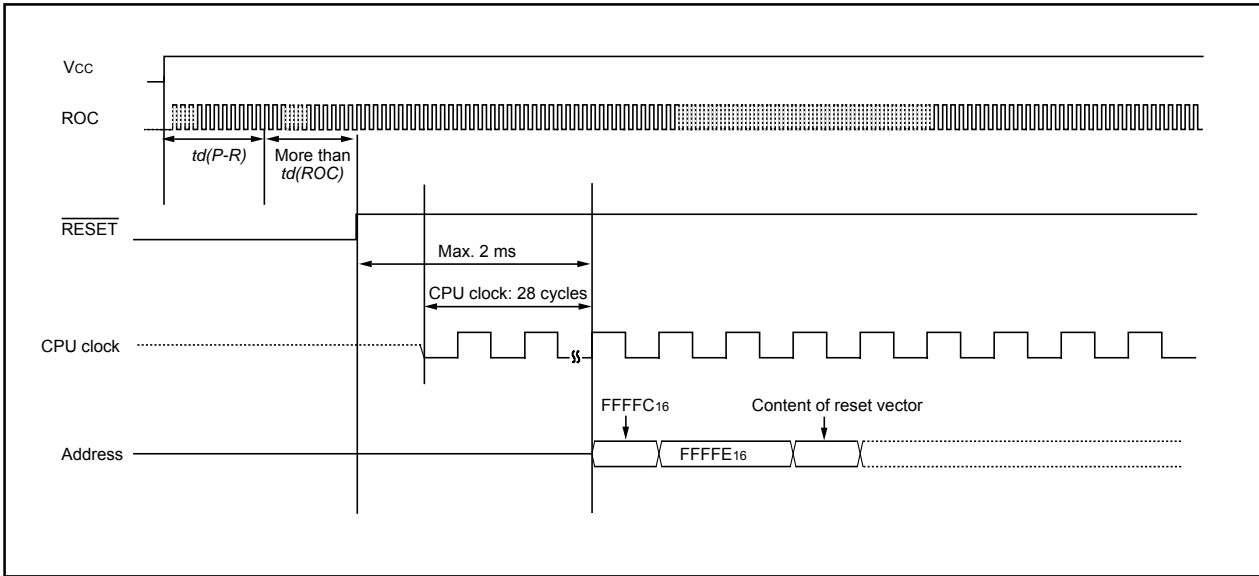


Figure 5.2 Reset Sequence

Table 5.1 Pin Status When RESET Pin Level is “L”

| Pin name | Status |
|------------------------|-----------------------------|
| P0 to P3, P6 to P10 | Input port (high impedance) |

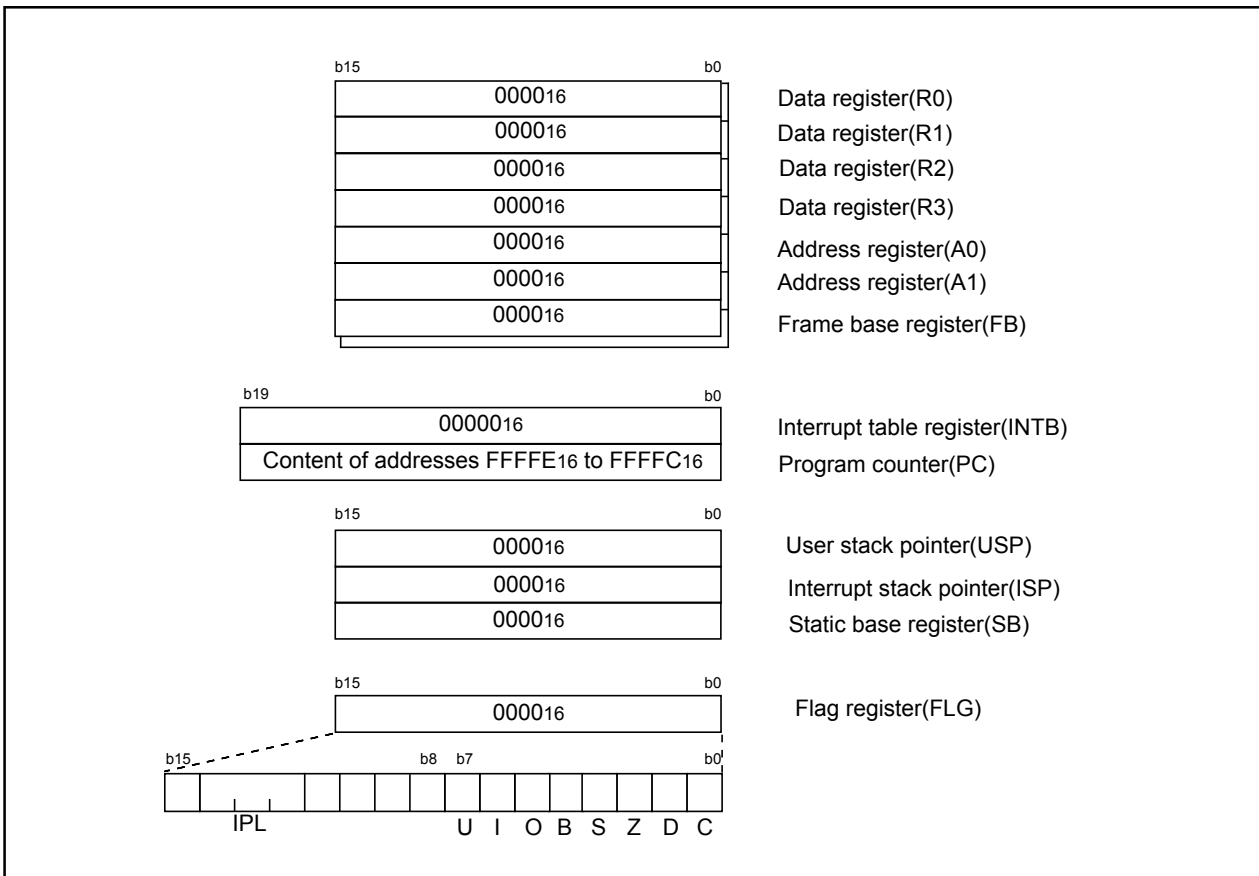


Figure 5.3 CPU Register Status After Reset

5.5 Voltage Detection Circuit

Note

VCC = 5 V is assumed in **5.5 Voltage Detection Circuit**.

Voltage detection circuit in the M16C/29 Group, T-ver. and V-ver. cannot be used.

The voltage detection circuit has the reset level detection circuit and the low voltage detection circuit. The reset level detection circuit monitors the voltage applied to the VCC pin. The MCU is reset if the reset level detection circuit detects VCC is Vdet3 or below. Use bits VC27 and VC26 in the VCR2 register to determine whether the individual circuit is enabled.

Use the reset level detection circuit for brown-out detection reset.

The low voltage detection circuit also monitors the voltage applied to the VCC pin. The low voltage detection circuit use the VC13 bit in the VCR1 register to detect VCC is above or below Vdet4. The low voltage detection interrupt can be used in the voltage detection circuit.

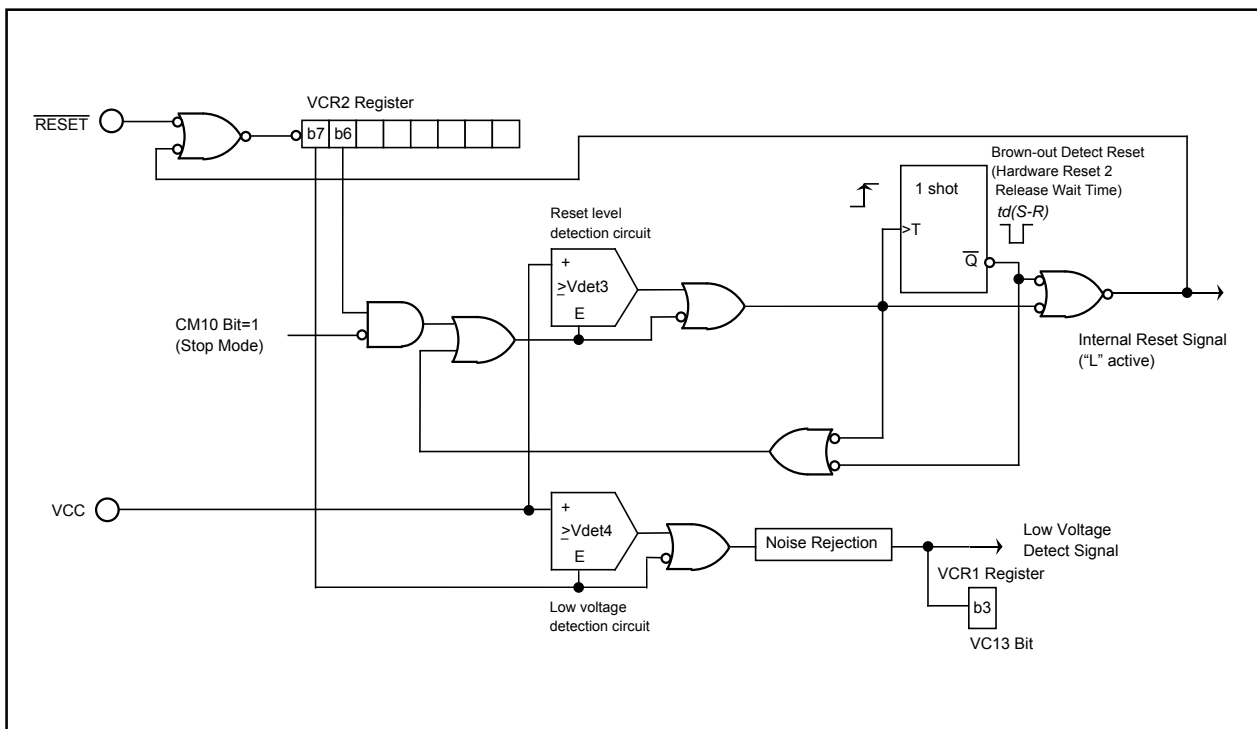


Figure 5.4 Voltage Detection Circuit Block

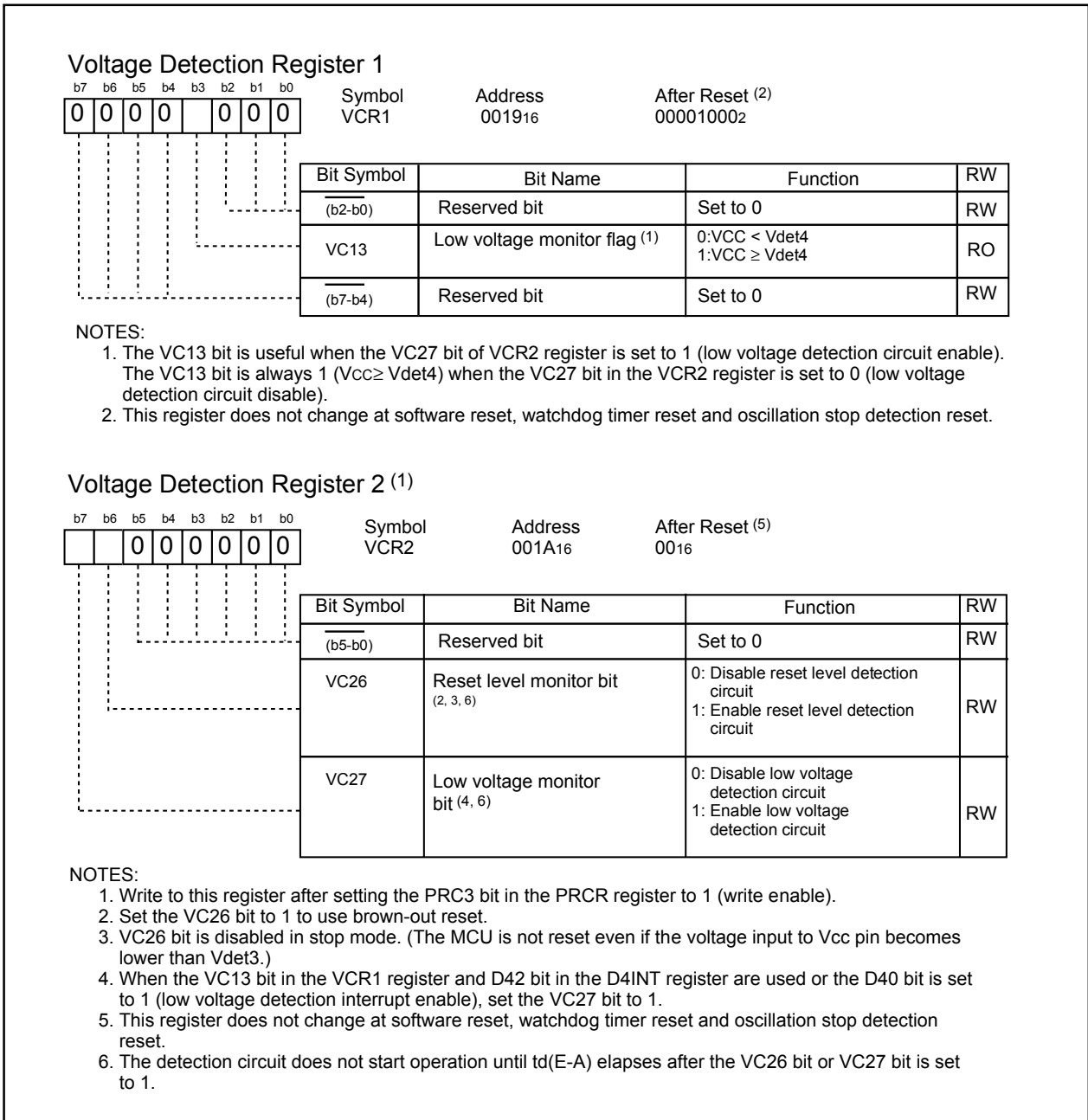


Figure 5.5 VCR1 Register and VCR2 Register

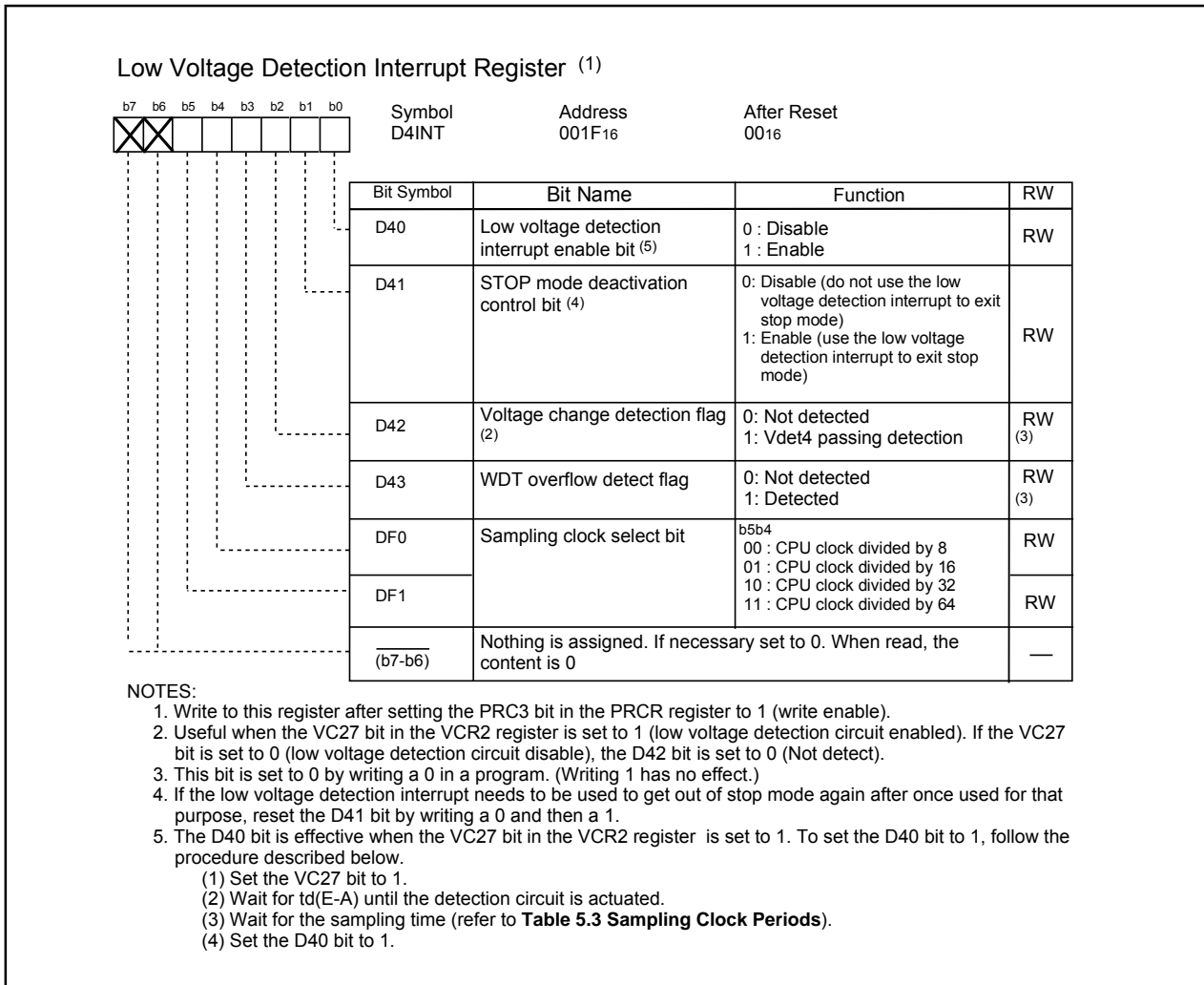


Figure 5.6 D4INT Register

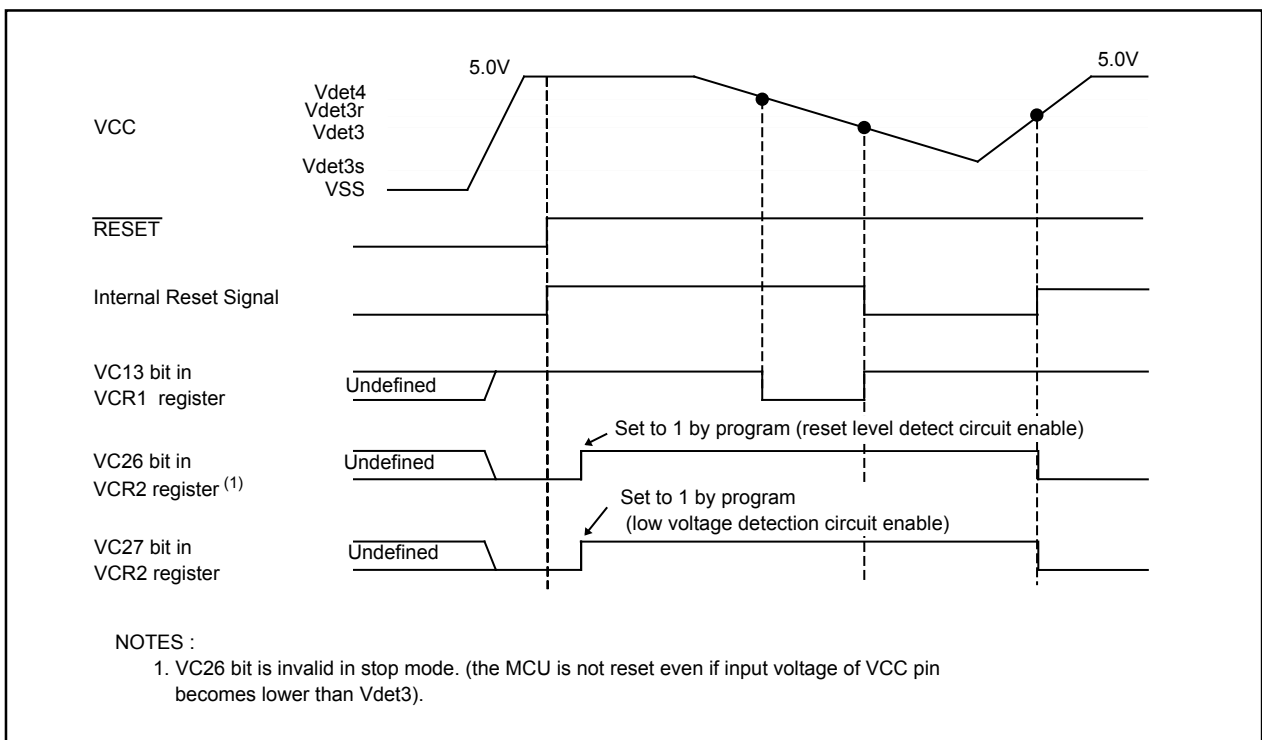


Figure 5.7 Typical Operation of Brown-Out Detection Reset (Hardware Reset 2)

5.5.1 Low Voltage Detection Interrupt

If the D40 bit in the D4INT register is set to 1 (low voltage detection interrupt enabled), a low voltage detection interrupt request is generated when voltage applied to the VCC pin is above or below Vdet4. The low voltage detection interrupt shares the same interrupt vector with watchdog timer interrupt and oscillation stop, re-oscillation detection interrupt.

Set the D41 bit in the D4INT register to 1 (enabled) to use the low voltage detection interrupt to exit stop mode, set the D41 bit in the D4INT register to 1 (enable).

The D42 bit in the D4INT register is set to 1 (above or below Vdet4 detected) as soon as voltage applied to the VCC pin goes above or below Vdet4 due to the voltage change. When the D42 bit setting changes 0 to 1, a low voltage detection interrupt is generated. Set the D42 bit to 0 (not detected) by program. However, when the D41 bit is set to 1 and the MCU is in stop mode, a low voltage detection interrupt request is generated, regardless of the D42 bit setting, if voltage applied to the VCC pin is detected to rise above or drop below Vdet4. The MCU then exits stop mode.

Table 5.2 shows how a low voltage detection interrupt request is generated.

Bits DF1 and DF0 in the D4INT register determine sampling period that detects voltage applied to the VCC pin rises above or drops below Vdet4. **Table 5.3** shows sampling periods.

Table 5.2 Voltage Detection Interrupt Request Generation Conditions

| Operation Mode | VC27 bit | D40 bit | D41 bit | D42 bit | CM02 bit | VC13 bit |
|--------------------------|----------|---------|---------|------------|----------|------------|
| Normal operation mode(1) | 1 | 1 | — | 0 to 1 | — | 0 to 1 (3) |
| | | | | 1 to 0 (3) | | |
| Wait mode (2) | | | — | 0 to 1 | 0 | 0 to 1 (3) |
| | | | | — | 1 | 1 to 0 (3) |
| Stop mode (2) | | | 1 | — | 0 | 0 to 1 |
| | | | | | | 0 to 1 |

— : 0 or 1

NOTES:

1. The status except the wait mode and stop mode is handled as the normal mode. (Refer to **7. Clock generating circuit**)
2. Refer to **5.5.2 Limitations on stop mode** and **5.5.3 Limitations on wait mode**.
3. An interrupt request for voltage reduction is generated a sampling time after the value of the VC13 bit has changed. Refer to the **Figure 5.9** for details.

Table 5.3 Sampling Clock Periods

| CPU clock (MHz) | Sampling clock (μs) | | | |
|-----------------|---|--|--|--|
| | DF1 to DF0=00 (CPU clock divided by 8) | DF1 to DF0=01 (CPU clock divided by 16) | DF1 to DF0=10 (CPU clock divided by 32) | DF1 to DF0=11 (CPU clock divided by 64) |
| 16 | 3.0 | 6.0 | 12.0 | 24.0 |

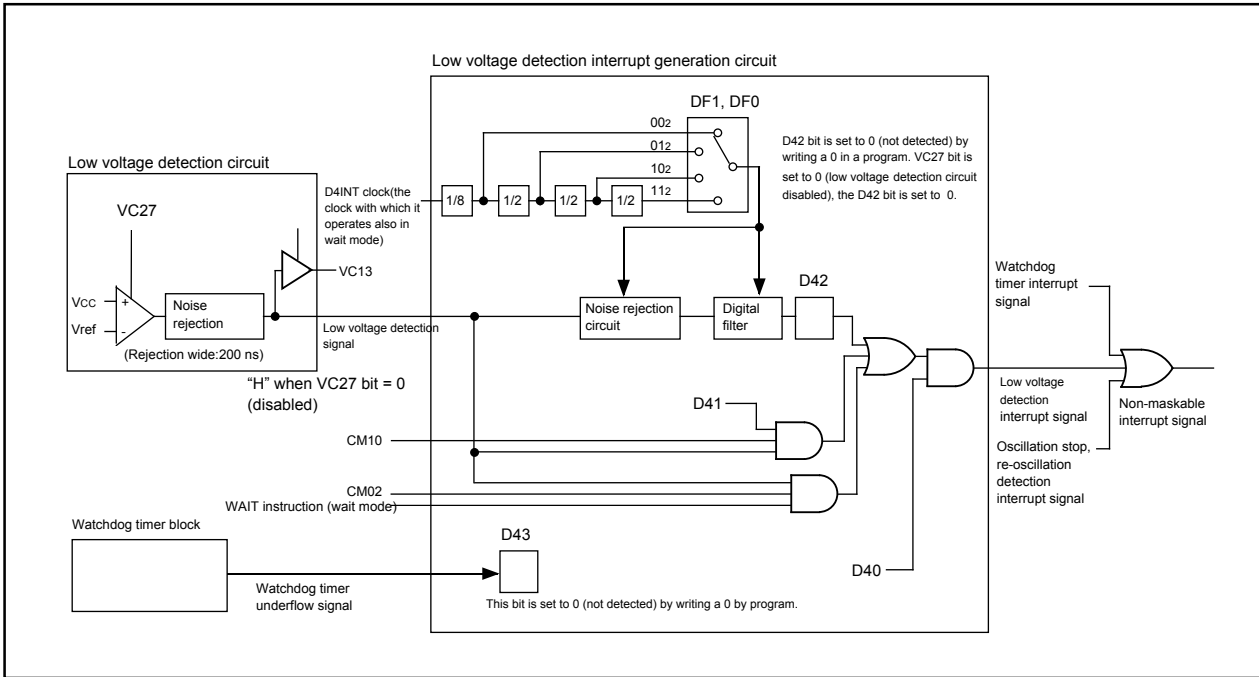


Figure 5.8 Low Voltage Detection Interrupt Generation Block

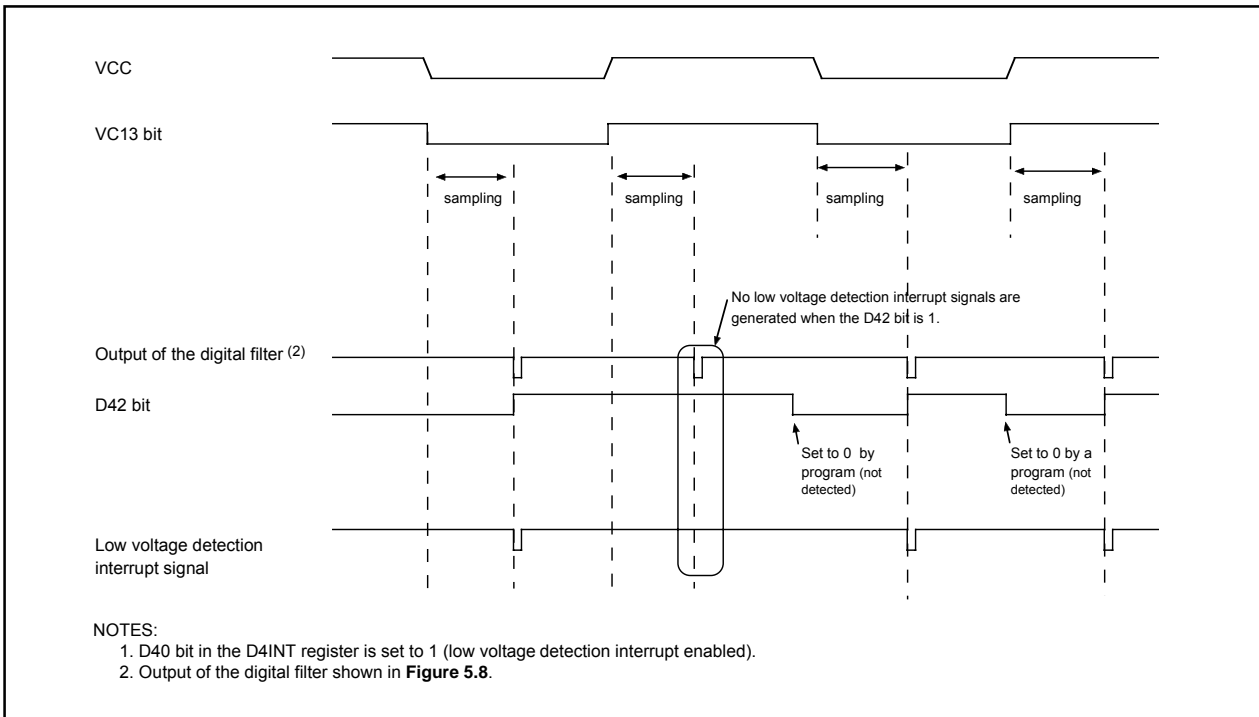


Figure 5.9 Low voltage Detection Interrupt Generation Circuit Operation Example

5.5.2. Limitations on Stop Mode

When all the conditions below are met, the low voltage detection interrupt is generated and the MCU exits stop mode as soon as the CM10 bit in the CM1 register is set to 1 (all clocks stopped).

- the VC27 bit in the VCR2 register is set to 1 (low voltage detection circuit enabled)
- the D40 bit in the D4INT register is set to 1 (low voltage detection interrupt enabled)
- the D41 bit in the D4INT register is set to 1 (low voltage detection interrupt is used to exit stop mode)
- the voltage applied to the Vcc pin is higher than Vdet4 (the VC13 bit in the VCR1 register is 1)

Set the CM10 bit to 1 when the VC13 bit is set to set to 0 ($V_{CC} < V_{det4}$), if the MCU is configured to enter stop mode when voltage applied to the Vcc pin drops Vdet4 or below and to exit stop mode when the voltage applied rises to Vdet4 or above.

5.5.3. Limitations on WAIT Instruction

When all the conditions below are met, the low voltage detection interrupt is generated and the MCU exits wait mode as soon as WAIT instruction is executed.

- the CM02 bit in the CM0 register is set to 1 (stop peripheral function clock)
- the VC27 bit in the VCR2 register is set to 1 (low voltage detection circuit enabled)
- the D40 bit in the D4INT register is set to 1 (low voltage detection interrupt enabled)
- the D41 bit in the D4INT register is set to 1 (low voltage detection interrupt is used to exit wait mode)
- the voltage applied to the Vcc pin is higher than Vdet4 (the VC13 bit in the VCR1 register is 1)

Execute the WAIT instruction when the VC13 bit is set to set to 0 ($V_{CC} < V_{det4}$), if the MCU is configured to enter wait mode when voltage applied to the Vcc pin drops Vdet4 or below and to exit wait mode when the voltage applied rises to Vdet4 or above.

6. Processor Mode

The MCU supports single-chip mode only. **Figures 6.1 and 6.2** show the associated registers.

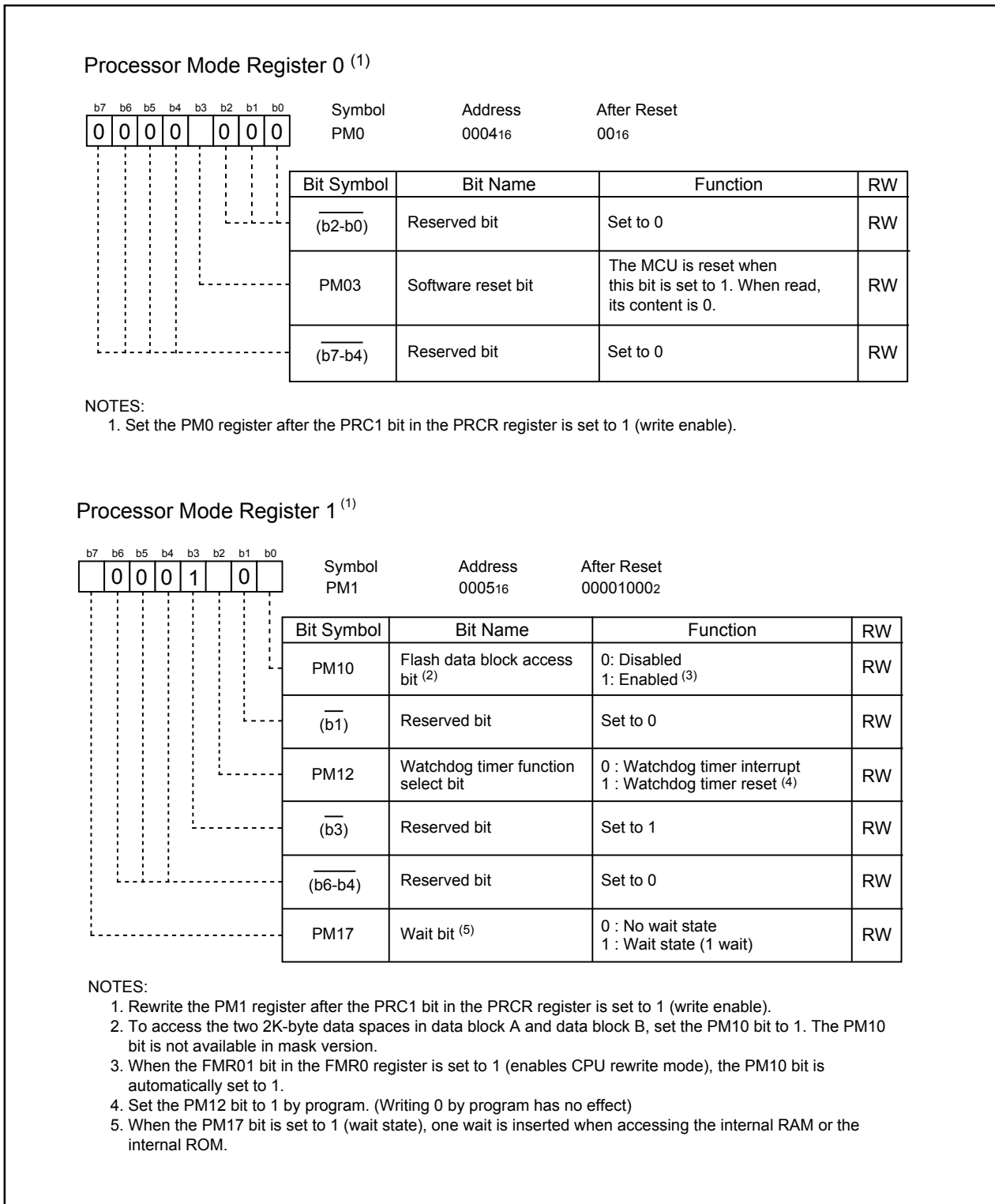


Figure 6.1 PM0 Register and PM1 Register

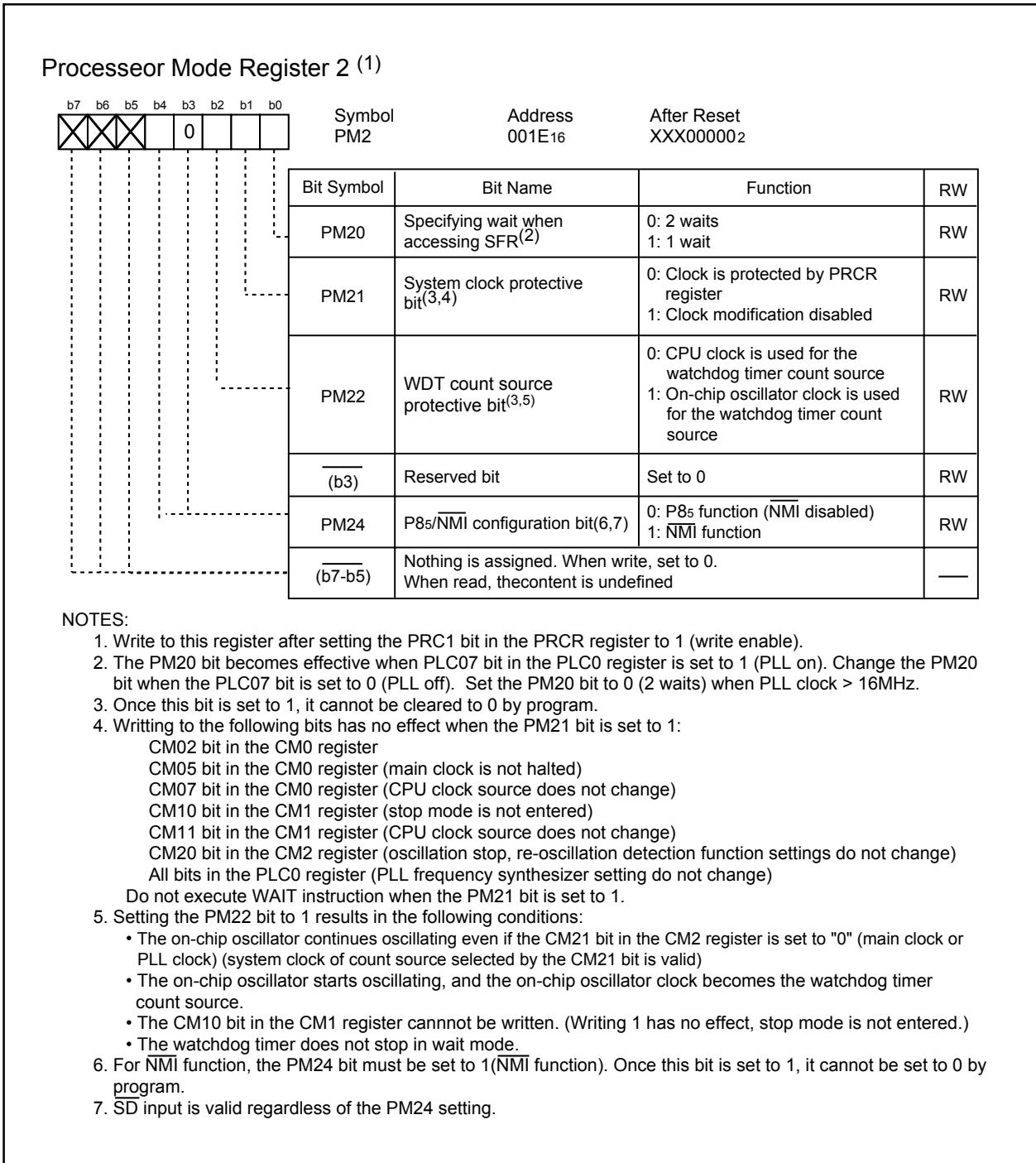


Figure 6.2 PM2 Register

The internal bus consists of CPU bus, memory bus, and peripheral bus. Bus Interface Unit (BIU) is used to interfere with CPU, ROM/RAM, and peripheral functions by controlling CPU bus, memory bus, and peripheral bus. **Figure 6.3** shows the block diagram of the internal bus.

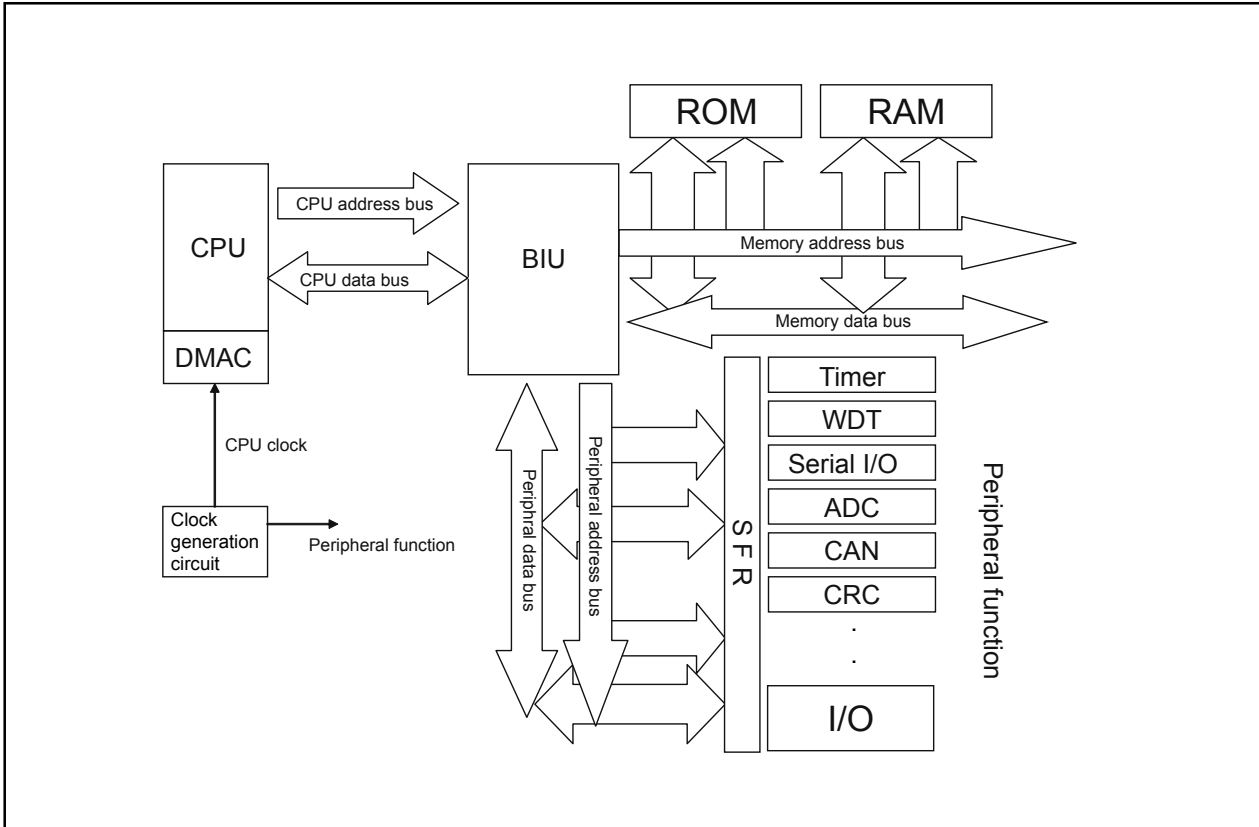


Figure 6.3 Bus Block Diagram

The number of bus cycle varies by the internal bus. Table 6.1 lists the accessible area and bus cycle.

Table 6.1 Accessible Area and Bus Cycle

| | Accessible Area | Bus Cycle |
|---------|------------------------|--------------------|
| SFR | PM20 bit = 0 (2 waits) | 3 CPU clock cycles |
| | PM20 bit = 1 (1 wait) | 2 CPU clock cycles |
| ROM/RAM | PM17 bit = 0 (no wait) | 1 CPU clock cycle |
| | PM17 bit = 1 (1 wait) | 2 CPU clock cycles |

7. Clock Generation Circuit

The clock generation circuit contains four oscillator circuits as follows:

- (1) Main clock oscillation circuit
- (2) Sub clock oscillation circuit
- (3) Variable on-chip oscillators
- (4) PLL frequency synthesizer

Table 7.1 lists the specifications of the clock generation circuit. **Figure 7.1** shows the clock generation circuit. **Figures 7.2** to **7.7** show clock-associated registers.

Table 7.1 Clock Generation Circuit Specifications

| Item | Main Clock Oscillation Circuit | Sub Clock Oscillation Circuit | Variable On-chip Oscillator | PLL Frequency Synthesizer |
|------------------------------------|--|---|---|--|
| Use of clock | - CPU clock source - Peripheral function clock source | - CPU clock source - Timer A, B's clock source | - CPU clock source - Peripheral function clock source - CPU and peripheral function clock sources when the main clock stops oscillating | - CPU clock source - Peripheral function clock source |
| Clock frequency | 0 to 20 MHz | 32.768 kHz | Selectable source frequency: f ₁ (ROC), f ₂ (ROC), f ₃ (ROC) Selectable divider: by 2, by 4, by 8 | 10 to 20 MHz |
| Usable oscillator | - Ceramic oscillator - Crystal oscillator | - Crystal oscillator | _____ | _____ |
| Pins to connect oscillator | XIN, XOUT | XCIN, XCOUT | _____ | _____ |
| Oscillation stop, restart function | Available | Available | Available | Available |
| Oscillator status after reset | Oscillating | Stopped | Oscillating (CPU clock source) | Stopped |
| Other | Externally derived clock can be input | | _____ | _____ |

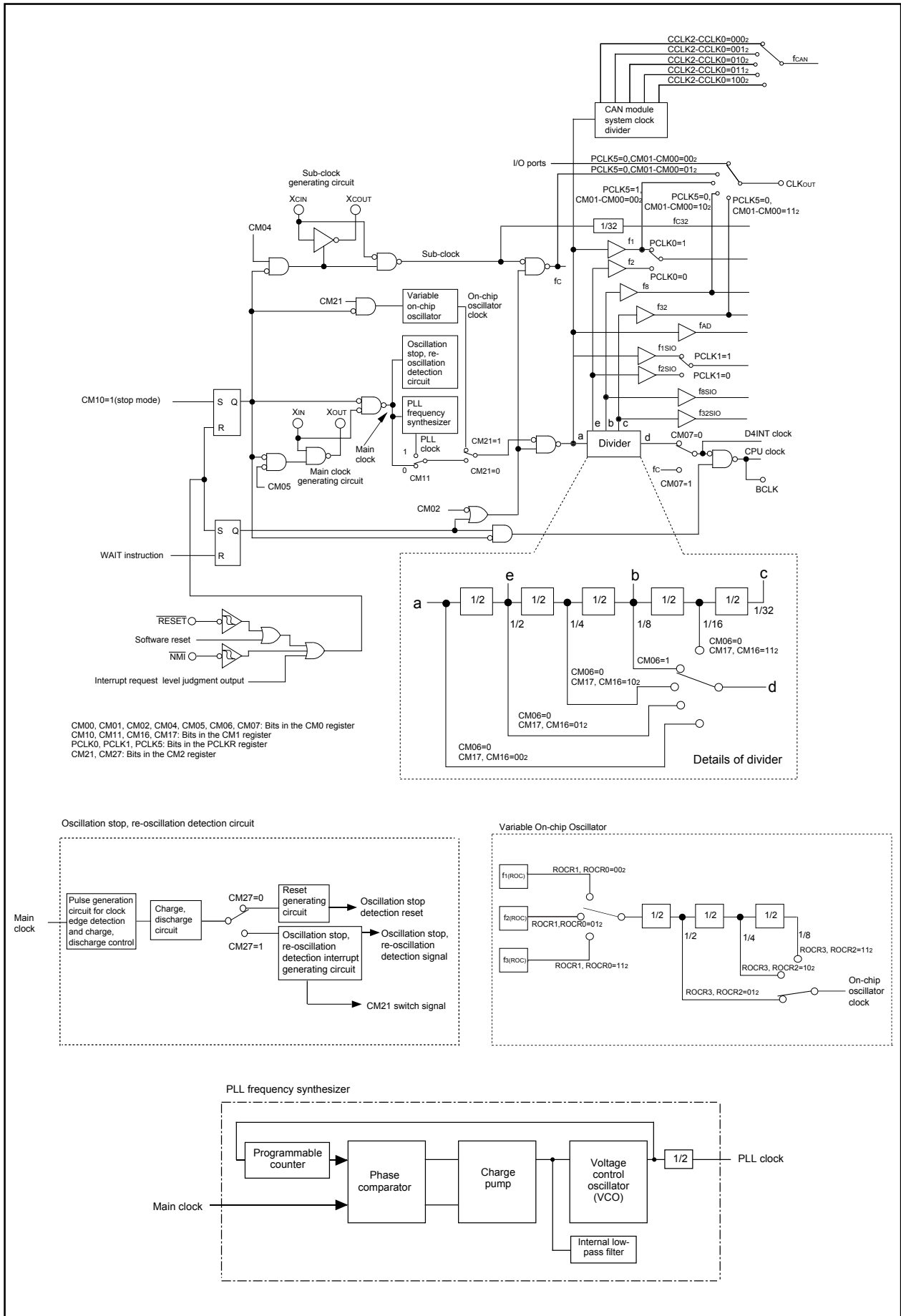


Figure 7.1 Clock Generation Circuit

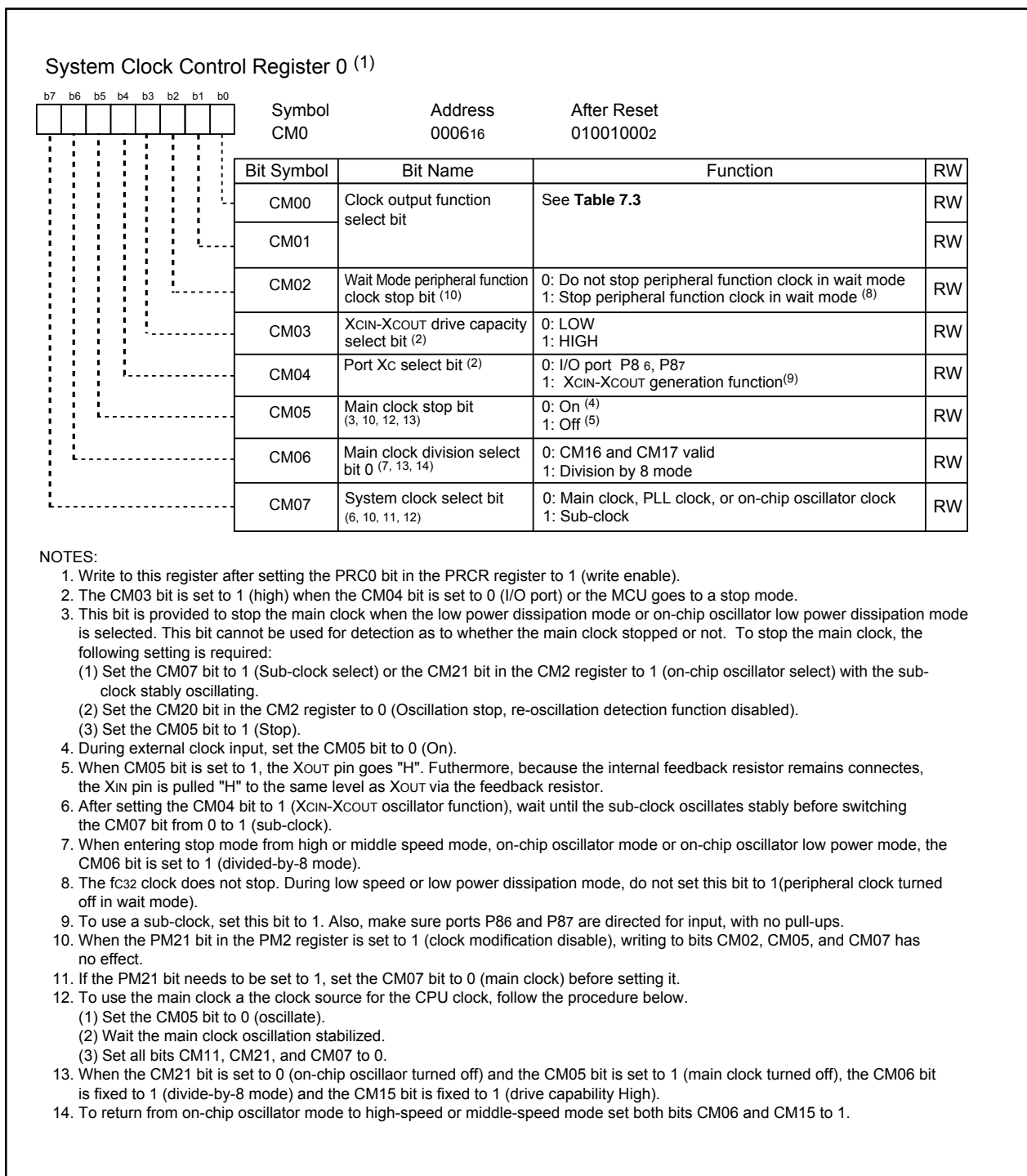


Figure 7.2 CM0 Register

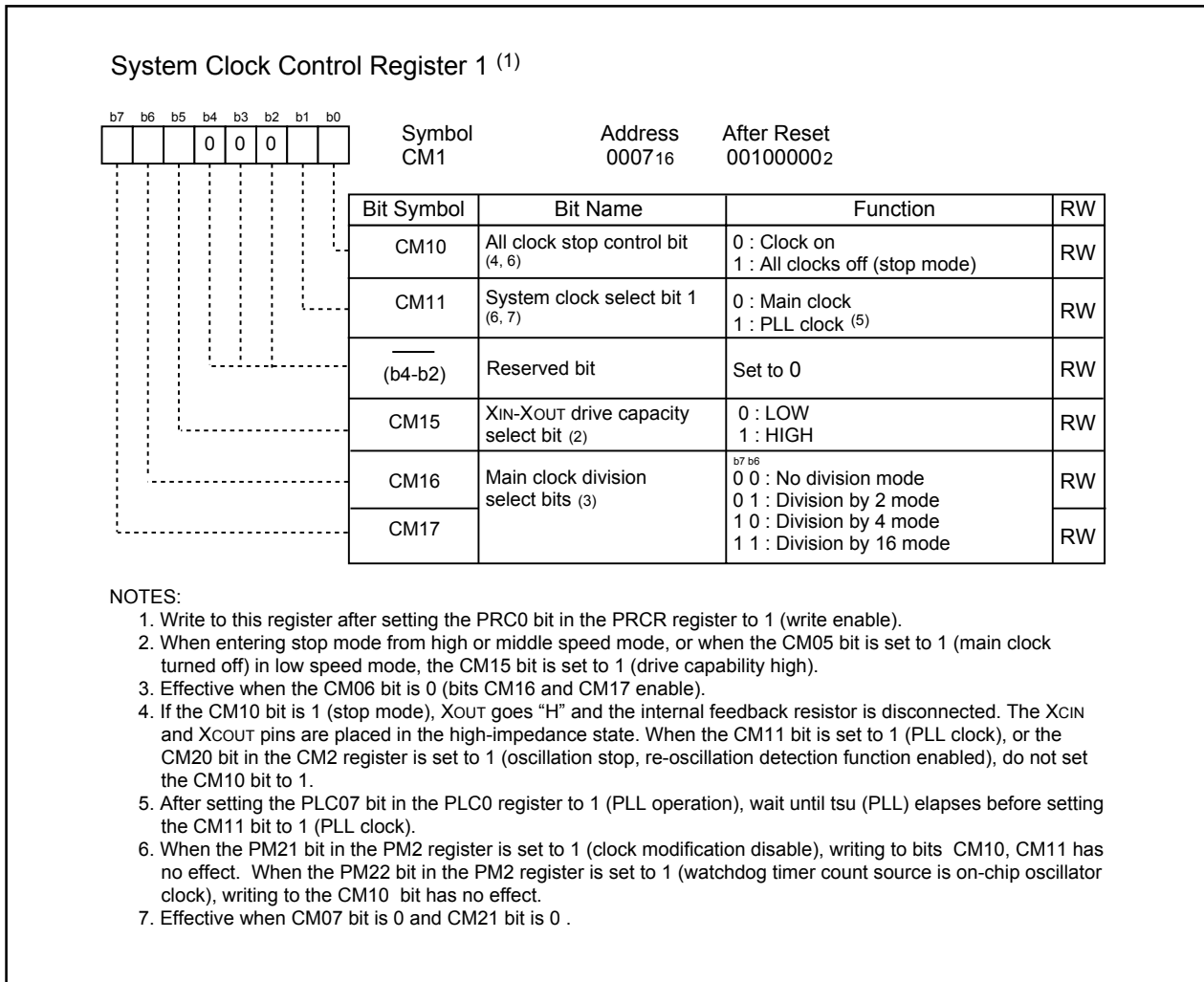


Figure 7.3 CM1 Register

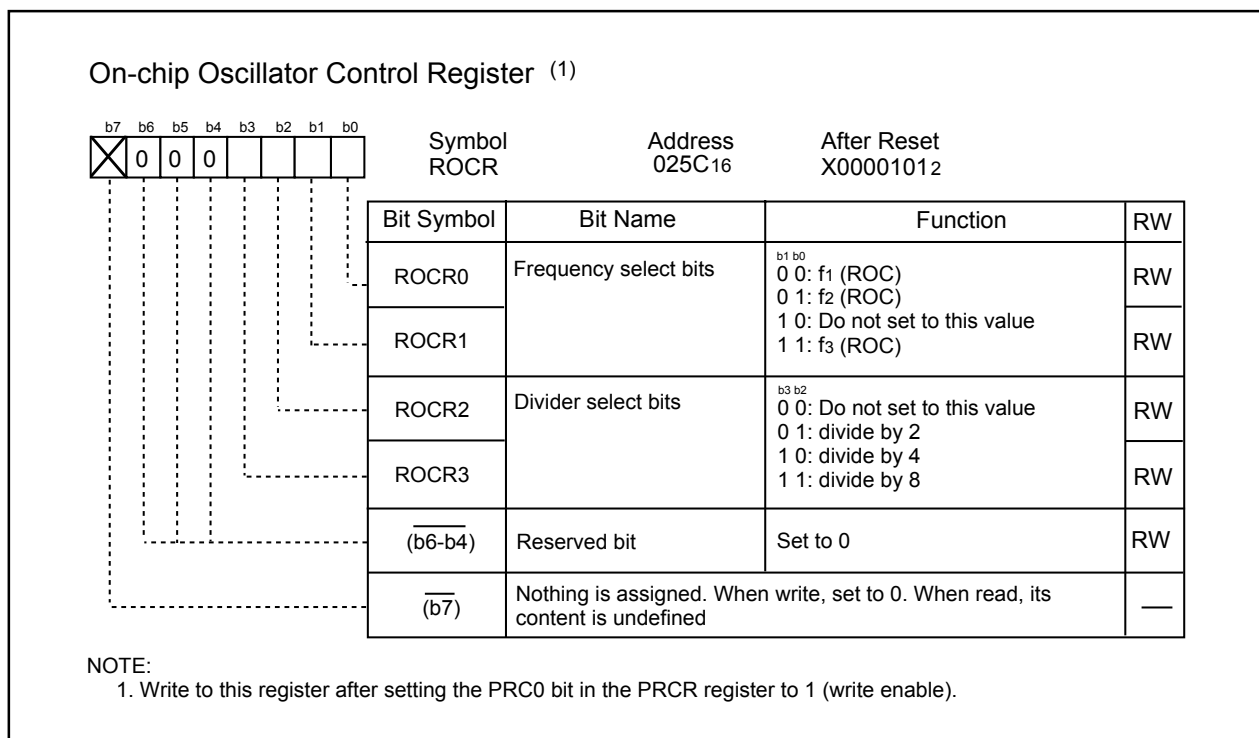


Figure 7.4 ROCR Register

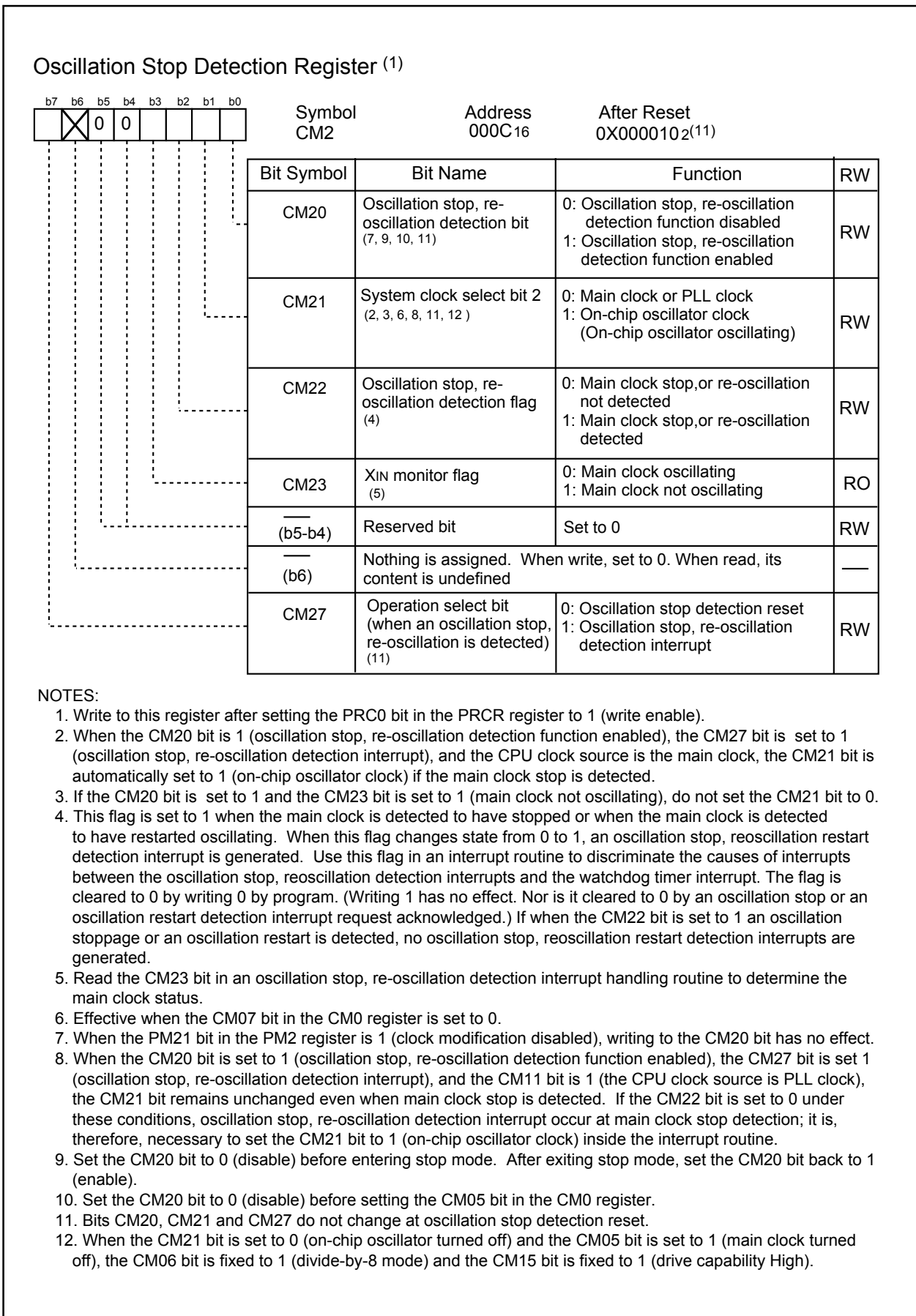


Figure 7.5 CM2 Register

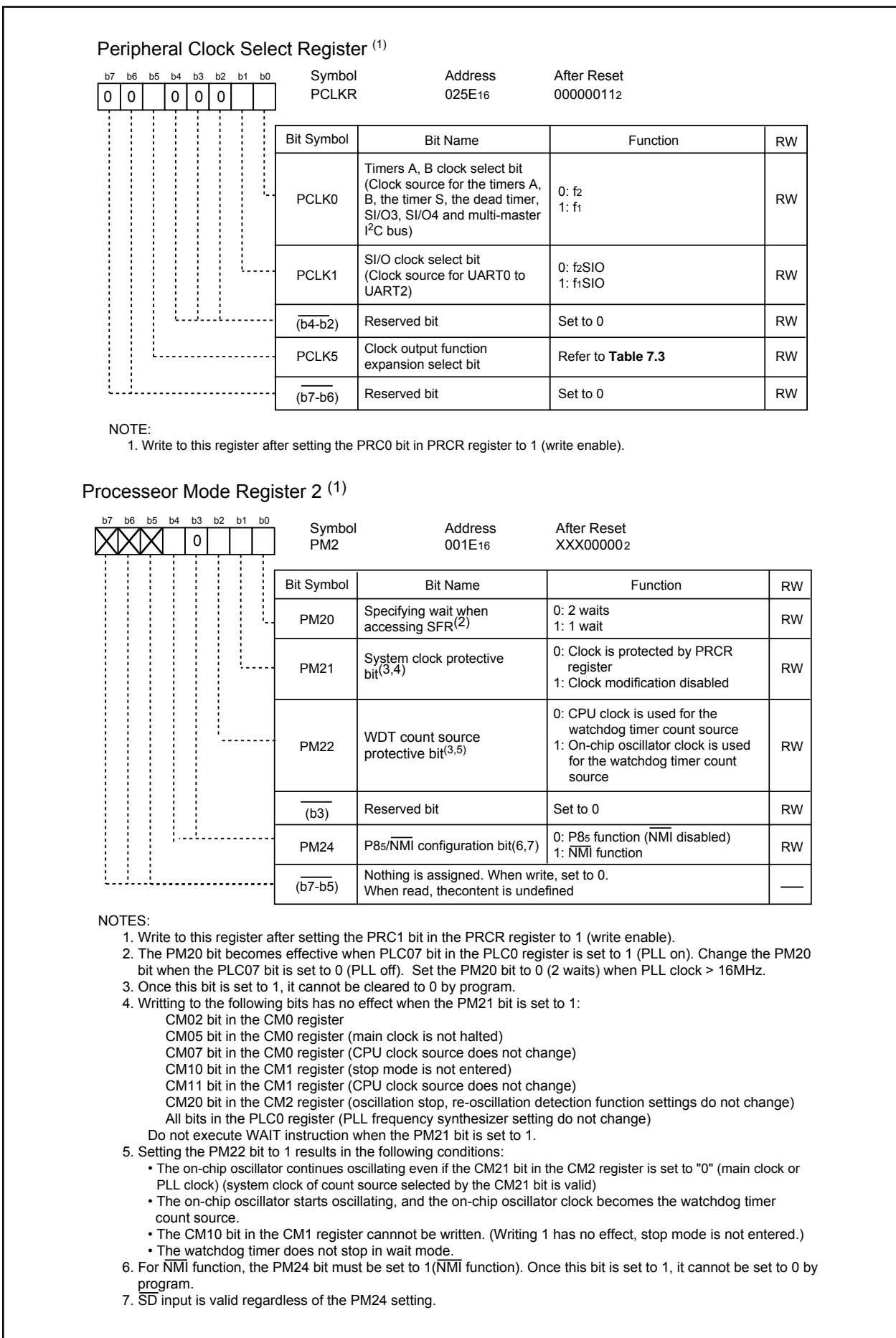


Figure 7.6 PCLKR Register and PM2 Register

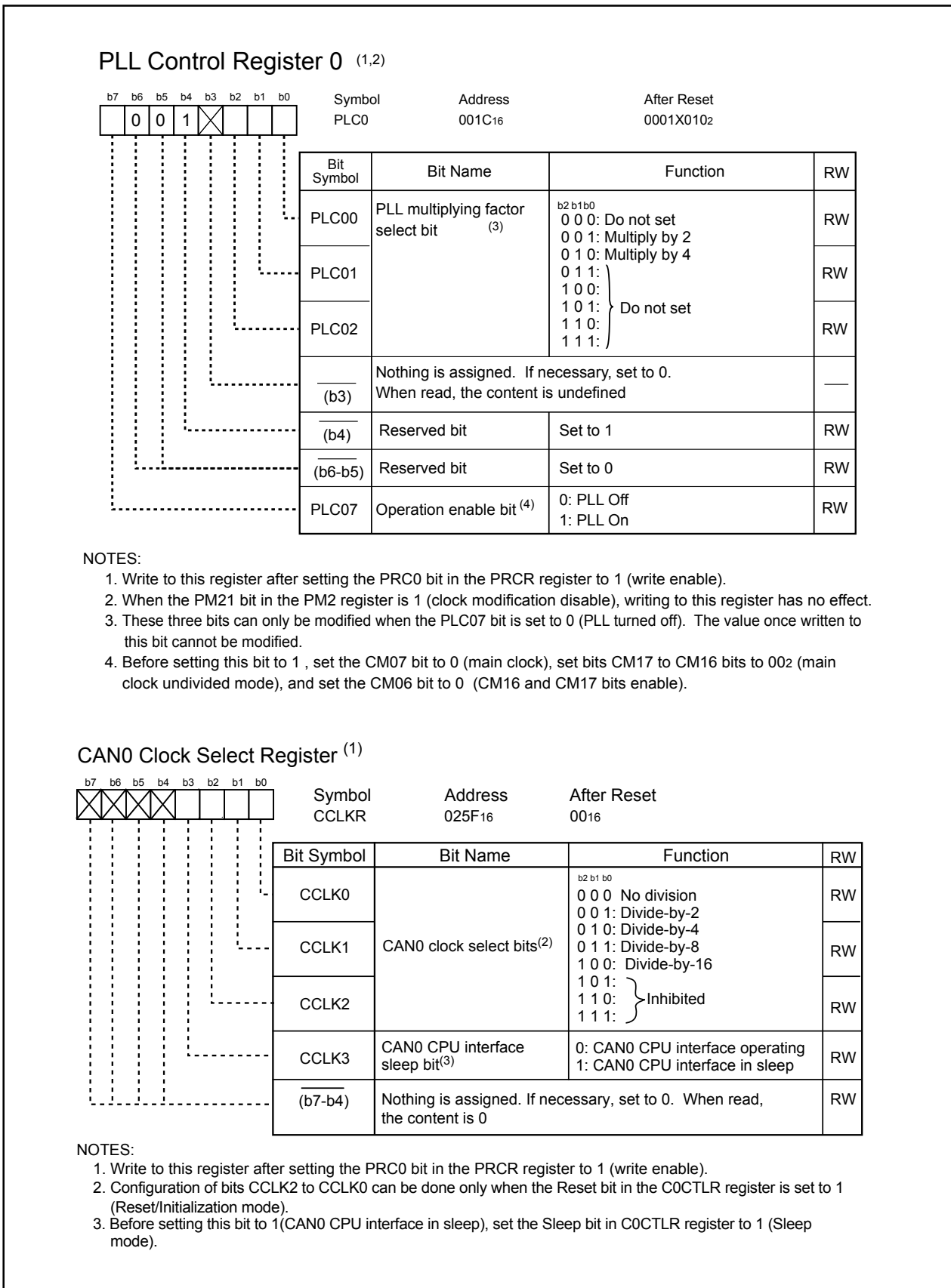


Figure 7.7 PLC0 Register and CCLKR register

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

7.1 Main Clock

The main clock is generated by the main clock oscillation circuit. This clock is used as the clock source for the CPU and peripheral function clocks. The main clock oscillator circuit is configured by connecting a resonator between the XIN and XOUT pins. The main clock oscillator circuit contains a feedback resistor, which is disconnected from the oscillator circuit during stop mode in order to reduce the amount of power consumed in the chip. The main clock oscillator circuit may also be configured by feeding an externally generated clock to the XIN pin. **Figure 7.8** shows the examples of main clock connection circuit.

The power consumption in the chip can be reduced by setting the CM05 bit in the CM0 register to 1 (main clock oscillator circuit turned off) after switching the clock source for the CPU clock to a sub clock or on-chip oscillator clock. In this case, XOUT goes "H". Furthermore, because the internal feedback resistor remains on, XIN is pulled "H" to XOUT via the feedback resistor.

During stop mode, all clocks including the main clock are turned off. Refer to "power control".

If the main clock is not used, it is recommended to connect the XIN pin to VCC to reduce power consumption during reset.

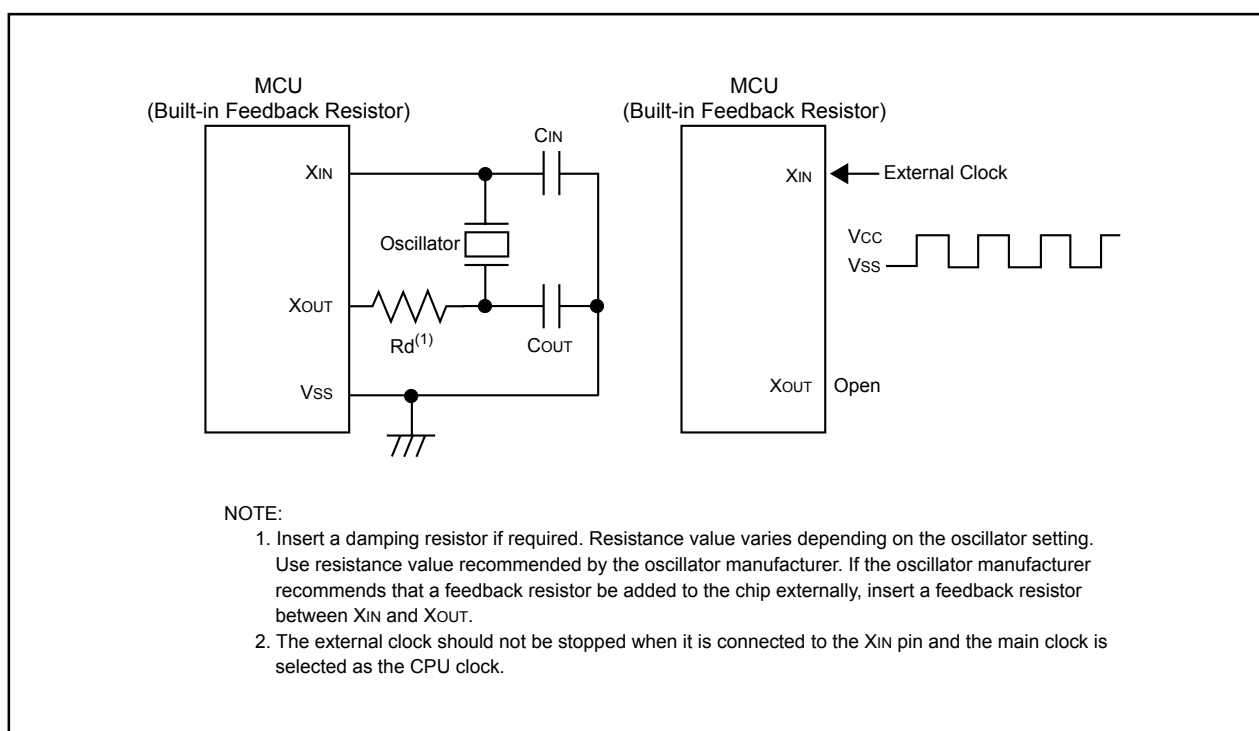


Figure 7.8 Examples of Main Clock Connection Circuit

7.2 Sub Clock

The sub clock is generated by the sub clock oscillation circuit. This clock is used as the clock source for the CPU clock, as well as the timer A and timer B count sources.

The sub clock oscillator circuit is configured by connecting a crystal resonator between the XCIN and XCOUT pins. The sub clock oscillator circuit contains a feedback resistor, which is disconnected from the oscillator circuit during stop mode in order to reduce the amount of power consumed in the chip. The sub clock oscillator circuit may also be configured by feeding an externally generated clock to the XCIN pin. **Figure 7.9** shows the examples of sub clock connection circuit.

After reset, the sub clock is turned off. At this time, the feedback resistor is disconnected from the oscillator circuit.

To use the sub clock for the CPU clock, set the CM07 bit in the CM0 register to 1 (sub clock) after the sub clock becomes oscillating stably.

During stop mode, all clocks including the sub clock are turned off. Refer to “power control”.

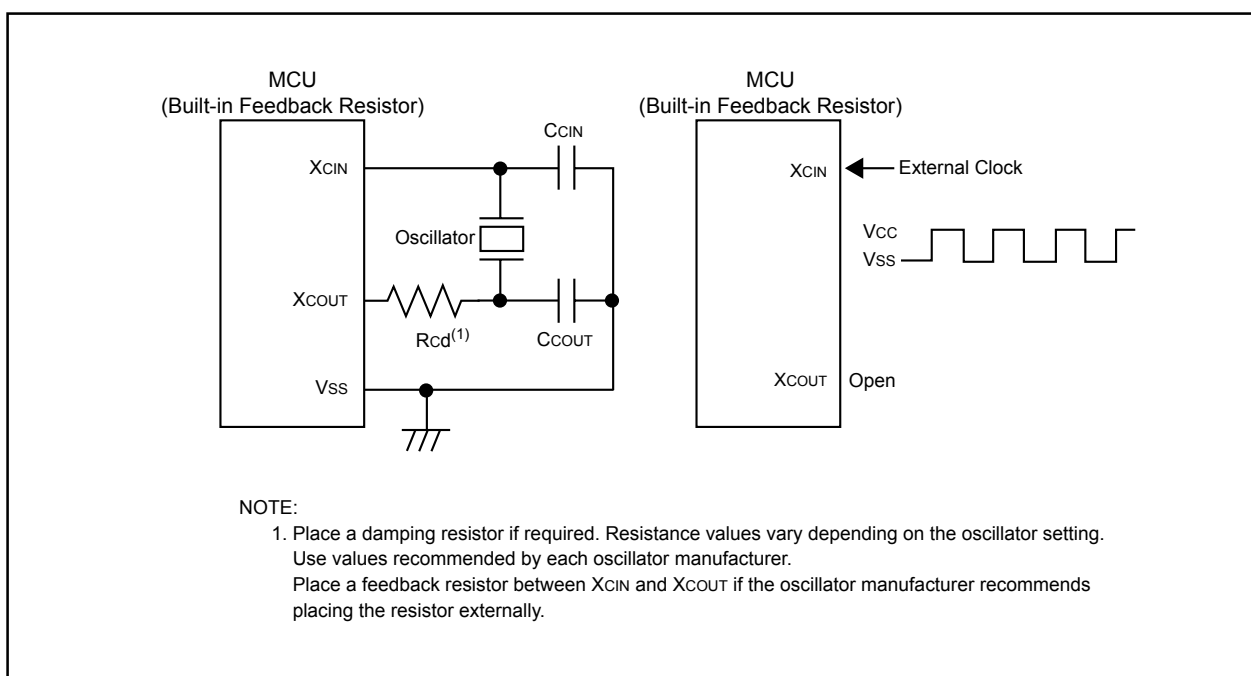


Figure 7.9 Examples of Sub Clock Connection Circuit

7.3 On-chip Oscillator Clock

This clock is supplied by a variable on-chip oscillator. This clock is used as the clock source for the CPU and peripheral function clocks. In addition, if the PM22 bit in the PM2 register is 1 (on-chip oscillator clock for the watchdog timer count source), this clock is used as the count source for the watchdog timer (Refer to 10. Watchdog Timer • Count source protective mode”).

After reset, the on-chip oscillator clock divided by 16 is used for the CPU clock. It can also be turned on by setting the CM21 bit in the CM2 register to 1 (on-chip oscillator clock), and is used as the clock source for the CPU and peripheral function clocks. If the main clock stops oscillating when the CM20 bit in the CM2 register is 1 (oscillation stop, re-oscillation detection function enabled) and the CM27 bit is 1 (oscillation stop, re-oscillation detection interrupt), the on-chip oscillator automatically starts operating, supplying the necessary clock for the MCU.

7.4 PLL Clock

The PLL clock is generated from the main clock by a PLL frequency synthesizer. This clock is used as the clock source for the CPU and peripheral function clocks. After reset, the PLL clock is turned off. The PLL frequency synthesizer is activated by setting the PLC07 bit to 1 (PLL operation). When the PLL clock is used as the clock source for the CPU clock, wait $t_{su}(PLL)$ for the PLL clock to be stable, and then set the CM11 bit in the CM1 register to 1.

Before entering wait mode or stop mode, be sure to set the CM11 bit to 0 (CPU clock source is the main clock). Furthermore, before entering stop mode, be sure to set the PLC07 bit in the PLC0 register to 0 (PLL stops). **Figure 7.10** shows the procedure for using the PLL clock as the clock source for the CPU.

The PLL clock frequency is determined by the equation below.

$$PLL \text{ clock frequency} = f(X_{IN}) \times (\text{multiplying factor set by bits PLC02 to PLC00 in the PLC0 register})$$

(However, $10 \text{ MHz} \leq PLL \text{ clock frequency} \leq 20 \text{ MHz}$)

Bits PLC02 to PLC00 can be set only once after reset. **Table 7.2** shows the example for setting PLL clock frequencies.

Table 7.2 Example for Setting PLL Clock Frequencies

| XIN (MHz) | PLC02 | PLC01 | PLC00 | Multiplying factor | PLL clock (MHz) ⁽¹⁾ |
|-----------|-------|-------|-------|--------------------|--------------------------------|
| 10 | 0 | 0 | 1 | 2 | 20 |
| 5 | 0 | 1 | 0 | 4 | |

NOTE:

1. $10\text{MHz} \leq \text{PLL clock frequency} \leq 20\text{MHz}$.

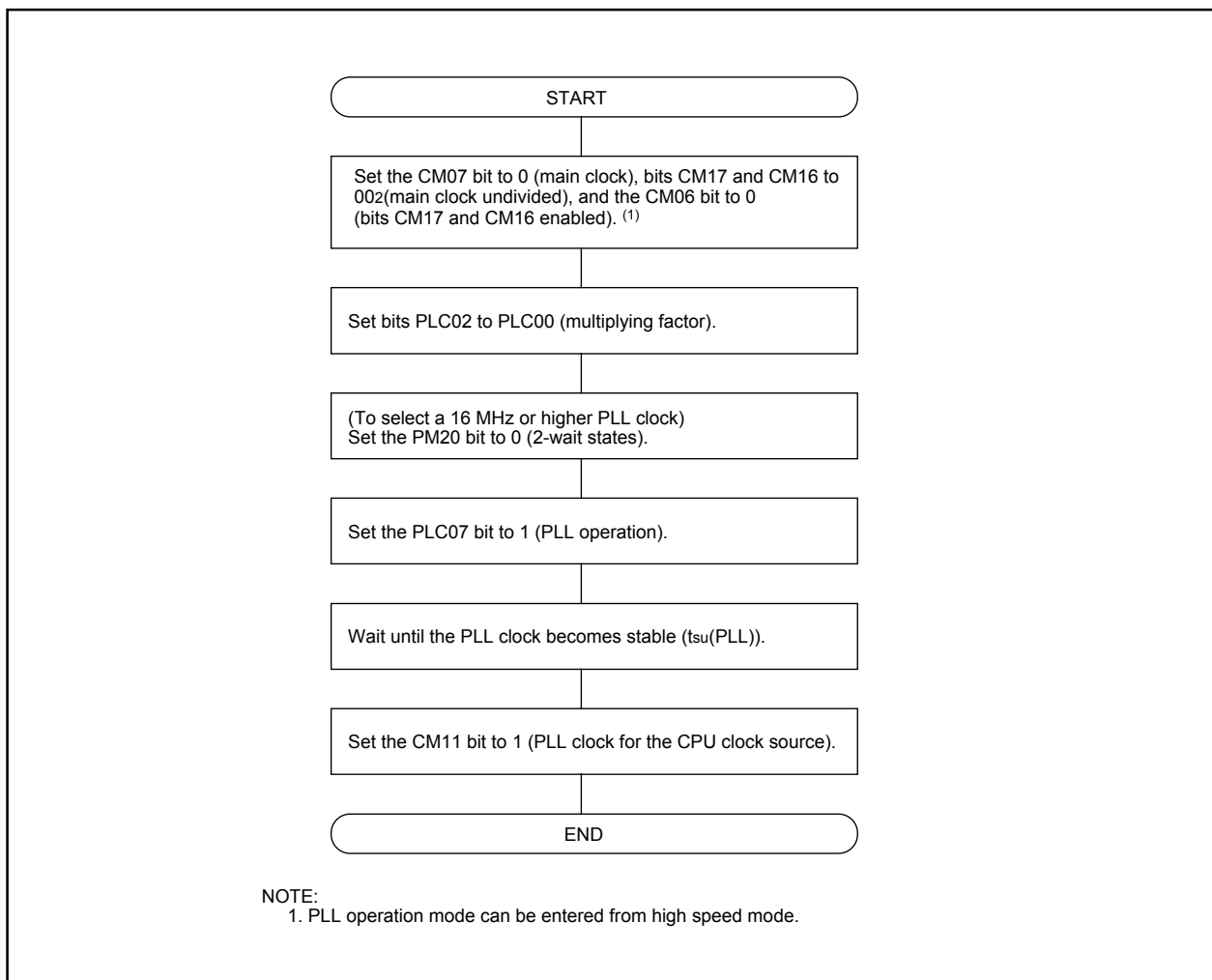


Figure 7.10 Procedure to Use PLL Clock as CPU Clock Source

7.5 CPU Clock and Peripheral Function Clock

The CPU clock is used to operate the CPU and peripheral function clocks are used to operate the peripheral functions.

7.5.1 CPU Clock

This is the operating clock for the CPU and watchdog timer.

The clock source for the CPU clock can be chosen to be the main clock, sub clock, on-chip oscillator clock or the PLL clock.

If the main clock or on-chip oscillator clock is selected as the clock source for the CPU clock, the selected clock source can be divided by 1 (undivided), 2, 4, 8 or 16 to produce the CPU clock. Use the CM06 bit in CM0 register and bits CM17 to CM16 in CM1 register to select the divide-by-n value.

When the PLL clock is selected as the clock source for the CPU clock, the CM06 bit should be set to 0 and bits CM17 and CM16 to 00₂ (undivided).

After reset, the on-chip oscillator clock divided by 16 provides the CPU clock.

Note that when entering stop mode from high or middle speed mode, on-chip oscillator mode or on-chip oscillator low power dissipation mode, or when the CM05 bit in the CM0 register is set to 1 (main clock turned off) in low-speed mode, the CM06 bit in the CM0 register is set to 1 (divide-by-8 mode).

7.5.2 Peripheral Function Clock(f₁, f₂, f₈, f₃₂, f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}, f_{AD}, f_{C32}, f_{CAN0})

These are operating clocks for the peripheral functions.

Of these, f_i (i = 1, 2, 8, 32) and f_{SIO} are derived from the main clock, PLL clock, or on-chip oscillator clock divided by i. The clock f_i is used for Timer A, Timer B, SI/O3 and SI/O4 while f_{SIO} is used for UART0 to UART2. Additionally, the f₁ and f₂ clocks are also used for dead time timer, Timer S, multi-master I²C bus. The f_{AD} clock is produced from the main clock, PLL clock or on-chip oscillator clock, and is used for the A/D converter.

The f_{CAN0} clock is derived from the main clock, PLL clock or on-chip oscillator clock divided by 1 (undivided), 2, 4, 8, or 16, and is used for the CAN module.

When the WAIT instruction is executed after setting the CM02 bit in the CM0 register to 1 (peripheral function clock turned off during wait mode), or when the MCU is in low power dissipation mode, the f_i, f_{SIO}, f_{AD}, and f_{CAN0} clocks are turned off. (Note 1)

The f_{C32} clock is produced from the sub clock, and is used for timers A and B. This clock can only be used when the sub clock is on.

Note 1: f_{CAN0} clock stops at "H" in CAN0 sleep mode.

7.5.3 ClockOutput Function

The f₁, f₈, f₃₂ or f_C clock can be output from the CLKOUT pin. Use the PCLK5 bit in the PCLKR register and bits CM01 to CM00 in the CM0 register to select. **Table 7.3** shows the function of the CLKOUT pin.

Table 7.3 The function of the CLKOUT pin

| PCLK5 | CM01 | CM00 | The function of the CLKOUT pin |
|-------|------|------|--------------------------------|
| 0 | 0 | 0 | I/O port P9 ₀ |
| 0 | 0 | 1 | f _C |
| 0 | 1 | 0 | f ₈ |
| 0 | 1 | 1 | f ₃₂ |
| 1 | 0 | 0 | f ₁ |
| 1 | 0 | 1 | Do not set |
| 1 | 1 | 0 | Do not set |
| 1 | 1 | 1 | Do not set |

7.6 Power Control

There are three power control modes. In this chapter, all modes other than wait and stop modes are referred to as normal operation mode.

7.6.1 Normal Operation Mode

Normal operation mode is further classified into seven modes.

In normal operation mode, because the CPU clock and the peripheral function clocks both are on, the CPU and the peripheral functions are operating. Power control is exercised by controlling the CPU clock frequency. The higher the CPU clock frequency, the greater the processing capability. The lower the CPU clock frequency, the smaller the power consumption in the chip. If the unnecessary oscillator circuits are turned off, the power consumption is further reduced.

Before the clock sources for the CPU clock can be switched over, the new clock source must be in stable oscillation. If the new clock source is the main clock, sub clock or PLL clock, allow a sufficient wait time in a program until it becomes oscillating stably.

Note that operation modes cannot be changed directly from low power dissipation mode to on-chip oscillator mode or on-chip oscillator low power dissipation mode. Nor can operation modes be changed directly from on-chip oscillator mode or on-chip oscillator low power dissipation mode to low power dissipation mode.

When the CPU clock source is changed from the on-chip oscillator to the main clock, change the operation mode to the medium speed mode (divided by 8 mode) after the clock was divided by 8 (the CM06 bit in the CM0 register was set to 1) in the on-chip oscillator mode.

7.6.1.1 High-speed Mode

The main clock divided by 1 provides the CPU clock. If the sub clock is on, fc32 can be used as the count source for timers A and B.

7.6.1.2 PLL Operation Mode

The main clock multiplied by 2 or 4 provides the PLL clock, and this PLL clock serves as the CPU clock. If the sub clock is on, fc32 can be used as the count source for timers A and B. PLL operation mode can be entered from high speed mode. If PLL operation mode is to be changed to wait or stop mode, first go to high speed mode before changing.

7.6.1.3 Medium-speed Mode

The main clock divided by 2, 4, 8 or 16 provides the CPU clock. If the sub clock is on, fc32 can be used as the count source for timers A and B.

7.6.1.4 Low-speed Mode

The sub clock provides the CPU clock. The main clock is used as the clock source for the peripheral function clock when the CM21 bit is set to 0 (on-chip oscillator turned off), and the on-chip oscillator clock is used when the CM21 bit is set to 1 (on-chip oscillator oscillating).

The fc32 clock can be used as the count source for timers A and B.

7.6.1.5 Low Power Dissipation Mode

In this mode, the main clock is turned off after being placed in low speed mode. The sub clock provides the CPU clock. The fc32 clock can be used as the count source for timers A and B. Peripheral function clock can use only fc32.

Simultaneously when this mode is selected, the CM06 bit in the CM0 register becomes 1 (divided by 8 mode). In the low power dissipation mode, do not change the CM06 bit. Consequently, the medium speed (divided by 8) mode is to be selected when the main clock is operated next.

7.6.1.6 On-chip Oscillator Mode

The selected on-chip oscillator clock divided by 1 (undivided), 2, 4, 8 or 16 provides the CPU clock. The on-chip oscillator clock is also the clock source for the peripheral function clocks. If the sub clock is on, fc32 can be used as the count source for timers A and B. The on-chip oscillator frequency can be selected by bits ROCR3 to ROCR0 in the ROCR register. When the operation mode is returned to the high and medium speed modes, set the CM06 bit to 1 (divided by 8 mode).

7.6.1.7 On-chip Oscillator Low Power Dissipation Mode

The main clock is turned off after being placed in on-chip oscillator mode. The CPU clock can be selected as in the on-chip oscillator mode. The on-chip oscillator clock is the clock source for the peripheral function clocks. If the sub clock is on, fc32 can be used as the count source for timers A and B.

Table 7.4 Setting Clock Related Bit and Modes

| Modes | | CM2 Register | CM1 Register | | CM0 Register | | | |
|---|---------------|--------------|--------------|------------|--------------|------------------|------------------|------|
| | | CM21 | CM11 | CM17, CM16 | CM07 | CM06 | CM05 | CM04 |
| PLL operation mode | | 0 | 1 | 002 | 0 | 0 | 0 | — |
| High-speed mode | | 0 | 0 | 002 | 0 | 0 | 0 | — |
| Medium-speed mode | divided by 2 | 0 | 0 | 012 | 0 | 0 | 0 | — |
| | divided by 4 | 0 | 0 | 102 | 0 | 0 | 0 | — |
| | divided by 8 | 0 | 0 | — | 0 | 1 | 0 | — |
| | divided by 16 | 0 | 0 | 112 | 0 | 0 | 0 | — |
| Low-speed mode | | — | — | — | 1 | — | 0 | 1 |
| Low power dissipation mode | | — | — | — | 1 | 1 ⁽¹⁾ | 1 ⁽¹⁾ | 1 |
| On-chip oscillator mode ⁽³⁾ | divided by 1 | 1 | — | 002 | 0 | 0 | 0 | — |
| | divided by 2 | 1 | — | 012 | 0 | 0 | 0 | — |
| | divided by 4 | 1 | — | 102 | 0 | 0 | 0 | — |
| | divided by 8 | 1 | — | — | 0 | 1 | 0 | — |
| | divided by 16 | 1 | — | 112 | 0 | 0 | 0 | — |
| On-chip oscillator low power dissipation mode | | 1 | — | (2) | 0 | (2) | 1 | — |

NOTES:

- When the CM05 bit is set to 1 (main clock turned off) in low-speed mode, the mode goes to low power dissipation mode and CM06 bit is set to 1 (divided by 8 mode) simultaneously.
- The divide-by-n value can be selected the same way as in on-chip oscillator mode.
- On-chip oscillator frequency can be any of those described in the section **7.6.1.6 On-chip Oscillator Mode**.

7.6.2 Wait Mode

In wait mode, the CPU clock is turned off, so are the CPU (because operated by the CPU clock) and the watchdog timer. However, if the PM22 bit in the PM2 register is 1 (on-chip oscillator clock for the watchdog timer count source), the watchdog timer remains active. Because the main clock, sub clock, on-chip oscillator clock and PLL clock all are on, the peripheral functions using these clocks keep operating.

7.6.2.1 Peripheral Function Clock Stop Function

When the CM02 bit is 1 (peripheral function clocks turned off during wait mode), f1, f2, f8, f32, f1SIO, f2SIO, f8SIO, f32SIO, and fAD stop running in wait mode to reduce power consumption. However, fc32 remains active.

7.6.2.2 Entering Wait Mode

The MCU enters wait mode by executing the WAIT instruction.

When the CM11 bit is set to 1 (CPU clock source is the PLL clock), be sure to clear the CM11 bit to 0 (CPU clock source is the main clock) before going to wait mode. The power consumption of the chip can be reduced by clearing the PLC07 bit to 0 (PLL stops).

7.6.2.3 Pin Status During Wait Mode

Table 7.5 lists pin status during wait mode.

Table 7.5 Pin Status in Wait Mode

| Pin | | Status |
|-----------|---------------------------|---|
| I/O ports | | Retains status before wait mode |
| CLKOUT | When fC selected | Does not stop |
| | When f1, f8, f32 selected | Does not stop when the CM02 bit is set to 0 |
| | | Retains status before wait mode when the CM02 bit is set to 1 |

7.6.2.4 Exiting Wait Mode

The MCU exits from wait mode by a hardware reset, $\overline{\text{NMI}}$ interrupt, or peripheral function interrupt.

If wait mode is exited by a hardware reset or $\overline{\text{NMI}}$ interrupt, set the peripheral function interrupt priority bits ILVL2 to ILVL0 to 0002 (interrupts disabled) before executing the WAIT instruction.

The CM02 bit affects the peripheral function interrupts. If the CM02 bit is 0 (peripheral function clocks not turned off during wait mode), all peripheral function interrupts can be used to exit wait mode. If the CM02 bit is 1 (peripheral function clock stops during wait mode), the peripheral functions using the peripheral function clock stops operating, so that only the peripheral functions clocked by external signals can be used to exit wait mode.

Table 7.6 lists the interrupts to exit wait mode.

Table 7.6 Interrupts to Exit Wait Mode

| Interrupt | CM02 = 0 | CM02 = 1 |
|---|--|--|
| NMI interrupt | Available | Available |
| Serial I/O interrupt | Available when internal and external clocks are used | Available when external clock is used |
| Multi-master I ² C interrupt | Available | Do not used |
| Key input interrupt | Available | Available |
| A/D conversion interrupt | Available in one-shot or single sweep mode | Do not use |
| Timer A interrupt Timer B interrupt | Available in all modes | Available in event counter mode or when count source is fC32 |
| Timer S interrupt | Available in all modes | Do not use |
| $\overline{\text{INT}}$ interrupt | Available | Available |
| CAN0 wake_up interrupt | Available in CAN sleep mode | Available in CAN sleep mode |

To use peripheral function interrupts to exit wait mode, set the followings before executing the WAIT instruction.

1. Set the interrupt priority level to the bits ILVL2 to ILVL0 in the interrupt control register of the peripheral function interrupts that are used to exit wait mode. Also, set bits ILVL2 to ILVL0 of all peripheral function interrupts that are not used to exit wait mode to 0002 (interrupt disabled).
2. Set the I flag to 1.
3. Operate the peripheral functions that are used to exit wait mode.

When the peripheral function interrupts are used to exit wait mode, an interrupt routine is executed after an interrupt request is generated and the CPU is clocked.

The CPU clock used when exiting wait mode by a peripheral function interrupt is the same CPU clock that is used when executing the WAIT instruction.

7.6.3 Stop Mode

In stop mode, all oscillator circuits are turned off, so are the CPU clock and the peripheral function clocks. Therefore, the CPU and the peripheral functions clocked by these clocks stop operating. The least amount of power is consumed in this mode. If the voltage applied to Vcc pin is VRAM or more, the internal RAM is retained. When applying 2.7 or less voltage to Vcc pin, make sure $V_{CC} \geq V_{RAM}$.

However, the peripheral functions clocked by external signals keep operating. The following interrupts can be used to exit stop mode.

- \overline{NMI} interrupt
- Key interrupt
- \overline{INT} interrupt
- Timer A, Timer B interrupt (when counting external pulses in event counter mode)
- Serial I/O interrupt (when external clock is selected)
- Low voltage detection interrupt (refer to "**Low Voltage Detection Interrupt**" for an operating condition)
- CAN0 Wake_up interrupt (in CAN sleep mode)

7.6.3.1 Entering Stop Mode

The MCU is placed into stop mode by setting the CM10 bit in the CM1 register to 1 (all clocks turned off). 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 (main clock oscillator circuit drive capability high).

Before entering stop mode, set the CM20 bit to 0 (oscillation stop, re-oscillation detection function disable).

Also, if the CM11 bit is 1 (PLL clock for the CPU clock source), set the CM11 bit to 0 (main clock for the CPU clock source) and the PLC07 bit to 0 (PLL turned off) before entering stop mode.

7.6.3.2 Pin Status during Stop Mode

The I/O pins retain their status held just prior to entering stop mode.

7.6.3.3 Exiting Stop Mode

The MCU is moved out of stop mode by a hardware reset, \overline{NMI} interrupt or peripheral function interrupt. If the MCU is to be moved out of stop mode by a hardware reset or \overline{NMI} interrupt, set the peripheral function interrupt priority bits ILVL2 to ILVL0 to 0002 (interrupts disable) before setting the CM10 bit to 1. If the MCU is to be moved out of stop mode by a peripheral function interrupt, set up the following before setting the CM10 bit to 1.

1. In bits ILVL2 to ILVL0 of the interrupt control register, set the interrupt priority level of the peripheral function interrupt to be used to exit stop mode.

Also, for all of the peripheral function interrupts not used to exit stop mode, set bits ILVL2 to ILVL0 to 0002.

2. Set the I flag to 1.
3. Enable the peripheral function whose interrupt is to be used to exit stop mode.

In this case, when an interrupt request is generated and the CPU clock is thereby turned on, an interrupt service routine is executed.

Which CPU clock will be used after exiting stop mode by a peripheral function or \overline{NMI} interrupt is determined by the CPU clock that was on when the MCU was placed into stop mode as follows:

If the CPU clock before entering stop mode was derived from the sub clock: sub clock

If the CPU clock before entering stop mode was derived from the main clock: main clock divide-by-8

If the CPU clock before entering stop mode was derived from the on-chip oscillator clock: on-chip oscillator clock divide-by-8

Figure 7.11 shows the state transition from normal operation mode to stop mode and wait mode. Figure 7.12 shows the state transition in normal operation mode. Table 7.7 shows a state transition matrix describing allowed transition and setting. The vertical line shows current state and horizontal line shows state after transition.

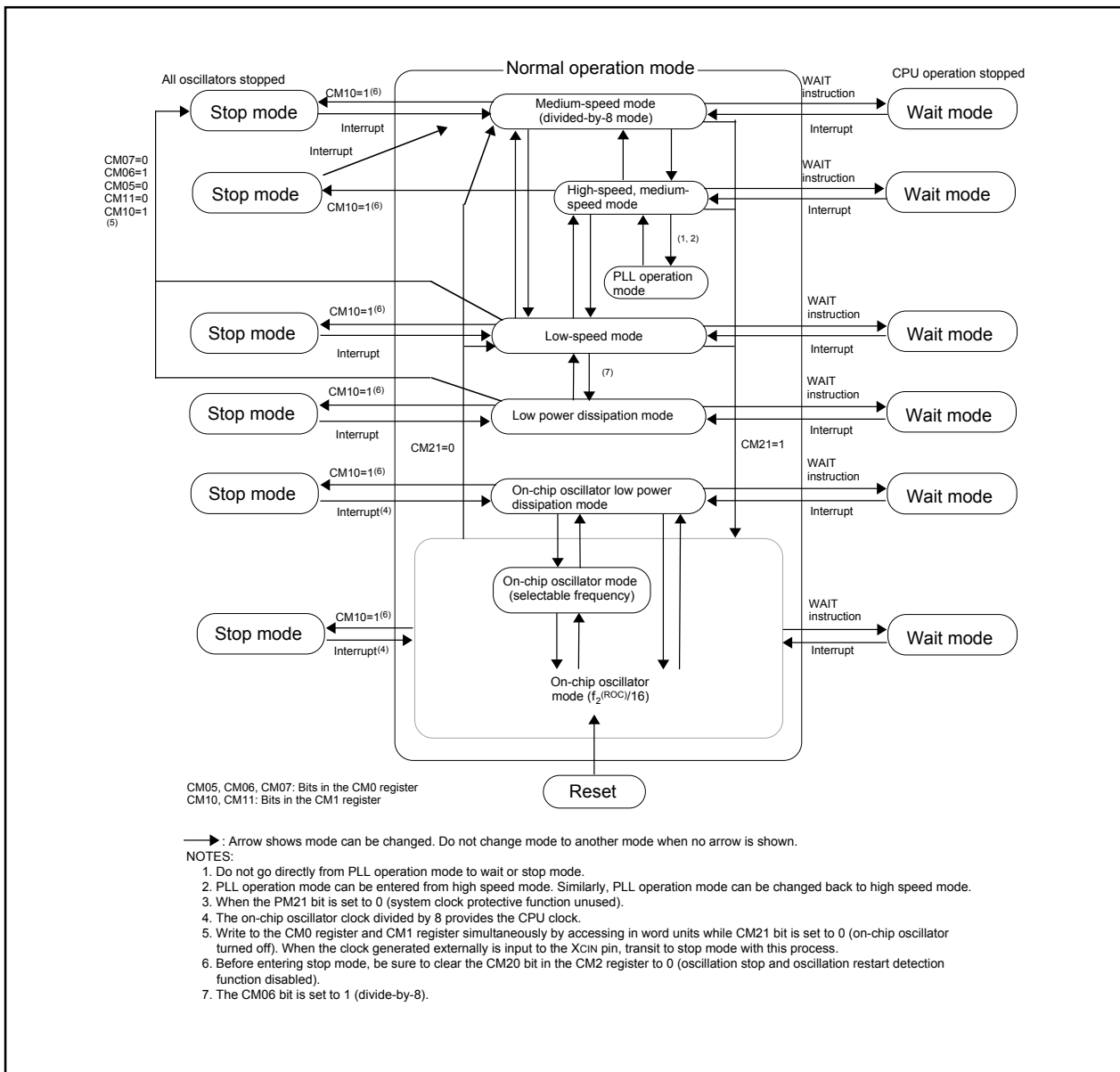


Figure 7.11 State Transition to Stop Mode and Wait Mode

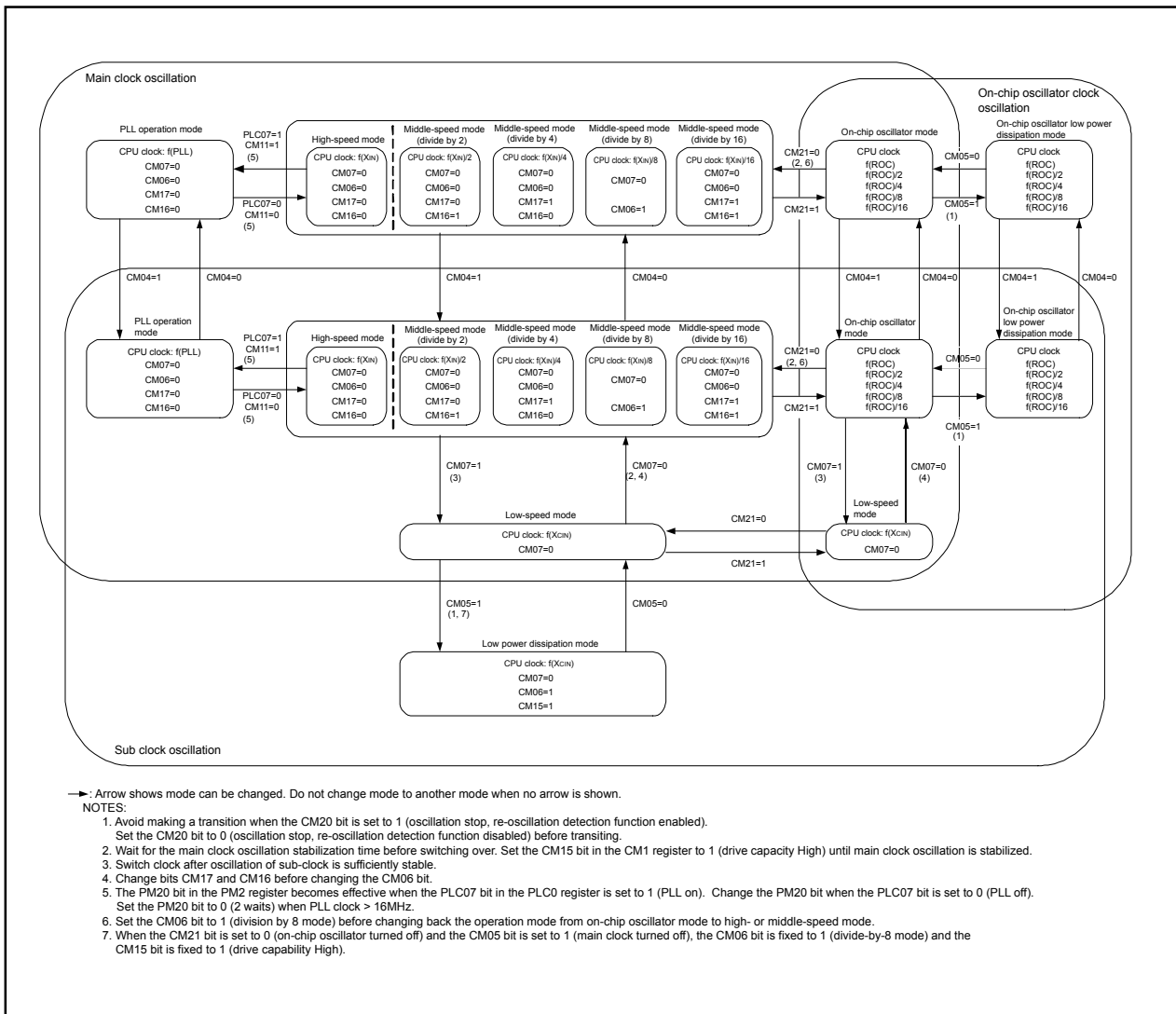


Figure 7.12 State Transition in Normal Mode

Table 7.7 Allowed Transition and Setting

| | | State after transition | | | | | | | |
|---------------|---|------------------------------------|-----------------------------|----------------------------|---------------------------------|-------------------------|---|-------------------|-----------|
| | | High-speed mode, middle-speed mode | Low-speed mode ² | Low power dissipation mode | PLL operation mode ² | On-chip oscillator mode | On-chip oscillator low power dissipation mode | Stop mode | Wait mode |
| Current state | High-speed mode, middle-speed mode | 8 | (9) ⁷ | -- | (13) ³ | (15) | -- | (16) ¹ | (17) |
| | Low-speed mode ² | (8) | | (11) ^{1, 6} | -- | (8) | -- | (16) ¹ | (17) |
| | Low power dissipation mode | -- | (10) | | -- | -- | -- | (16) ¹ | (17) |
| | PLL operation mode ² | (12) ³ | -- | -- | | -- | -- | -- | -- |
| | On-chip oscillator mode | (14) ⁴ | (9) ⁷ | -- | -- | 8 | (11) ¹ | (16) ¹ | (17) |
| | On-chip oscillator low power dissipation mode | -- | -- | -- | -- | (10) | 8 | (16) ¹ | (17) |
| | Stop mode | (18) ⁵ | (18) | (18) | -- | (18) ⁵ | (18) ⁵ | | -- |
| | Wait mode | (18) | (18) | (18) | -- | (18) | (18) | -- | |

NOTES:

1. Avoid making a transition when the CM20 bit is set to 1 (oscillation stop, re-oscillation detection function enabled).
Set the CM20 bit to 0 (oscillation stop, re-oscillation detection function disabled) before transitioning.
2. On-chip oscillator clock oscillates and stops in low-speed mode. In this mode, the on-chip oscillator can be used as peripheral function clock. Sub clock oscillates and stops in PLL operation mode. In this mode, sub clock can be used as a clock for the timers A and B.
3. PLL operation mode can only be entered from and changed to high-speed mode.
4. Set the CM06 bit to 1 (division by 8 mode) before transitioning from on-chip oscillator mode to high- or middle-speed mode.
5. When exiting stop mode, the CM06 bit is set to 1 (division by 8 mode).
6. If the CM05 bit is set to 1 (main clock stop), then the CM06 bit is set to 1 (division by 8 mode).
7. A transition can be made only when sub clock is oscillating.
8. State transitions within the same mode (divide-by-n values changed or subclock oscillation turned on or off) are shown in the table below.

| | | Sub clock oscillating | | | | | Sub clock turned off | | | | |
|-----------------------|---------------|-----------------------|--------------|--------------|--------------|---------------|----------------------|--------------|--------------|--------------|---------------|
| | | No division | Divided by 2 | Divided by 4 | Divided by 8 | Divided by 16 | No division | Divided by 2 | Divided by 4 | Divided by 8 | Divided by 16 |
| Sub clock oscillating | No division | | (4) | (5) | (7) | (6) | (1) | -- | -- | -- | -- |
| | Divided by 2 | (3) | | (5) | (7) | (6) | -- | (1) | -- | -- | -- |
| | Divided by 4 | (3) | (4) | | (7) | (6) | -- | -- | (1) | -- | -- |
| | Divided by 8 | (3) | (4) | (5) | | (6) | -- | -- | -- | (1) | -- |
| | Divided by 16 | (3) | (4) | (5) | (7) | | -- | -- | -- | -- | (1) |
| Sub clock turned off | No division | (2) | -- | -- | -- | -- | (4) | (5) | (7) | (6) | |
| | Divided by 2 | -- | (2) | -- | -- | -- | (3) | (4) | (5) | (7) | (6) |
| | Divided by 4 | -- | -- | (2) | -- | -- | (3) | (4) | (7) | (6) | |
| | Divided by 8 | -- | -- | -- | (2) | -- | (3) | (4) | (5) | (6) | |
| | Divided by 16 | -- | -- | -- | -- | (2) | (3) | (4) | (5) | (7) | |

9. () : setting method. Refer to following table. --: Cannot transit

| | Setting | Operation |
|------|------------------------------|---|
| (1) | CM04 = 0 | Sub clock turned off |
| (2) | CM04 = 1 | Sub clock oscillating |
| (3) | CM06 = 0, CM17 = 0, CM16 = 0 | CPU clock no division mode |
| (4) | CM06 = 0, CM17 = 0, CM16 = 1 | CPU clock division by 2 mode |
| (5) | CM06 = 0, CM17 = 1, CM16 = 0 | CPU clock division by 4 mode |
| (6) | CM06 = 0, CM17 = 1, CM16 = 1 | CPU clock division by 16 mode |
| (7) | CM06 = 1 | CPU clock division by 8 mode |
| (8) | CM07 = 0 | Main clock, PLL clock, or on-chip oscillator clock selected |
| (9) | CM07 = 1 | Sub clock selected |
| (10) | CM05 = 0 | Main clock oscillating |
| (11) | CM05 = 1 | Main clock turned off |
| (12) | PLC07 = 0, CM11 = 0 | Main clock selected |
| (13) | PLC07 = 1, CM11 = 1 | PLL clock selected |
| (14) | CM21 = 0 | Main clock or PLL clock selected |
| (15) | CM21 = 1 | On-chip oscillator clock selected |
| (16) | CM10 = 1 | Transition to stop mode |
| (17) | wait instruction | Transition to wait mode |
| (18) | Hardware interrupt | Exit stop mode or wait mode |

CM04, CM05, CM06, CM07 : Bits in the CM0 register
 CM10, CM11, CM16, CM17 : Bits in the CM1 register
 CM20, CM21 : Bits in the CM2 register
 PLC07 : Bit in the PLC0 register

7.7 System Clock Protective Function

When the main clock is selected for the CPU clock source, this function protects the clock from modifications in order to prevent the CPU clock from becoming halted by run-away.

If the PM21 bit in the PM2 register is set to 1 (clock modification disabled), the following bits are protected against writes:

- Bits CM02, CM05, and CM07 in CM0 register
- Bits CM10 and CM11 in CM1 register
- CM20 bit in CM2 register
- All bits in the PLC0 register

Before the system clock protective function can be used, the following register settings must be made while the CM05 bit in the CM0 register is 0 (main clock oscillating) and CM07 bit is 0 (main clock selected for the CPU clock source):

- (1) Set the PRC1 bit in the PRCR register to 1 (enable writes to PM2 register).
- (2) Set the PM21 bit in the PM2 register to 1 (disable clock modification).
- (3) Set the PRC1 bit in the PRCR register to 0 (disable writes to PM2 register).

Do not execute the WAIT instruction when the PM21 bit is 1.

7.8 Oscillation Stop and Re-oscillation Detect Function

The oscillation stop and re-oscillation detect function detects the re-oscillation after stop of main clock oscillation circuit. When the oscillation stop and re-oscillation detection occurs, the oscillation stop detect function is reset or oscillation stop and re-oscillation detection interrupt is generated, depending on the CM27 bit set in the CM2 register. The oscillation stop detect function is enabled or disabled by the CM20 bit in the CM2 register. **Table 7.8** lists a specification overview of the oscillation stop and re-oscillation detect function.

Table 7.8 Specification Overview of Oscillation Stop and Re-oscillation Detect Function

| Item | Specification |
|--|--|
| Oscillation stop detectable clock and frequency bandwidth | $f(X_{IN}) \geq 2 \text{ MHz}$ |
| Enabling condition for oscillation stop, re-oscillation detection function | Set CM20 bit to 1(enable) |
| Operation at oscillation stop, re-oscillation detection | <ul style="list-style-type: none"> •Reset occurs (when CM27 bit =0) •Oscillation stop, re-oscillation detection interrupt occurs(when CM27 bit =1) |

7.8.1 Operation When CM27 bit = 0 (Oscillation Stop Detection Reset)

When main clock stop is detected when the CM20 bit is 1 (oscillation stop, re-oscillation detection function enabled), the MCU is initialized, coming to a halt (oscillation stop reset; refer to “SFR”, “Reset”).

This status is reset with hardware reset 1. Also, even when re-oscillation is detected, the MCU can be initialized and stopped; it is, however, necessary to avoid such usage. (During main clock stop, do not set the CM20 bit to 1 and the CM27 bit to 0.)

7.8.2 Operation When CM27 bit = 1 (Oscillation Stop and Re-oscillation Detect Interrupt)

When the main clock corresponds to the CPU clock source and the CM20 bit is 1 (oscillation stop and re-oscillation detect function enabled), the system is placed in the following state if the main clock comes to a halt:

- Oscillation stop and re-oscillation detect interrupt request occurs.
- The on-chip oscillator starts oscillation, and the on-chip oscillator clock becomes the CPU clock and clock source for peripheral functions in place of the main clock.
- CM21 bit = 1 (on-chip oscillator clock for CPU clock source)
- CM22 bit = 1 (main clock stop detected)
- CM23 bit = 1 (main clock stopped)

When the PLL clock corresponds to the CPU clock source and the CM20 bit is 1, the system is placed in the following state if the main clock comes to a halt: Since the CM21 bit remains unchanged, set it to 1 (on-chip oscillator clock) inside the interrupt routine.

- Oscillation stop and re-oscillation detect interrupt request occurs.
- CM22 bit = 1 (main clock stop detected)
- CM23 bit = 1 (main clock stopped)
- CM21 bit remains unchanged

When the CM20 bit is 1, the system is placed in the following state if the main clock re-oscillates from the stop condition:

- Oscillation stop and re-oscillation detect interrupt request occurs.
- CM22 bit = 1 (main clock re-oscillation detected)
- CM23 bit = 0 (main clock oscillation)
- CM21 bit remains unchanged

7.8.3 How to Use Oscillation Stop and Re-oscillation Detect Function

- The oscillation stop and re-oscillation detect interrupt shares the vector with the watchdog timer interrupt. If the oscillation stop, re-oscillation detection and watchdog timer interrupts both are used, read the CM22 bit in an interrupt routine to determine which interrupt source is requesting the interrupt.
- Where the main clock re-oscillated after oscillation stop, return the main clock to the CPU clock and peripheral function clock source by program. **Figure 7.13** shows the procedure for switching the clock source from the on-chip oscillator to the main clock.
- Simultaneously with oscillation stop, re-oscillation detection interrupt occurrence, the CM22 bit becomes 1. When the CM22 bit is set at 1, oscillation stop, re-oscillation detection interrupt are disabled. By setting the CM22 bit to 0 by program, oscillation stop, re-oscillation detection interrupt are enabled.
- If the main clock stops during low speed mode where the CM20 bit is 1, an oscillation stop, re-oscillation detection interrupt request is generated. At the same time, the on-chip oscillator starts oscillating. In this case, although the CPU clock is derived from the sub clock as it was before the interrupt occurred, the peripheral function clocks now are derived from the on-chip oscillator clock.
- To enter wait mode while using the oscillation stop, re-oscillation detection function, set the CM02 bit to 0 (peripheral function clocks not turned off during wait mode).
- Since the oscillation stop, re-oscillation detection function is provided in preparation for main clock stop due to external factors, set the CM20 bit to 0 (Oscillation stop, re-oscillation detection function disabled) where the main clock is stopped or oscillated by program, that is where the stop mode is selected or the CM05 bit is altered.
- This function cannot be used if the main clock frequency is 2 MHz or less. In that case, set the CM20 bit to 0.

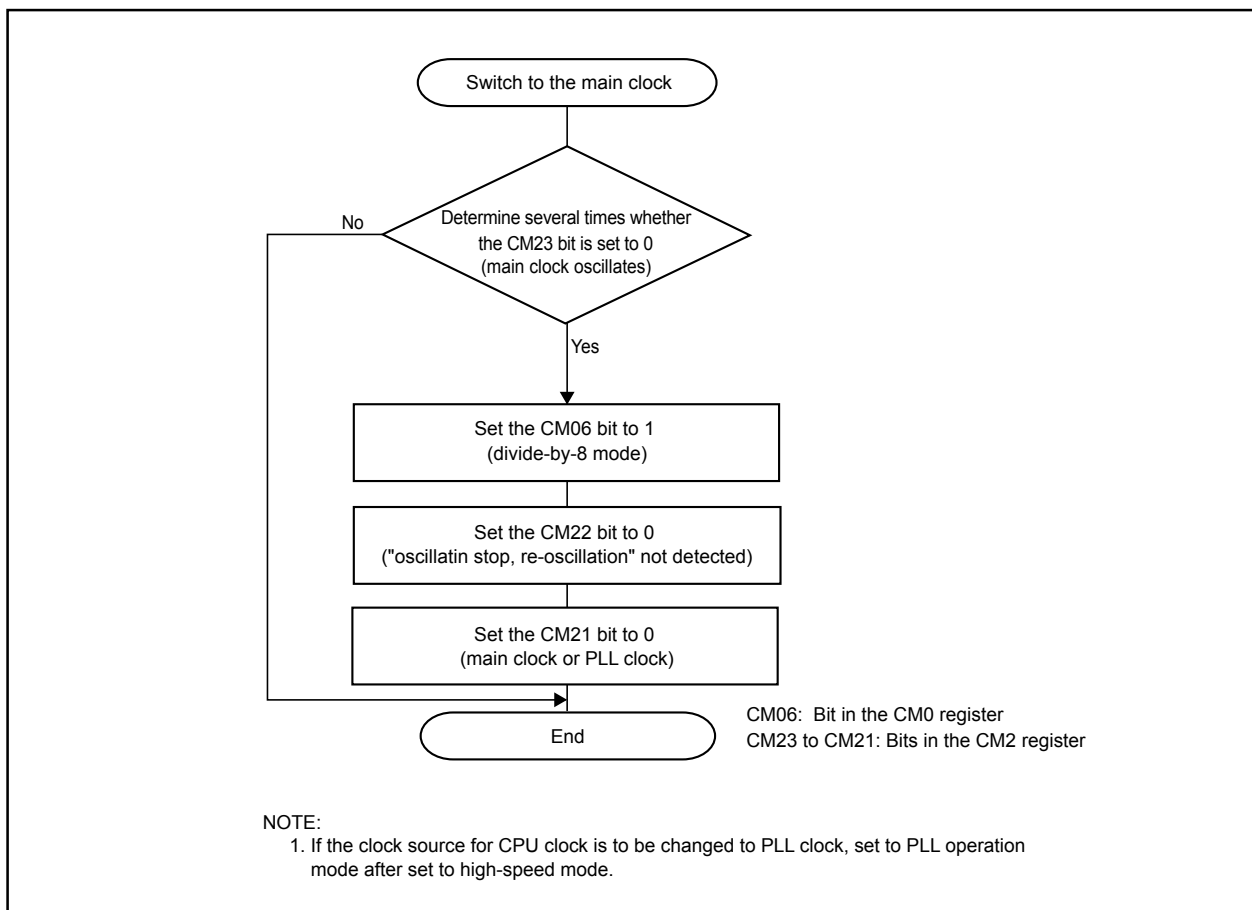


Figure 7.13 Procedure to Switch Clock Source From On-chip Oscillator to Main Clock

8. Protection

In the event that a program runs out of control, this function protects the important registers so that they will not be rewritten easily. **Figure 8.1** shows the PRCR register. The following lists the registers protected by the PRCR register.

- Registers protected by the PRC0 bit: CM0, CM1, CM2, PLC0, ROCR, PCLKR, and CCLKR
- Registers protected by the PRC1 bit: PM0, PM1, PM2, TB2SC, INVC0, and INVC1
- Registers protected by the PRC2 bit: PD9, PACR, S4C, and NDDR
- Registers protected by the PRC3 bit: VCR2 and D4INT

The PRC2 bit is set to 0 (write enabled) when data is written to the SFR area after setting the PRC2 bit to 1 (write enable). Set registers PD9, PACR, S4C and NDDR immediately after setting the PRC2 bit in the PRCR register to 1 (write enable). Do not generate an interrupt or a DMA transfer between the instruction to set the PRC2 bit to 1 and the following instruction. Bits PRC3, PRC1, and PRC0 are not set to 0 even if data is written to the SFR area. Set bits PRC3, PRC1, and PRC0 to 0 by program.

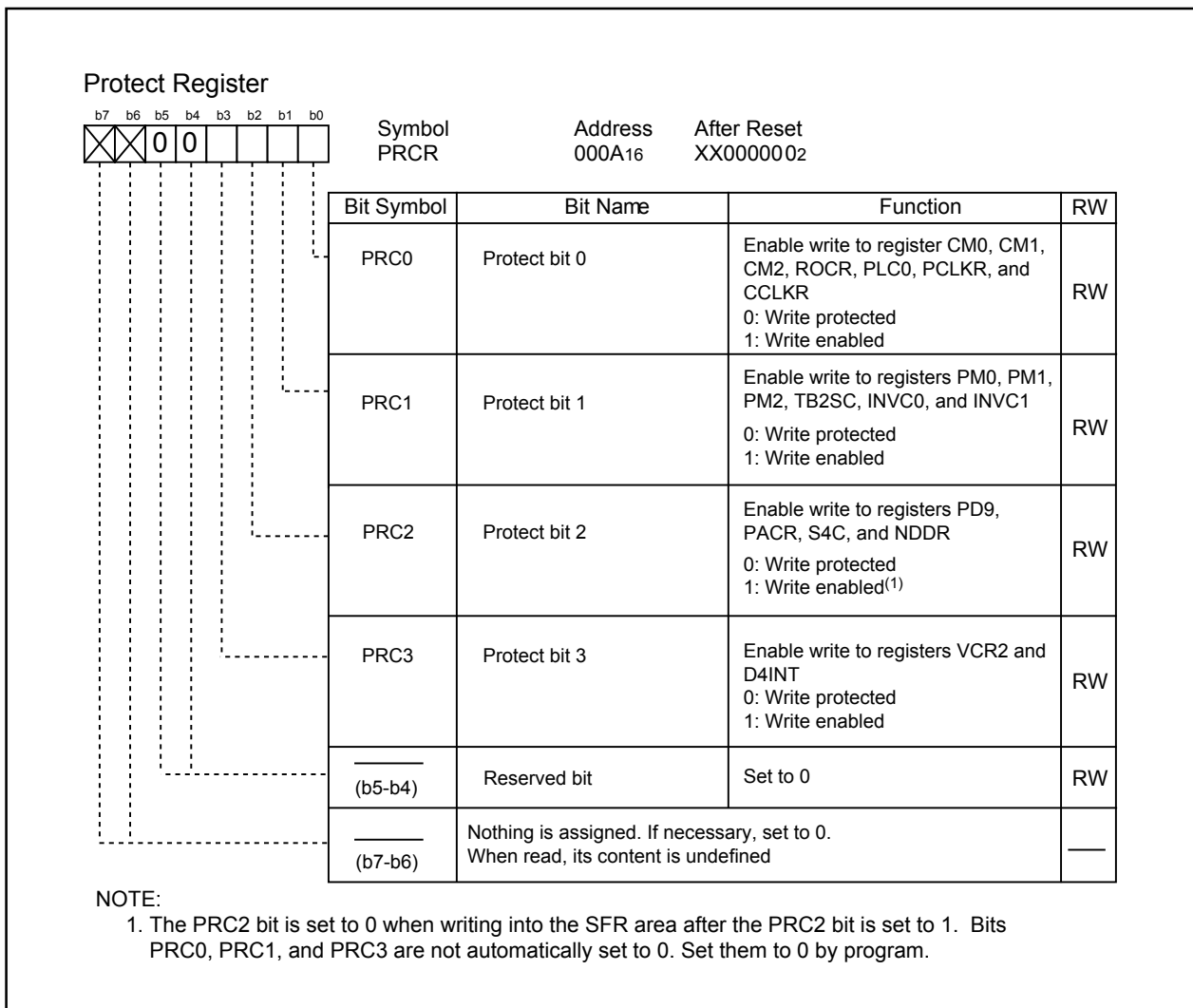


Figure 8.1 PRCR Register

9. Interrupts

Note

The SI/O4 interrupt of peripheral function interrupts is not available in the 64-pin package.
The low voltage detection function is not available in M16C/29 T-ver. and V-ver..

9.1 Type of Interrupts

Figure 9.1 shows types of interrupts.

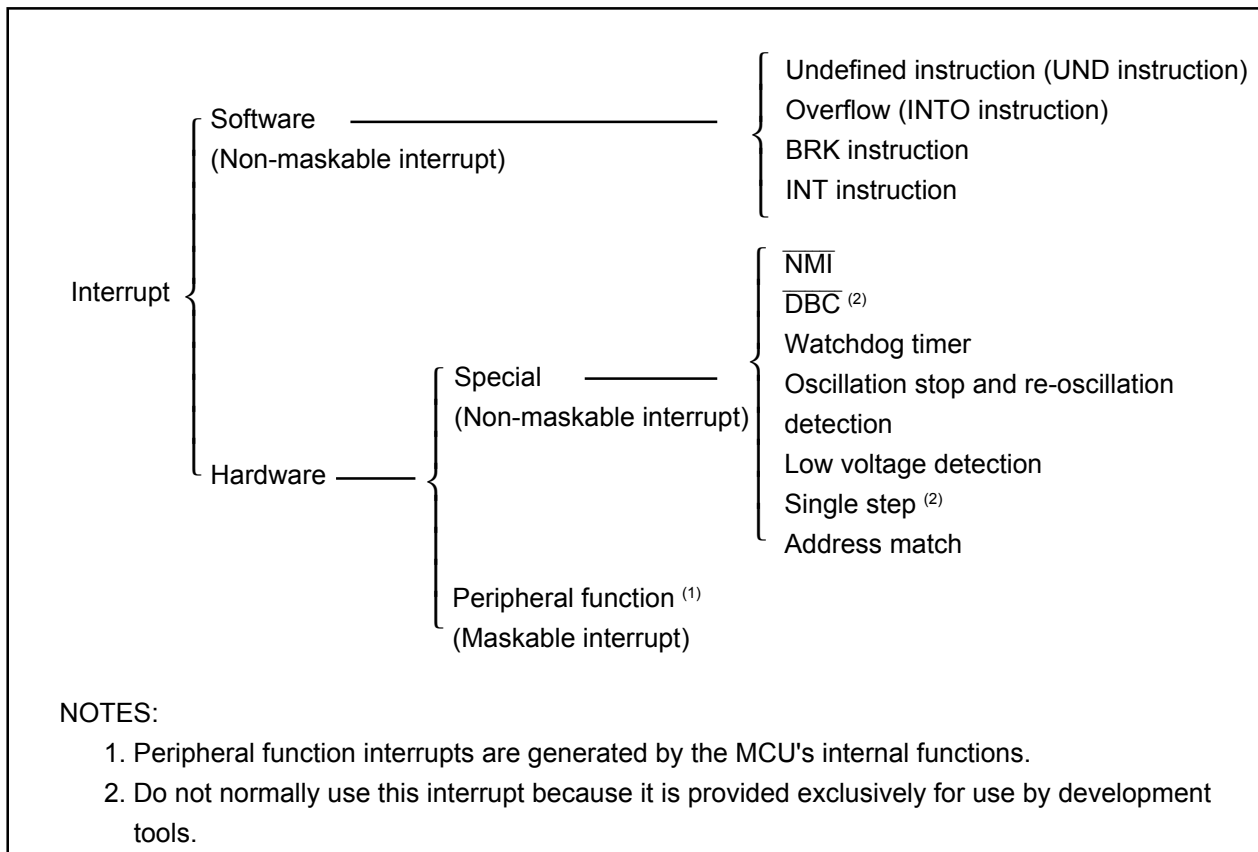


Figure 9.1 Interrupts

- Maskable Interrupt: An interrupt which can be enabled (disabled) by the interrupt enable flag (I flag) or whose interrupt priority **can be changed** by priority level.
- Non-maskable Interrupt: An interrupt which cannot be enabled (disabled) by the interrupt enable flag (I flag) or whose interrupt priority **cannot be changed** by priority level.

9.1.1 Software Interrupts

A software interrupt occurs when executing certain instructions. Software interrupts are non-maskable interrupts.

9.1.1.1 Undefined Instruction Interrupt

An undefined instruction interrupt occurs when executing the UND instruction.

9.1.1.2 Overflow Interrupt

An overflow interrupt occurs when executing the INTO instruction with the O flag set to 1 (the operation resulted in an overflow). The following are instructions whose O flag changes by arithmetic: ABS, ADC, ADCF, ADD, CMP, DIV, DIVU, DIVX, NEG, RMPA, SBB, SHA, SUB

9.1.1.3 BRK Interrupt

A BRK interrupt occurs when executing the BRK instruction.

9.1.1.4 INT Instruction Interrupt

An INT instruction interrupt occurs when executing the INT instruction. Software interrupt Nos. 0 to 63 can be specified for the INT instruction. Because software interrupt Nos. 1 to 31 are assigned to peripheral function interrupts, the same interrupt routine as for peripheral function interrupts can be executed by executing the INT instruction.

In software interrupt Nos. 0 to 31, the U flag is saved to the stack during instruction execution and is cleared 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 Nos. 32 to 63, the U flag does not change state during instruction execution, and the SP then selected is used.

9.1.2 Hardware Interrupts

Hardware interrupts are classified into two types — special interrupts and peripheral function interrupts.

9.1.2.1 Special Interrupts

Special interrupts are non-maskable interrupts.

9.1.2.1.1 $\overline{\text{NMI}}$ Interrupt

An $\overline{\text{NMI}}$ interrupt is generated when input on the $\overline{\text{NMI}}$ pin changes state from high to low. For details about the $\overline{\text{NMI}}$ interrupt, refer to the section "NMI interrupt".

9.1.2.1.2 $\overline{\text{DBC}}$ Interrupt

This interrupt is exclusively for debugger, do not use in any other circumstances.

9.1.2.1.3 Watchdog Timer Interrupt

Generated by the watchdog timer. Once a watchdog timer interrupt is generated, be sure to initialize the watchdog timer. For details about the watchdog timer, refer to the section "watchdog timer".

9.1.2.1.4 Oscillation Stop and Re-oscillation Detection Interrupt

Generated by the oscillation stop and re-oscillation detection function. For details about the oscillation stop and re-oscillation detection function, refer to the section "clock generating circuit".

9.1.2.1.5 Low Voltage Detection Interrupt

Generated by the voltage detection circuit. For details about the voltage detection circuit, refer to the section "voltage detection circuit".

9.1.2.1.6 Single-step Interrupt

Do not normally use this interrupt because it is provided exclusively for use by development tools.

9.1.2.1.7 Address Match Interrupt

An address match interrupt is generated immediately before executing the instruction at the address indicated by the RMAD0 or RMAD1 register, if the corresponding enable bit (AIER0 or AIER1 bit in the AIER register) is set to 1. For details about the address match interrupt, refer to the section "address match interrupt".

9.1.2.2 Peripheral Function Interrupts

Peripheral function interrupts are maskable interrupts and generated by the MCU's internal functions. The interrupt sources for peripheral function interrupts are listed in **Table 9.2** Relocatable Vector Tables. For details about the peripheral functions, refer to the description of each peripheral function in this manual.

9.2 Interrupts and Interrupt Vector

One interrupt vector consists of 4 bytes. Set the start address of each interrupt routine in the respective interrupt vectors. When an interrupt request is accepted, the CPU branches to the address set in the corresponding interrupt vector. **Figure 9.2** shows the interrupt vector.

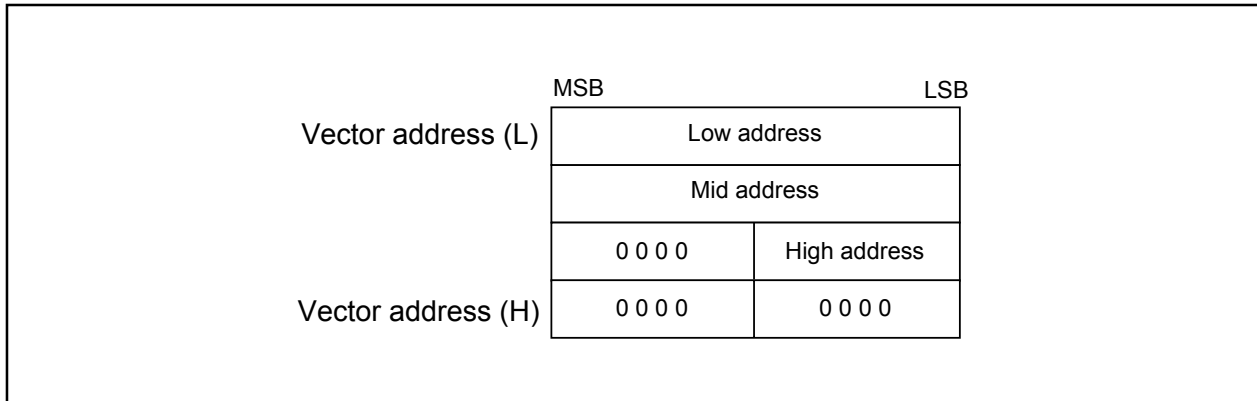


Figure 9.2 Interrupt Vector

9.2.1 Fixed Vector Tables

The fixed vector tables are allocated to the addresses from FFFDC₁₆ to FFFFF₁₆. **Table 9.1** lists the fixed vector tables. In the flash memory version of MCU, the vector addresses (H) of fixed vectors are used by the ID code check function. For details, refer to the section "flash memory rewrite disabling function".

Table 9.1 Fixed Vector Tables

| Interrupt source | Vector table addresses Address (L) to address (H) | Remarks | Reference |
|--|--|--|---|
| Undefined instruction | FFFD ₁₆ to FFFD ₁₆ F | Interrupt on UND instruction | M16C/60, M16C/20 serise software maual |
| Overflow | FFFE0 ₁₆ to FFFE3 ₁₆ | Interrupt on INTO instruction | |
| BRK instruction | FFFE4 ₁₆ to FFFE7 ₁₆ | If the contents of address FFFE7 ₁₆ is FF ₁₆ , program execution starts from the address shown by the vector in the relocatable vector table | |
| Address match | FFFE8 ₁₆ to FFFEB ₁₆ | | Address match interrupt |
| Single step ⁽¹⁾ | FFFE ₁₆ C to FFFE ₁₆ F | | |
| Watchdog timer Oscillation stop and re-oscillation detection, low voltage detection | FFFF0 ₁₆ to FFFF3 ₁₆ | | Watchdog timer, clock generating circuit, voltage detection circuit |
| DBC ⁽¹⁾ | FFFF4 ₁₆ to FFFF7 ₁₆ | | |
| NMI | FFFF8 ₁₆ to FFFF ₁₆ B | | NMI interrupt |
| Reset ⁽²⁾ | FFFF ₁₆ C to FFFFF ₁₆ | | Reset |

NOTE:

- Do not normally use this interrupt because it is provided exclusively for use by development tools.

9.2.2 Relocatable Vector Tables

The 256 bytes beginning with the start address set in the INTB register comprise a relocatable vector table area. **Table 9.2** lists the relocatable vector tables. Setting an even address in the INTB register results in the interrupt sequence being executed faster than in the case of odd addresses.

Table 9.2 Relocatable Vector Tables

| Interrupt source | Vector address ⁽¹⁾ Address (L) to address (H) | Software interrupt number | Reference |
|---|--|------------------------------|---|
| BRK instruction ⁽²⁾ | +0 to +3 (0000 ₁₆ to 0003 ₁₆) | 0 | M16C/60, M16C/20 series software manual |
| CAN0 wakeup ⁽³⁾ | +4 to +7 (0004 ₁₆ to 0007 ₁₆) | 1 | CAN module |
| CAN0 receive completion | +8 to +11 (0008 ₁₆ to 000B ₁₆) | 2 | |
| CAN0 transmit completion | +12 to +15 (000C ₁₆ to 000F ₁₆) | 3 | |
| $\overline{\text{INT}}3$ | +16 to +19 (0010 ₁₆ to 0013 ₁₆) | 4 | $\overline{\text{INT}}$ interrupt |
| IC/OC interrupt 0 | +20 to +23 (0014 ₁₆ to 0017 ₁₆) | 5 | Timer S |
| IC/OC interrupt 1, I ² C bus interface ⁽⁴⁾ | +24 to +27 (0018 ₁₆ to 001B ₁₆) | 6 | Timer S Multi-Master I ² C bus interface |
| IC/OC base timer, S _{CL} /S _{DA} ⁽⁴⁾ | +28 to +31 (001C ₁₆ to 001F ₁₆) | 7 | |
| SI/O4, $\overline{\text{INT}}5$ ⁽⁵⁾ | +32 to +35 (0020 ₁₆ to 0023 ₁₆) | 8 | $\overline{\text{INT}}$ interrupt |
| SI/O3, $\overline{\text{INT}}4$ ⁽⁵⁾ | +36 to +39 (0024 ₁₆ to 0027 ₁₆) | 9 | Serial I/O |
| UART 2 bus collision detection ⁽⁶⁾ | +40 to +43 (0028 ₁₆ to 002B ₁₆) | 10 | Serial I/O |
| DMA0 | +44 to +47 (002C ₁₆ to 002F ₁₆) | 11 | DMAC |
| DMA1 | +48 to +51 (0030 ₁₆ to 0033 ₁₆) | 12 | |
| CAN0 state, error | +52 to +55 (0034 ₁₆ to 0037 ₁₆) | 13 | CAN module |
| A/D, Key input interrupt ⁽⁷⁾ | +56 to +59 (0038 ₁₆ to 003B ₁₆) | 14 | A/D convertor, Key input interrupt |
| UART2 transmit, NACK2 ⁽⁸⁾ | +60 to +63 (003C ₁₆ to 003F ₁₆) | 15 | Serial I/O |
| UART2 receive, ACK2 ⁽⁸⁾ | +64 to +67 (0040 ₁₆ to 0043 ₁₆) | 16 | |
| UART0 transmit | +68 to +71 (0044 ₁₆ to 0047 ₁₆) | 17 | |
| UART0 receive | +72 to +75 (0048 ₁₆ to 004B ₁₆) | 18 | |
| UART1 transmit | +76 to +79 (004C ₁₆ to 004F ₁₆) | 19 | |
| UART1 receive | +80 to +83 (0050 ₁₆ to 0053 ₁₆) | 20 | |
| Timer A0 | +84 to +87 (0054 ₁₆ to 0057 ₁₆) | 21 | Timer |
| Timer A1 | +88 to +91 (0058 ₁₆ to 005B ₁₆) | 22 | |
| Timer A2 | +92 to +95 (005C ₁₆ to 005F ₁₆) | 23 | |
| Timer A3 | +96 to +99 (0060 ₁₆ to 0063 ₁₆) | 24 | |
| Timer A4 | +100 to +103 (0064 ₁₆ to 0067 ₁₆) | 25 | |
| Timer B0 | +104 to +107 (0068 ₁₆ to 006B ₁₆) | 26 | |
| Timer B1 | +108 to +111 (006C ₁₆ to 006F ₁₆) | 27 | |
| Timer B2 | +112 to +115 (0070 ₁₆ to 0073 ₁₆) | 28 | |
| $\overline{\text{INT}}0$ | +116 to +119 (0074 ₁₆ to 0077 ₁₆) | 29 | $\overline{\text{INT}}$ interrupt |
| $\overline{\text{INT}}1$ | +120 to +123 (0078 ₁₆ to 007B ₁₆) | 30 | |
| $\overline{\text{INT}}2$ | +124 to +127 (007C ₁₆ to 007F ₁₆) | 31 | |
| Software interrupt ⁽²⁾ | +128 to +131 (0080 ₁₆ to 0083 ₁₆) to +252 to +255 (00FC ₁₆ to 00FF ₁₆) | 32 to 63 | M16C/60, M16C/20 series software manual |

NOTES:

1. Address relative to address in INTB.
2. These interrupts cannot be disabled using the I flag.
3. Set the IFSR22 bit in the IFSR register to 0.
4. Use bits IFSR26 and IFSR27 in the IFSR2A register to select.
5. Use bits IFSR6 and IFSR7 in the IFSR register to select.
6. Bus collision detection: In IEBus mode, this bus collision detection constitutes the cause of an interrupt. In I²C bus mode, however, a start condition or a stop condition detection constitutes the cause of an interrupt.
7. Use the IFSR21 bit in the IFSR2A register to select.
8. During I²C bus mode, NACK and ACK interrupts comprise the interrupt source.

9.3 Interrupt Control

The following describes how to enable/disable the maskable interrupts, and how to set the priority in which order they are accepted. What is explained here does not apply to nonmaskable interrupts.

Use I flag in the the FLG register, IPL, and bits ILVL2 to ILVL0 in the 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 9.3 shows the interrupt control registers.

Also, the following interrupts share a vector and an interrupt control register.

- $\overline{\text{INT4}}$ and SIO3
- $\overline{\text{INT5}}$ and SIO4
- A/D converter and key input interrupt
- IC/OC base timer and SCL/SDA
- IC/OC interrupt 1 and I²C bus interface

An interrupt request is set by bits IFSR6 and IFSR7 in the IFSR register and bits IFSR27, IFSR26, and IFSR21 in the IFSR2A register. **Figure 9.4** shows registers IFSR register and IFSR2A.

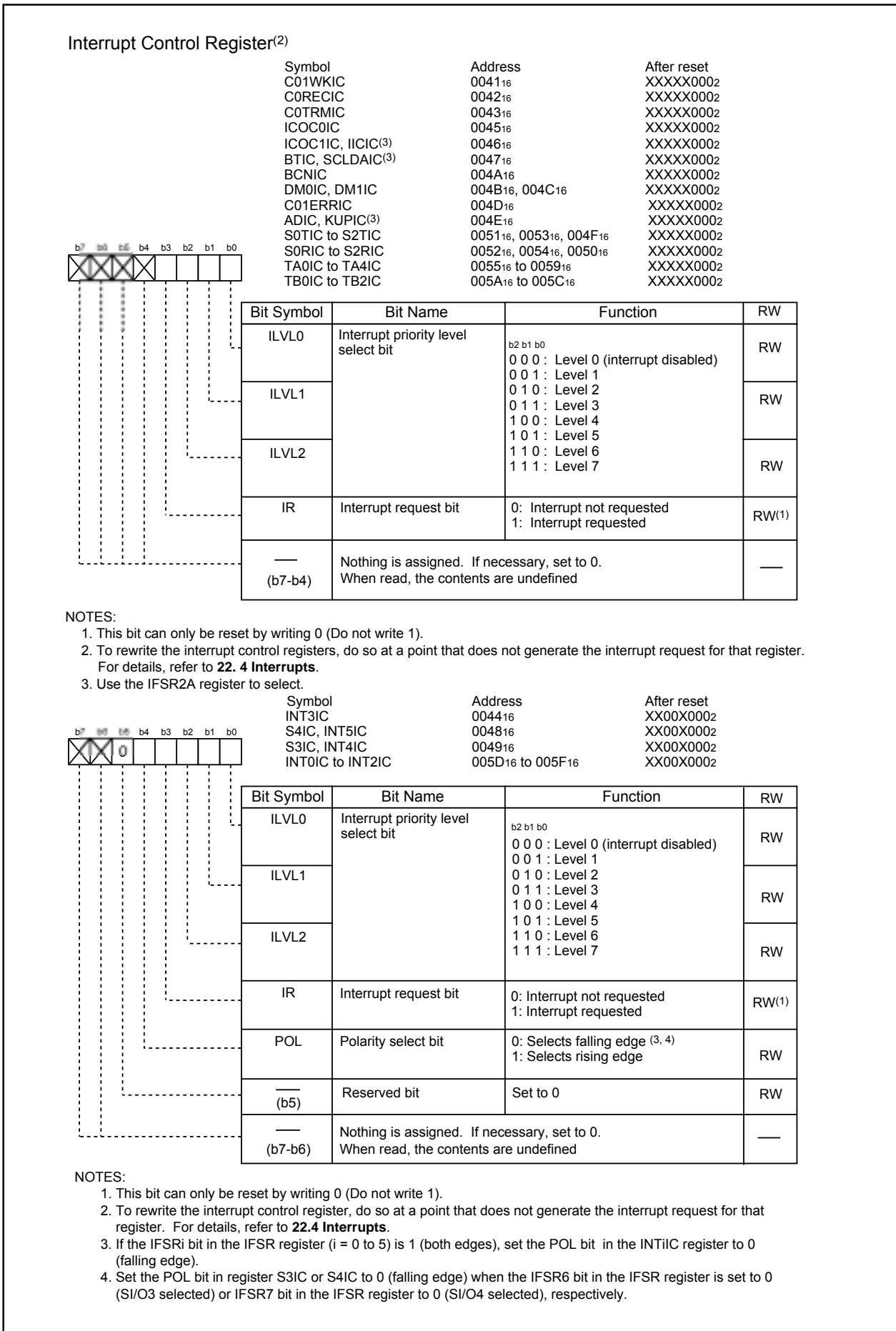


Figure 9.3 Interrupt Control Registers

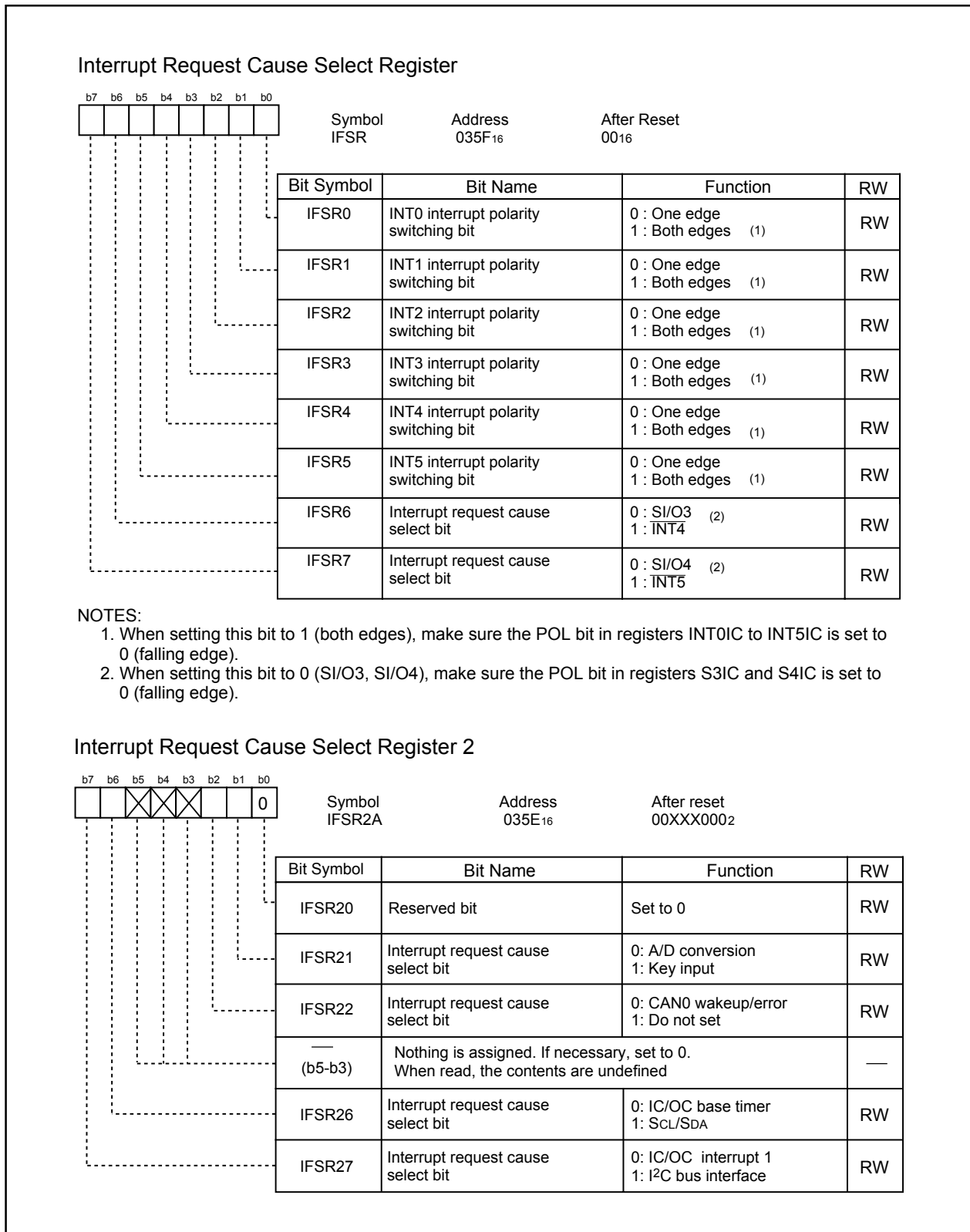


Figure 9.4 IFSR Register and IFSR2A Register

9.3.1 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.

9.3.2 IR Bit

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

The IR bit can be cleared to 0 in a program. Note that do not write 1 to this bit.

9.3.3 ILVL2 to ILVL0 Bits and IPL

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

Table 9.3 shows the settings of interrupt priority levels and **Table 9.4** shows the interrupt priority levels enabled by the IPL.

The following are conditions under which an interrupt is accepted:

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

The I flag, IR bit, bits ILVL2 to ILVL0, and IPL are independent of each other. In no case do they affect one another.

Table 9.3 Settings of Interrupt Priority Levels

| ILVL2 to ILVL0 bits | Interrupt priority level | Priority order |
|---------------------|------------------------------|---|
| 0002 | Level 0 (interrupt disabled) | ——— |
| 0012 | Level 1 | ↓ Low ↓ High |
| 0102 | Level 2 | |
| 0112 | Level 3 | |
| 1002 | Level 4 | |
| 1012 | Level 5 | |
| 1102 | Level 6 | |
| 1112 | Level 7 | |

Table 9.4 Interrupt Priority Levels Enabled by IPL

| IPL | Enabled interrupt priority levels |
|------|--|
| 0002 | Interrupt levels 1 and above are enabled |
| 0012 | Interrupt levels 2 and above are enabled |
| 0102 | Interrupt levels 3 and above are enabled |
| 0112 | Interrupt levels 4 and above are enabled |
| 1002 | Interrupt levels 5 and above are enabled |
| 1012 | Interrupt levels 6 and above are enabled |
| 1102 | Interrupt levels 7 and above are enabled |
| 1112 | All maskable interrupts are disabled |

9.4 Interrupt Sequence

An interrupt sequence (the device behavior from the instant an interrupt is accepted to the instant the interrupt routine is executed) is described here.

If an interrupt occurs during execution of an instruction, the processor determines its priority when the execution of the instruction is completed, and transfers control to the interrupt sequence from the next cycle. If an interrupt occurs during execution of either the SMOVB, SMOVF, SSTR or RMPA instruction, the processor temporarily suspends the instruction being executed, and transfers control to the interrupt sequence.

The CPU behavior during the interrupt sequence is described below. **Figure 9.5** shows time required for executing the interrupt sequence.

- (1) The CPU gets interrupt information (interrupt number and interrupt request priority level) by reading the address 00000₁₆. Then it clears the IR bit for the corresponding interrupt to 0 (interrupt not requested).
- (2) The FLG register immediately before entering the interrupt sequence is saved to the CPU's internal temporary register^(Note).
- (3) The I, D and U flags in the FLG register become as follows:
 - The I flag is cleared to 0 (interrupts disabled).
 - The D flag is cleared to 0 (single-step interrupt disabled).
 - The U flag is cleared to 0 (ISP selected).
 However, the U flag does not change state if an INT instruction for software interrupt Nos. 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 accepted interrupt is set in the IPL.
- (7) The start address of the relevant interrupt routine set in the interrupt vector is stored in the PC.

After the interrupt sequence is completed, the processor resumes executing instructions from the start address of the interrupt routine.

NOTE:

1. This register cannot be used by user.

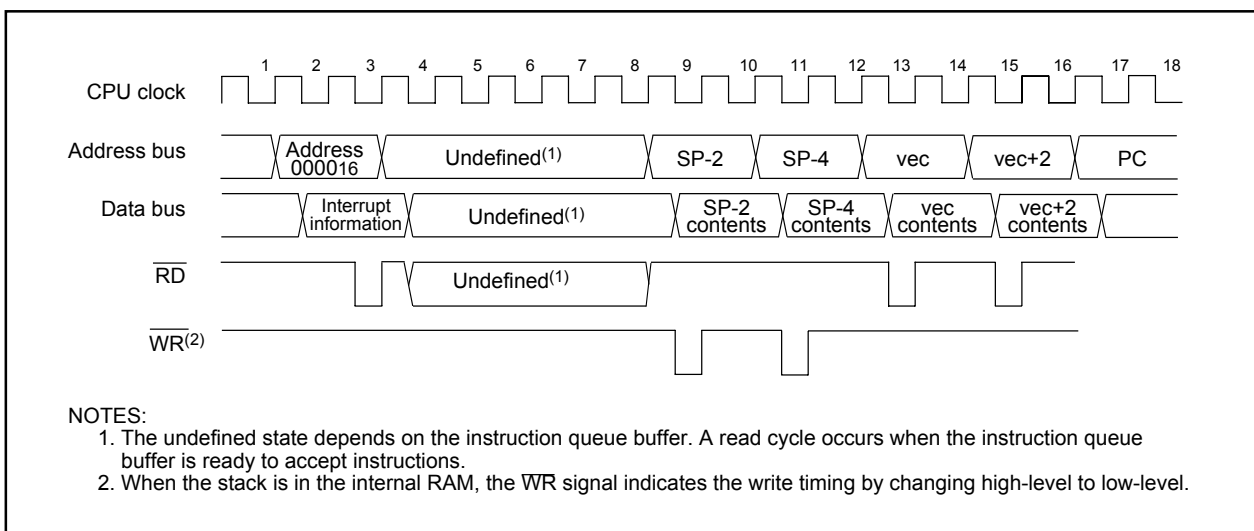


Figure 9.5 Time Required for Executing Interrupt Sequence

9.4.1 Interrupt Response Time

Figure 9.6 shows the interrupt response time. The interrupt response or interrupt acknowledge time denotes time from when an interrupt request is generated till when the first instruction in the interrupt routine is executed. Specifically, it consists of the time from when an interrupt request is generated till when the instruction then executing is completed ((a) in **Figure 9.6**) and the time during which the interrupt sequence is executed ((b) in **Figure 9.6**).

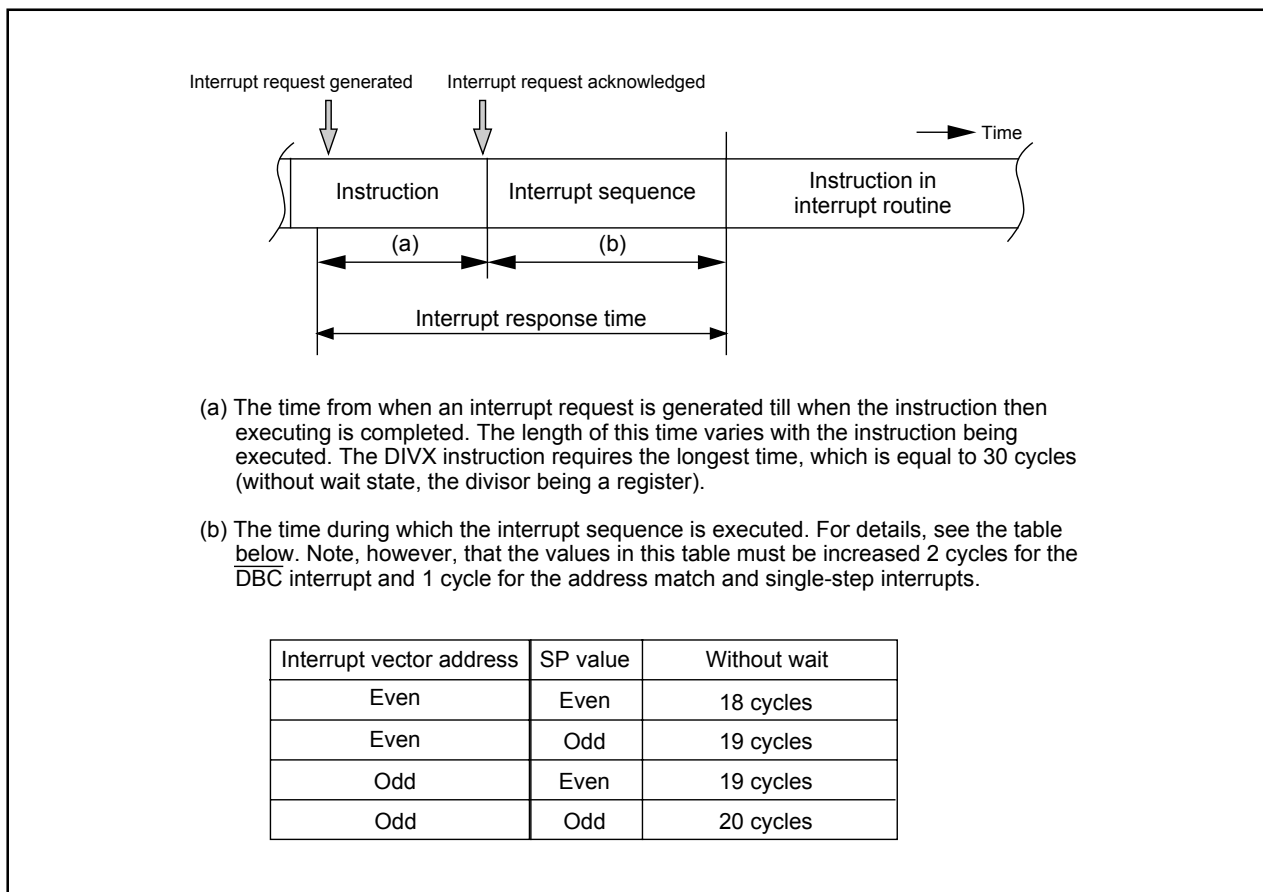


Figure 9.6 Interrupt response time

9.4.2 Variation of IPL when Interrupt Request is Accepted

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

When a software interrupt or special interrupt request is accepted, one of the interrupt priority levels listed in **Table 9.5** is set in the IPL. Shown in **Table 9.5** are the IPL values of software and special interrupts when they are accepted.

Table 9.5 IPL Level That is Set to IPL When A Software or Special Interrupt Is Accepted

| Interrupt sources | IPL setting |
|---|-------------|
| Watchdog timer, \overline{NMI} , Oscillation stop and re-oscillation detection, Low voltage detection | 7 |
| Software, address match, \overline{DBC} , single-step | No change |

9.4.3 Saving Registers

In the interrupt sequence, the FLG register and PC are saved to the stack.

At this time, the 4 high-order bits of the PC and the 4 high-order (IPL) and 8 low-order bits of the FLG register, 16 bits in total, are saved to the stack first. Next, the 16 low-order bits of the PC are saved.

Figure 9.7 shows the stack status before and after an interrupt request is accepted.

The other necessary registers must be saved in a program at the beginning of the interrupt routine. Use the PUSHM instruction, and all registers except SP can be saved with a single instruction.

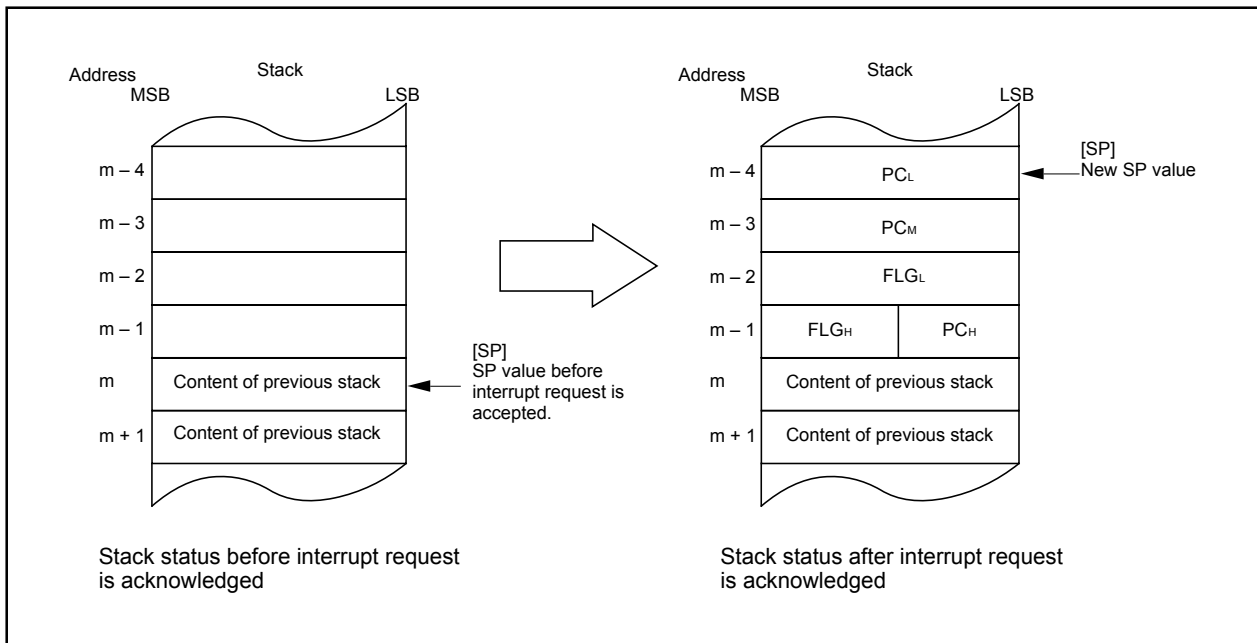


Figure 9.7 Stack Status Before and After Acceptance of Interrupt Request

The operation of saving registers carried out in the interrupt sequence is dependent on whether the SP⁽¹⁾, at the time of acceptance of an interrupt request, is even or odd. If the stack pointer ⁽¹⁾ is even, the FLG register and the PC are saved, 16 bits at a time. If odd, they are saved in two steps, 8 bits at a time. **Figure 9.8** shows the operation of the saving registers.

NOTE:

1. When any INT instruction in software numbers 32 to 63 has been executed, this is the SP indicated by the U flag. Otherwise, it is the ISP.

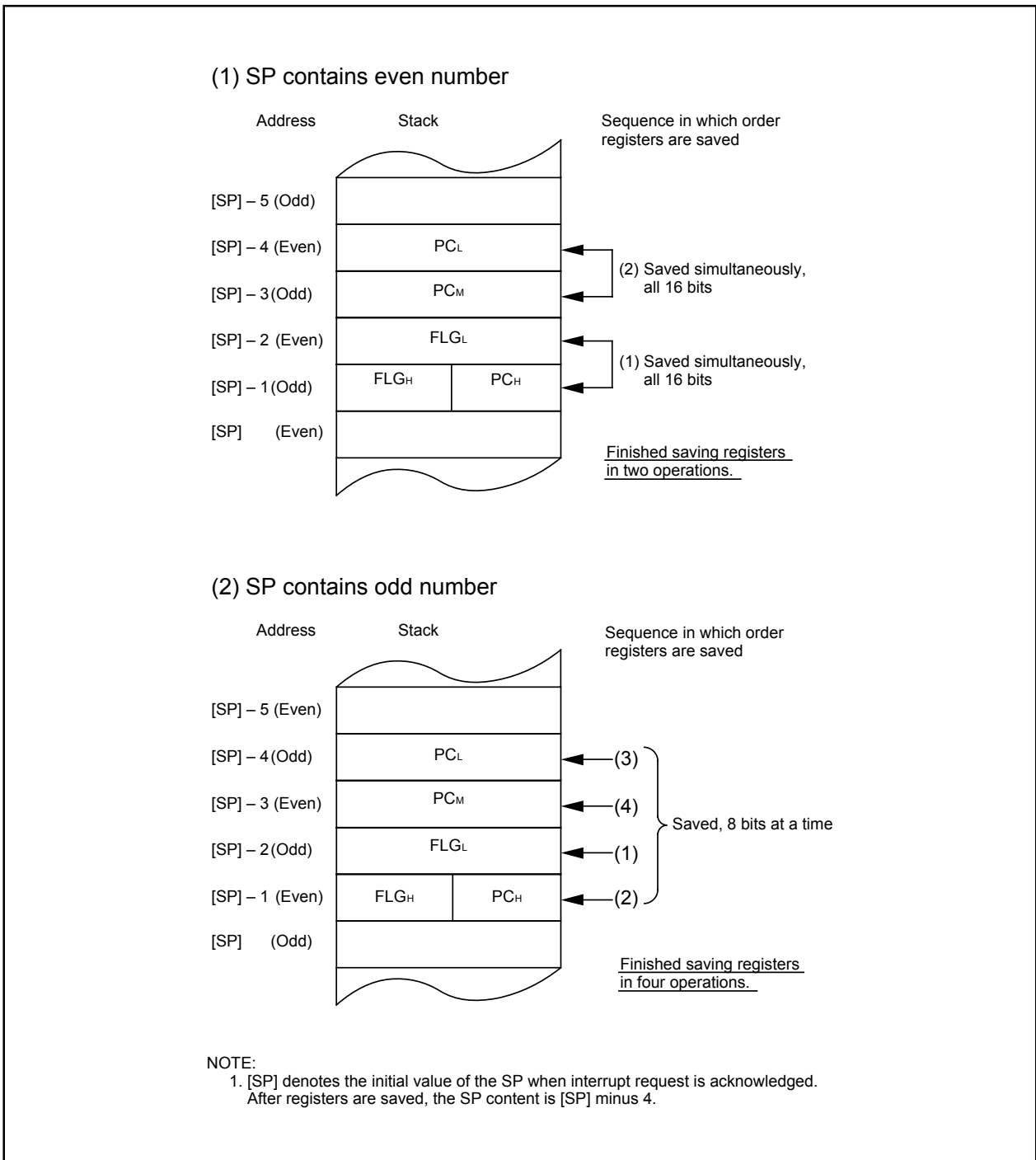


Figure 9.8 Operation of Saving Register

9.4.4 Returning from an Interrupt Routine

The FLG register and PC in the state in which they were immediately before entering the interrupt sequence are restored from the stack by executing the REIT instruction at the end of the interrupt routine. Thereafter the CPU returns to the program which was being executed before accepting the interrupt request.

Return the other registers saved by a program within the interrupt routine using the POPM or similar instruction before executing the REIT instruction.

9.5 Interrupt Priority

If two or more interrupt requests are generated while executing one instruction, the interrupt request that has the highest priority is accepted.

For maskable interrupts (peripheral functions), any desired priority level can be selected using bits ILVL2 to ILVL0. However, if two or more maskable interrupts have the same priority level, their interrupt priority is resolved by hardware, with the highest priority interrupt accepted.

The watchdog timer and other special interrupts have their priority levels set in hardware. **Figure 9.9** shows the priorities of hardware interrupts.

Software interrupts are not affected by the interrupt priority. If an instruction is executed, control branches invariably to the interrupt routine.

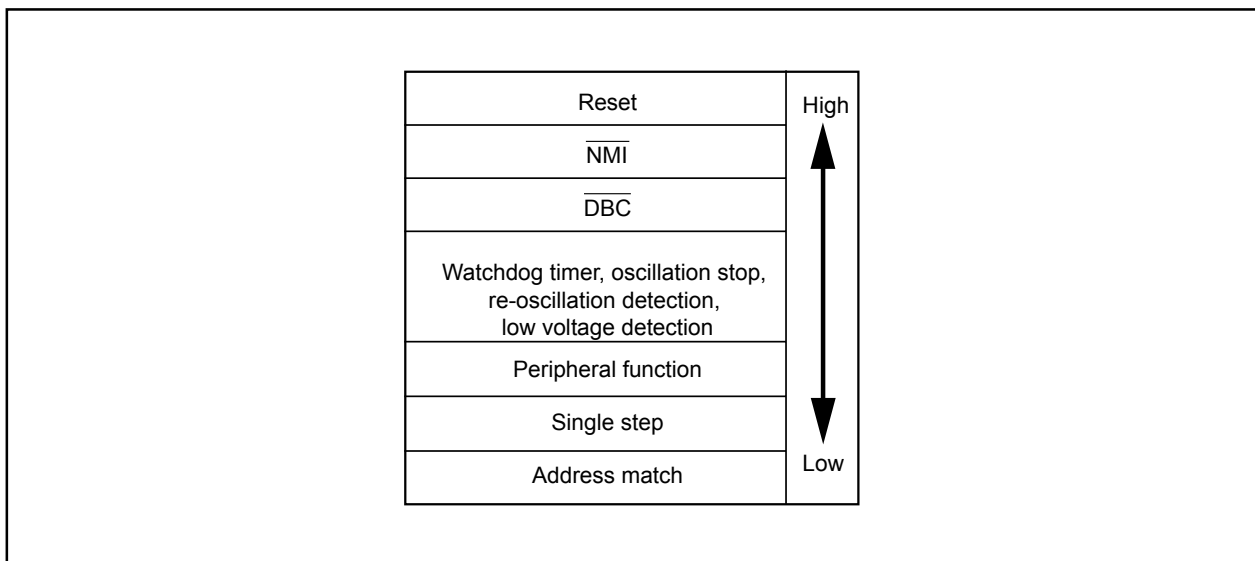


Figure 9.9 Hardware Interrupt Priority

9.5.1 Interrupt Priority Resolution Circuit

The interrupt priority resolution circuit is used to select the interrupt with the highest priority among those requested.

Figure 9.10 shows the circuit that judges the interrupt priority level.

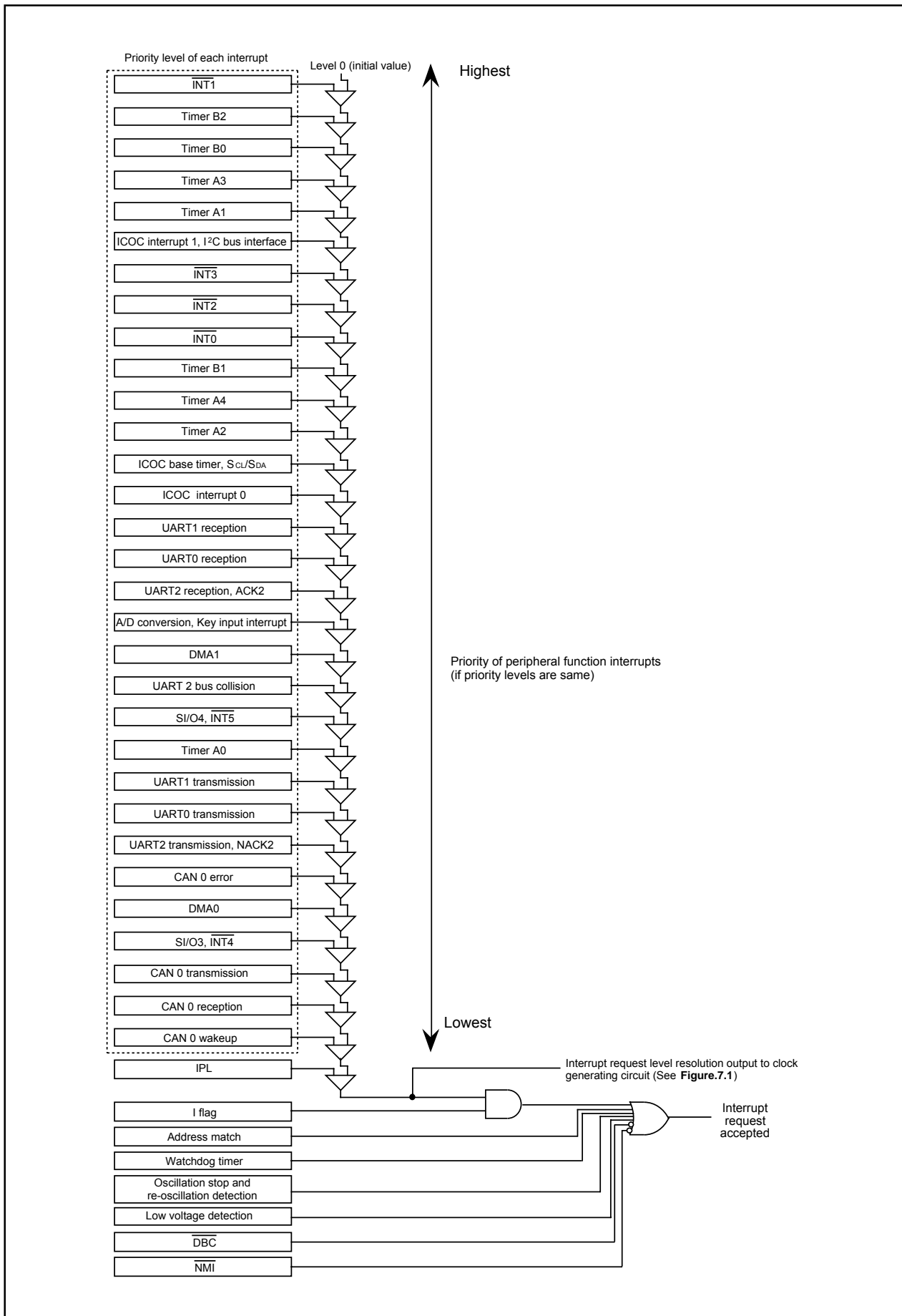


Figure 9.10 Interrupts Priority Select Circuit

9.6 INT Interrupt

INT_i interrupt (i=0 to 5) is triggered by the edges of external inputs. The edge polarity is selected using the IFSR_i bit in the IFSR register.

The INT₅ input has an effective digital debounce function for a noise rejection. Refer to "19.6 Digital Debounce function" for this detail. When using INT₅ interrupt to exit stop mode, set the P17DDR register to FF₁₆ before entering stop mode.

To use the INT₄ interrupt, set the IFSR6 bit in the IFSR register to 1 (INT₄). To use the INT₅ interrupt, set the IFSR7 bit in the IFSR register to 1 (INT₅).

After modifying bit IFSR6 or IFSR7, clear the corresponding IR bit to 0 (interrupt not requested) before enabling the interrupt.

Figure 9.11 shows the IFSR registers.

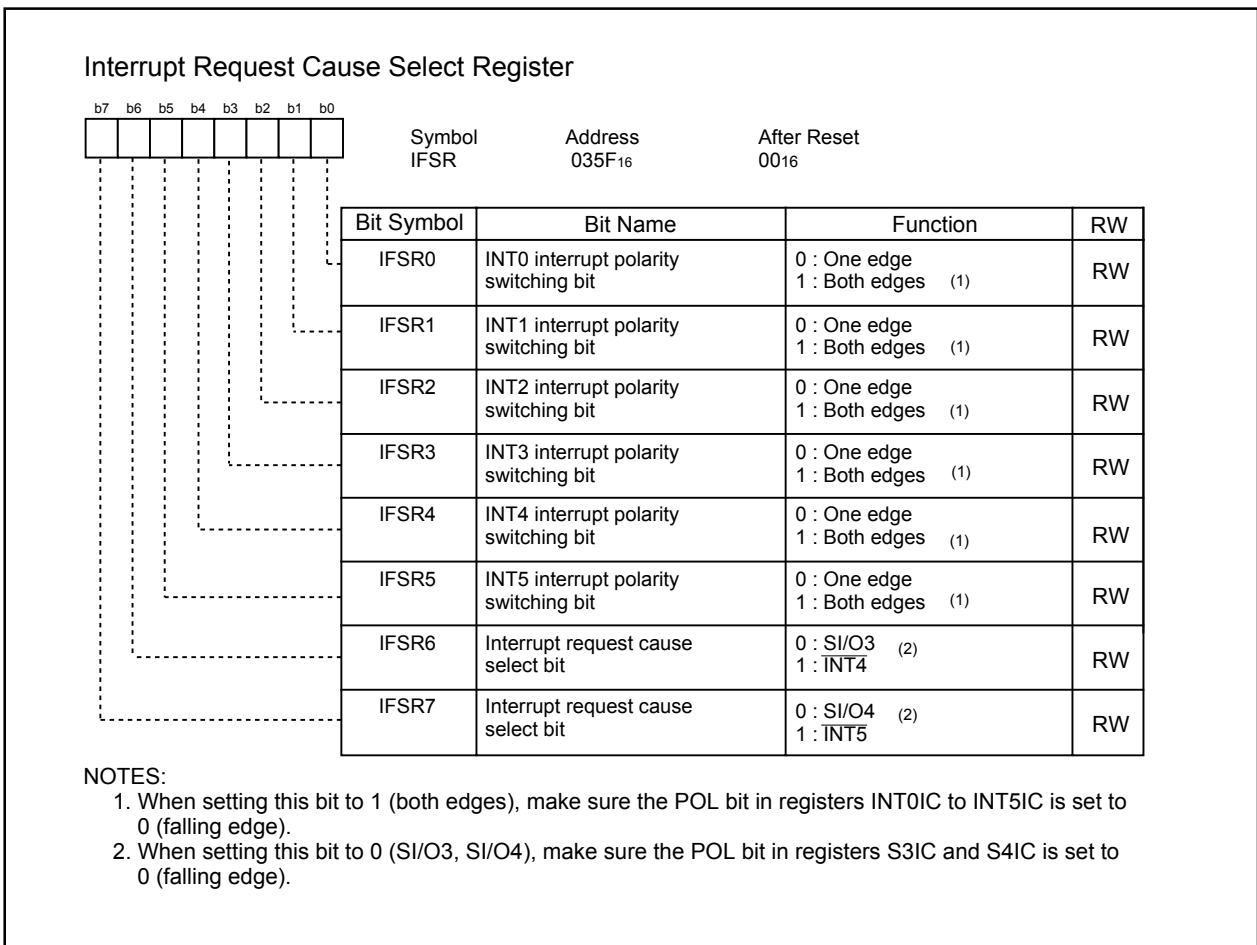


Figure 9.11 IFSR Register

9.7 $\overline{\text{NMI}}$ Interrupt

An $\overline{\text{NMI}}$ interrupt request is generated when input on the $\overline{\text{NMI}}$ pin changes state from high to low, after the $\overline{\text{NMI}}$ interrupt was enabled by writing a 1 to bit 4 in the register PM2. The $\overline{\text{NMI}}$ interrupt is a non-maskable interrupt, once it is enabled.

The input level of this $\overline{\text{NMI}}$ interrupt input pin can be read by accessing the P8_5 bit in the P8 register.

$\overline{\text{NMI}}$ is disabled by default after reset (the pin is a GPIO pin, P85) and can be enabled using bit 4 in the PM2 register. Once enabled, it can only be disabled by a reset signal.

The $\overline{\text{NMI}}$ input has a digital debounce function for noise rejection. Refer to "19.6 Digital Debounce function" for details. When using $\overline{\text{NMI}}$ interrupt to exit stop mode, set the NDDR register to FF16 before entering stop mode.

9.8 Key Input Interrupt

A key input interrupt is generated when input on any of the P104 to P107 pins which has had bits PD10_7 to PD10_4 in the PD10 register set to 0 (= input) goes low. Key input interrupts can be used for a key-on wakeup function to get the MCU to exit stop or wait modes. However, if you intend to use the key input interrupt, do not use P104 to P107 as analog input ports. **Figure 9.12** shows the block diagram of the key input interrupt. Note, however, that while input on any pin which has had bits PD10_7 to PD10_4 set to 0 (= input mode) is pulled low, inputs on all other pins of the port are not detected as interrupts.

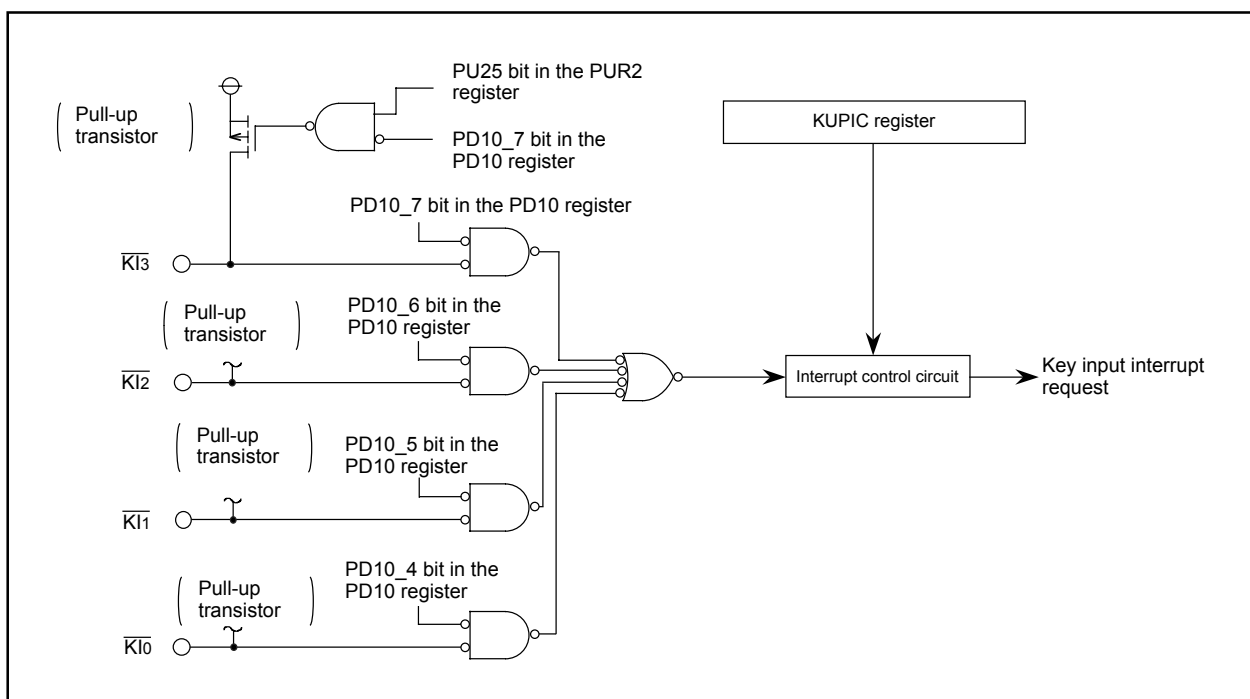


Figure 9.12 Key Input Interrupt

9.9 CAN0 Wake-up Interrupt

CAN0 wake-up interrupt occurs when a falling edge is input to CRX. The CAN0 wake-up interrupt is enabled when the PortEn bit is set to 1 (CTX/CRX function) and Sleep bit is set to 1 (Sleep mode enabled) in the C0CTRL register. **Figure 9.13** shows the block diagram of the CAN0 wake-up interrupt.

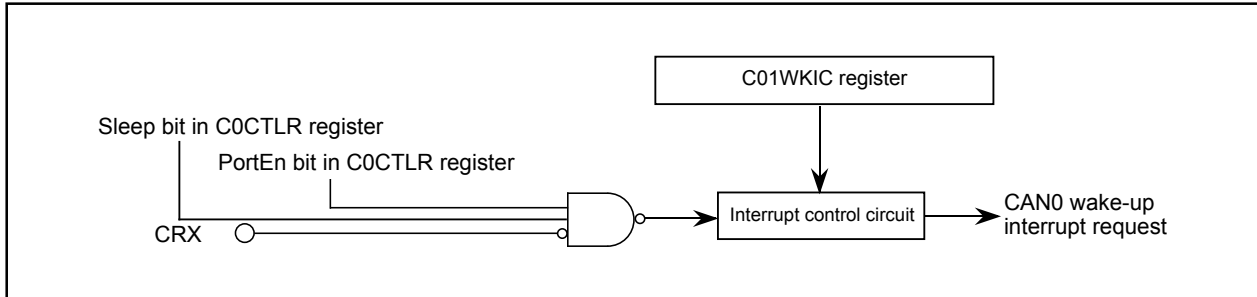


Figure 9.13 CAN0 Wake-up Interrupt Block Diagram

9.10 Address Match Interrupt

An address match interrupt request is generated immediately before executing the instruction at the address indicated by the RMAD_i register (i=0 to 1). Set the start address of any instruction in the RMAD_i register. Use bits AIER1 and AIER0 in the AIER register to enable or disable the interrupt. Note that the address match interrupt is unaffected by the I flag and IPL. For address match interrupts, the value of the PC that is saved to the stack area varies depending on the instruction being executed (refer to “**Saving Registers**”).

(The value of the PC that is saved to the stack area is not the correct return address.) Therefore, follow one of the methods described below to return from the address match interrupt.

- Rewrite the content of the stack and then use the REIT instruction to return.
- Restore the stack to its previous state before the interrupt request was accepted by using the POP or similar other instruction and then use a jump instruction to return.

Table 9.6 shows the value of the PC that is saved to the stack area when an address match interrupt request is accepted.

Figure 9.14 shows registers AIER, RMAD₀, and RMAD₁.

Table 9.6 PC Value Saved in Stack Area When Address Match Interrupt Request Is Acknowledged

| Instruction at the address indicated by the RMADi register | Value of the PC that is saved to the stack area |
|--|---|
| <ul style="list-style-type: none"> • 2-byte op-code instruction • 1-byte op-code instructions which are followed: 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 #IMM8,dest STNZ.B #IMM8,dest STZX.B #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) | The address indicated by the RMADi register +2 |
| Instructions other than the above | The address indicated by the RMADi register +1 |

Value of the PC that is saved to the stack area : Refer to “Saving Registers”.
 Op-code is an abbreviation of Operation Code. It is a portion of instruction code.
 Refer to Chapter 4 Instruction Code/Number of Cycles in M16C/60, M16C/20 Series Software Manual. Op-code is shown as a bold-framed figure directly below the Syntax.

Table 9.7 Relationship Between Address Match Interrupt Sources and Associated Registers

| Address match interrupt sources | Address match interrupt enable bit | Address match interrupt register |
|---------------------------------|------------------------------------|----------------------------------|
| Address match interrupt 0 | AIER0 | RMAD0 |
| Address match interrupt 1 | AIER1 | RMAD1 |

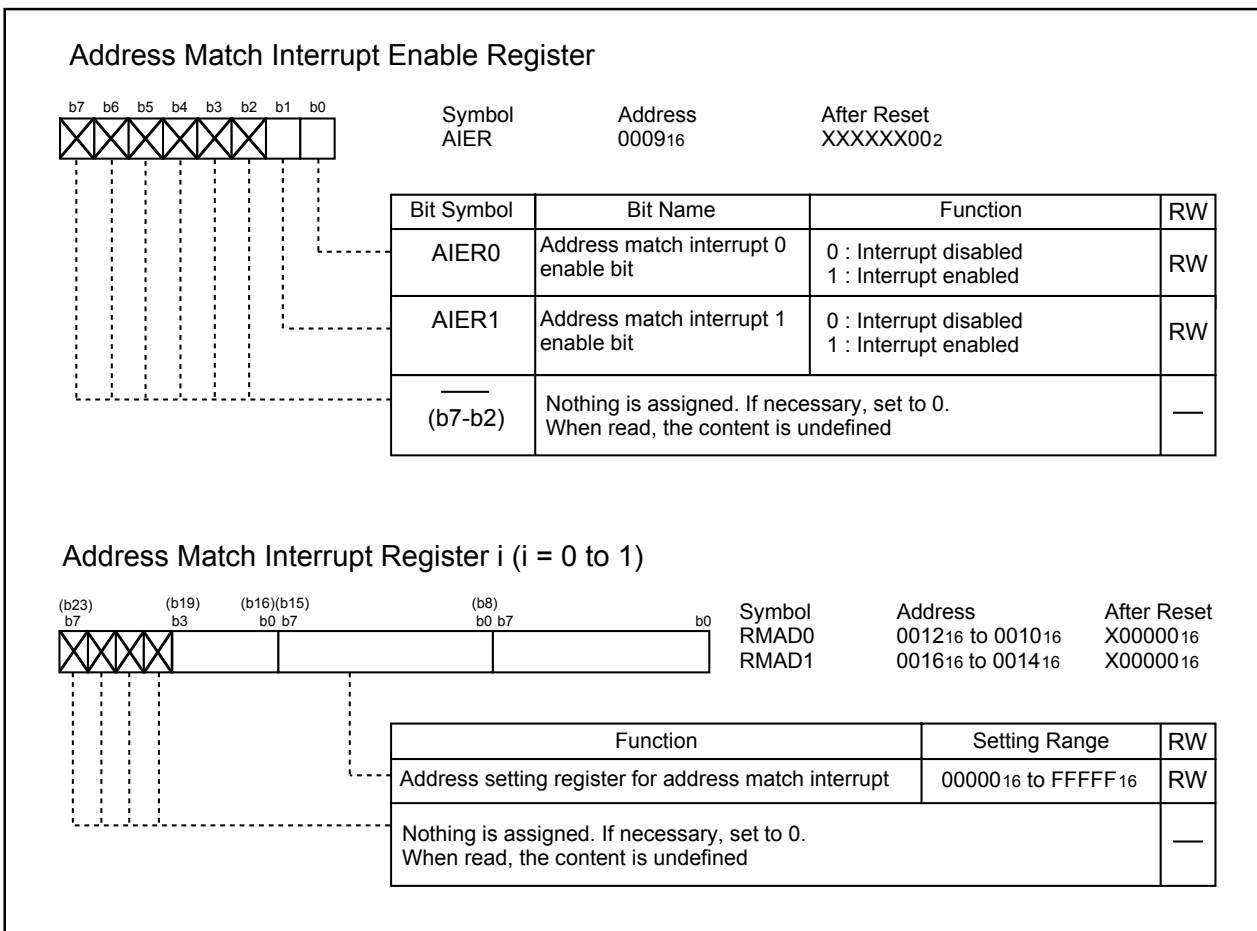


Figure 9.14 AIER Register, RMAD0 and RMAD1 Registers

10. Watchdog Timer

The watchdog timer is the function of detecting when the program is out of control. Therefore, we recommend using the watchdog timer to improve reliability of a system. The watchdog timer contains a 15-bit counter which counts down the clock derived by dividing the CPU clock using the prescaler. Whether to generate a watchdog timer interrupt request or apply a watchdog timer reset as an operation to be performed when the watchdog timer underflows after reaching the terminal count can be selected using the PM12 bit in the PM1 register. The PM12 bit can only be set to 1 (reset). Once this bit is set to 1, it cannot be set to 0 (watchdog timer interrupt) in a program. Refer to **5.3 Watchdog Timer Reset** for the details of watchdog timer reset.

When the main clock source is selected for CPU clock, on-chip oscillator clock, PLL clock, the WDC7 bit in the WDC register value for prescaler can be chosen to be 16 or 128. If a sub-clock is selected for CPU clock, the prescaler is always 2 no matter how the WDC7 bit is set. The period of watchdog timer can be calculated as given below. The period of watchdog timer is, however, subject to an error due to the prescaler.

With main clock source chosen for CPU clock, on-chip oscillator clock, PLL clock

$$\text{Watchdog timer period} = \frac{\text{Prescaler dividing (16 or 128) X Watchdog timer count (32768)}}{\text{CPU clock}}$$

With sub-clock chosen for CPU clock

$$\text{Watchdog timer period} = \frac{\text{Prescaler dividing (2) X Watchdog timer count (32768)}}{\text{CPU clock}}$$

For example, when CPU clock is set to 16 MHz and the divide-by-N value for the prescaler is set to 16, the watchdog timer period is approx. 32.8 ms.

The watchdog timer is initialized by writing to the WDTS register. The prescaler is initialized after reset. Note that the watchdog timer and the prescaler both are inactive after reset, so that the watchdog timer is activated to start counting by writing to the WDTS register.

Write the WDTS register with shorter cycle than the watchdog timer cycle. Set the WDTS register also in the beginning of the watchdog timer interrupt routine.

In stop mode and wait mode, the watchdog timer and prescaler are stopped. Counting is resumed from the held value when the modes or state are released.

Figure 10.1 shows the block diagram of the watchdog timer. Figure 10.2 shows the watchdog timer-related registers.

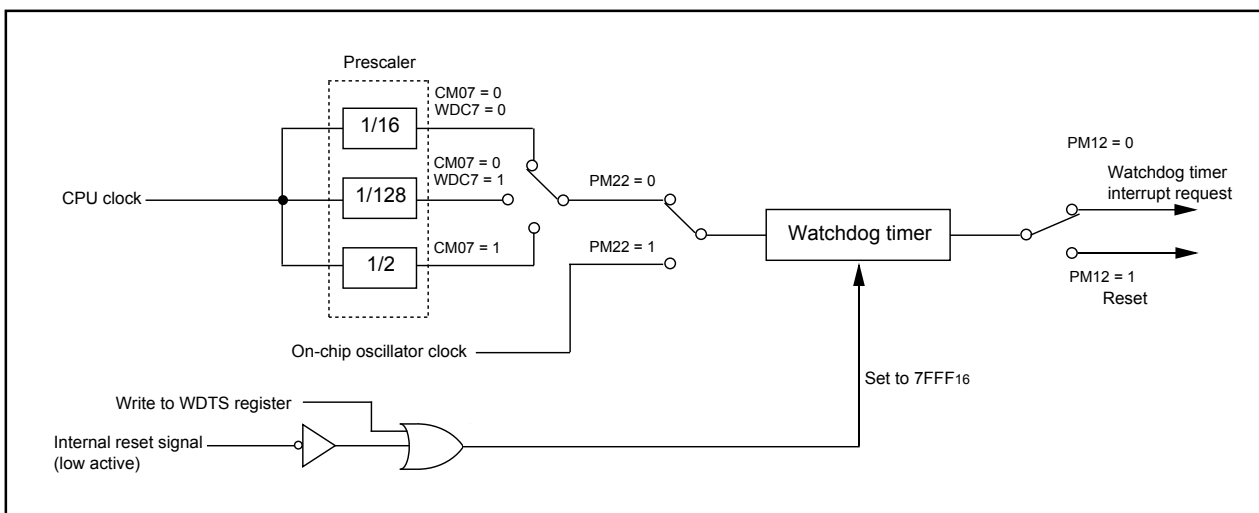


Figure 10.1 Watchdog Timer Block Diagram

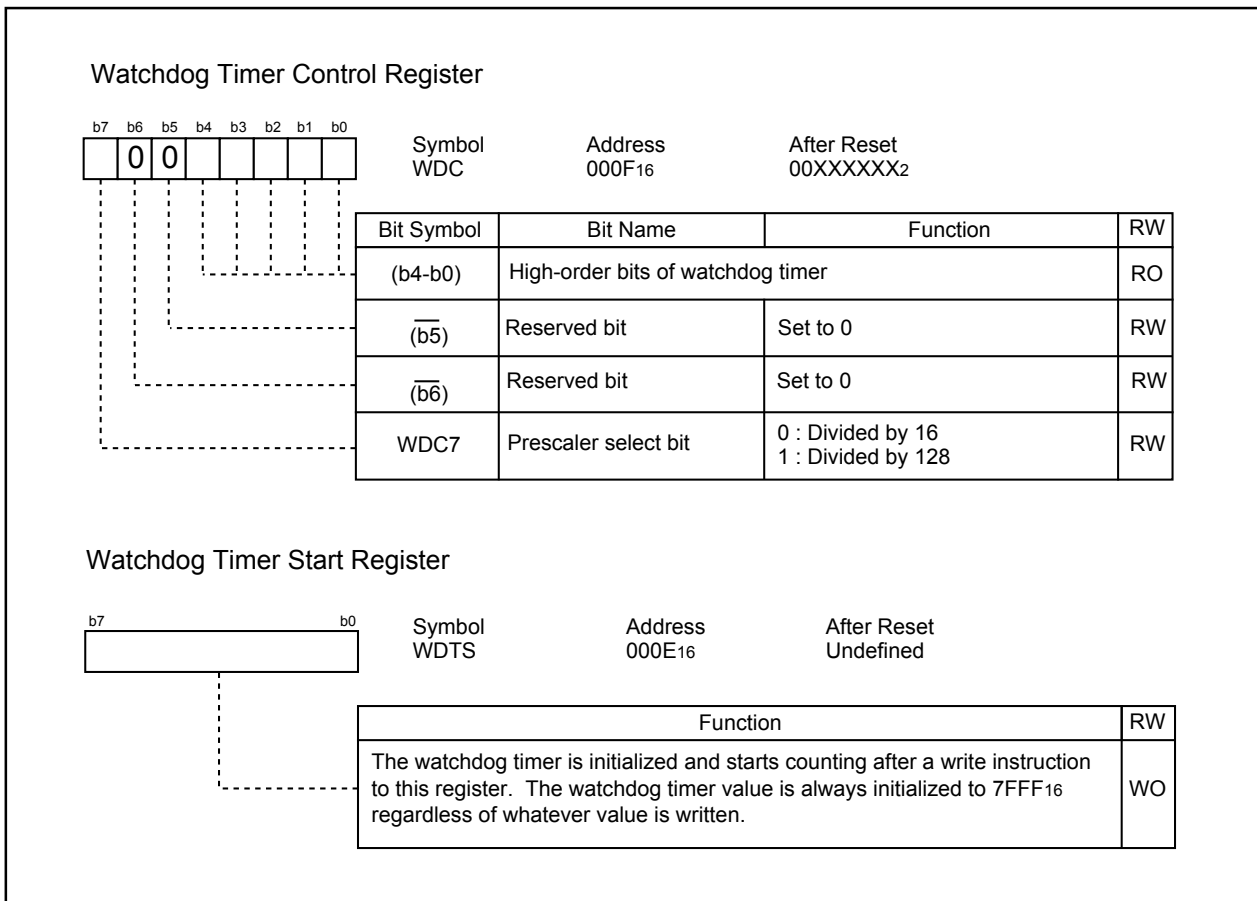


Figure 10.2 WDC Register and WDTS Register

10.1 Count Source Protective Mode

In this mode, a on-chip oscillator clock is used for the watchdog timer count source. The watchdog timer can be kept being clocked even when CPU clock stops as a result of run-away.

Before this mode can be used, the following register settings are required:

- (1) Set the PRC1 bit in the PRCR register to 1 (enable writes to PM1 and PM2 registers).
- (2) Set the PM12 bit in the PM1 register to 1 (reset when the watchdog timer underflows).
- (3) Set the PM22 bit in the PM2 register to 1 (on-chip oscillator clock used for the watchdog timer count source).
- (4) Set the PRC1 bit in the PRCR register to 0 (disable writes to PM1 and PM2 registers).
- (5) Write to the WDTS register (watchdog timer starts counting).

Setting the PM22 bit to 1 results in the following conditions

- The on-chip oscillator continues oscillating even if the CM21 bit in the CM2 register is set to "0" (main clock or PLL clock) (system clock of count source selected by the CM21 bit is valid)
- The on-chip oscillator starts oscillating, and the on-chip oscillator clock becomes the watchdog timer count source.

$$\text{Watchdog timer period} = \frac{\text{Watchdog timer count (32768)}}{\text{on-chip oscillator clock}}$$

- The CM10 bit in the CM1 register is disabled against write. (Writing a 1 has no effect, nor is stop mode entered.)
- The watchdog timer does not stop when in wait mode.

11. DMAC

Note

Do not use SI/O4 interrupt request as a DMA request in the 64-pin package.

The DMAC (Direct Memory Access Controller) allows data to be transferred without the CPU intervention. Two DMAC channels are included. Each time a DMA request occurs, the DMAC transfers one (8 or 16-bit) data from the source address to the destination address. The DMAC uses the same data bus as used by the CPU. Because the DMAC has higher priority of bus control than the CPU and because it makes use of a cycle steal method, it can transfer one word (16 bits) or one byte (8 bits) of data within a very short time after a DMA request is generated. **Figure 11.1** shows the block diagram of the DMAC. **Table 11.1** shows the DMAC specifications. **Figures 11.2 to 11.4** show the DMAC-related registers.

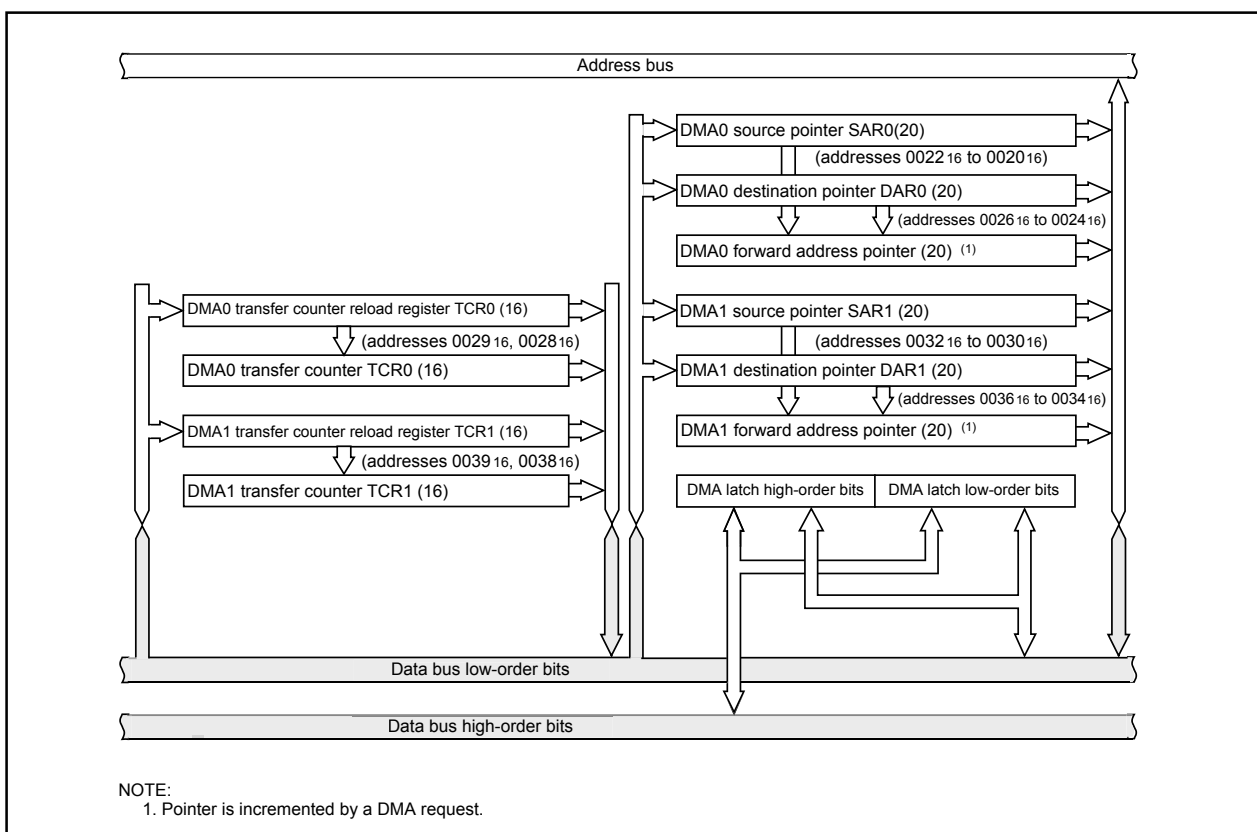


Figure 11.1 DMAC Block Diagram

A DMA request is generated by a write to the DSR bit in the DMiSL register ($i = 0, 1$), as well as by an interrupt request which is generated by any function specified by the DMS and bits DSEL3 to DSEL0 in the DMiSL register. However, unlike in the case of interrupt requests, DMA requests are not affected by the I flag and the interrupt control register, so that even when interrupt requests are disabled and no interrupt request can be accepted, DMA requests are always accepted. Furthermore, because the DMAC does not affect interrupts, the IR bit in the interrupt control register does not change state due to a DMA transfer. A data transfer is initiated each time a DMA request is generated when the DMAE bit in the DMiCON register is set to 1 (DMA enabled). However, if the cycle in which a DMA request is generated is faster than the DMA transfer cycle, the number of transfer requests generated and the number of times data is transferred may not match. For details, refer to “DMA Requests”.

Table 11.1 DMAC Specifications

| Item | | Specification |
|--|-----------------|---|
| No. of channels | | 2 (cycle steal method) |
| Transfer memory space | | <ul style="list-style-type: none"> • From any address in the 1M bytes space to a fixed address • From a fixed address to any address in the 1M bytes space • From a fixed address to a fixed address |
| Maximum No. of bytes transferred | | 128K bytes (with 16-bit transfers) or 64K bytes (with 8-bit transfers) |
| DMA request factors ^(1, 2) | | Falling edge of $\overline{\text{INT0}}$ or $\overline{\text{INT1}}$ Both edge of $\overline{\text{INT0}}$ or $\overline{\text{INT1}}$ Timer A0 to timer A4 interrupt requests Timer B0 to timer B2 interrupt requests UART0 transfer, UART0 reception interrupt requests UART1 transfer, UART1 reception interrupt requests UART2 transfer, UART2 reception interrupt requests SI/O3, SI/O4 interrupt requests A/D conversion interrupt requests Timer S(IC/OC) requests Software triggers |
| Channel priority | | DMA0 > DMA1 (DMA0 takes precedence) |
| Transfer unit | | 8 bits or 16 bits |
| Transfer address direction | | forward or fixed (The source and destination addresses cannot both be in the forward direction) |
| Transfer mode | Single transfer | Transfer is completed when the DMA _i transfer counter (i = 0,1) underflows after reaching the terminal count |
| | Repeat transfer | When the DMA _i transfer counter underflows, it is reloaded with the value of the DMA _i transfer counter reload register and a DMA transfer is continued with it |
| DMA interrupt request generation timing | | When the DMA _i transfer counter underflowed |
| DMA startup the | | Data transfer is initiated each time a DMA request is generated when DMAE bit in the DMA _i CON register = 1 (enabled) |
| DMA shutdown | Single transfer | <ul style="list-style-type: none"> • When the DMAE bit is set to 0 (disabled) • After the DMA_i transfer counter underflows |
| | Repeat transfer | When the DMAE bit is set to 0 (disabled) |
| Reload timing for forward address pointer and transfer counter | | When a data transfer is started after setting the DMAE bit to 1 (enabled), the forward address pointer is reloaded with the value of the SAR _i or the DAR _i pointer whichever is specified to be in the forward direction and the DMA _i transfer counter is reloaded with the value of the DMA _i transfer counter reload register |

NOTES:

1. DMA transfer is not effective to any interrupt. DMA transfer is affected neither by the I flag nor by the interrupt control register.
2. The selectable causes of DMA requests differ with each channel.
3. Make sure that no DMAC-related registers (addresses 0020₁₆ to 003F₁₆) are accessed by the DMAC.

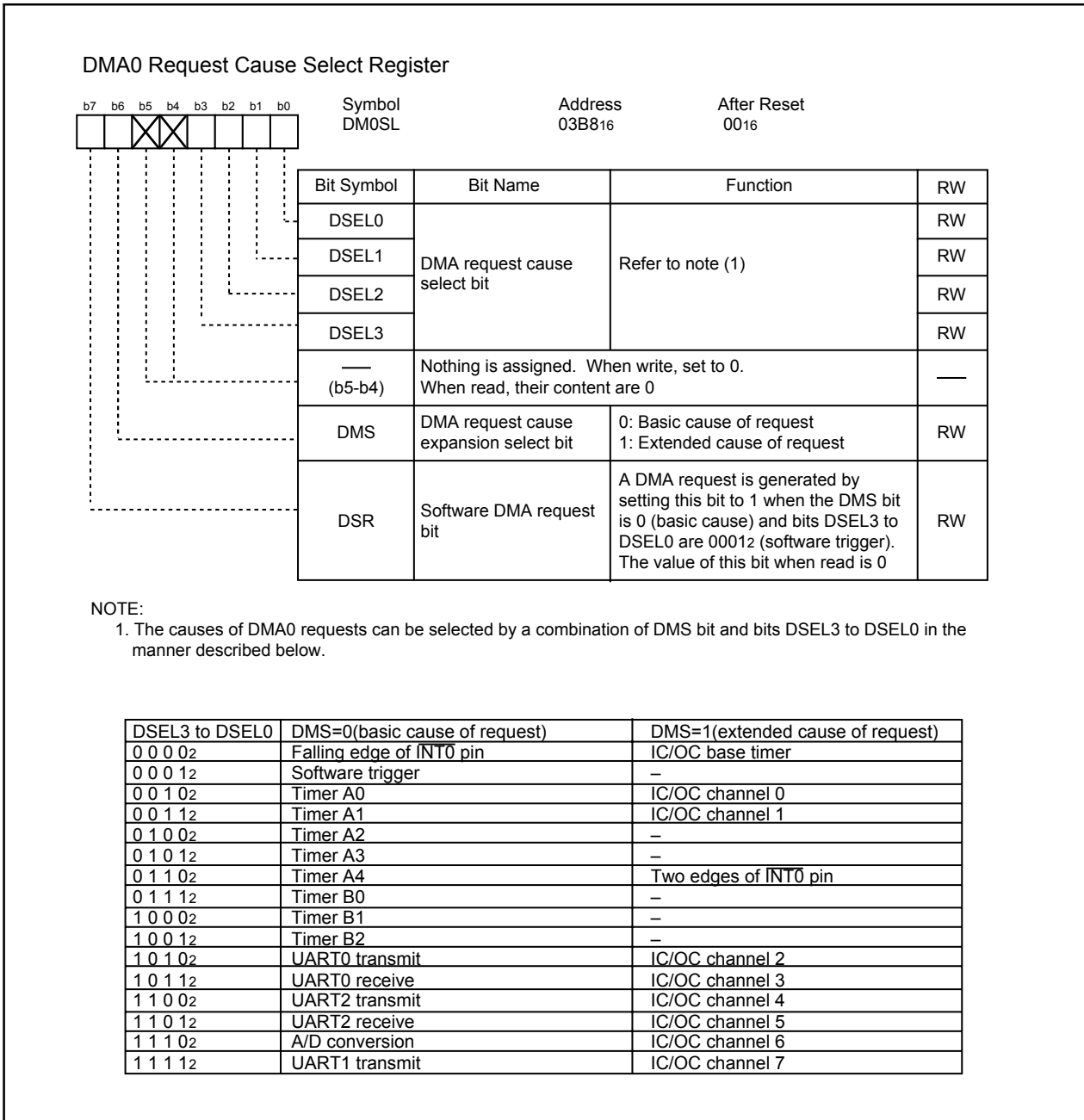


Figure 11.2 DM0SL Register

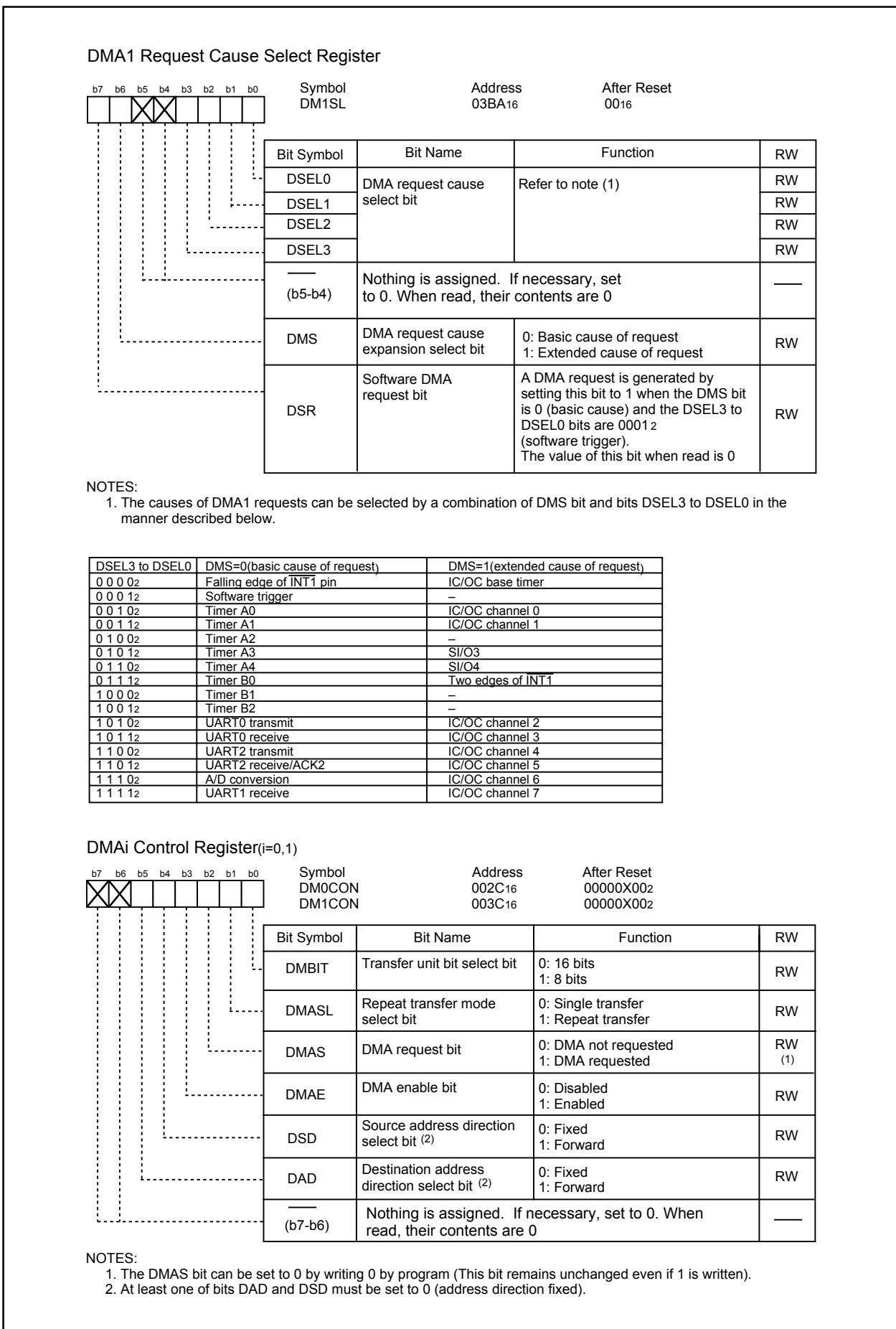


Figure 11.3 DM1SL Register, DM0CON Register, and DM1CON Registers

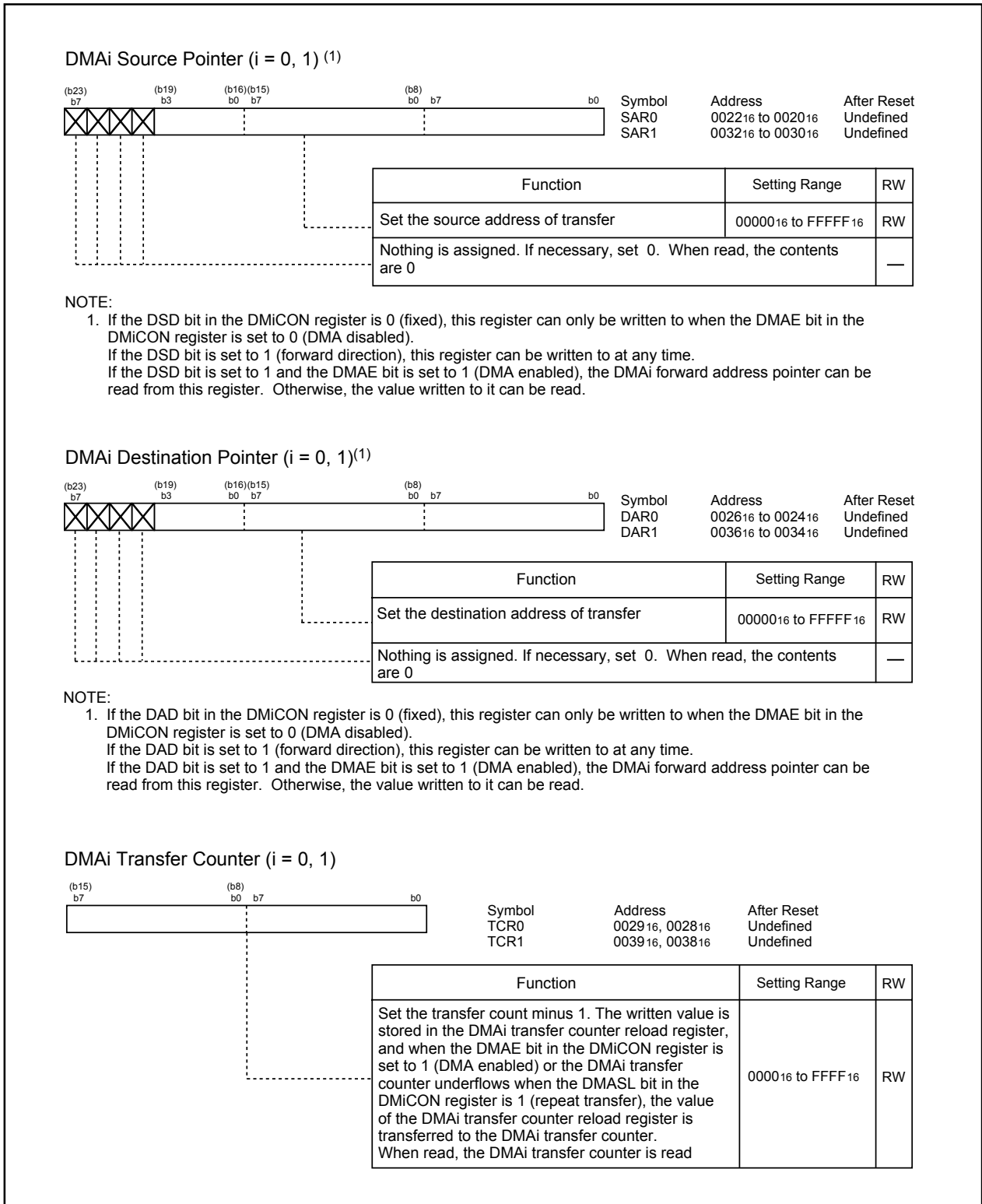


Figure 11.4 SAR0, SAR1, DAR0, DAR1, TCR0, and TCR1 Registers

11.1 Transfer Cycles

The transfer cycle consists of a memory or SFR read (source read) bus cycle and a write (destination write) bus cycle. The number of read and write bus cycles is affected by the source and destination addresses of transfer. Furthermore, the bus cycle itself is extended by a software wait.

11.1.1 Effect of Source and Destination Addresses

If the transfer unit is 16 bits and the source address of transfer begins with an odd address, the source read cycle consists of one more bus cycle than when the source address of transfer begins with an even address.

Similarly, if the transfer unit is 16 bits and the destination address of transfer begins with an odd address, the destination write cycle consists of one more bus cycle than when the destination address of transfer begins with an even address.

11.1.2 Effect of Software Wait

For memory or SFR accesses in which one or more software wait states are inserted, the number of bus cycles required for that access increases by an amount equal to software wait states.

Figure 11.5 shows the example of the cycles for a source read. For convenience, the destination write cycle is shown as one cycle and the source read cycles for the different conditions are shown. In reality, the destination write cycle is subject to the same conditions as the source read cycle, with the transfer cycle changing accordingly. When calculating transfer cycles, take into consideration each condition for the source read and the destination write cycle, respectively. For example, when data is transferred in 16 bit units and when both the source address and destination address are an odd address ((2) in **Figure 11.5**), two source read bus cycles and two destination write bus cycles are required.

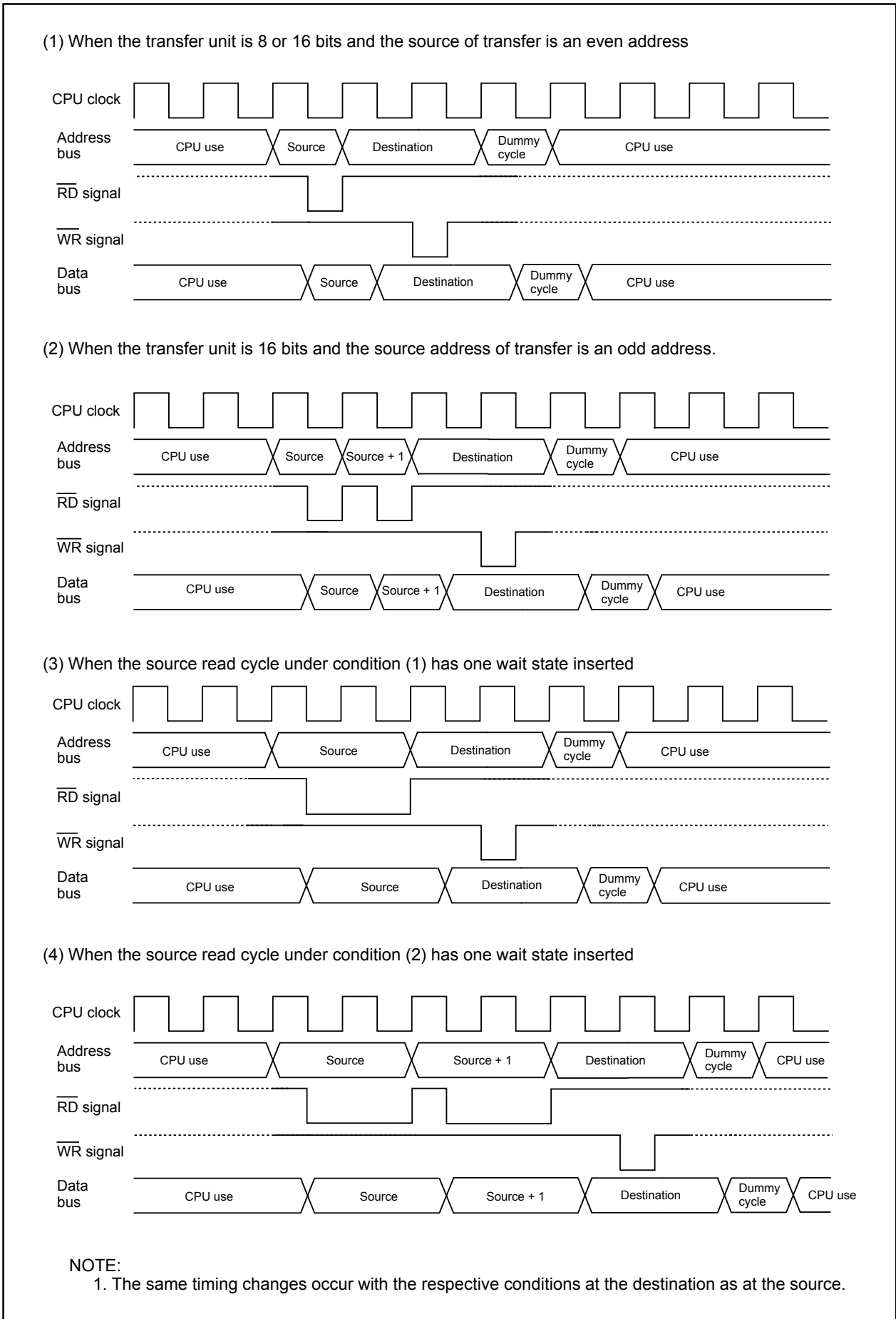


Figure 11.5 Transfer Cycles for Source Read

11.2. DMA Transfer Cycles

Any combination of even or odd transfer read and write addresses is possible. **Table 11.2** shows the number of DMA transfer cycles. **Table 11.3** shows the Coefficient j, k.

The number of DMAC transfer cycles can be calculated as follows:

$$\text{No. of transfer cycles per transfer unit} = \text{No. of read cycles} \times j + \text{No. of write cycles} \times k$$

Table 11.2 DMA Transfer Cycles

| Transfer unit | Access address | No. of read cycles | No. of write cycles |
|--------------------------------|----------------|--------------------|---------------------|
| 8-bit transfers (DMBIT= 1) | Even | 1 | 1 |
| | Odd | 1 | 1 |
| 16-bit transfers (DMBIT= 0) | Even | 1 | 1 |
| | Odd | 2 | 2 |

Table 11.3 Coefficient j, k

| | Internal Area | | | |
|---|-------------------|-----------|---------------|---------------|
| | Internal ROM, RAM | | SFR | |
| | No wait | With wait | 1 wait (1) | 2 wait (1) |
| j | 1 | 2 | 2 | 3 |
| k | 1 | 2 | 2 | 3 |

NOTE:

1. Depends on the set value of PM20 bit in PM2 register

11.3 DMA Enable

When a data transfer starts after setting the DMAE bit in the DMiCON register ($i = 0, 1$) to 1 (enabled), the DMAC operates as follows:

- (a) Reload the forward address pointer with the SAR_i register value when the DSD bit in DMiCON register is 1 (forward) or the DAR_i register value when the DAD bit in the DMiCON register is 1 (forward).
- (b) Reload the DMA_i transfer counter with the DMA_i transfer counter reload register value.

If the DMAE bit is set to 1 again while it remains set, the DMAC performs the above operation. However, if a DMA request may occur simultaneously when the DMAE bit is being written, follow the steps below.

- (1) Write 1 to bits DMAE and DMAS in DMiCON register simultaneously.
- (2) Make sure that the DMA_i is in an initial state as described above (a) and (b) by program.

If the DMA_i is not in an initial state, the above steps should be repeated.

11.4 DMA Request

The DMAC can generate a DMA request as triggered by the cause of request that is selected with the DMS bit and bits DSEL3 to DSEL0 in the DMiSL register ($i = 0, 1$) on either channel. **Table 11.4** shows the timing at which the DMAS bit changes state.

Whenever a DMA request is generated, the DMAS bit is set to 1 (DMA requested) regardless of whether or not the DMAE bit is set. If the DMAE bit was set to 1 (enabled) when this occurred, the DMAS bit is set to 0 (DMA not requested) immediately before a data transfer starts. This bit cannot be set to 1 by program (it can only be set to 0).

The DMAS bit may be set to 1 when the DMS or the DSEL3 to DSEL0 bits change state. Therefore, always be sure to set the DMAS bit to 0 after changing the DMS or the DSEL3 to DSEL0 bits.

Because if the DMAE bit is set to 1, a data transfer starts immediately after a DMA request is generated, the DMAS bit in almost all cases is 0 when read by program. Read the DMAE bit to determine whether the DMAC is enabled.

Table 11.4 Timing at Which the DMAS Bit Changes State

| DMA Factor | DMAS Bit in the DMiCON Register | |
|---------------------|---|---|
| | Timing at which the bit is set to 1 | Timing at which the bit is set to 0 |
| Software trigger | When the DSR bit in the DMiSL register is set to 1 | <ul style="list-style-type: none"> • Immediately before a data transfer starts • When set by writing 0 by program |
| Peripheral function | When the interrupt control register for the peripheral function that is selected by bits DSEL3 to DSEL0 and the DMS bit in the DMiSL register has its IR bit set to 1 | |

11.5 Channel Priority and DMA Transfer Timing

If both DMA0 and DMA1 are enabled and DMA transfer request signals from DMA0 and DMA1 are detected active in the same sampling period (one period from a falling edge to the next falling edge of CPU clock), the DMAS bit on each channel is set to 1 (DMA requested) at the same time. In this case, the DMA requests are arbitrated according to the channel priority, DMA0 > DMA1. The following describes DMAC operation when DMA0 and DMA1 requests are detected active in the same sampling period. **Figure 11.6** shows an example of DMA transfer effected by external factors.

DMA0 request having priority is received first to start a transfer when a DMA0 request and DMA1 request are generated simultaneously. After one DMA0 transfer is completed, a bus arbitration is returned to the CPU. When the CPU has completed one bus access, a DMA1 transfer starts. After one DMA1 transfer is completed, the bus arbitration is again returned to the CPU.

In addition, DMA requests cannot be counted up since each channel has one DMAS bit. Therefore, when DMA requests, as DMA1 in **Figure 11.6** occurs more than one time, the DAMS bit is set to 0 as soon as getting the bus arbitration. The bus arbitration is returned to the CPU when one transfer is completed.

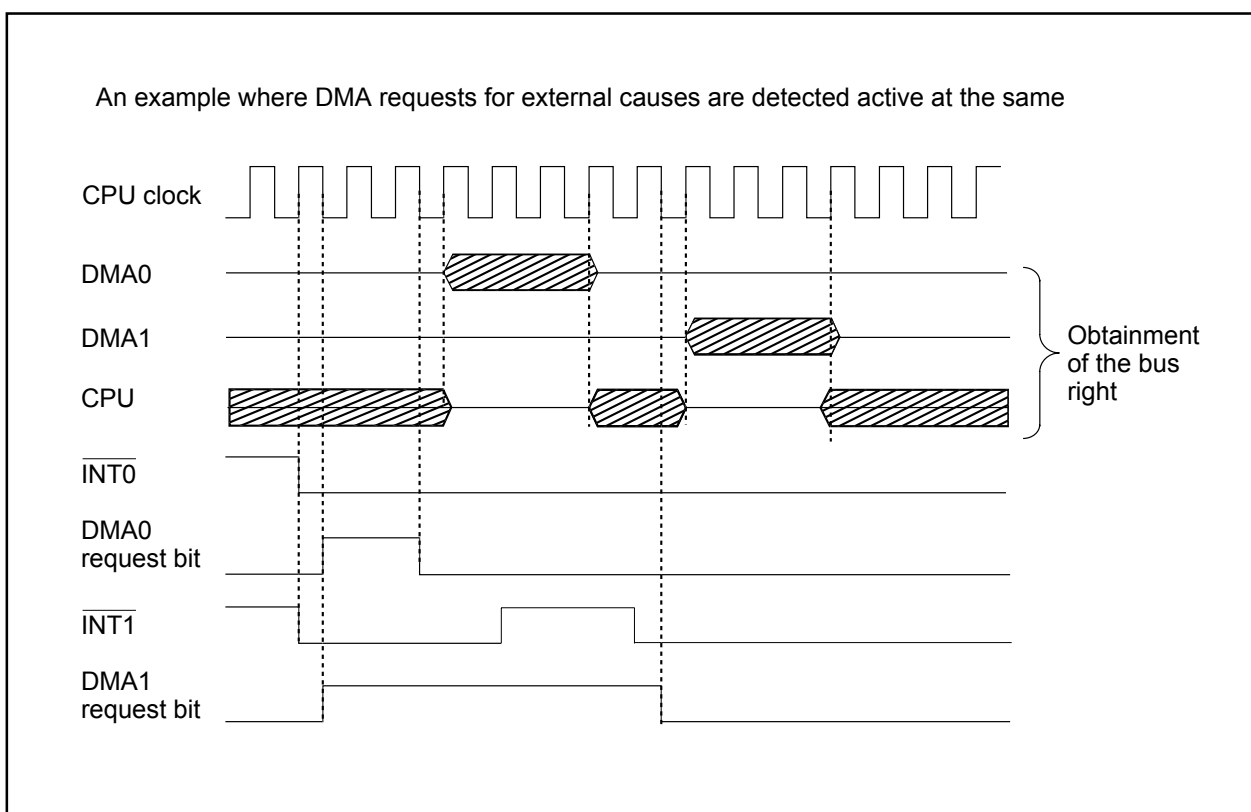


Figure 11.6 DMA Transfer by External Factors

12. Timers

Eight 16-bit timers, each capable of operating independently of the others, can be classified by function as either timer A (five) and timer B (three). The count source for each timer acts as a clock, to control such timer operations as counting, reloading, etc. **Figures 12.1** and **12.2** show block diagrams of timer A and timer B configuration, respectively.

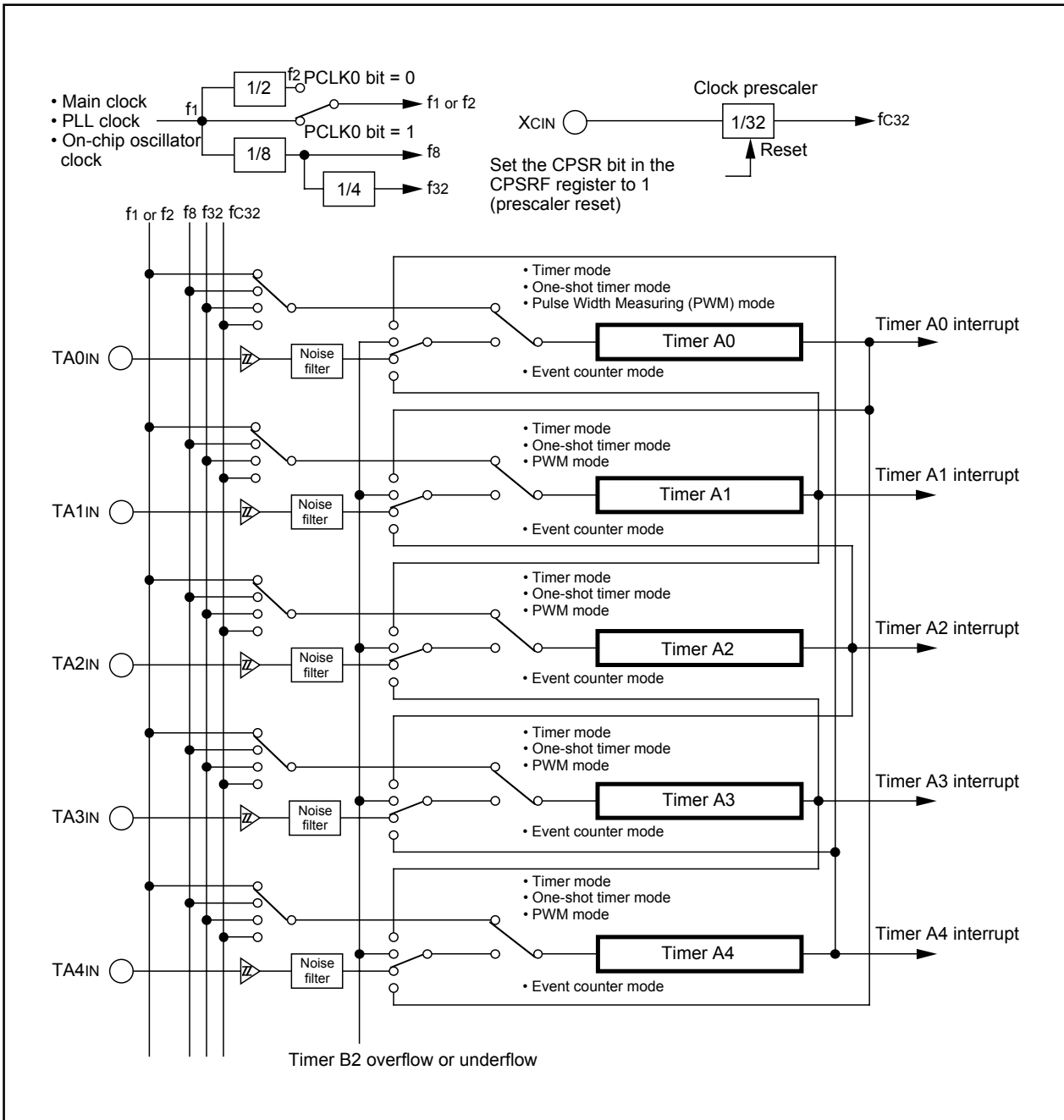


Figure 12.1 Timer A Configuration

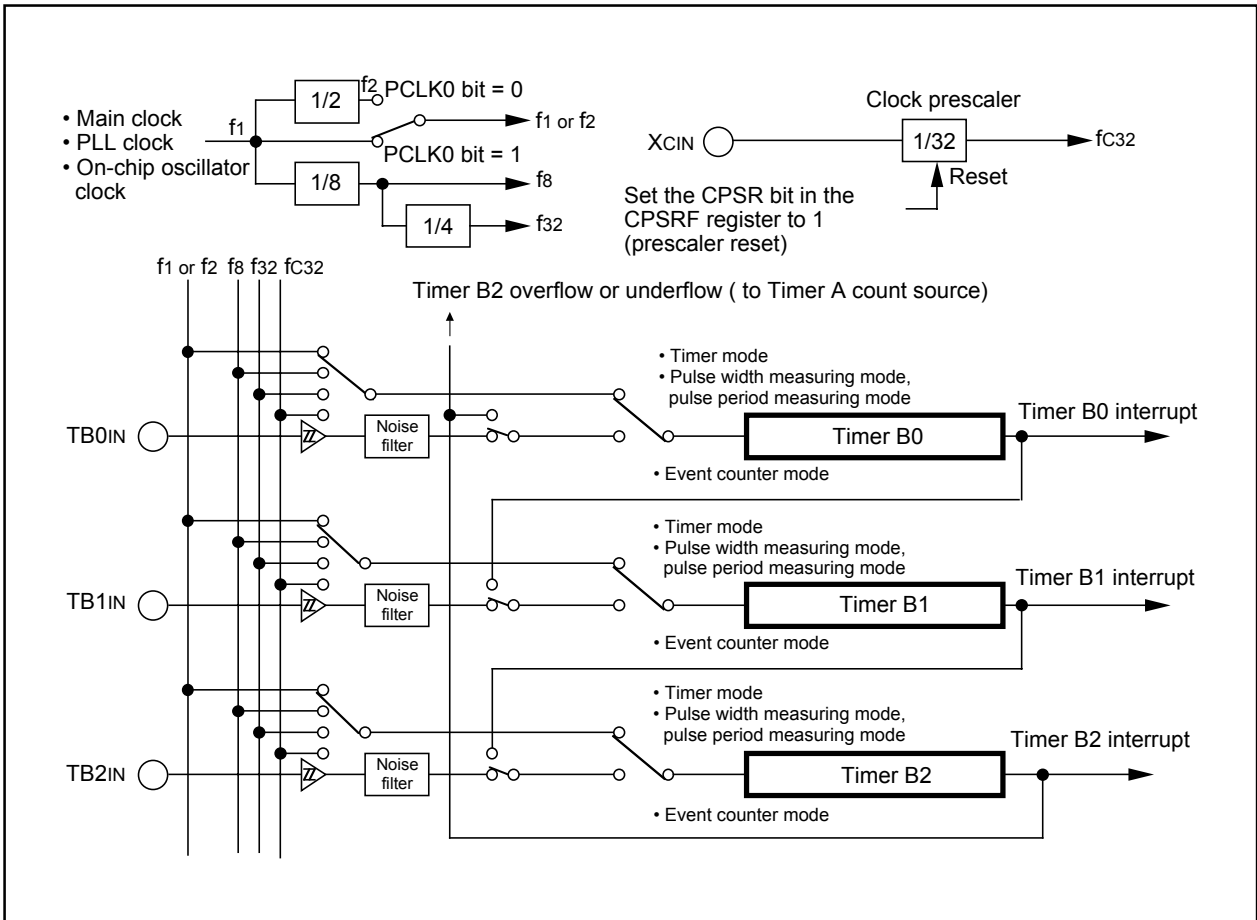


Figure 12.2. Timer B Configuration

12.1 Timer A

Figure 12.3 shows a block diagram of the timer A. Figures 12.4 to 12.6 show registers related to the timer A. The timer A supports the following four modes. Except in event counter mode, timers A0 to A4 all have the same function. Use bits TMOD1 to TMOD0 in the TAI_{MR} register (i = 0 to 4) to select the desired mode.

- Timer mode: The timer counts an internal count source.
- Event counter mode: The timer counts pulses from an external device or overflows and underflows of other timers.
- One-shot timer mode: The timer outputs a pulse only once before it reaches the minimum count 0000₁₆.
- Pulse width modulation (PWM) mode: The timer outputs pulses in a given width successively.

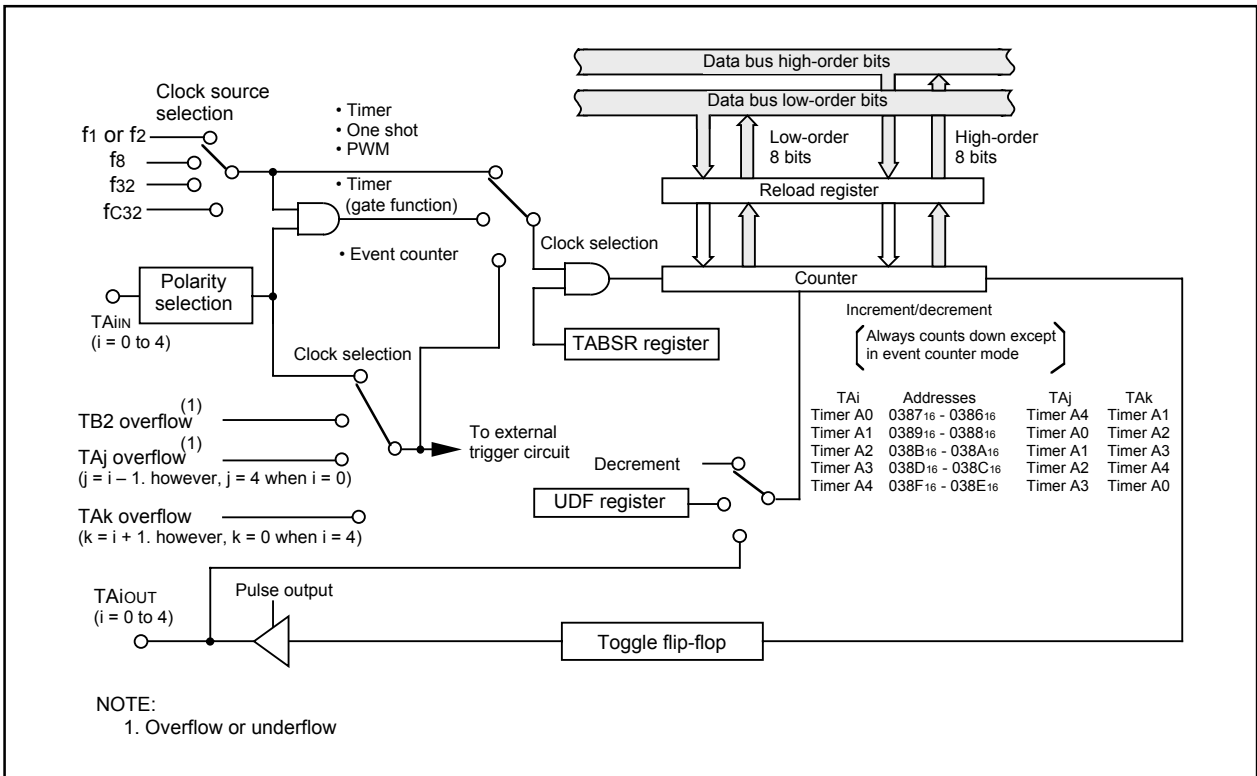


Figure 12.3 Timer A Block Diagram

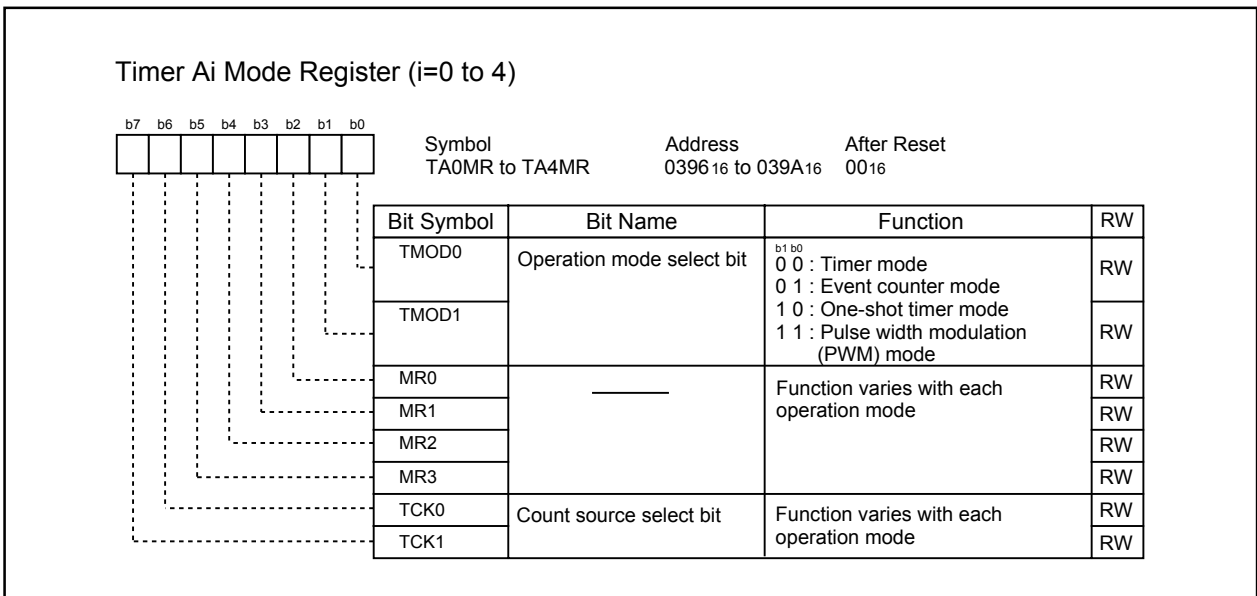


Figure 12.4 TA0MR to TA4MR Registers

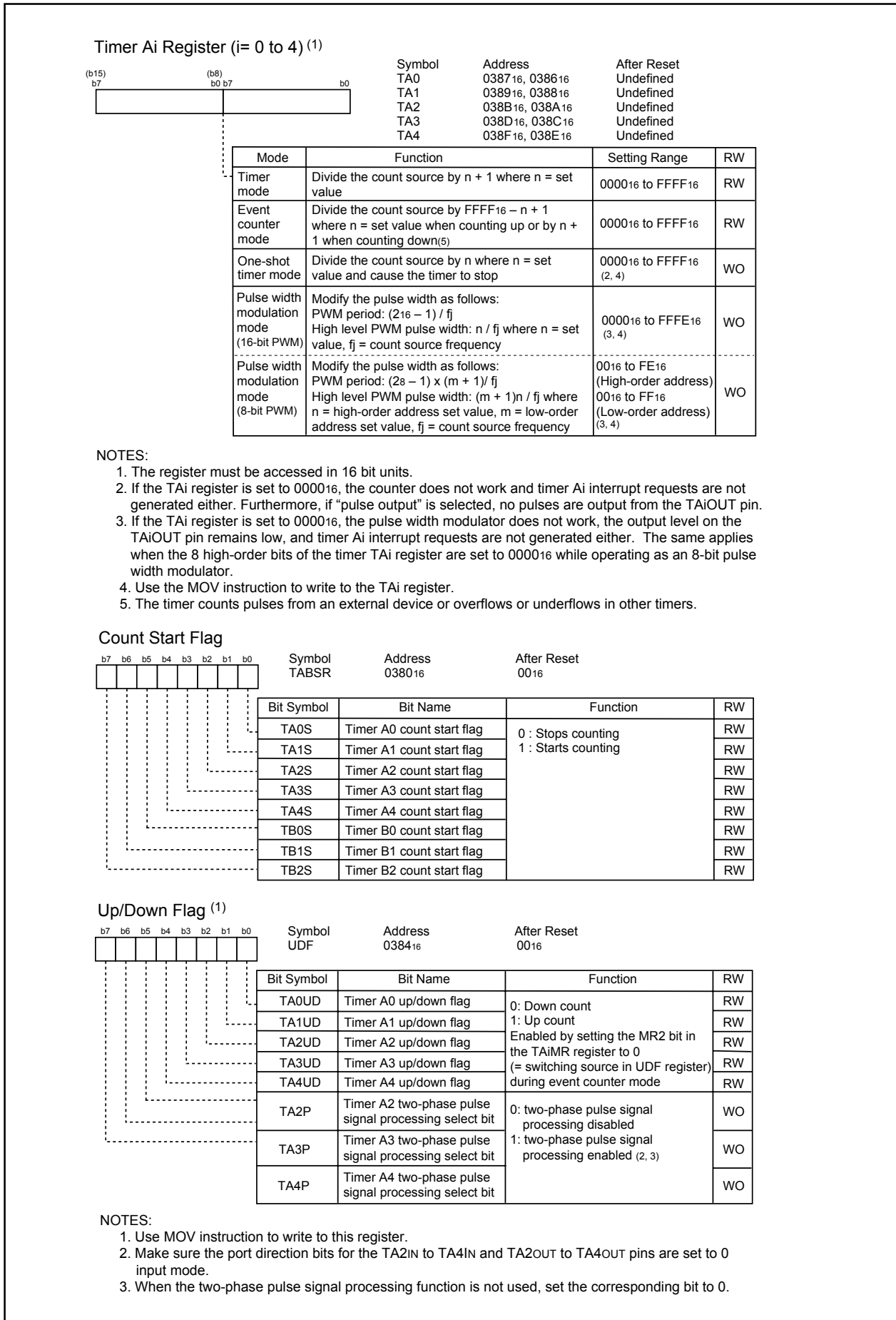


Figure 12.5 TA0 to TA4 Registers, TABSR Register, and UDF Register

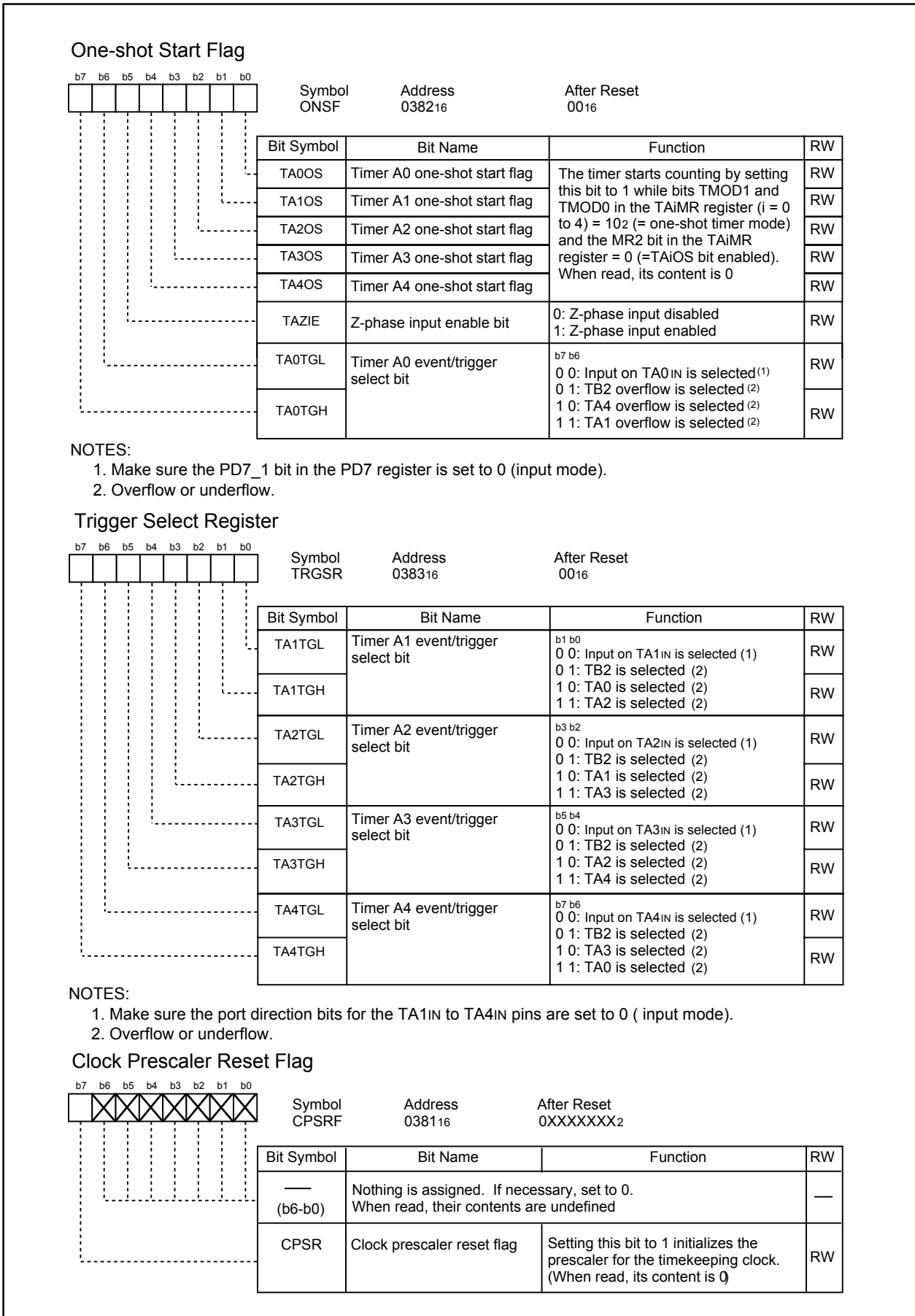


Figure 12.6 ONSF Register, TRGSR Register, and CPSRF Register

12.1.1 Timer Mode

In timer mode, the timer counts a count source generated internally (see **Table 12.1**). **Figure 12.7** shows TAI_iMR register in timer mode.

Table 12.1 Specifications in Timer Mode

| Item | Specification |
|-------------------------------------|--|
| Count source | f1, f2, f8, f32, fc32 |
| Count operation | <ul style="list-style-type: none"> Decrement When the timer underflows, it reloads the reload register contents and continues counting |
| Divide ratio | $1/(n+1)$ n: set value of TAI register (i= 0 to 4) 0000 ₁₆ to FFFF ₁₆ |
| Count start condition | Set TAI _S bit in the TABSR register to 1 (start counting) |
| Count stop condition | Set TAI _S bit to 0 (stop counting) |
| Interrupt request generation timing | Timer underflow |
| TAI _i N pin function | I/O port or gate input |
| TAI _i OUT pin function | I/O port or pulse output |
| Read from timer | Count value can be read by reading TAI register |
| Write to timer | <ul style="list-style-type: none"> When not counting and until the 1st count source is input after counting start Value written to TAI register is written to both reload register and counter When counting (after 1st count source input) Value written to TAI register is written to only reload register (Transferred to counter when reloaded next) |
| Select function | <ul style="list-style-type: none"> Gate function Counting can be started and stopped by an input signal to TAI_iN pin Pulse output function Whenever the timer underflows, the output polarity of TAI_iOUT pin is inverted. When not counting, the pin outputs a low. |

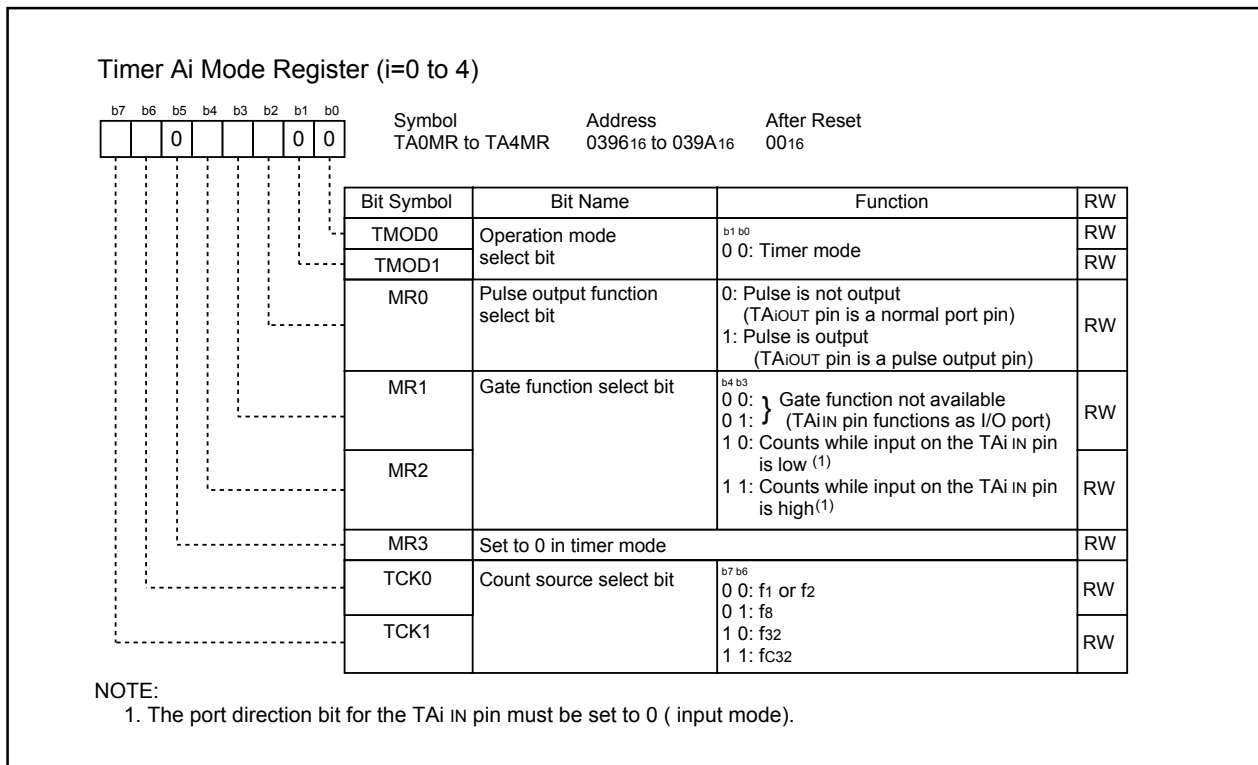


Figure 12.7 Timer Ai Mode Register in Timer Mode

12.1.2 Event Counter Mode

In event counter mode, the timer counts pulses from an external device or overflows and underflows of other timers. Timers A2, A3, and A4 can count two-phase external signals. **Table 12.2** lists specifications in event counter mode (when not processing two-phase pulse signal). **Table 12.3** lists specifications in event counter mode (when processing two-phase pulse signal with the timers A2, A3 and A4). **Figure 12.8** shows TAI_{MR} register in event counter mode (when not processing two-phase pulse signal). **Figure 12.9** shows TA2_{MR} to TA4_{MR} registers in event counter mode (when processing two-phase pulse signal with the timers A2, A3 and A4).

Table 12.2 Specifications in Event Counter Mode (when not processing two-phase pulse signal)

| Item | Specification |
|-------------------------------------|--|
| Count source | <ul style="list-style-type: none"> External signals input to TAI_{IN} pin (i=0 to 4) (effective edge can be selected in program) Timer B2 overflows or underflows, timer A_j (j=i-1, except j=4 if i=0) overflows or underflows, timer A_k (k=i+1, except k=0 if i=4) overflows or underflows |
| Count operation | <ul style="list-style-type: none"> Increment or decrement can be selected by external signal or program When the timer overflows or underflows, it reloads the reload register contents and continues counting. When operating in free-running mode, the timer continues counting without reloading. |
| Divided ratio | $1 / (FFFF_{16} - n + 1)$ for increment $1 / (n + 1)$ for down-count n : set value of TAI register 0000 ₁₆ to FFFF ₁₆ |
| Count start condition | Set TAI _S bit in the TABSR register to 1 (start counting) |
| Count stop condition | Set TAI _S bit to 0 (stop counting) |
| Interrupt request generation timing | Timer overflow or underflow |
| TAI _{IN} pin function | I/O port or count source input |
| TAI _{OUT} pin function | I/O port, pulse output, or up/down-count select input |
| Read from timer | Count value can be read by reading TAI register |
| Write to timer | <ul style="list-style-type: none"> When not counting and until the 1st count source is input after counting start Value written to TAI register is written to both reload register and counter When counting (after 1st count source input) Value written to TAI register is written to only reload register (Transferred to counter when reloaded next) |
| Select function | <ul style="list-style-type: none"> Free-run count function Even when the timer overflows or underflows, the reload register content is not reloaded to it Pulse output function Whenever the timer underflows or underflows, the output polarity of TAI_{OUT} pin is inverted . When not counting, the pin outputs a low. |

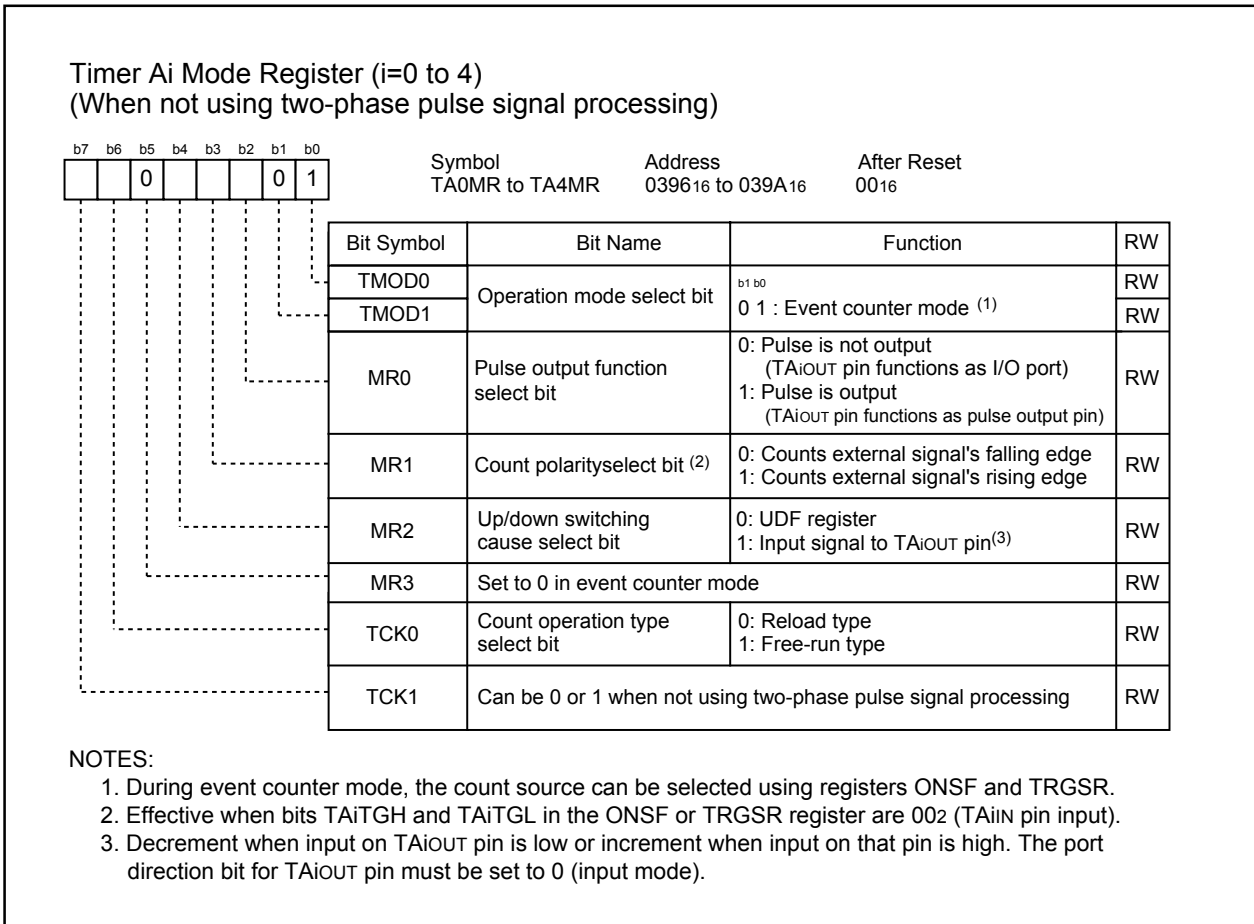
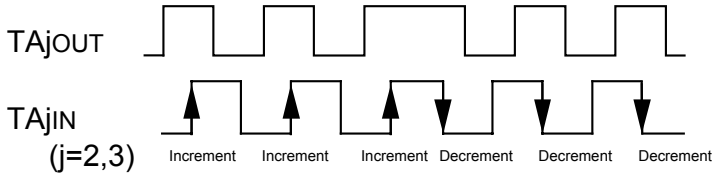
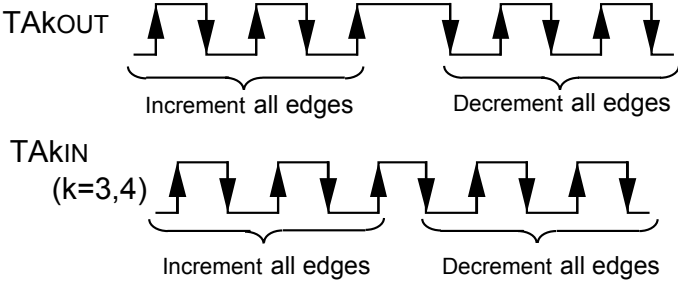


Figure 12.8 TA_iMR Register in Event Counter Mode (when not using two-phase pulse signal processing)

Table 12.3 Specifications in Event Counter Mode
(when processing two-phase pulse signal with timers A2, A3 and A4)

| Item | Specification |
|-------------------------------------|---|
| Count source | <ul style="list-style-type: none"> Two-phase pulse signals input to TAIIN or TAIOUT pins (i = 2 to 4) |
| Count operation | <ul style="list-style-type: none"> Increment or down-count can be selected by two-phase pulse signal When the timer overflows or underflows, it reloads the reload register contents and continues counting. When operating in free-running mode, the timer continues counting without reloading. |
| Divide ratio | $1 / (FFFF_{16} - n + 1)$ for increment $1 / (n + 1)$ for down-count n : set value of TAI register 0000_{16} to $FFFF_{16}$ |
| Count start condition | Set TAI _S bit in the TABSR register to 1 (start counting) |
| Count stop condition | Set TAI _S bit to 0 (stop counting) |
| Interrupt request generation timing | Timer overflow or underflow |
| TAIIN pin function | Two-phase pulse input |
| TAIOUT pin function | Two-phase pulse input |
| Read from timer | Count value can be read by reading timer A2, A3 or A4 register |
| Write to timer | <ul style="list-style-type: none"> When not counting and until the 1st count source is input after counting start Value written to TAI register is written to both reload register and counter When counting (after 1st count source input) Value written to TAI register is written to reload register (Transferred to counter when reloaded next) |
| Select function (Note) | <ul style="list-style-type: none"> Normal processing operation (timer A2 and timer A3) The timer counts up rising edges or counts down falling edges on TAJIN pin when input signals on TAJOUT pin is "H".  Multiply-by-4 processing operation (timer A3 and timer A4) If the phase relationship is such that TAKIN(k=3, 4) pin goes "H" when the input signal on TAKOUT pin is "H", the timer counts up rising and falling edges on TAKOUT and TAKIN pins. If the phase relationship is such that TAKIN pin goes "L" when the input signal on TAKOUT pin is "H", the timer counts down rising and falling edges on TAKOUT and TAKIN pins.  Counter initialization by Z-phase input (timer A3) The timer count value is initialized to 0 by Z-phase input. |

NOTE:

- Only timer A3 is selectable. Timer A2 is fixed to normal processing operation, and timer A4 is fixed to multiply-by-4 processing operation.

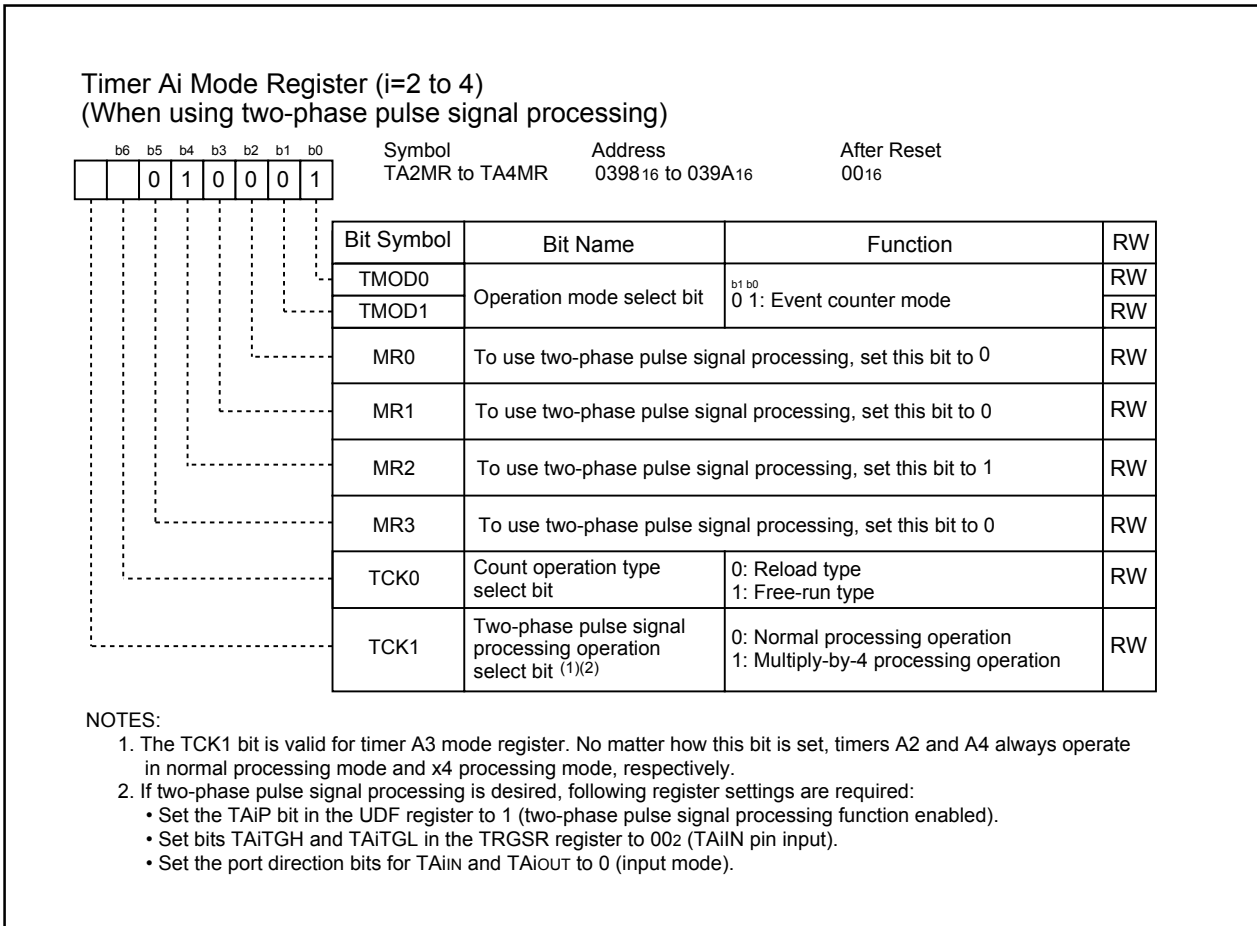


Figure 12.9 TA2MR to TA4MR Registers in Event Counter Mode (when using two-phase pulse signal processing with timer A2, A3 or A4)

12.1.2.1 Counter Initialization by Two-Phase Pulse Signal Processing

This function initializes the timer count value to 0 by Z-phase (counter initialization) input during two-phase pulse signal processing.

This function can only be used in timer A3 event counter mode during two-phase pulse signal processing, free-running type, x4 processing, with Z-phase entered from the $\overline{\text{INT2}}$ pin.

Counter initialization by Z-phase input is enabled by writing 0000₁₆ to the TA3 register and setting the TAZIE bit in ONSF register to 1 (Z-phase input enabled).

Counter initialization is accomplished by detecting Z-phase input edge. The active edge can be chosen to be the rising or falling edge by using the POL bit in the INT2IC register. The Z-phase pulse width applied to the $\overline{\text{INT2}}$ pin must be equal to or greater than one clock cycle of the timer A3 count source.

The counter is initialized at the next count timing after recognizing Z-phase input. **Figure 12.10** shows the relationship between the two-phase pulse (A phase and B phase) and the Z phase.

If timer A3 overflow or underflow coincides with the counter initialization by Z-phase input, a timer A3 interrupt request is generated twice in succession. Do not use the timer A3 interrupt when using this function.

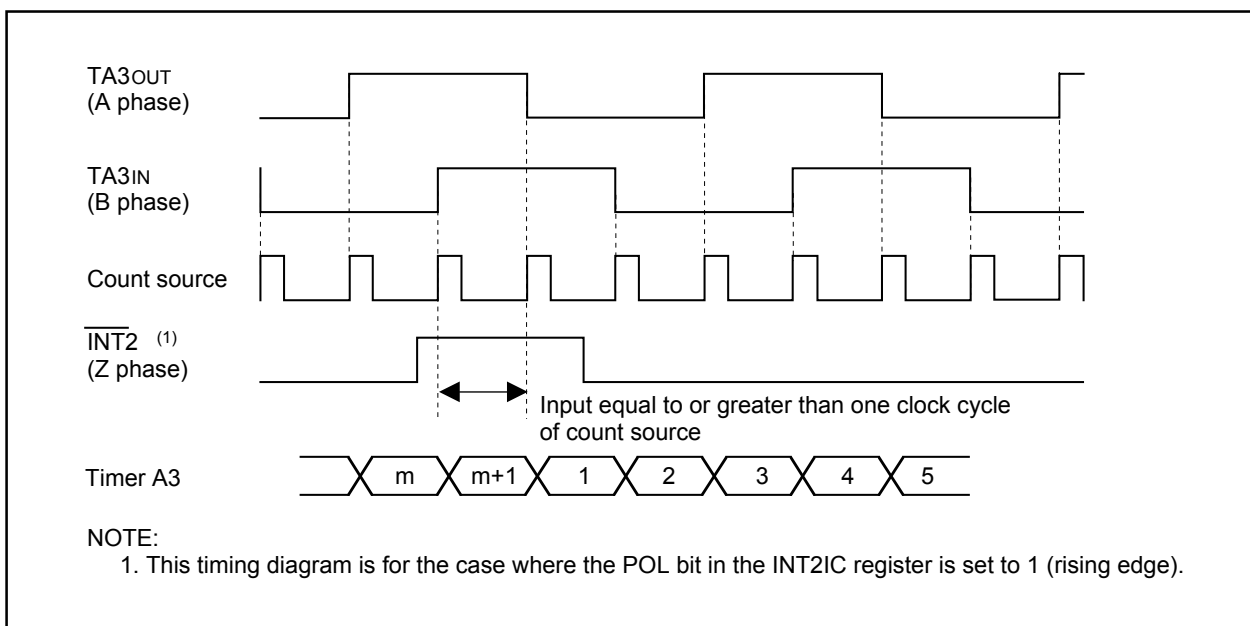


Figure 12.10 Two-phase Pulse (A phase and B phase) and the Z Phase

12.1.3 One-shot Timer Mode

In one-shot timer mode, the timer is activated only once by one trigger. (See **Table 12.4**) When the trigger occurs, the timer starts up and continues operating for a given period. **Figure 12.11** shows the TAI_{MR} register in one-shot timer mode.

Table 12.4 Specifications in One-shot Timer Mode

| Item | Specification |
|-------------------------------------|--|
| Count source | f1, f2, f8, f32, fC32 |
| Count operation | <ul style="list-style-type: none"> Decrement When the counter reaches 0000₁₆, it stops counting after reloading a new value If a trigger occurs when counting, the timer reloads a new count and restarts counting |
| Divide ratio | $1/n$ n : set value of TAI register 0000 ₁₆ to FFFF ₁₆ However, the counter does not work if the divide-by-n value is set to 0000 ₁₆ . |
| Count start condition | TAI _S bit in the TABSR register is set to 1 (start counting) and one of the following triggers occurs. <ul style="list-style-type: none"> External trigger input from the TAI_{IN} pin Timer B2 overflow or underflow, timer A_j ($j=i-1$, except $j=4$ if $i=0$) overflow or underflow, timer A_k ($k=i+1$, except $k=0$ if $i=4$) overflow or underflow The TAI_{OS} bit in the ONSF register is set to 1 (timer starts) |
| Count stop condition | <ul style="list-style-type: none"> When the counter is reloaded after reaching 0000₁₆ TAI_S bit is set to 0 (stop counting) |
| Interrupt request generation timing | When the counter reaches 0000 ₁₆ |
| TAI _{IN} pin function | I/O port or trigger input |
| TAI _{OUT} pin function | I/O port or pulse output |
| Read from timer | An undefined value is read by reading TAI register |
| Write to timer | <ul style="list-style-type: none"> When not counting and until the 1st count source is input after counting start Value written to TAI register is written to both reload register and counter When counting (after 1st count source input) Value written to TAI register is written to only reload register (Transferred to counter when reloaded next) |
| Select function | <ul style="list-style-type: none"> Pulse output function The timer outputs a low when not counting and a high when counting. |

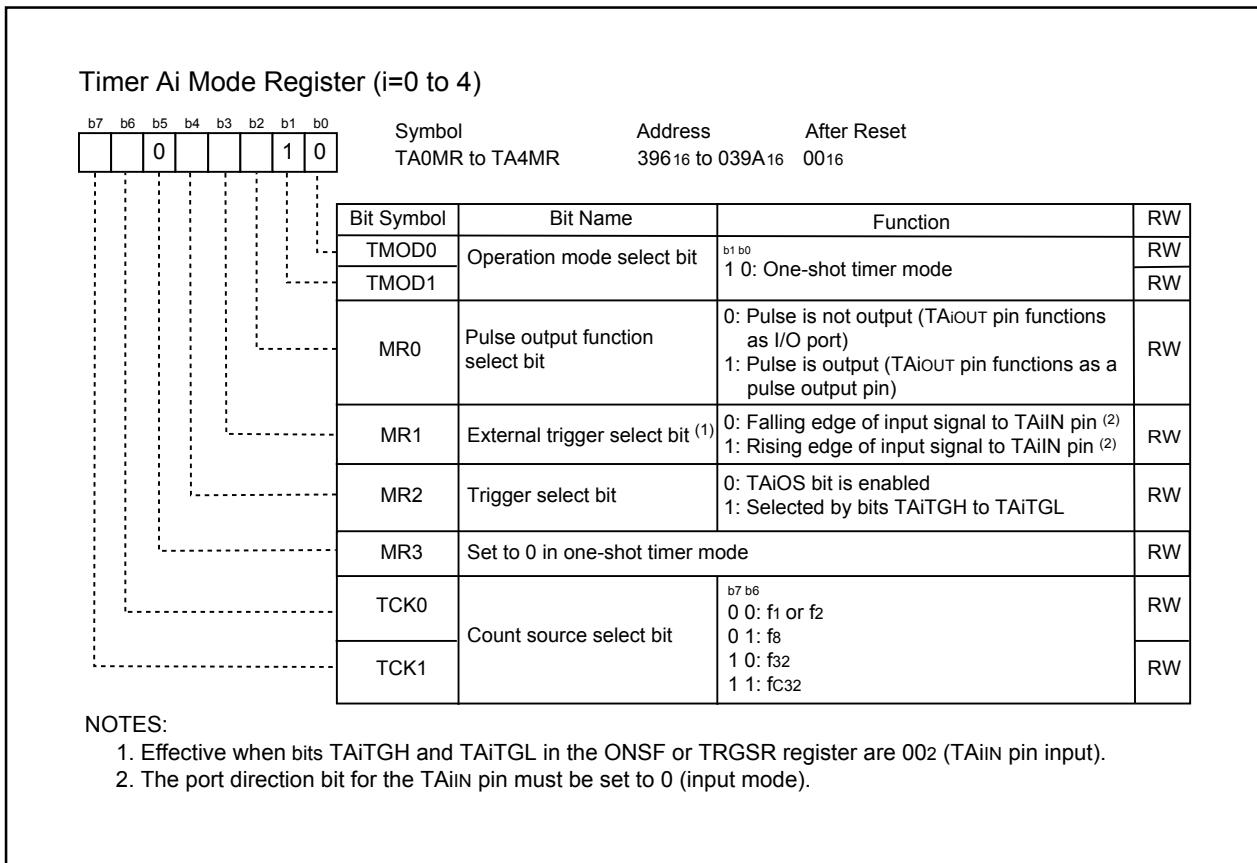


Figure 12.11 TAI_iMR Register in One-shot Timer Mode

12.1.4 Pulse Width Modulation (PWM) Mode

In PWM mode, the timer outputs pulses of a given width in succession (see **Table 12.5**). The counter functions as either 16-bit pulse width modulator or 8-bit pulse width modulator. **Figure 12.12** shows TAI_{MR} register in pulse width modulation mode. **Figures 12.13** and **12.14** show examples of how a 16-bit pulse width modulator operates and how an 8-bit pulse width modulator operates.

Table 12.5 Specifications in Pulse Width Modulation Mode

| Item | Specification |
|-------------------------------------|---|
| Count source | f1, f2, f8, f32, fC32 |
| Count operation | <ul style="list-style-type: none"> • Decrement (operating as an 8-bit or a 16-bit pulse width modulator) • The timer reloads a new value at a rising edge of PWM pulse and continues counting • The timer is not affected by a trigger that occurs during counting |
| 16-bit PWM | <ul style="list-style-type: none"> • High level width n / f_j n: set value of TAI register ($i=0$ to 4) • Cycle time $(2^{16}-1) / f_j$ fixed f_j: count source frequency (f1, f2, f8, f32, fC32) |
| 8-bit PWM | <ul style="list-style-type: none"> • High level width $n \times (m+1) / f_j$ n: set value of TAI register high-order address • Cycle time $(2^8-1) \times (m+1) / f_j$ m: set value of TAI register low-order address |
| Count start condition | <ul style="list-style-type: none"> • TAI_S bit in the TABSR register is set to 1 (= start counting) • The TAI_S bit = 1 and external trigger input from the TAI_{IN} pin • The TAI_S bit = 1 and one of the following external triggers occurs • Timer B2 overflow or underflow, timer A_j ($j=i-1$, except $j=4$ if $i=0$) overflow or underflow, timer A_k ($k=i+1$, except $k=0$ if $i=4$) overflow or underflow |
| Count stop condition | TAI _S bit is set to 0 (stop counting) |
| Interrupt request generation timing | PWM pulse goes "L" |
| TAI _{IN} pin function | I/O port or trigger input |
| TAI _{OUT} pin function | Pulse output |
| Read from timer | An undefined value is read by reading TAI register |
| Write to timer | <ul style="list-style-type: none"> • When not counting and until the 1st count source is input after counting start Value written to TAI register is written to both reload register and counter • When counting (after 1st count source input) Value written to TAI register is written to only reload register (Transferred to counter when reloaded next) |

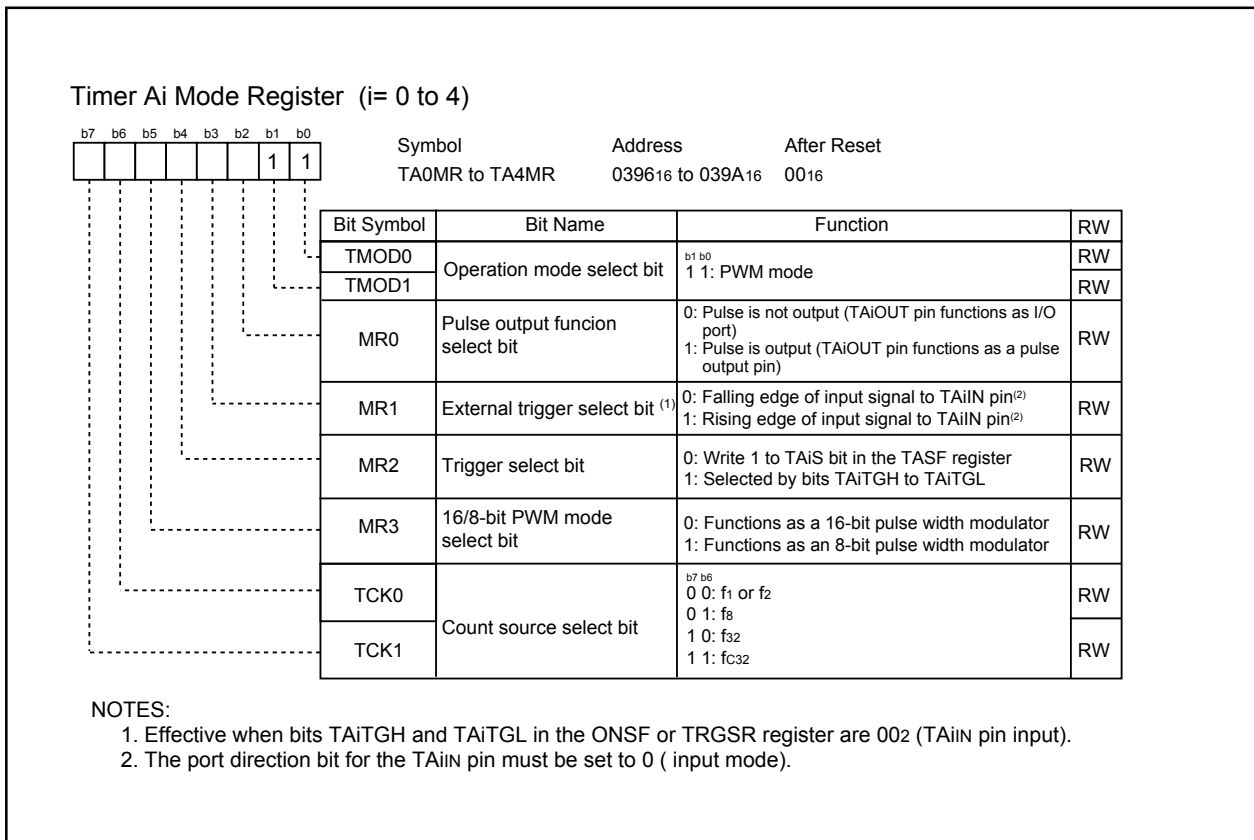


Figure 12.12 TAiMR Register in Pulse Width Modulation Mode

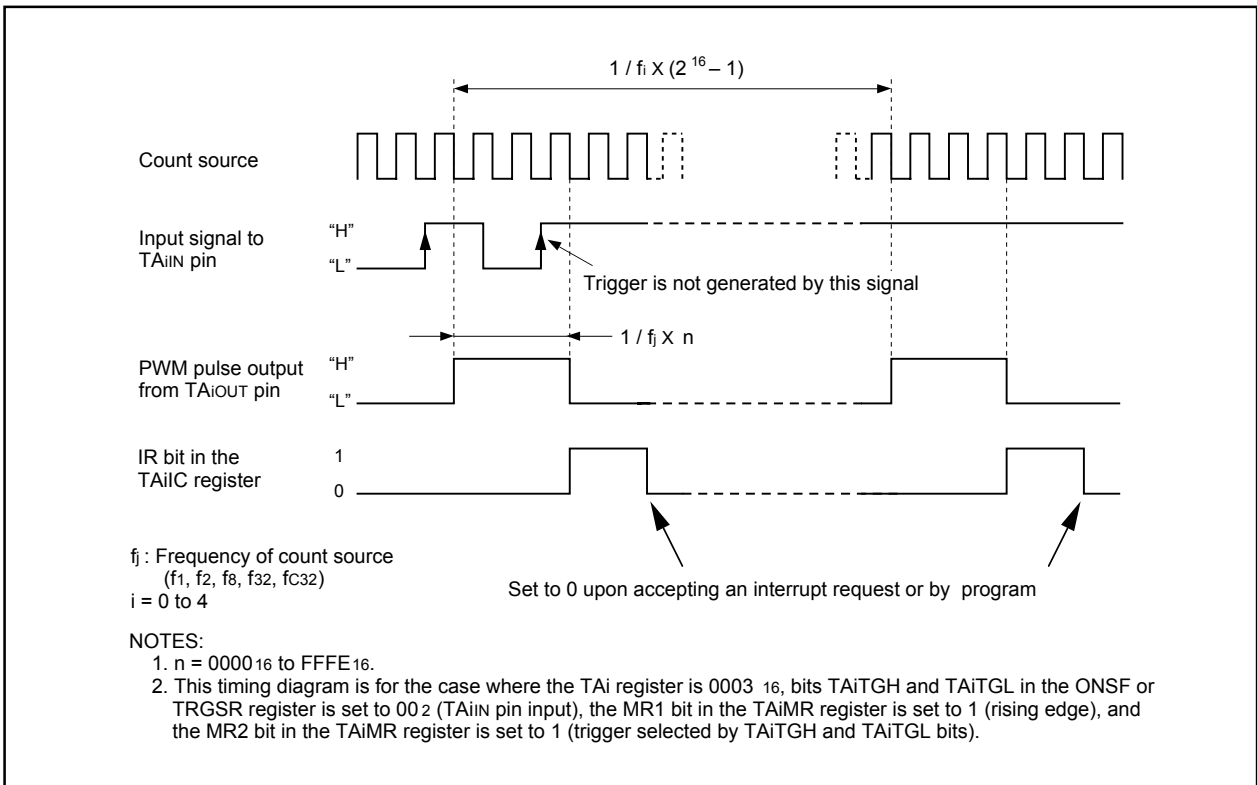


Figure 12.13 Example of 16-bit Pulse Width Modulator Operation

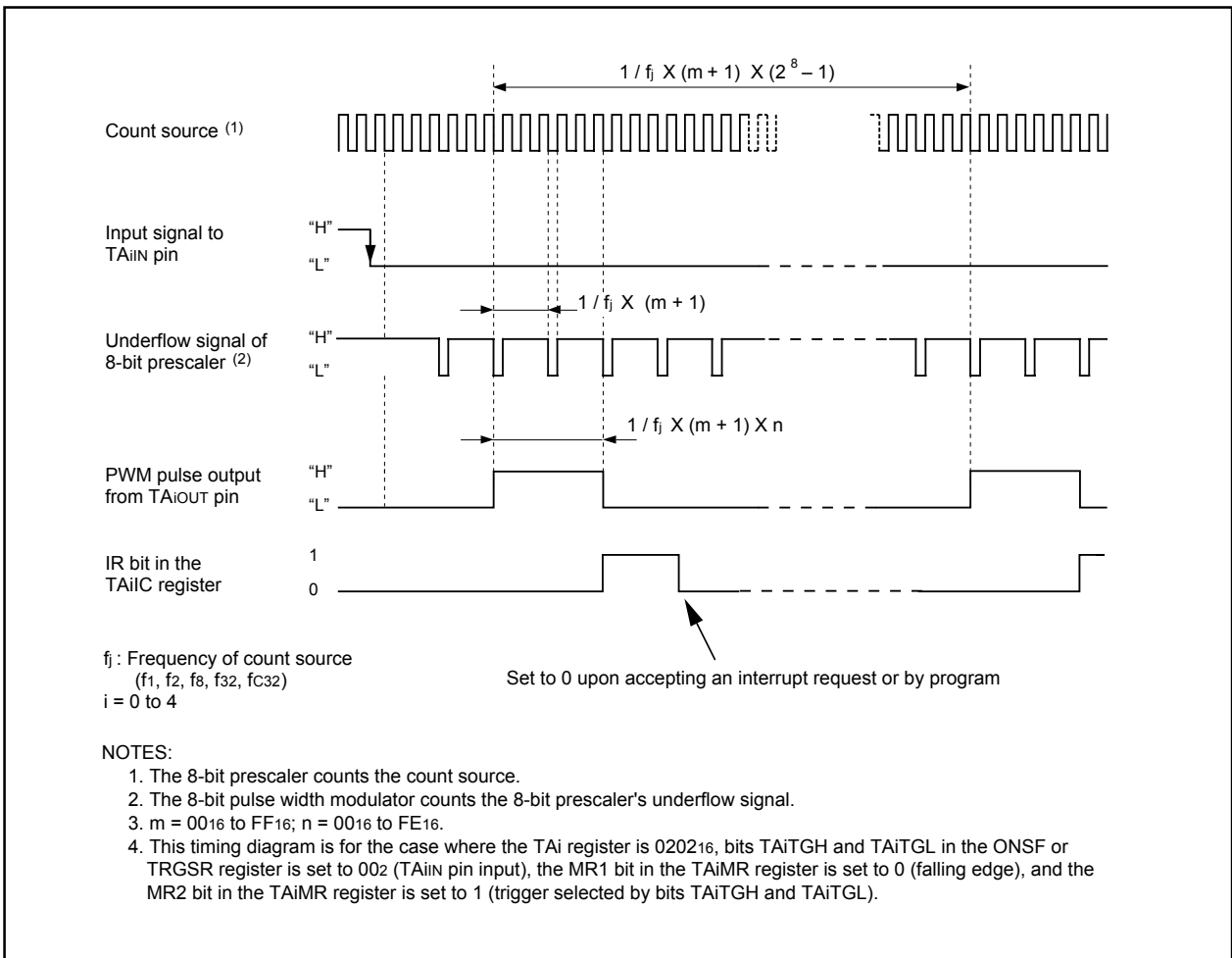


Figure 12.14 Example of 8-bit Pulse Width Modulator Operation

12.2 Timer B

Figure 12.15 shows a block diagram of the timer B. Figures 12.16 and 12.17 show registers related to the timer B.

Timer B supports the following four modes. Use bits TMOD1 and TMOD0 in the TBiMR register (i = 0 to 2) to select the desired mode.

- Timer mode: The timer counts the internal count source.
- Event counter mode: The timer counts the external pulses or overflows and underflows of other timers.
- Pulse period/pulse width measurement mode: The timer measures the pulse period or pulse width of external signal.
- A/D trigger mode: The timer starts counting by one trigger until the count value becomes 0000₁₆. This mode is used together with simultaneous sample sweep mode or delayed trigger mode 0 of A/D converter to start A/D conversion.

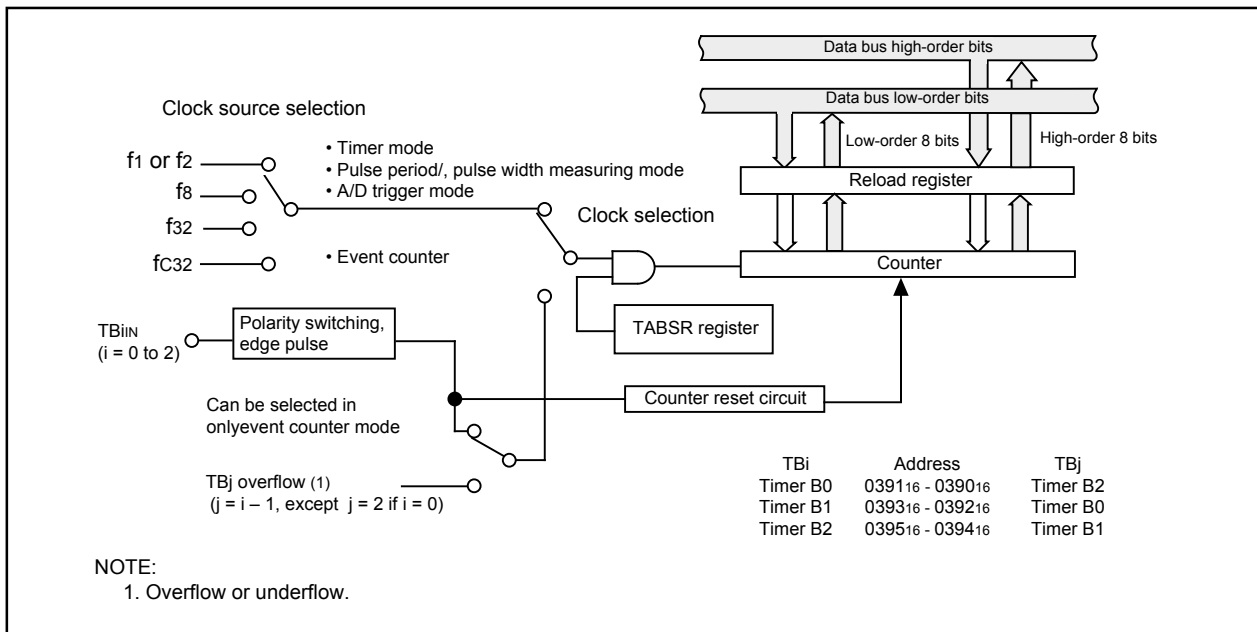


Figure 12.15 Timer B Block Diagram

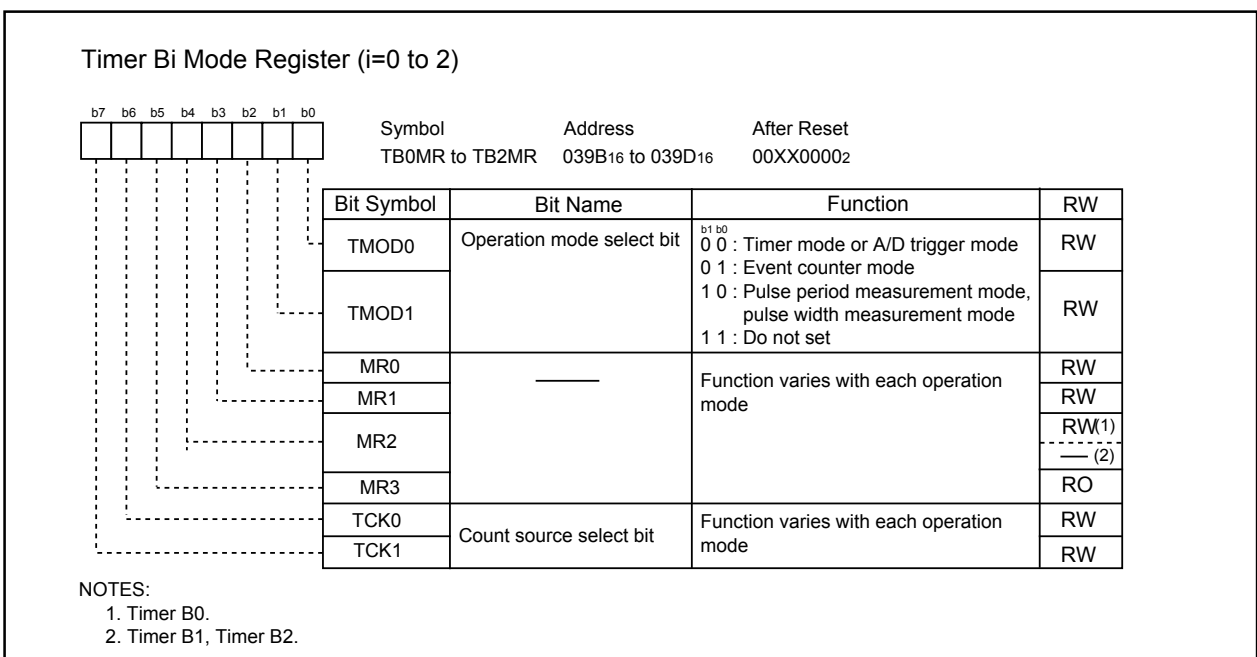


Figure 12.16 TB0MR to TB2MR Registers

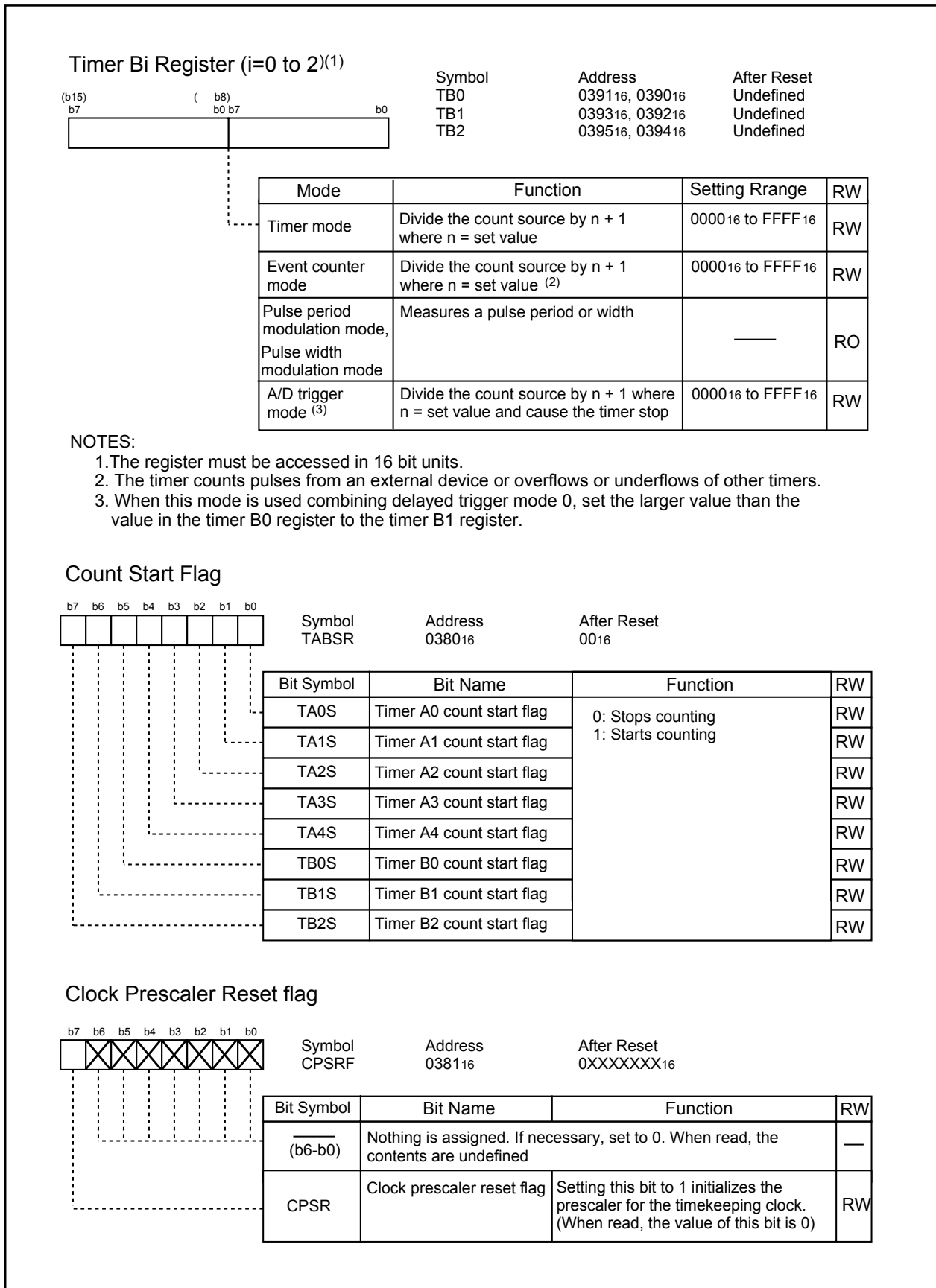


Figure 12.17 TB0 to TB2 Registers, TABSR Register, CPSRF Register

12.2.1 Timer Mode

In timer mode, the timer counts a count source generated internally (see **Table 12.6**). **Figure 12.18** shows TBiMR register in timer mode.

Table 12.6 Specifications in Timer Mode

| Item | Specification |
|-------------------------------------|--|
| Count source | f1, f2, f8, f32, fC32 |
| Count operation | <ul style="list-style-type: none"> Decrement When the timer underflows, it reloads the reload register contents and continues counting |
| Divide ratio | 1/(n+1) n: set value of TBi register (i= 0 to 2) 0000 ₁₆ to FFFF ₁₆ |
| Count start condition | Set TBiS bit ⁽¹⁾ to 1 (start counting) |
| Count stop condition | Set TBiS bit to 0 (stop counting) |
| Interrupt request generation timing | Timer underflow |
| TBiIN pin function | I/O port |
| Read from timer | Count value can be read by reading TBi register |
| Write to timer | <ul style="list-style-type: none"> When not counting and until the 1st count source is input after counting start Value written to TBi register is written to both reload register and counter When counting (after 1st count source input) Value written to TBi register is written to only reload register (Transferred to counter when reloaded next) |

NOTE:

- Bits TB0S to TB2S are assigned to the bit 7 to bit 5 in the TABSR register.

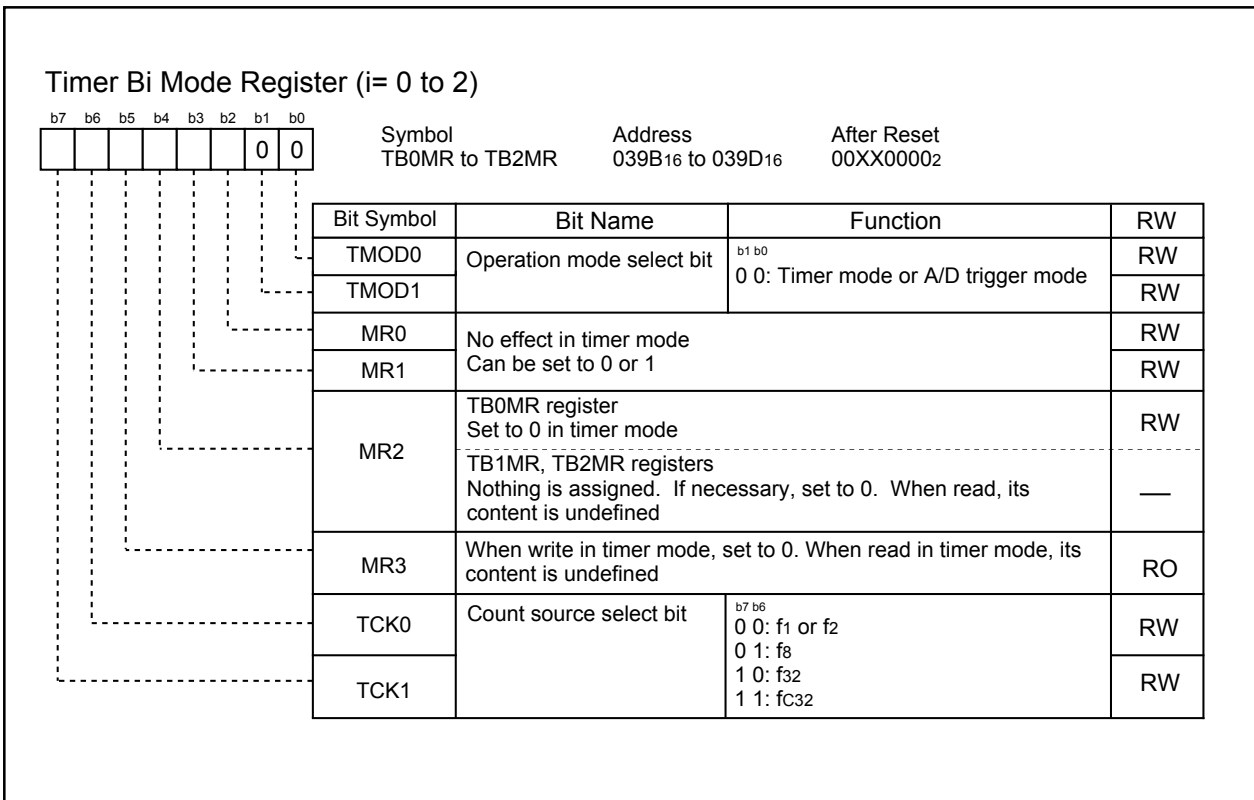


Figure 12.18 TBiMR Register in Timer Mode

12.2.2 Event Counter Mode

In event counter mode, the timer counts pulses from an external device or overflows and underflows of other timers (see Table 12.7). Figure 12.19 shows the TBiMR register in event counter mode.

Table 12.7 Specifications in Event Counter Mode

| Item | Specification |
|-------------------------------------|--|
| Count source | <ul style="list-style-type: none"> External signals input to TBiIN pin (i=0 to 2) (effective edge can be selected in program) Timer Bj overflow or underflow (j=i-1, except j=2 if i=0) |
| Count operation | <ul style="list-style-type: none"> Decrement When the timer underflows, it reloads the reload register contents and continues counting |
| Divide ratio | 1/(n+1) n: set value of TBi register 0000 ₁₆ to FFFF ₁₆ |
| Count start condition | Set TBiS bit ⁽¹⁾ to 1 (start counting) |
| Count stop condition | Set TBiS bit to 0 (stop counting) |
| Interrupt request generation timing | Timer underflow |
| TBiIN pin function | Count source input |
| Read from timer | Count value can be read by reading TBi register |
| Write to timer | <ul style="list-style-type: none"> When not counting and until the 1st count source is input after counting start Value written to TBi register is written to both reload register and counter When counting (after 1st count source input) Value written to TBi register is written to only reload register (Transferred to counter when reloaded next) |

NOTE:

- Bits TB2S to TB0S are assigned to the bit 7 to bit 5 in the TABSR register.

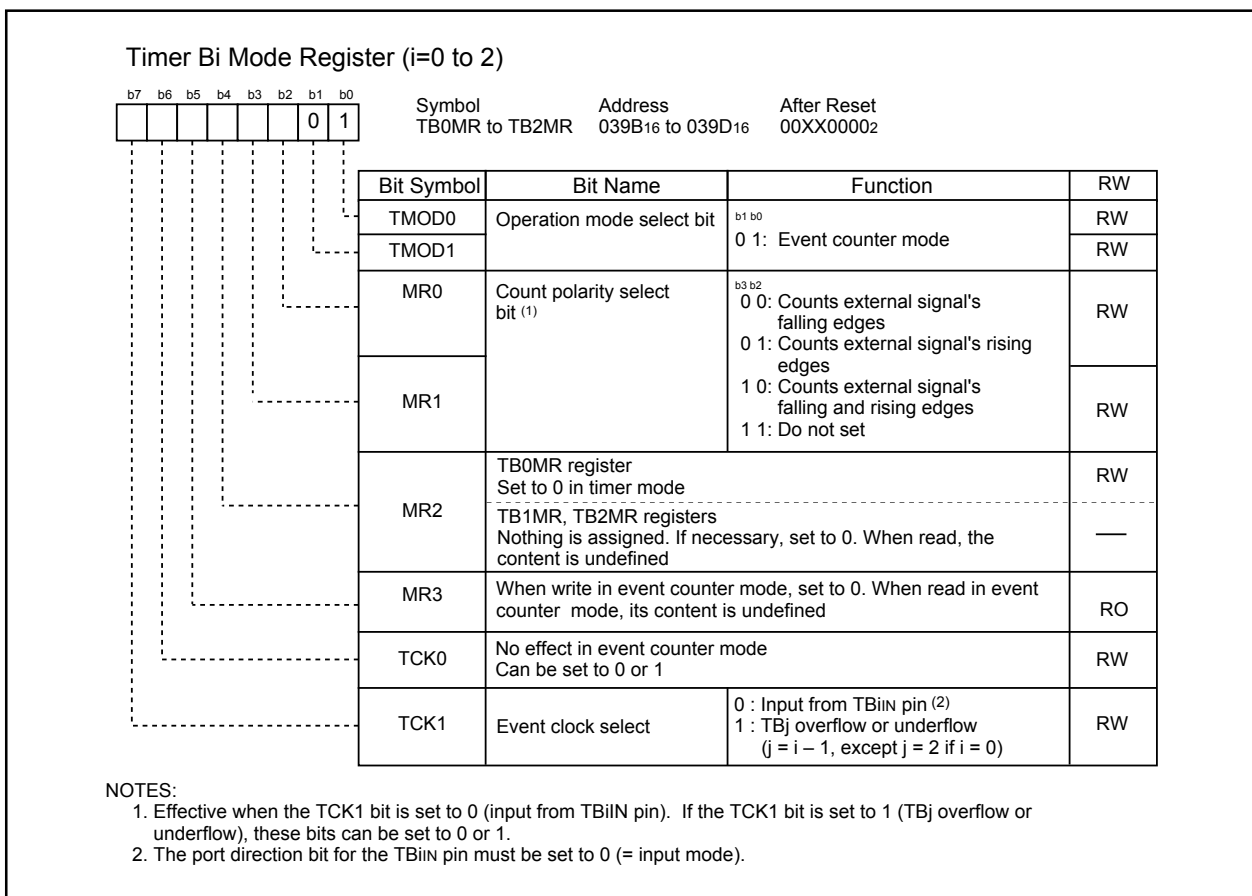


Figure 12.19 TBiMR Register in Event Counter Mode

12.2.3 Pulse Period and Pulse Width Measurement Mode

In pulse period and pulse width measurement mode, the timer measures pulse period or pulse width of an external signal (see **Table 12.8**). **Figure 12.20** shows the TBiMR register in pulse period and pulse width measurement mode. **Figure 12.21** shows the operation timing when measuring a pulse period. **Figure 12.22** shows the operation timing when measuring a pulse width.

Table 12.8 Specifications in Pulse Period and Pulse Width Measurement Mode

| Item | Specification |
|-------------------------------------|--|
| Count source | f1, f2, f8, f32, fc32 |
| Count operation | <ul style="list-style-type: none"> • Increment • Counter value is transferred to reload register at an effective edge of measurement pulse. The counter value is set to 0000₁₆ to continue counting. |
| Count start condition | Set TBiS (i=0 to 2) bit ⁽³⁾ to 1 (start counting) |
| Count stop condition | Set TBiS bit to 0 (stop counting) |
| Interrupt request generation timing | <ul style="list-style-type: none"> • When an effective edge of measurement pulse is input ⁽¹⁾ • Timer overflow. When an overflow occurs, MR3 bit in the TBiMR register is set to 1 (overflowed) simultaneously. MR3 bit is cleared to 0 (no overflow) by writing to TBiMR register at the next count timing or later after MR3 bit was set to 1. At this time, make sure TBiS bit is set to 1 (start counting). |
| TBiIN pin function | Measurement pulse input |
| Read from timer | Contents of the reload register (measurement result) can be read by reading TBi register ⁽²⁾ |
| Write to timer | Value written to TBi register is written to neither reload register nor counter |

NOTES:

1. Interrupt request is not generated when the first effective edge is input after the timer started counting.
2. Value read from TBi register is undefined until the second valid edge is input after the timer starts counting.
3. Bits TB0S to TB2S are assigned to the bit 5 to bit 7 in the TABSR register .

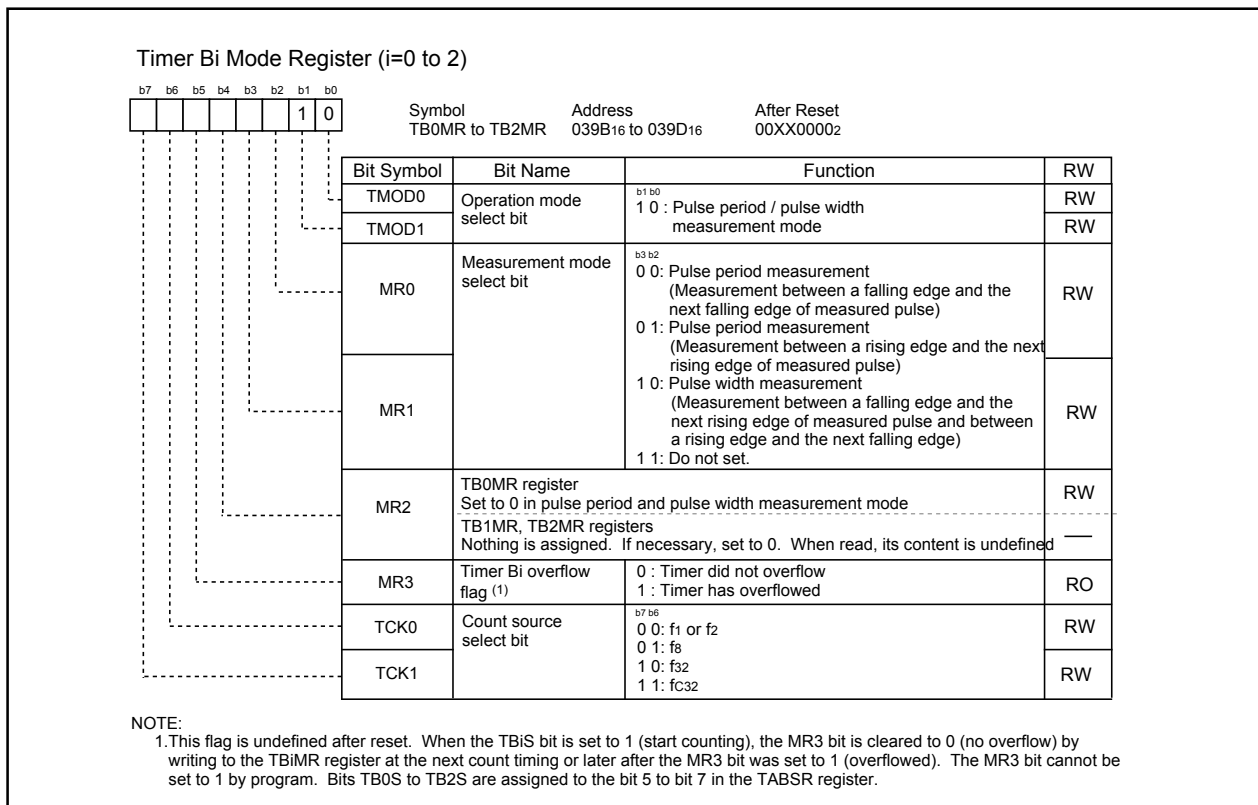


Figure 12.20 TBiMR Register in Pulse Period and Pulse Width Measurement Mode

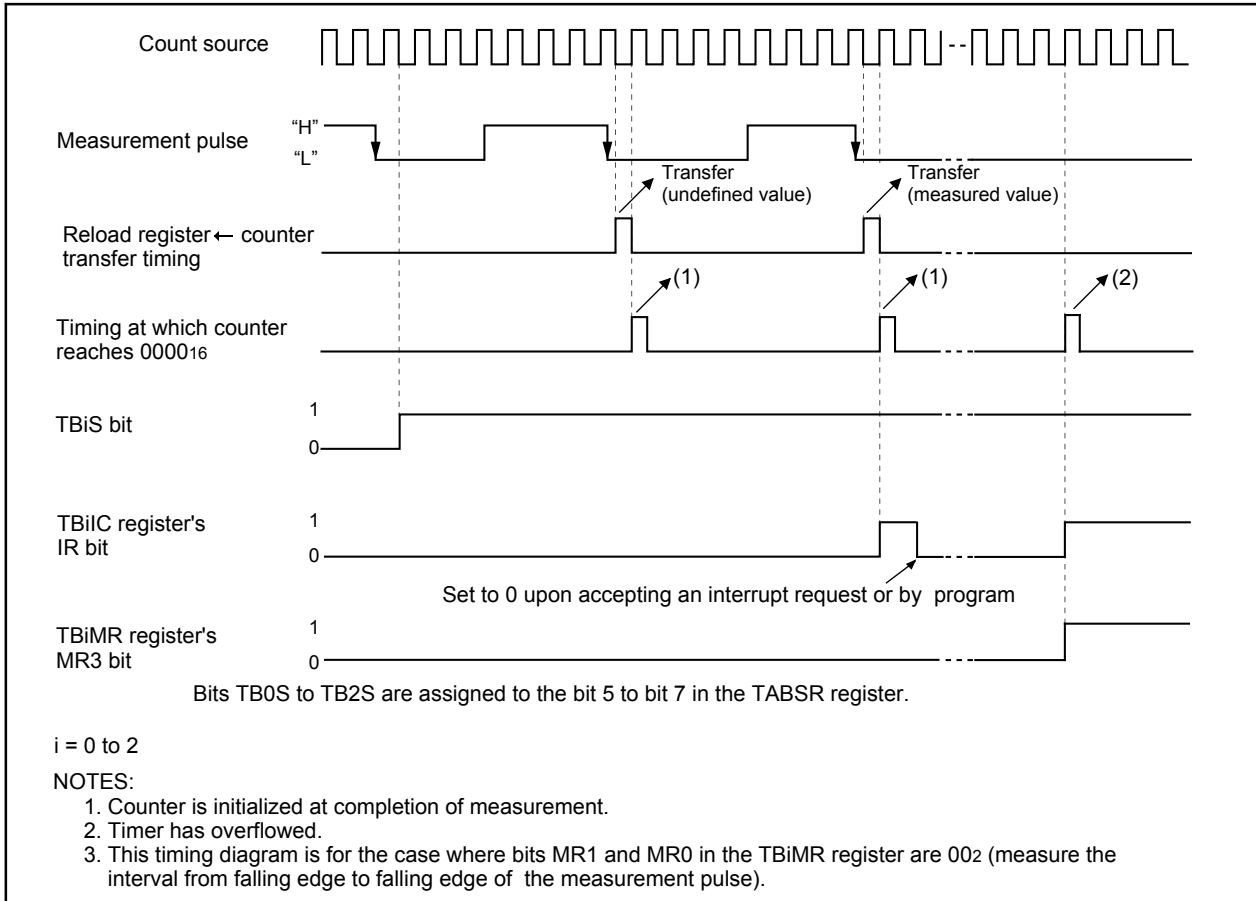


Figure 12.21 Operation timing when measuring a pulse period

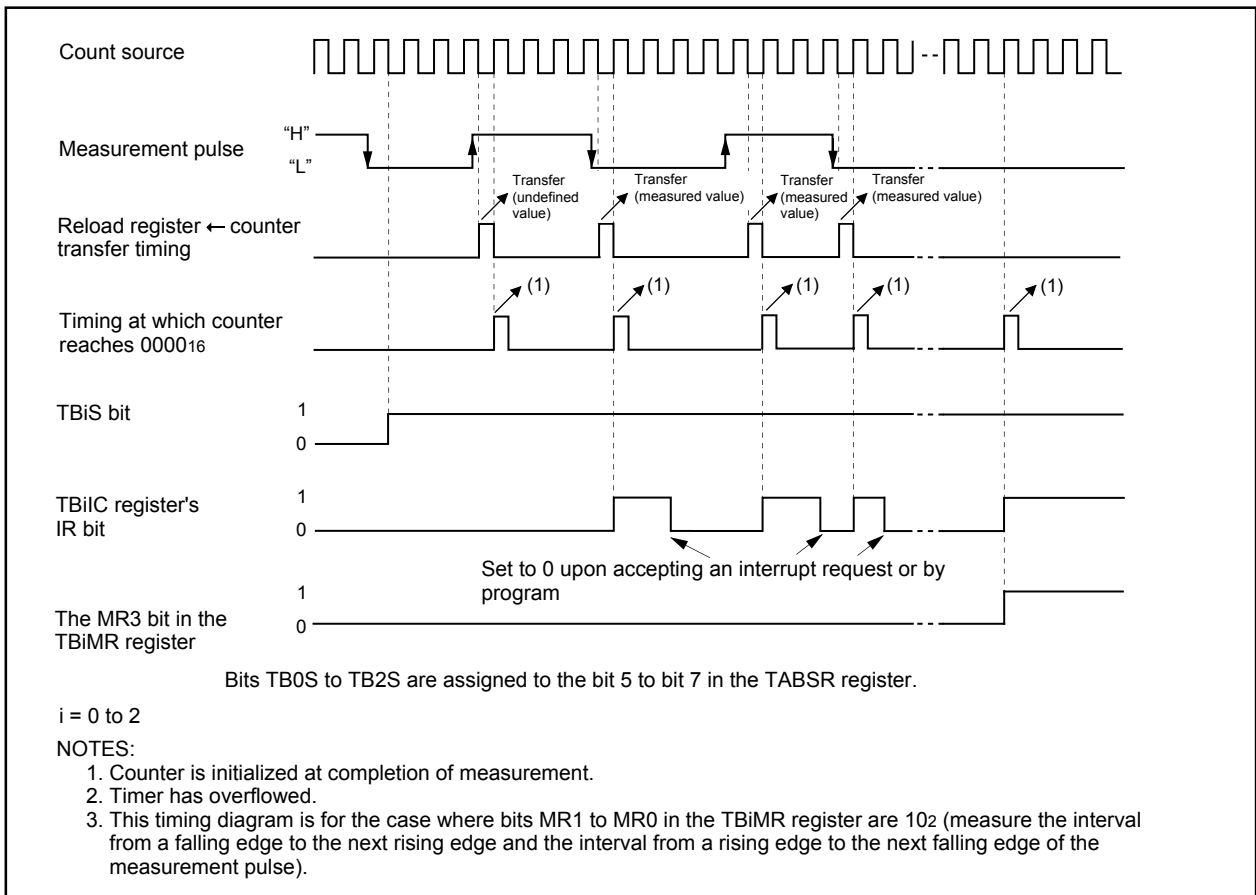


Figure 12.22 Operation timing when measuring a pulse width

12.2.4 A/D Trigger Mode

A/D trigger mode is used together with simultaneous sample sweep mode or delayed trigger mode 0 of A/D conversion to start A/D conversion. It is used in timer B0 and timer B1 only. In this mode, the timer starts counting by one trigger until the count value becomes 0000₁₆. **Figure 12.23** shows the TBiMR register in A/D trigger mode and **Figure 12.24** shows the TB2SC register.

Table 12.9 Specifications in A/D Trigger Mode

| Item | Specification |
|-------------------------------------|---|
| Count Source | f1, f2, f8, f32, and fc32 |
| Count Operation | <ul style="list-style-type: none"> • Decrement • When the timer underflows, reload register contents are reloaded before stopping counting • When a trigger is generated during the count operation, the count is not affected |
| Divide Ratio | 1/(n+1) n: Setting value of TBi register (i=0,1) 0000 ₁₆ -FFFF ₁₆ |
| Count Start Condition | When the TBiS (i=0,1) bit in the TABSR register is 1(count started), TBiEN(i=0,1) in TB2SC register is 1 (A/D trigger mode) and the following trigger is generated.(Selection based on bits TB2SEL in the TB2SC) <ul style="list-style-type: none"> • Timer B2 interrupt • Underflow of Timer B2 interrupt generation frequency counter setting |
| Count Stop Condition | <ul style="list-style-type: none"> • After the count value is 0000₁₆ and reload register contents are reloaded • Set the TBiS bit to 0 (count stopped) |
| Interrupt Request Generation Timing | Timer underflows ⁽¹⁾ |
| TBiIN Pin Function | I/O port |
| Read From Timer | Count value can be read by reading TBi register |
| Write To Timer ⁽²⁾ | <ul style="list-style-type: none"> • When writing in the TBi register during count stopped. Value is written to both reload register and counter • When writing in the TBi register during count. Value is written to only reload register (Transferred to counter when reloaded next) |

NOTES:

1: A/D conversion is started by the timer underflow. For details refer to **15. A/D Converter**.

2: When using in delayed trigger mode 0, set the larger value than the value of the timer B0 register to the timer B1 register.

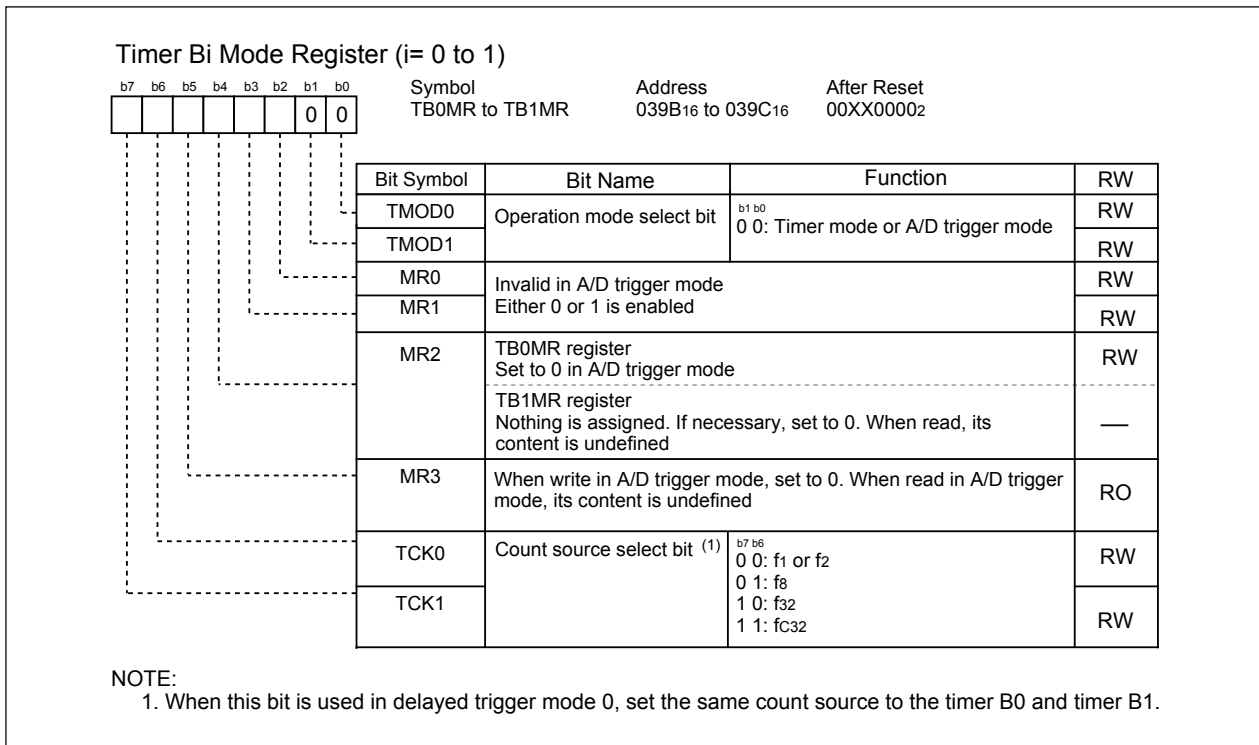


Figure 12.23 TBiMR Register in A/D Trigger Mode

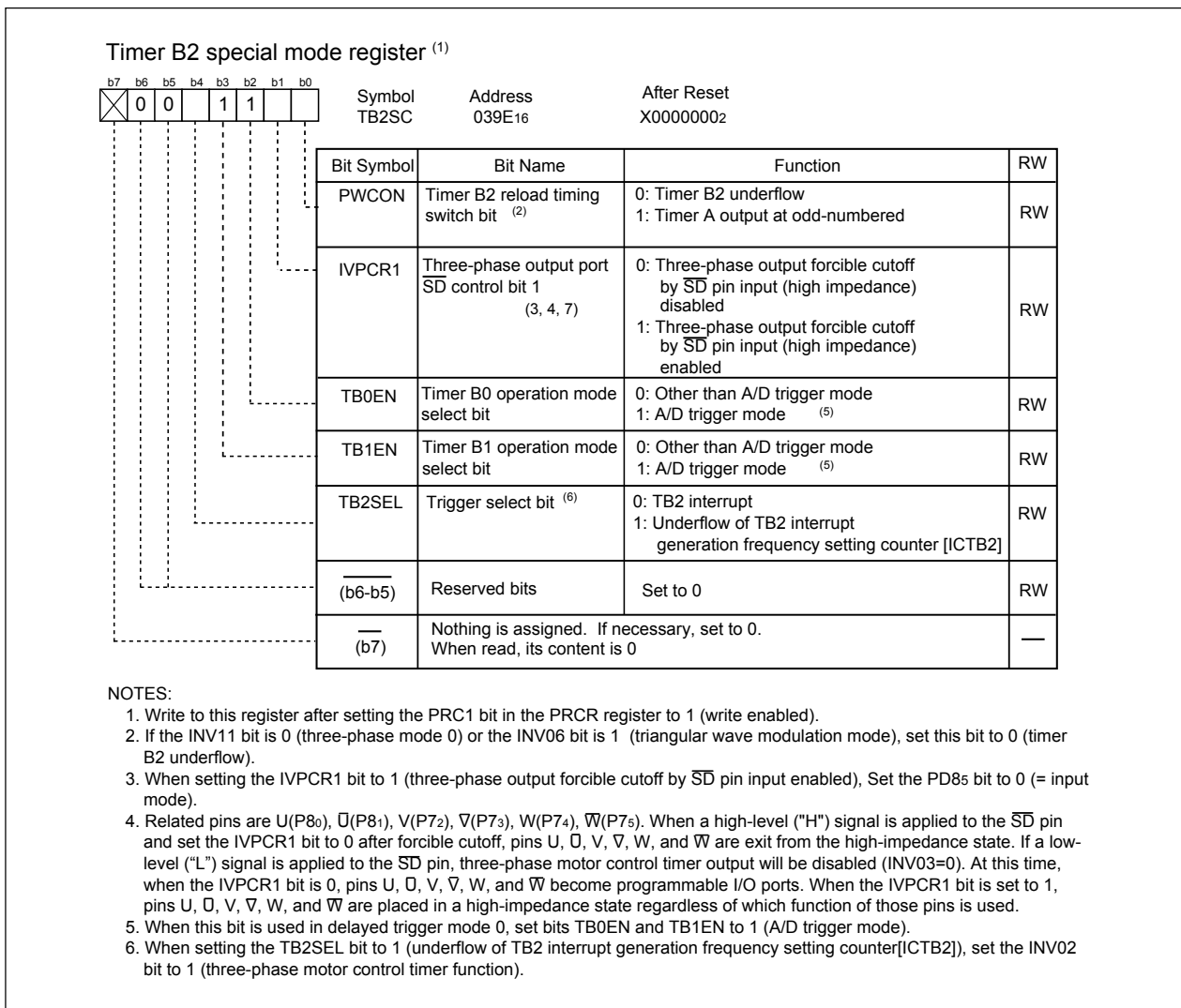


Figure 12.24 TB2SC Register in A/D Trigger Mode

12.3 Three-phase Motor Control Timer Function

Timers A1, A2, A4 and B2 can be used to output three-phase motor drive waveforms. **Table 12.10** lists the specifications of the three-phase motor control timer function. **Figure 12.24** shows the block diagram for three-phase motor control timer function. Also, the related registers are shown on **Figures 12.26 to 12.32**.

Table 12.10 Three-phase Motor Control Timer Function Specifications

| Item | Specification |
|--|--|
| Three-phase waveform output pin | Six pins (U, \bar{U} , V, \bar{V} , W, \bar{W}) |
| Forced cutoff input ⁽¹⁾ | Input "L" to \bar{SD} pin |
| Used Timers | Timer A4, A1, A2 (used in the one-shot timer mode) Timer A4: U- and \bar{U} -phase waveform control Timer A1: V- and \bar{V} -phase waveform control Timer A2: W- and \bar{W} -phase waveform control Timer B2 (used in the timer mode) Carrier wave cycle control Dead time timer (3 eight-bit timer and shared reload register) Dead time control |
| Output waveform | Triangular wave modulation, Sawtooth wave modification Enable to output "H" or "L" for one cycle Enable to set positive-phase level and negative-phase level respectively |
| Carrier wave cycle | Triangular wave modulation: count source x (m+1) x 2 Sawtooth wave modulation: count source x (m+1) m: Setting value of TB2 register, 0 to 65535 Count source: f ₁ , f ₂ , f ₈ , f ₃₂ , f _{c32} |
| Three-phase PWM output width | Triangular wave modulation: count source x n x 2 Sawtooth wave modulation: count source x n n: Setting value of TA4, TA1 and TA2 register (of TA4, TA41, TA1, TA11, TA2 and TA21 registers when setting the INV11 bit to 1), 1 to 65535 Count source: f ₁ , f ₂ , f ₈ , f ₃₂ , f _{c32} |
| Dead time | Count source x p, or no dead time p: Setting value of DTT register, 1 to 255 Count source: f ₁ , f ₂ , f ₁ divided by 2, f ₂ divided by 2 |
| Active level | Enable to select "H" or "L" |
| Positive and negative-phase concurrent | Positive and negative-phases concurrent active disable function Positive and negative-phases concurrent active detect function |
| Interrupt frequency | For Timer B2 interrupt, select a carrier wave cycle-to-cycle basis through 15 times carrier wave cycle-to-cycle basis |

NOTE:

- When the INV02 bit in the INVC0 register is set to 1 (three-phase motor control timer function), the \bar{SD} function of the P85/ \bar{SD} pin is enabled. At this time, the P85 pin cannot be used as a programmable I/O port. When the \bar{SD} function is not used, apply "H" to the P85/ \bar{SD} pin.

When the IVPCR1 bit in the TB2SC register is set to 1 (enable three-phase output forced cutoff by \bar{SD} pin input), and "L" is applied to the \bar{SD} pin, the related pins enter high-impedance state regardless of the functions which are used. When the IVPCR1 bit is set to 0 (disabled three-phase output forced cutoff by \bar{SD} pin input) and "L" is applied to the \bar{SD} pin, the related pins can be selected as a programmable I/O port and the setting of the port and port direction registers are enable.

Related pins: P72/CLK2/TA1OUT/V/RXD1 P73/CTS2/RTS2/TA1IN/V/TXD1
 P74/TA2OUT/W P75/TA2IN/ \bar{W}
 P80/TA4OUT/U P81/TA4IN/ \bar{U}

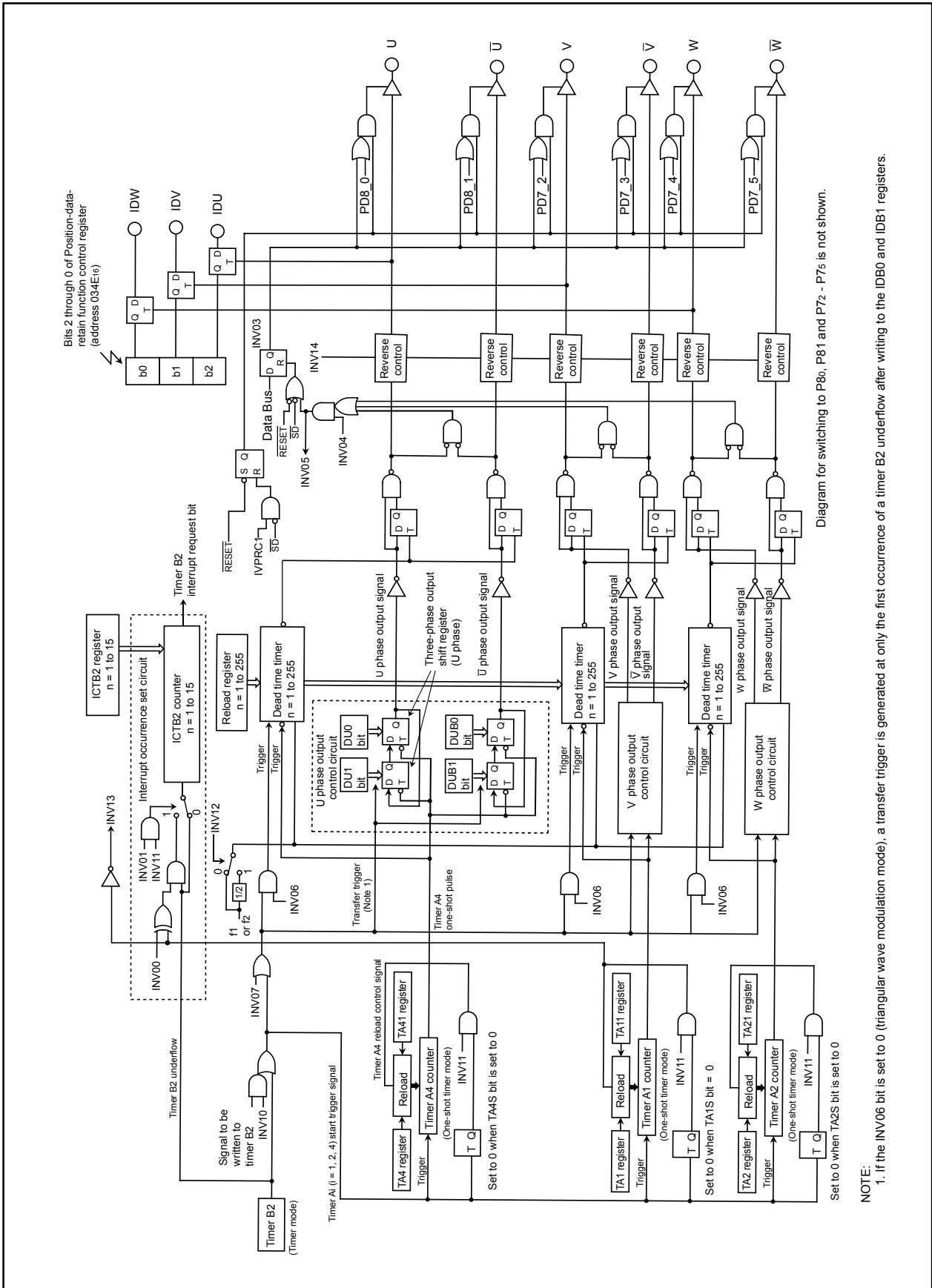


Figure 12.25 Three-phase Motor Control Timer Functions Block Diagram

NOTE:
 1. If the INV06 bit is set to 0 (triangular wave modulation mode), a transfer trigger is generated at only the first occurrence of a timer B2 underflow after writing to the IDB0 and IDB1 registers.

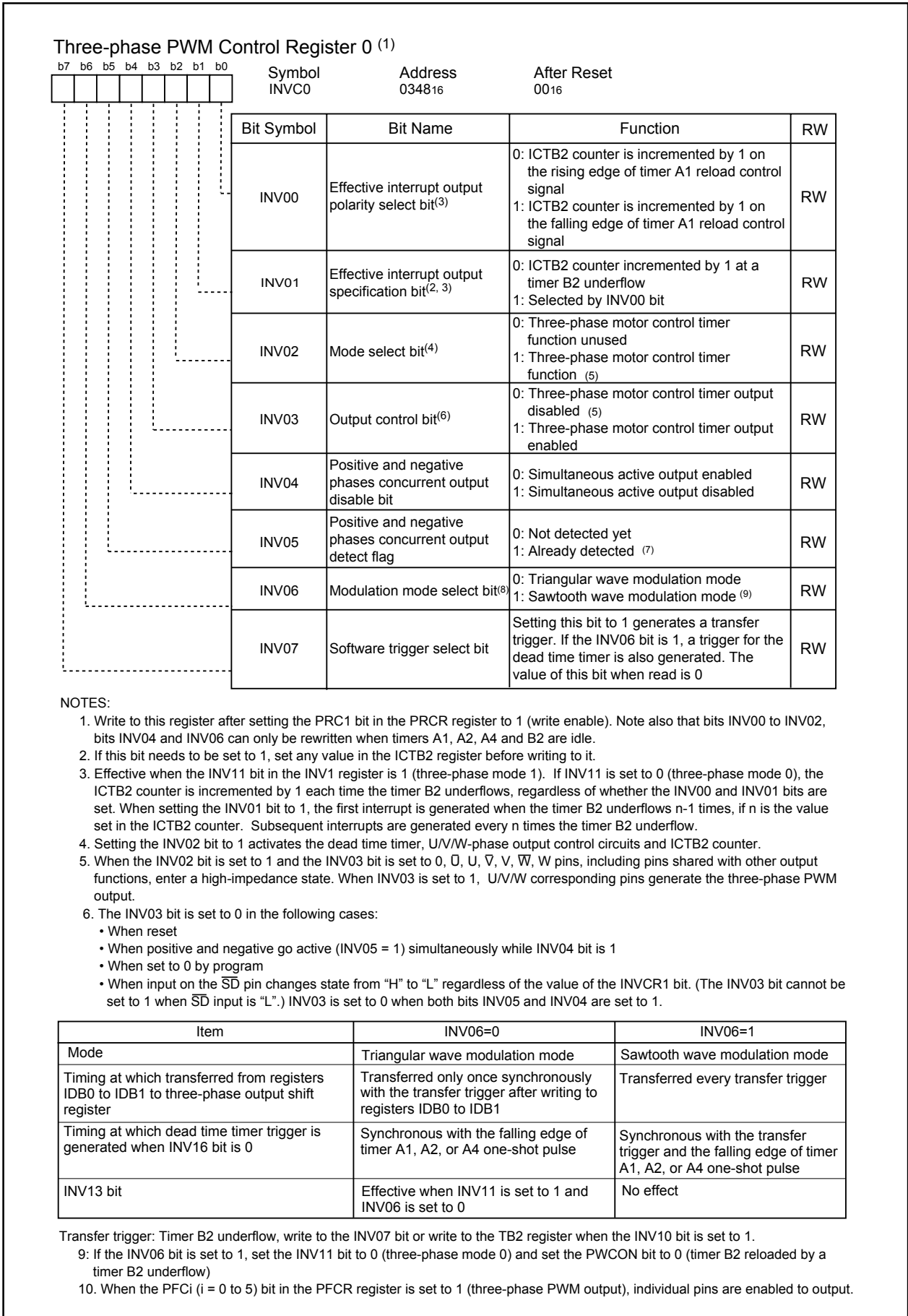


Figure 12.26 INVC0 Register

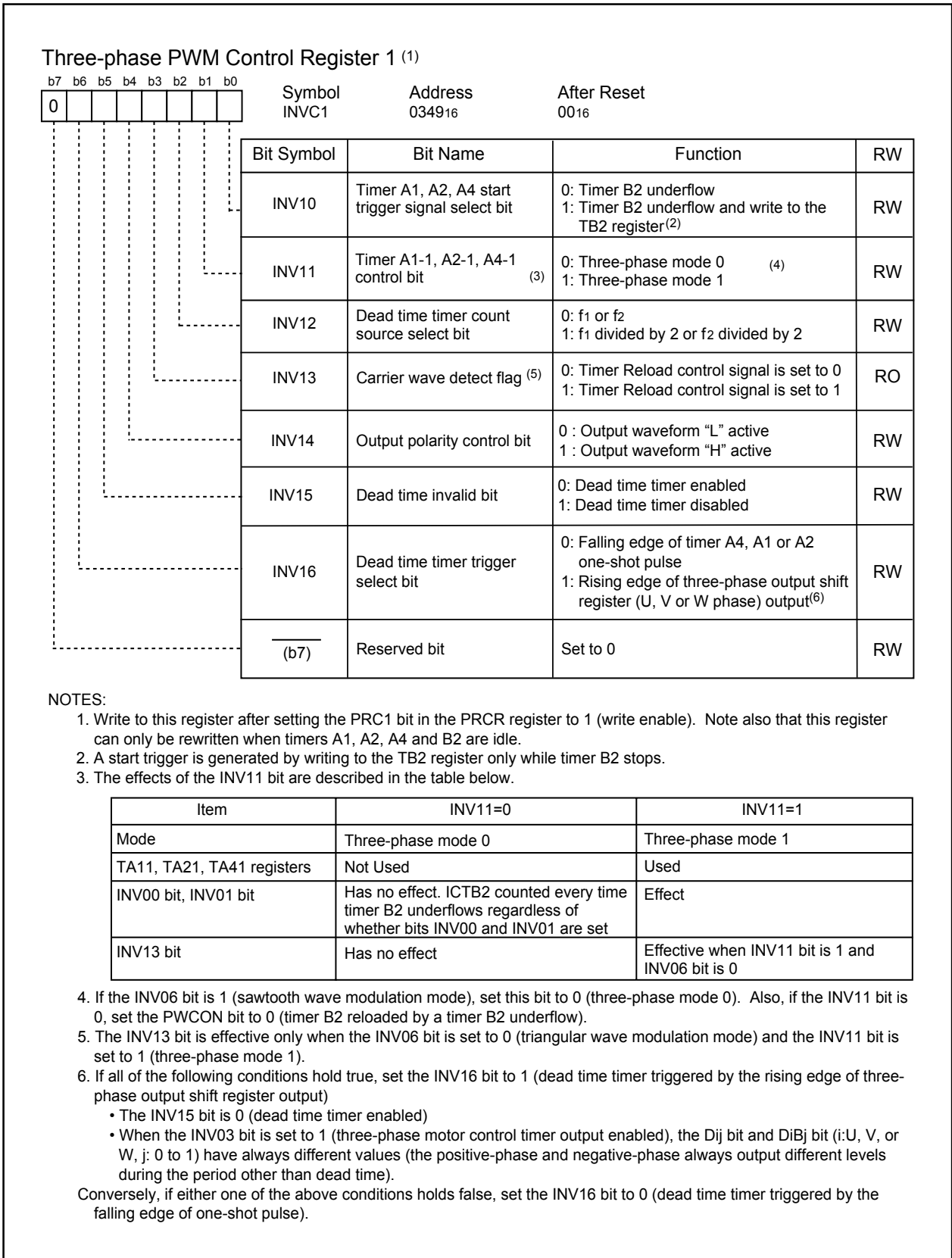


Figure 12.27 INVC1 Register

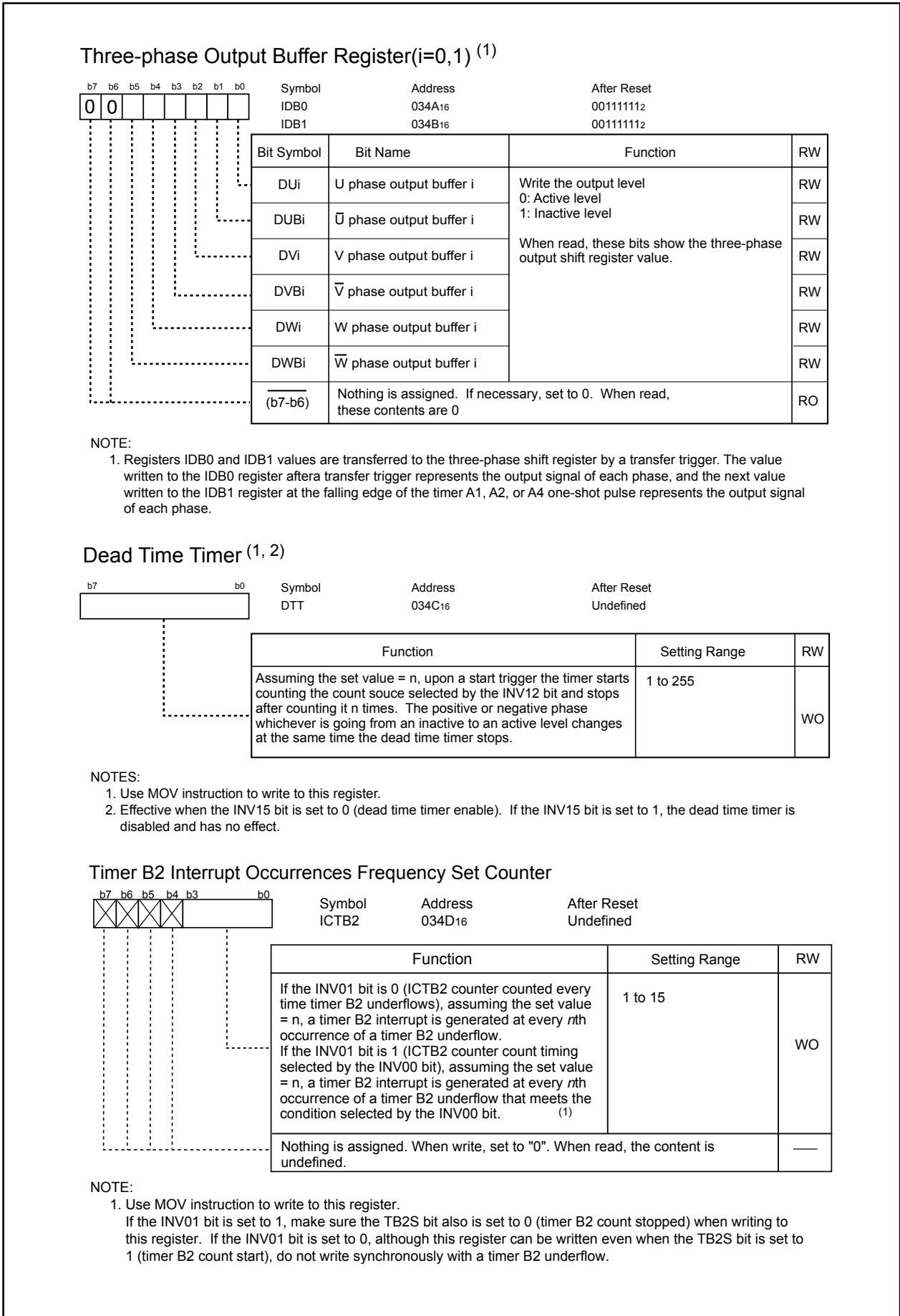


Figure 12.28 IDB0 Register, IDB1 Register, DTT Register, and ICTB2 Register

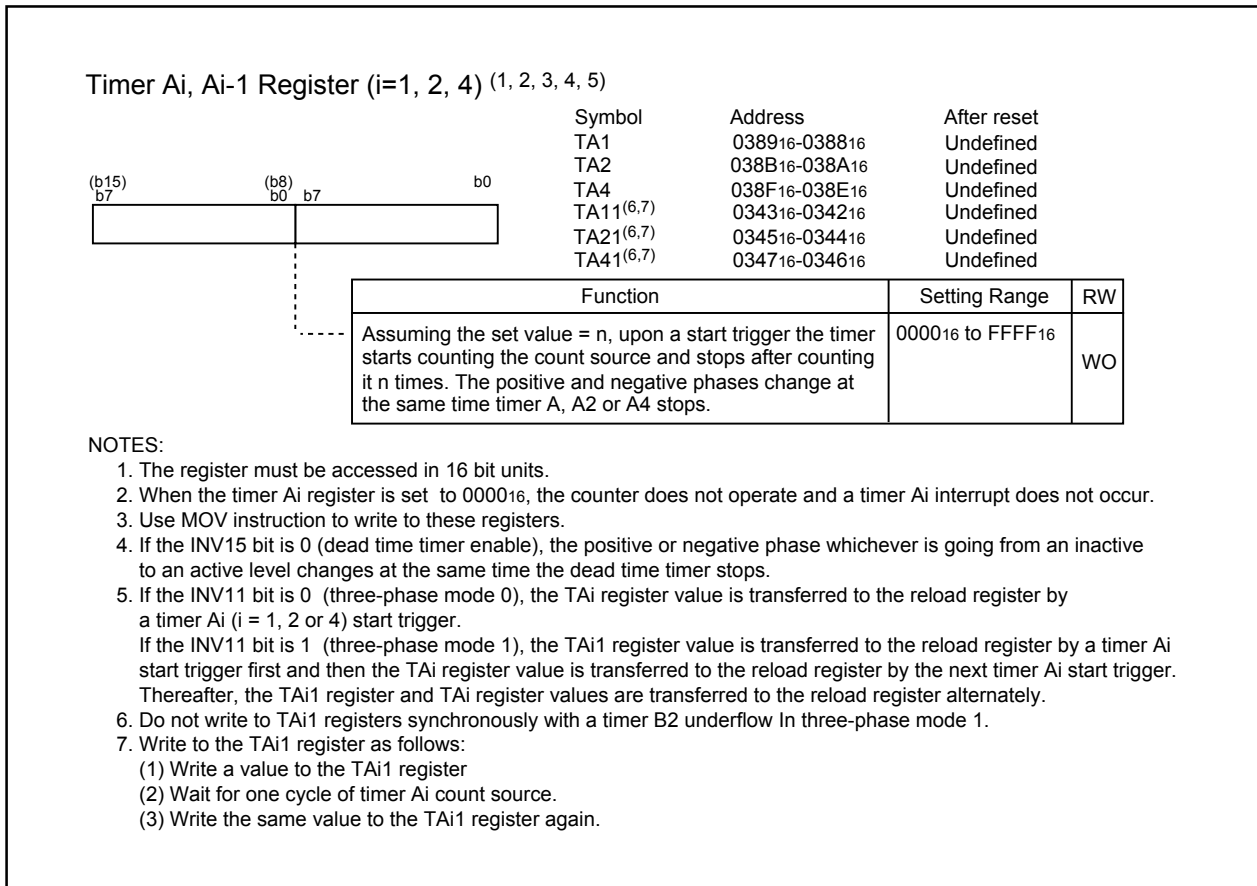


Figure 12.29 TA1, TA2, TA4, TA11, TA21, and TA41 Registers

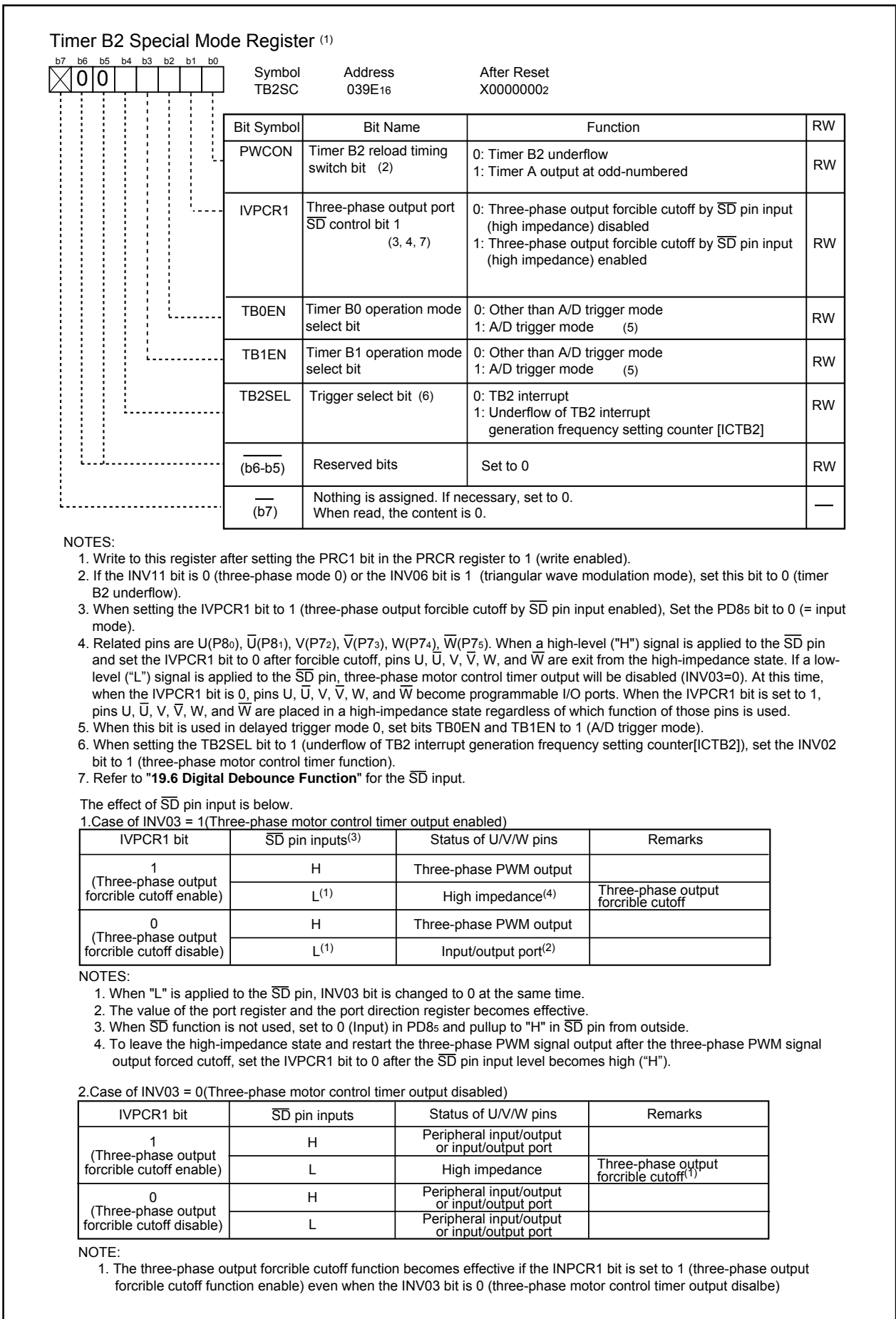


Figure 12.30 TB2SC Register

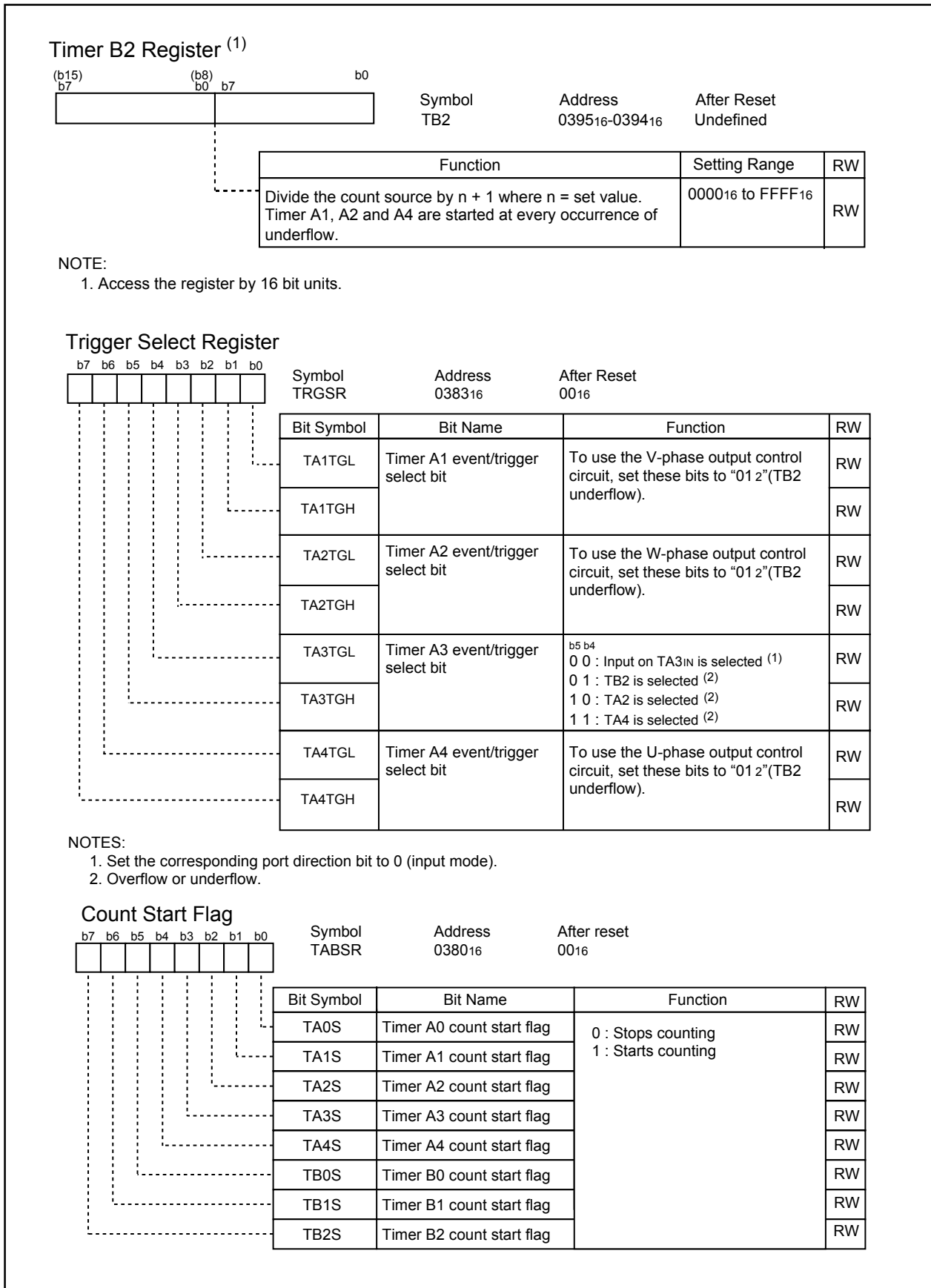


Figure 12.31 TB2 Register, TRGSR Register, and TABSR Register

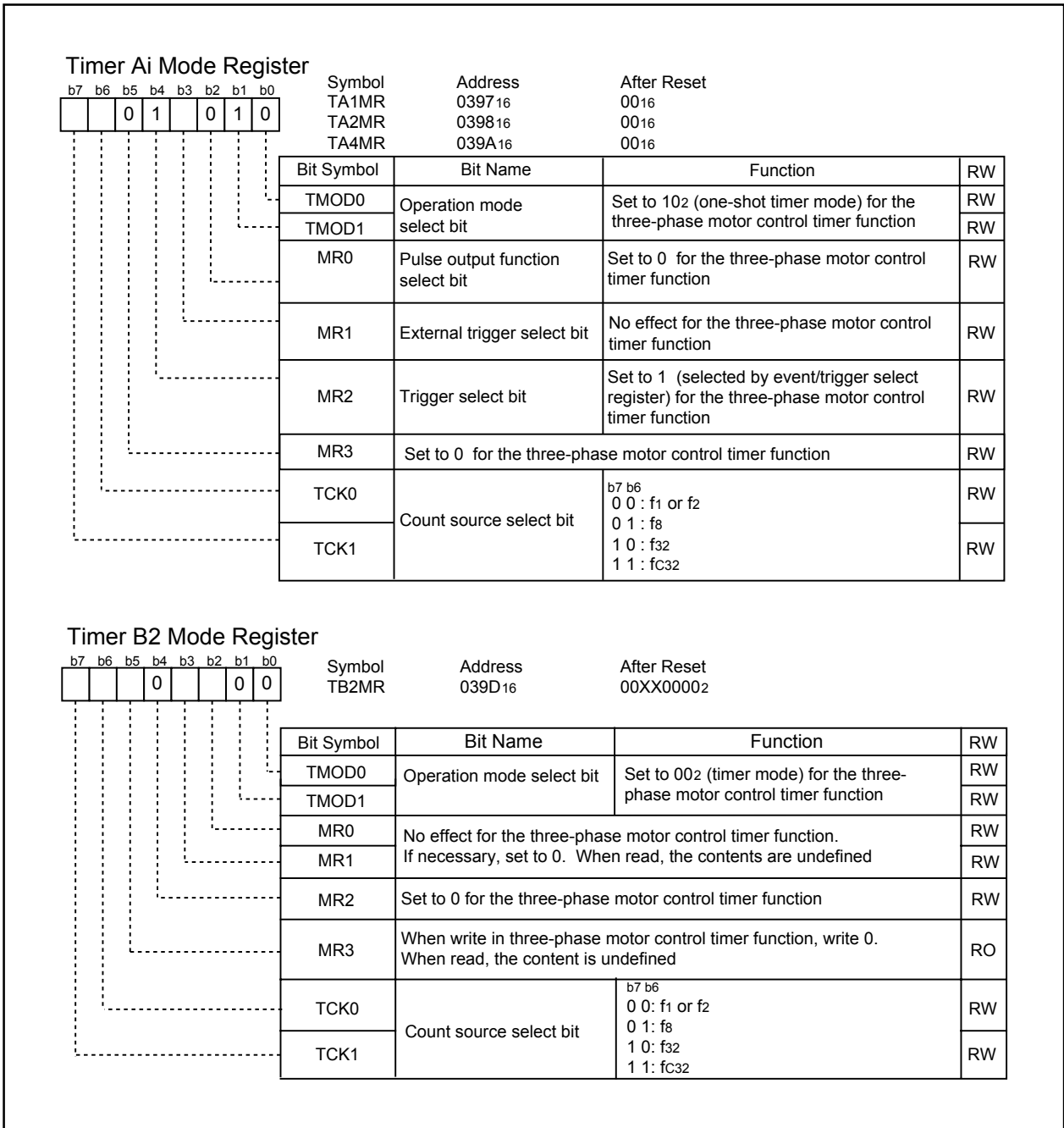


Figure 12.32 TA1MR, TA2MR, TA4MR, and TB2MR Registers

The three-phase motor control timer function is enabled by setting the INV02 bit in the INVC0 register to 1. When this function is on, timer B2 is used to control the carrier wave, and timers A4, A1 and A2 are used to control three-phase PWM outputs (U, \bar{U} , V, \bar{V} , W and \bar{W}). The dead time is controlled by a dedicated dead-time timer. **Figure 12.33** shows the example of triangular modulation waveform, and **Figure 12.34** shows the example of sawtooth modulation waveform.

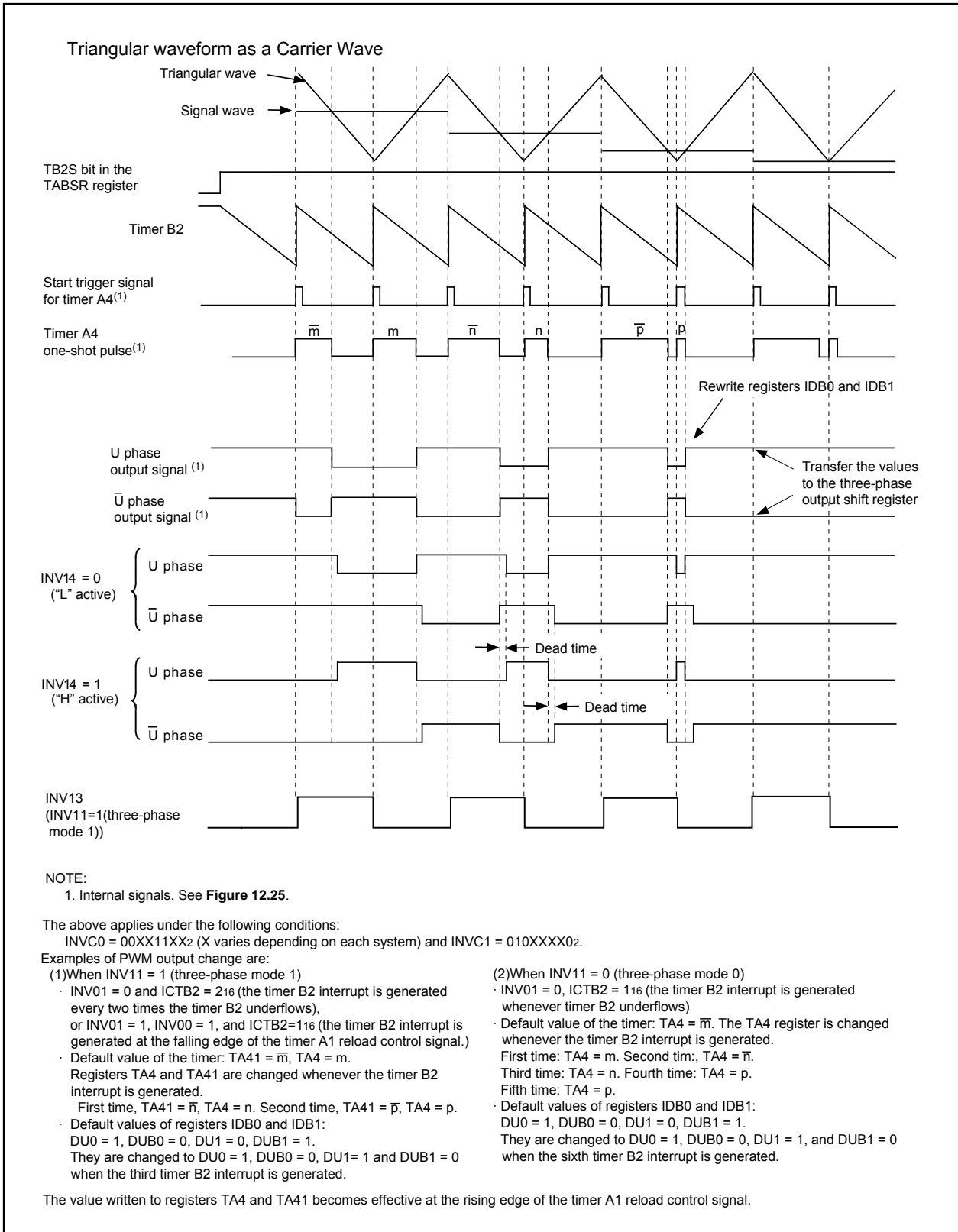
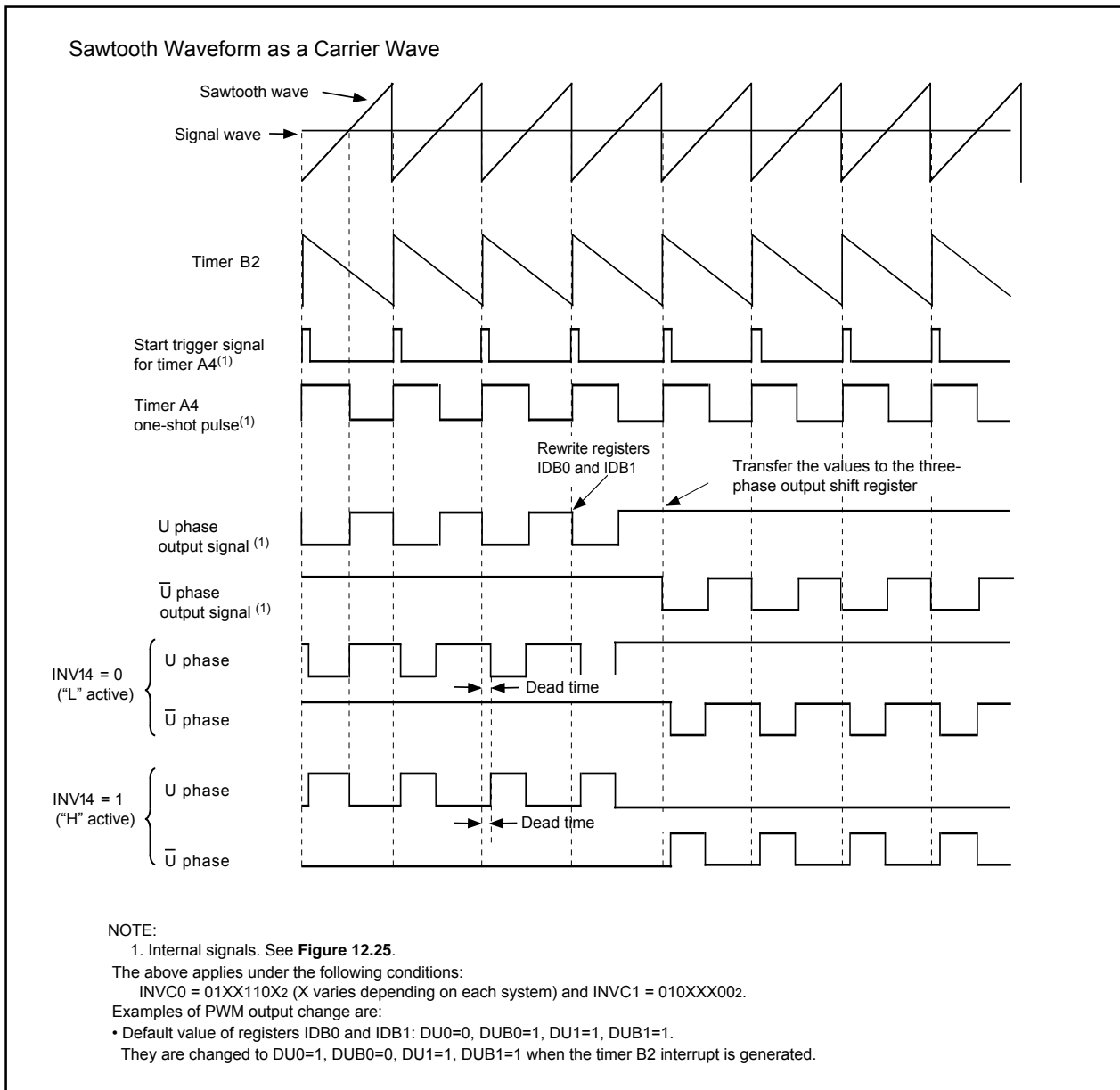


Figure 12.33 Triangular Wave Modulation Operation

**Figure 12.34 Sawtooth Wave Modulation Operation**

12.3.1 Position-Data-Retain Function

This function is used to retain the position data synchronously with the three-phase waveform output. There are three position-data input pins for U, V, and W phases.

A trigger to retain the position data (hereafter, this trigger is referred to as "retain trigger") can be selected by the PDRT bit in the PDRF register. This bit selects the retain trigger to be the falling edge of each positive phase, or the rising edge of each positive phase.

12.3.1.1 Operation of the Position-data-retain Function

Figure 12.35 shows a usage example of the position-data-retain function (U phase) when the retain trigger is selected as the falling edge of the positive signal.

- (1) At the falling edge of the U-phase waveform output, the state at pin IDU is transferred to the PDRU bit in the PDRF register.
- (2) Until the next falling edge of the Uphase waveform output, the above value is retained.

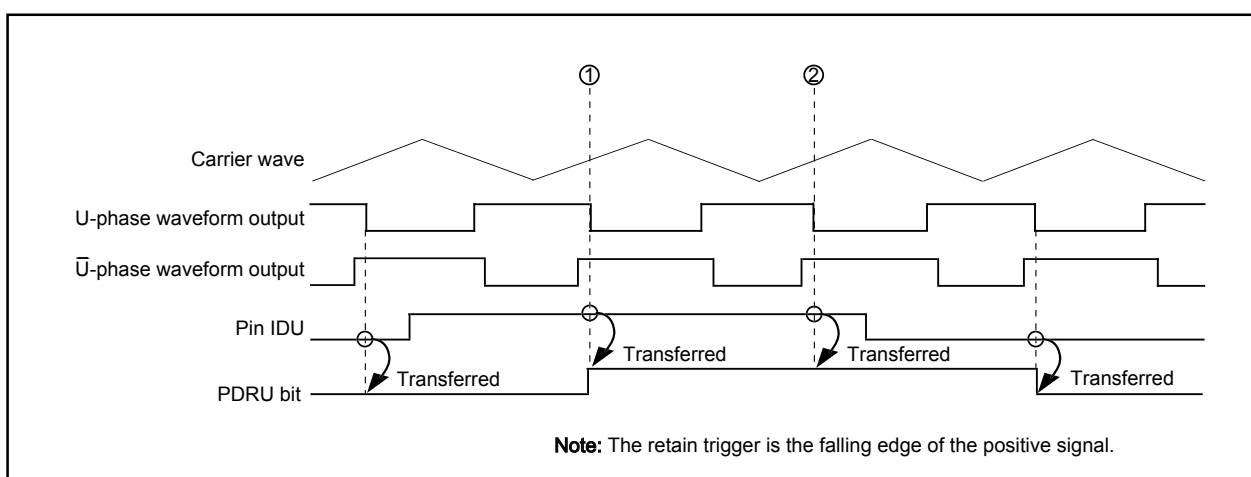


Figure 12.35 Usage Example of Position-data-retain Function (U phase)

12.3.1.2 Position-data-retain Function Control Register

Figure 12.36 shows the structure of the position-data-retain function control register.

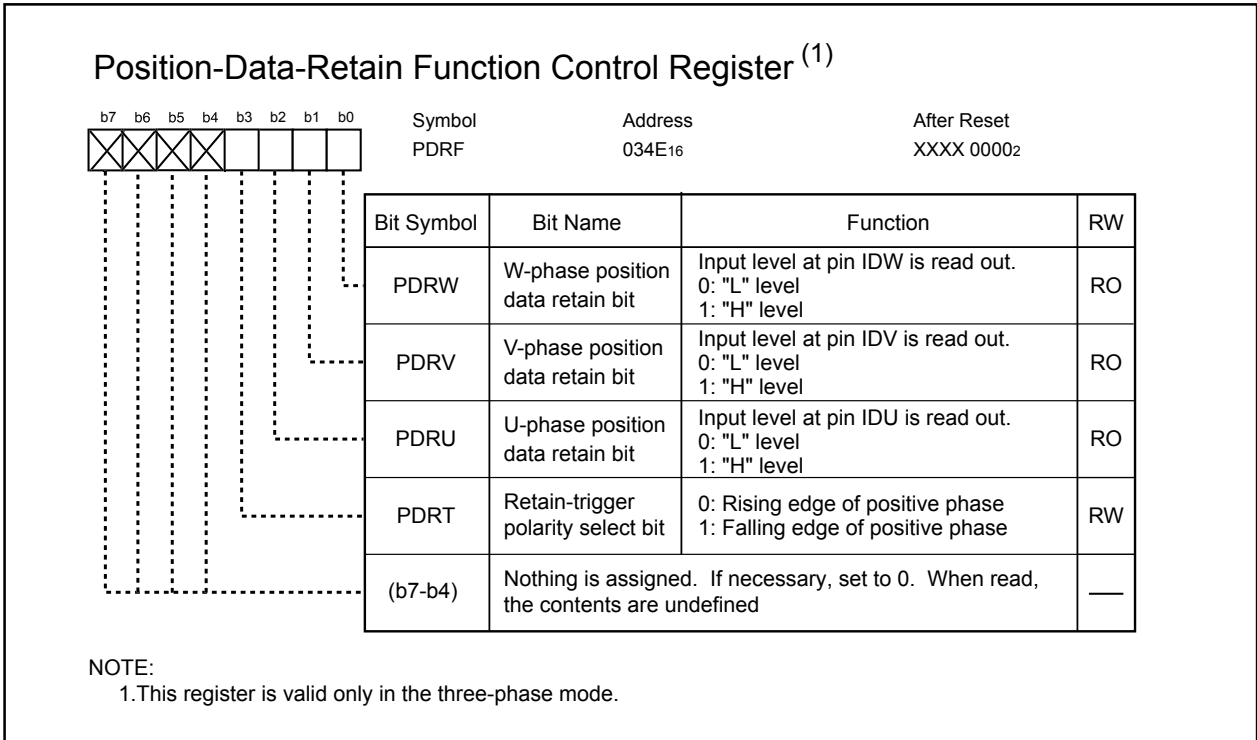


Figure 12.36 PDRF Register

12.3.1.2.1 W-phase Position Data Retain Bit (PDRW)

This bit is used to retain the input level at pin IDW.

12.3.1.2.2 V-phase Position Data Retain Bit (PDRV)

This bit is used to retain the input level at pin IDV.

12.3.1.2.3 U-phase Position Data Retain Bit (PDRU)

This bit is used to retain the input level at pin IDU.

12.3.1.2.4 Retain-trigger Polarity Select Bit (PDRT)

This bit is used to select the trigger polarity to retain the position data.
When this bit is set to 0, the rising edge of each positive phase selected.
When this bit is set to 1, the falling edge of each positive phase selected.

12.3.2 Three-phase/Port Output Switch Function

When the INVC03 bit in the INVC0 register set to 1 (Timer output enabled for three-phase motor control) and setting the PFCi (i=0 to 5) in the PFCR register to 0 (I/O port), the three-phase PWM output pin (U, \bar{U} , V, \bar{V} , W and \bar{W}) functions as I/O port. Each bit of the PFCi bits (i=0 to 5) is applicable for each one of three-phase PWM output pins. **Figure 12.37** shows the example of three-phase/port output switch function. **Figure 12.38** shows the PFCR register and the three-phase protect control register.

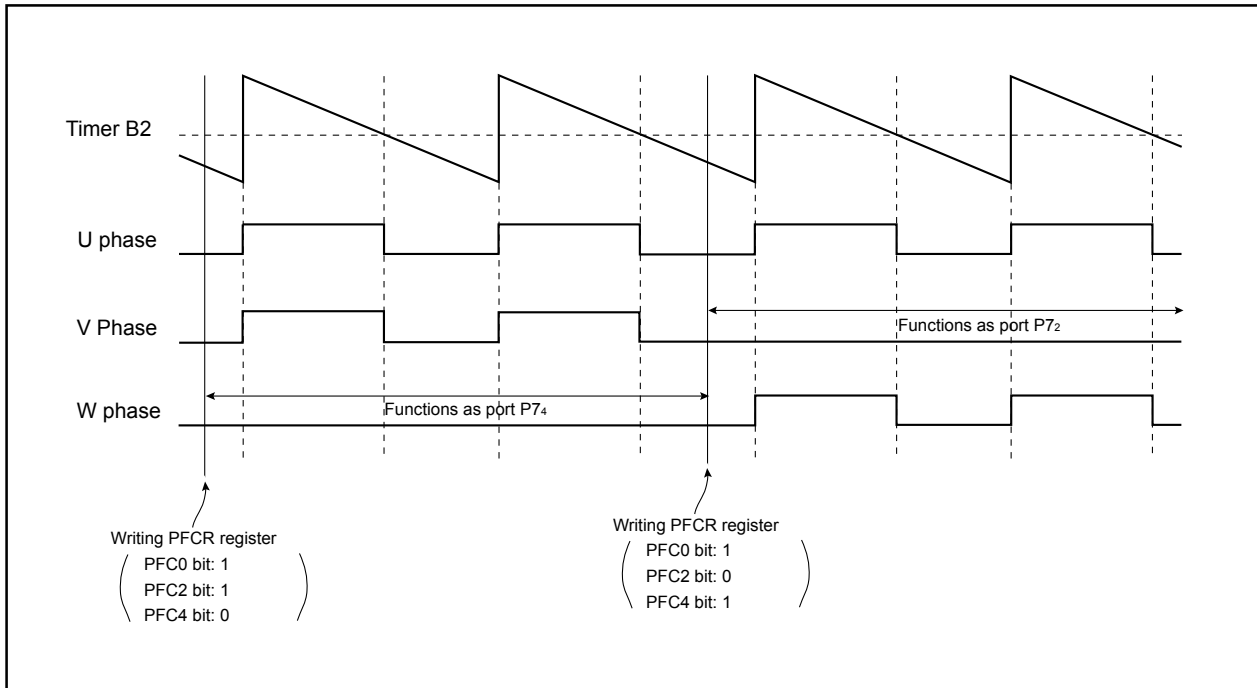
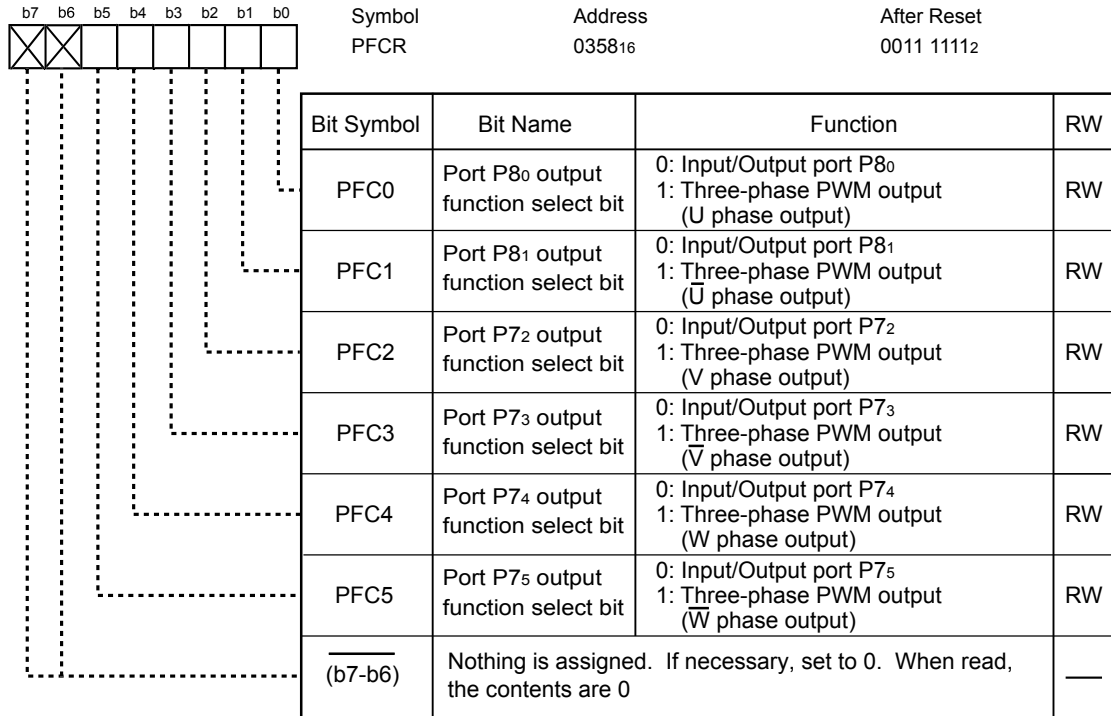


Figure 12.37 Usage Example of Three-phase/Port Output Switch Function

Port Function Control Register (1)



NOTE:
 1. This register is valid only when the INVC03 bit in the INVC0 register is 1 (Three-phase motor control timer).

Three-phase Protect Control Register

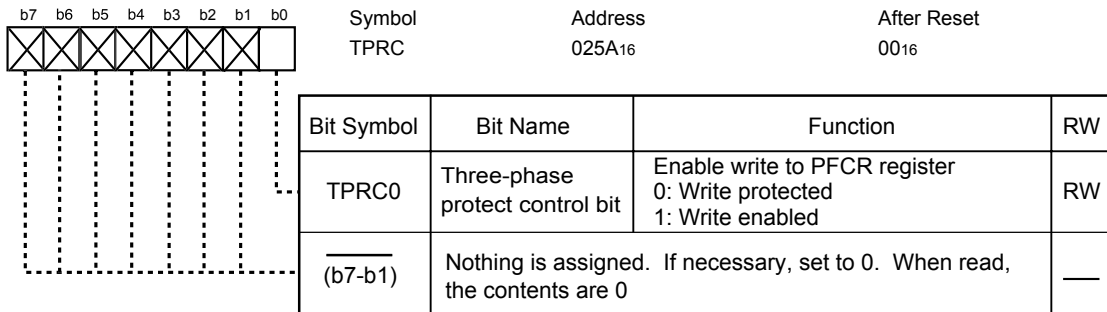


Figure 12.38 PFCR Register, and TPRC Register

13. Timer S

The Timer S (Input Capture/Output Compare : here after, Timer S is referred to as "IC/OC".) is a high-performance I/O port for time measurement and waveform generation.

The IC/OC has one 16-bit base timer for free-running operation and eight 16-bit registers for time measurement and waveform generation.

Table 13.1 lists functions and channels of the IC/OC.

Table 13.1 IC/OC Functions and Channels

| Function | Description |
|------------------------------------|-------------|
| Time measurement ⁽¹⁾ | 8 channels |
| Digital filter | 8 channels |
| Trigger input prescaler | 2 channels |
| Trigger input gate | 2 channels |
| Waveform generation ⁽¹⁾ | 8 channels |
| Single-phase waveform output | Available |
| Phase-delayed waveform output | Available |
| Set/Reset waveform output | Available |

NOTE:

- The time measurement function and the waveform generating function share a pin.
The time measurement function or waveform generating function can be selected for each channel.

Figure 13.1 shows the block diagram of the IC/OC.

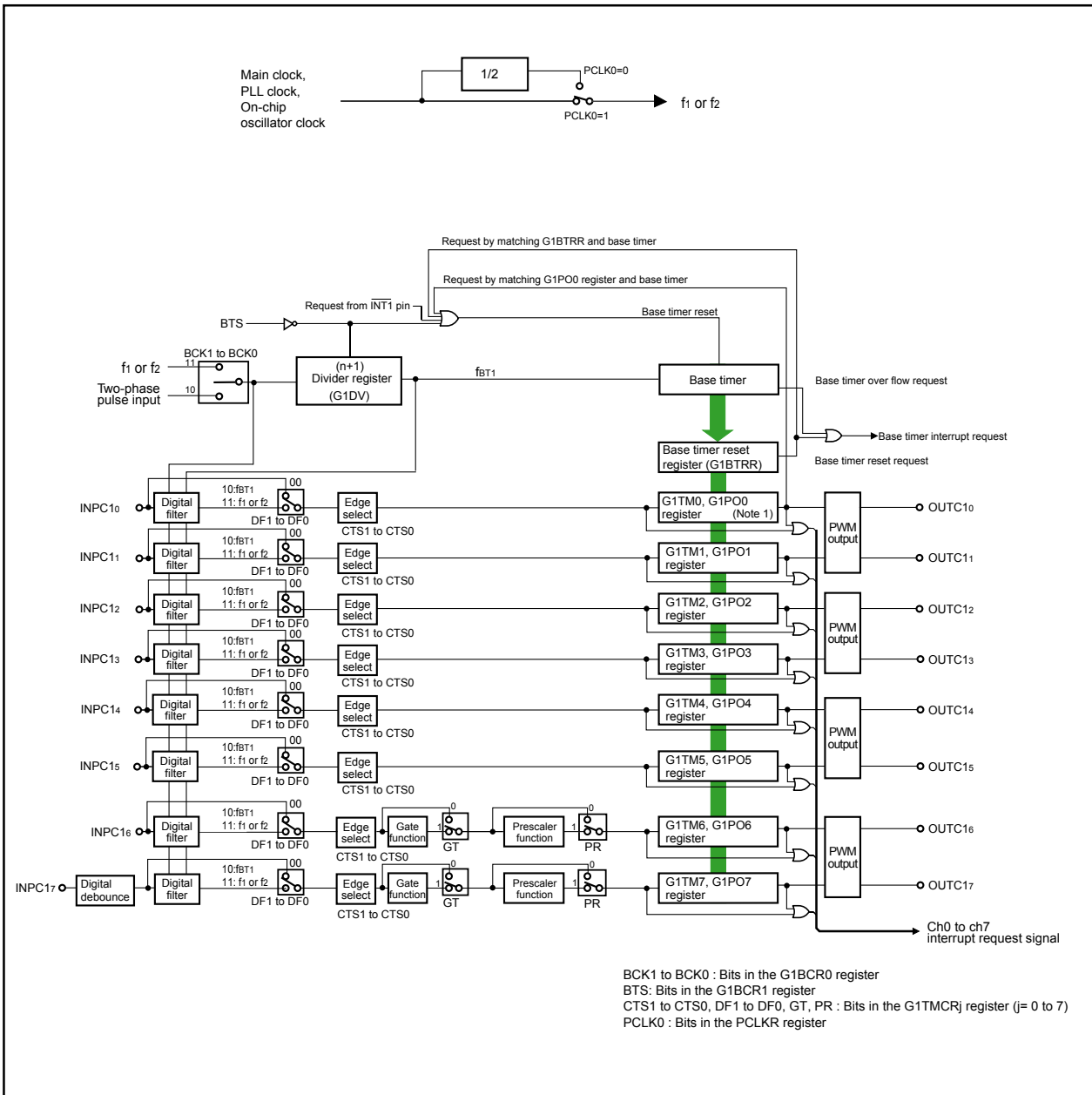


Figure 13.1 IC/OC Block Diagram

Figures 13.2 to 13.10 show registers associated with the IC/OC base timer, the time measurement function, and the waveform generating function.

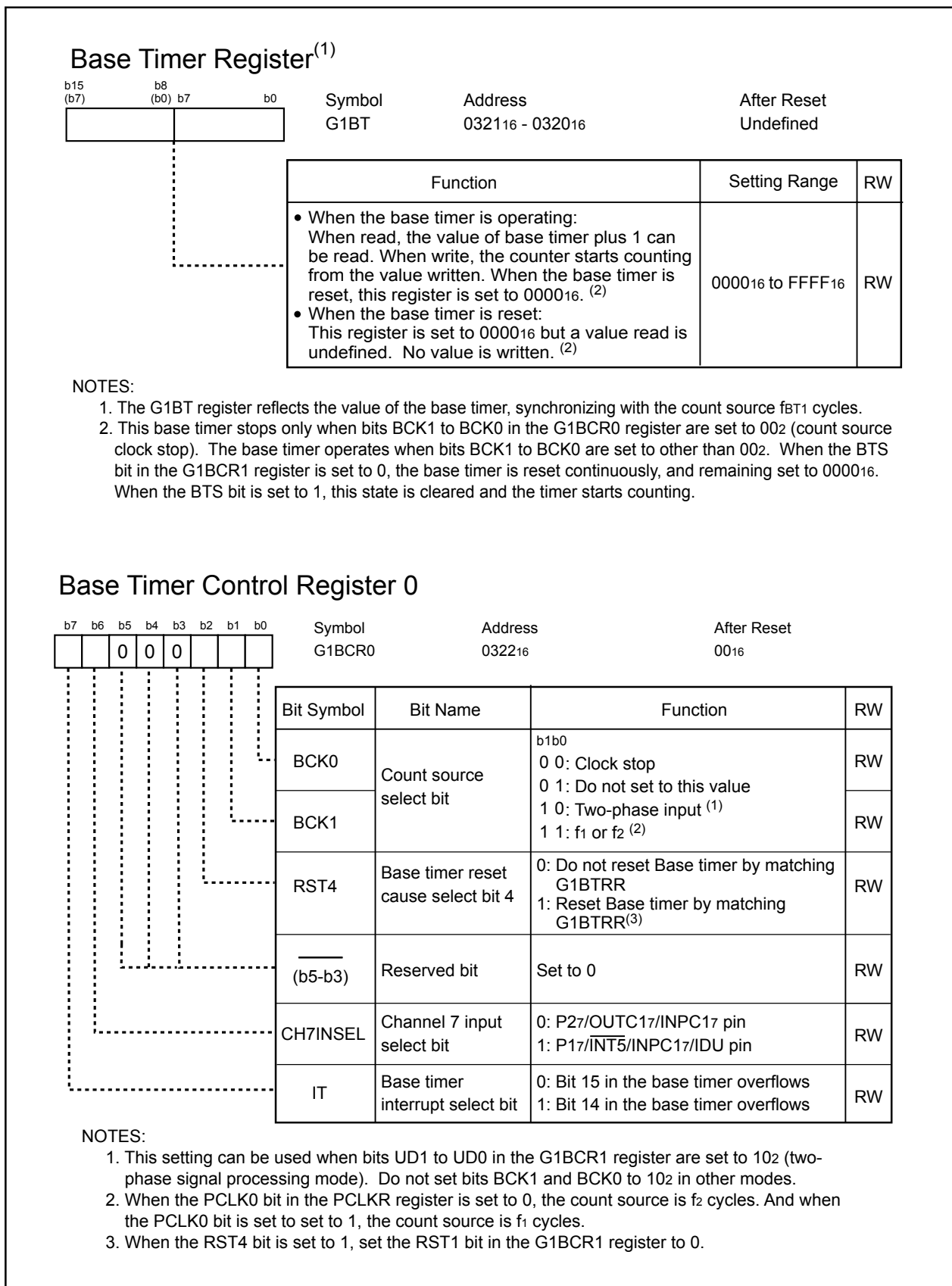


Figure 13.2 G1BT and G1BCR0 Registers

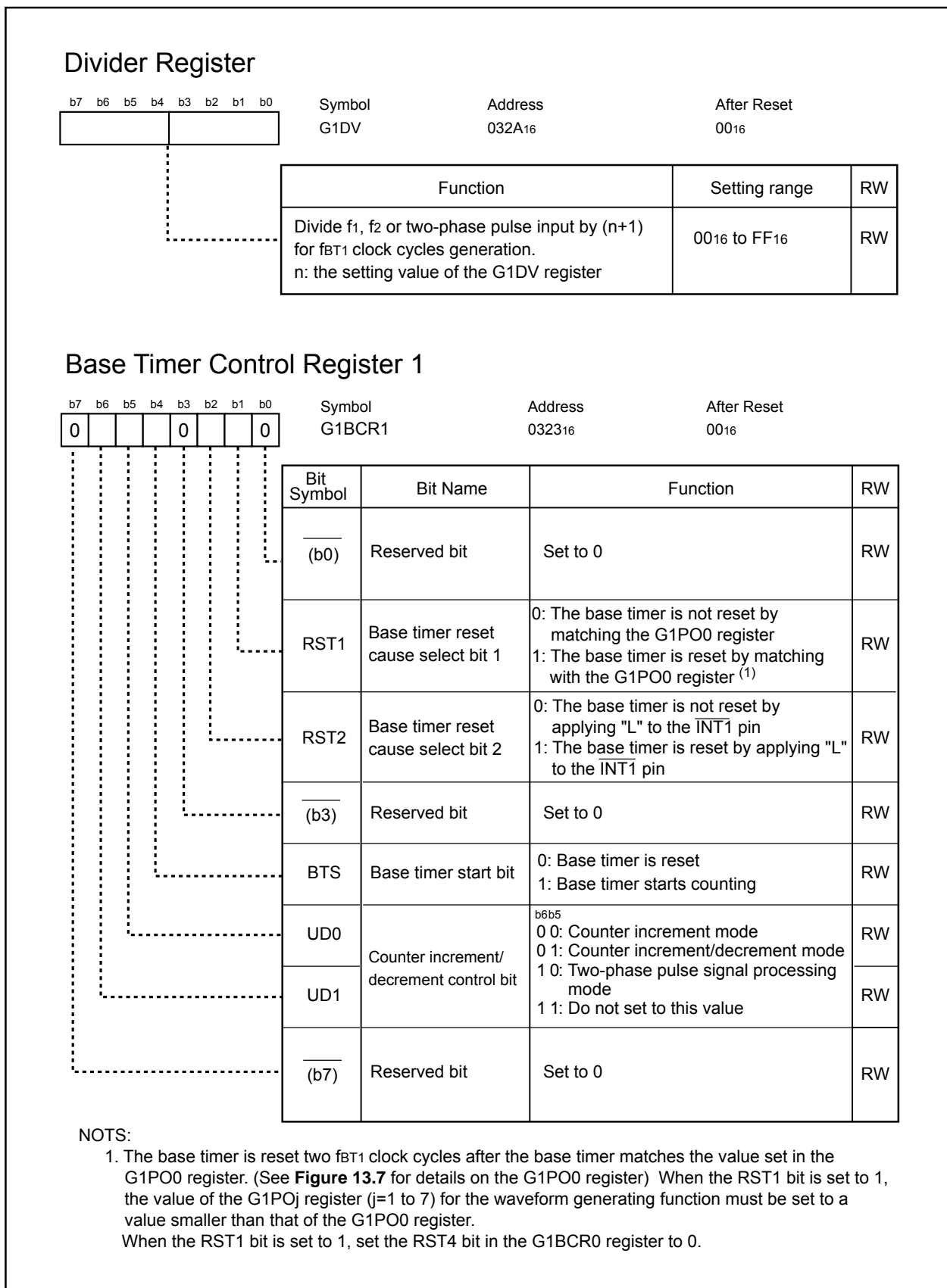


Figure 13.3 G1DV Register and G1BCR1 Register

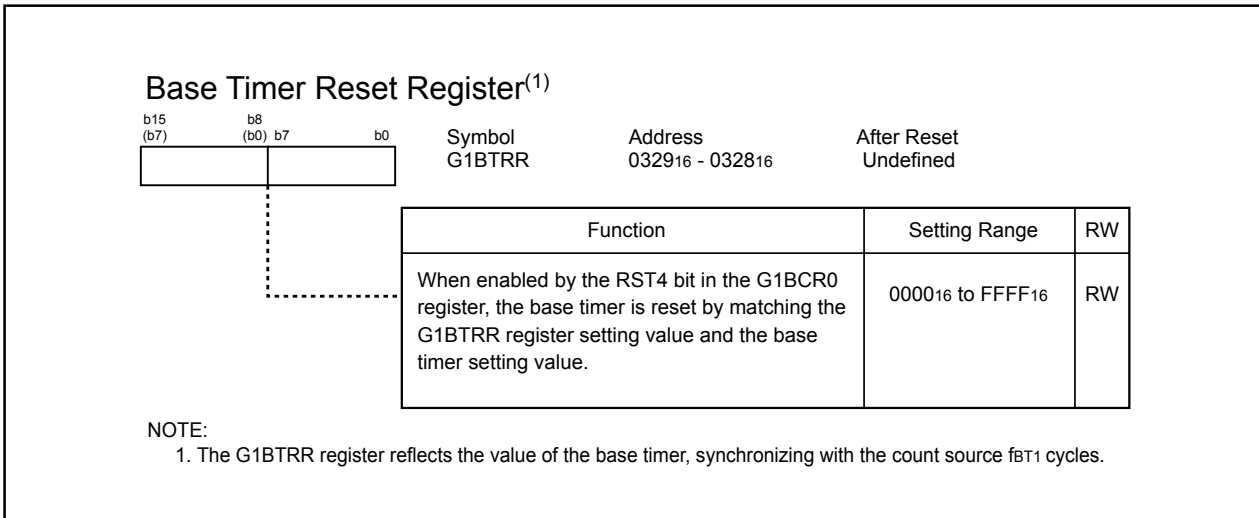


Figure 13.4 G1BTRR Register

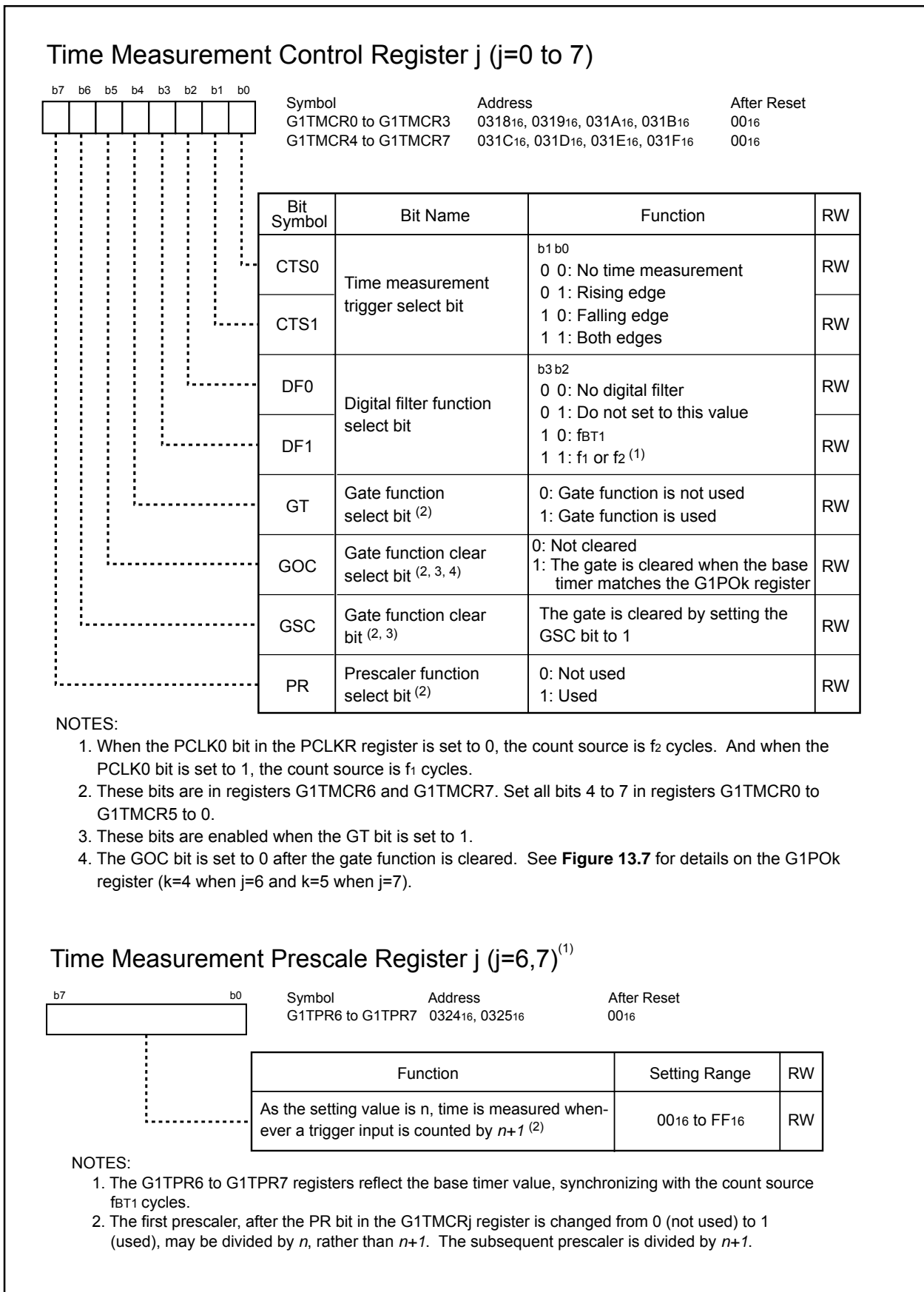


Figure 13.5 G1TMCR0 to G1TMCR7 Registers, and G1TPR6 to G1TPR7 Registers

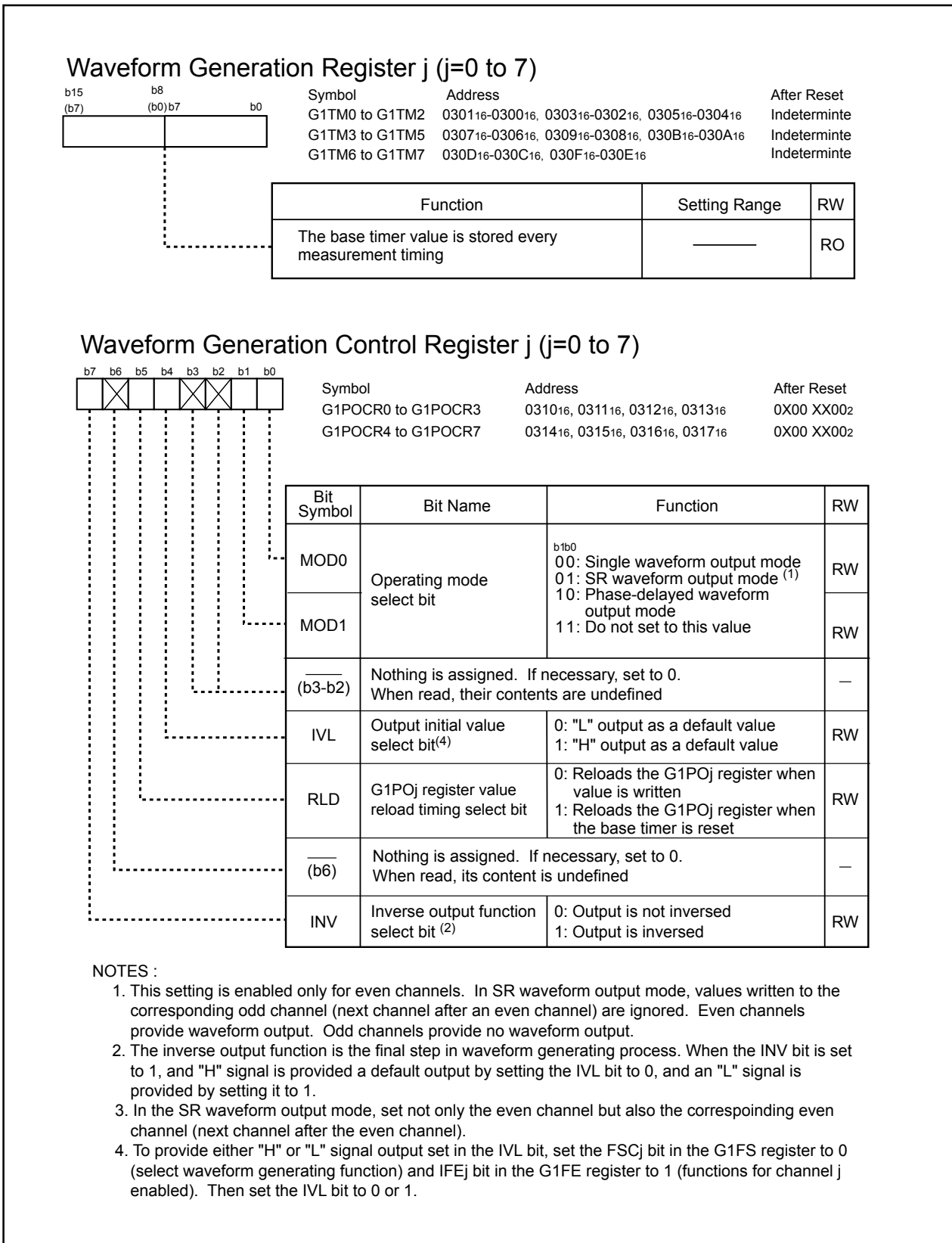


Figure 13.6 G1TM0 to G1TM7 Registers, and G1POCR0 to G1POCR7 Registers

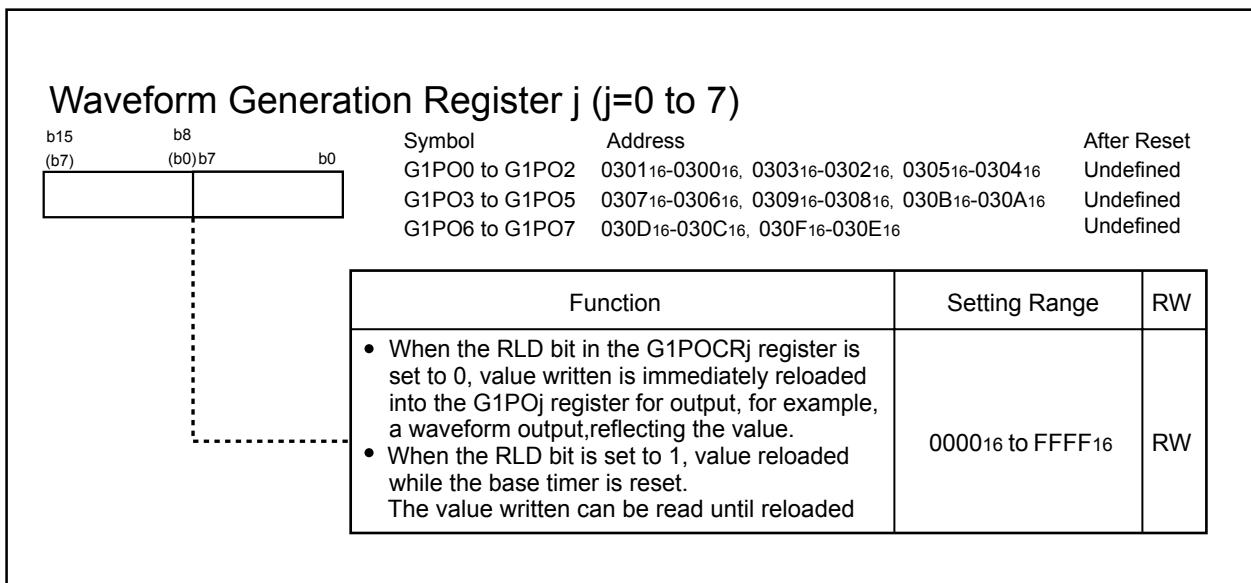
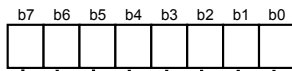


Figure 13.7 G1PO0 to G1PO7 Registers

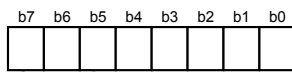
Function Select Register



Symbol: G1FS
 Address: 0327₁₆
 After Reset: 00₁₆

| Bit Symbol | Bit Name | Function | RW |
|------------|--|---|----|
| FSC0 | Channel 0 time measurement/waveform generating function select bit | 0: Select the waveform generating function 1: Select the time measurement function | RW |
| FSC1 | Channel 1 Time Measurement/Waveform Generating Function Select Bit | | RW |
| FSC2 | Channel 2 time measurement/waveform generating function select bit | | RW |
| FSC3 | Channel 3 time measurement/waveform generating function select bit | | RW |
| FSC4 | Channel 4 time measurement/waveform generating function select bit | | RW |
| FSC5 | Channel 5 time measurement/waveform generating function select bit | | RW |
| FSC6 | Channel 6 time measurement/waveform generating function select bit | | RW |
| FSC7 | Channel 7 time measurement/waveform generating function select bit | | RW |

Function Enable Register⁽¹⁾



Symbol: G1FE
 Address: 0326₁₆
 After Reset: 00₁₆

| Bit Symbol | Bit Name | Function | RW |
|------------|-------------------------------|--|----|
| IFE0 | Channel 0 function enable bit | 0 : Disable functions for channel j ⁽²⁾ 1 : Enable functions for channel j (j=0 to 7) | RW |
| IFE1 | Channel 1 function enable bit | | RW |
| IFE2 | Channel 2 function enable bit | | RW |
| IFE3 | Channel 3 function enable bit | | RW |
| IFE4 | Channel 4 function enable bit | | RW |
| IFE5 | Channel 5 function enable bit | | RW |
| IFE6 | Channel 6 function enable bit | | RW |
| IFE7 | Channel 7 function enable bit | | RW |

NOTES:

1. The G1FE register reflects the base timer value, synchronizing with the count source fBT1 cycles.
2. When functions for the channel j are disabled, each pin functions as an I/O port.

Figure 13.8 G1FS and G1FE Registers

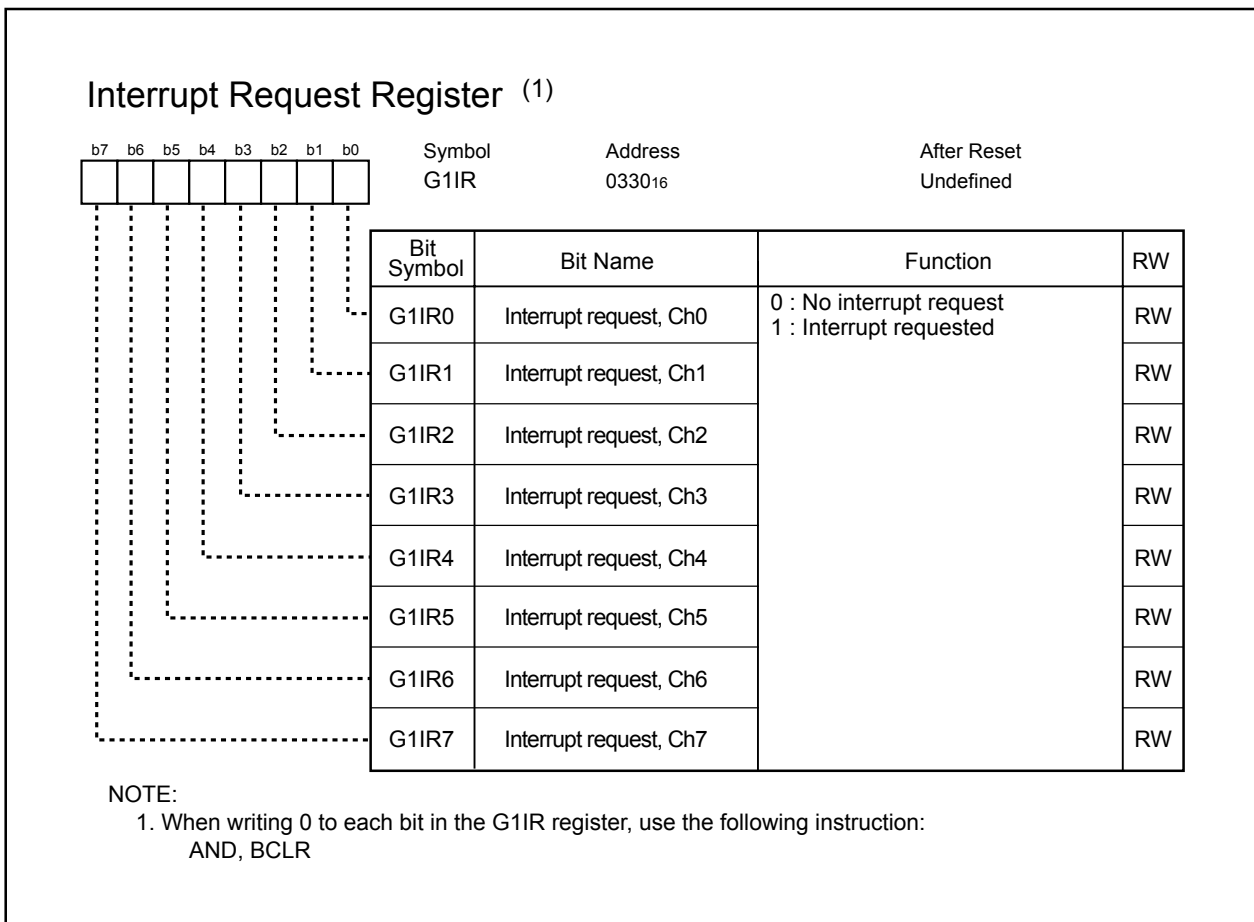


Figure 13.9 G1IR Register

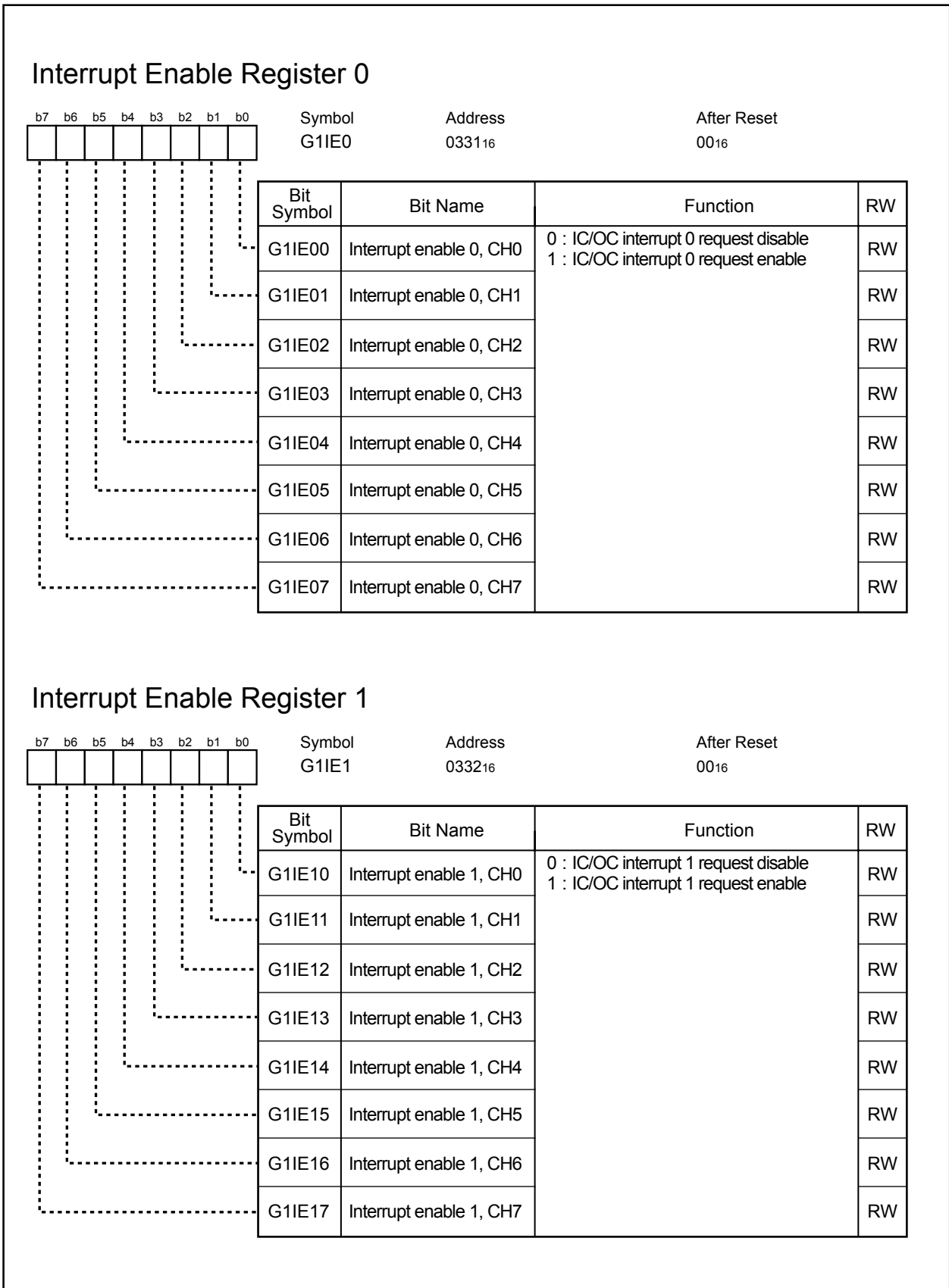


Figure 13.10 G1IE0 and G1IE1 Registers

13.1 Base Timer

The base timer is a free-running counter that counts an internally generated count source.

Table 13.2 lists specifications of the base timer. **Table 13.3** shows registers associated with the base timer.

Figure 13.11 shows a block diagram of the base timer. **Figure 13.12** shows an example of the base timer in counter increment mode. **Figure 13.13** shows an example of the base timer in counter increment/decrement mode. **Figure 13.14** shows an example of two-phase pulse signal processing mode.

Table 13.2 Base Timer Specifications

| Item | Specification |
|----------------------------|---|
| Count source(fBT1) | f1 or f2 divided by $(n+1)$, two-phase pulse input divided by $(n+1)$ n: determined by the DIV7 to DIV0 bits in the G1DV register. n=0 to 255 However, no division when n=0 |
| Counting operation | The base timer increments the counter value The base timer increments/decrements the counter value Two-phase pulse signal processing |
| Count start condition | The BTS bit in the G1BCR1 register is set to 1 (base timer starts counting) |
| Count stop condition | The BTS bit in the G1BCR1 register is set to 0 (base timer reset) |
| Base timer reset condition | (1) The value of the base timer matches the value of the G1BTRR register (2) The value of the base timer matches the value of G1PO0 register. (3) Apply a low-level signal ("L") to external interrupt pin, INT1 pin |
| Value for base timer reset | 0000 ₁₆ |
| Interrupt request | The base timer interrupt request is generated: (1) When the bit 14 or bit 15 in the base timer overflows (2) The value of the base timer value matches the value of the base timer reset register |
| Read from timer | • The G1BT register indicates a counter value while the base timer is running • The G1BT register is undefined when the base timer is reset |
| Write to timer | When a value is written while the base timer is running, the timer counter immediately starts counting from this value. No value can be written while the base timer is reset. |
| Selectable function | <ul style="list-style-type: none"> Counter increment/decrement mode The base timer starts counting from 0000₁₆. After incrementing to FFFF₁₆, the timer counter is then decremented back to 0000₁₆. The base timer increments the counter value again when the timer counter reaches 0000₁₆. (See Figure 13.13) Two-phase pulse processing mode Two-phase pulse signals from pins P80 and P81 are counted (See Figure 13.14) <div style="text-align: center;"> <p>The timer increments a counter on all edges The timer decrements a counter on all edges</p> </div> |

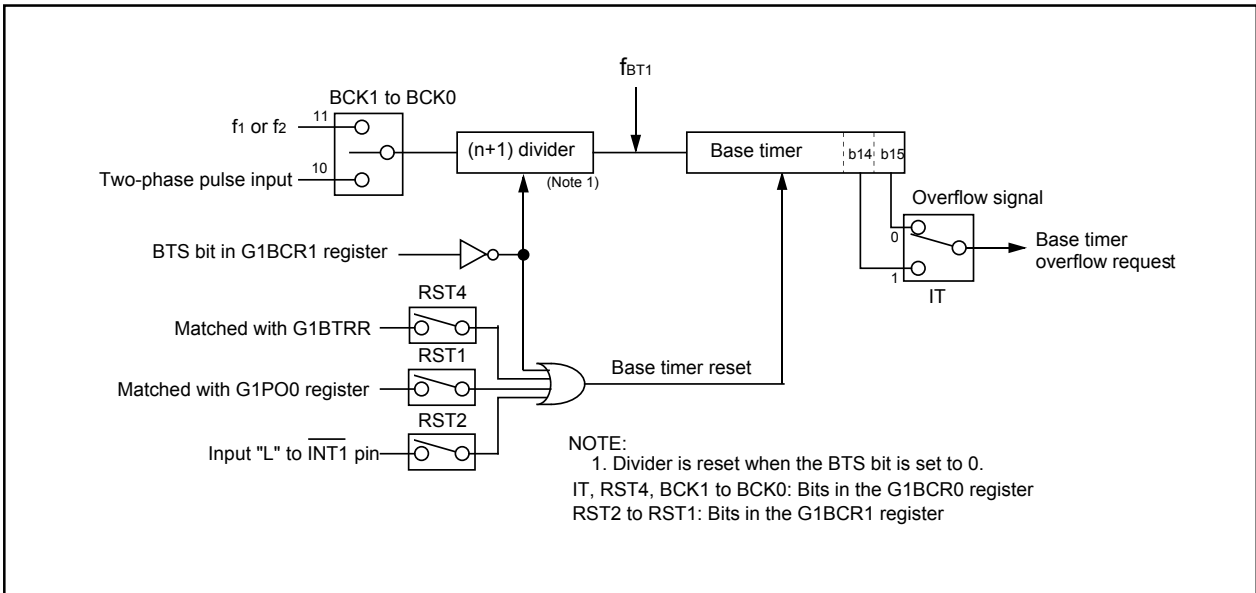


Figure 13.11 Base Timer Block Diagram

Table 13.3 Base Timer Associated Register Settings (Time Measurement Function, Waveform Generation Function, Communication Function)

| Register | Bit | Function |
|----------|--------------|--------------------------------|
| G1BCR0 | BCK1 to BCK0 | Select a count source |
| | RST4 | Select base timer reset timing |
| | IT | Select the base timer overflow |
| G1BCR1 | RST2 to RST1 | Select base timer reset timing |
| | BTS | Used to start the base timer |
| | UD1 to UD0 | Select how to count |
| G1BT | - | Read or write base timer value |
| G1DV | - | Divide ratio of a count source |

Set the following registers to set the RST1 bit to 1 (base timer reset by matching the base timer with the G1PO0 register)

| | | |
|---------|--------------|--|
| G1POCR0 | MOD1 to MOD0 | Set to 002 (single-phase waveform output mode) |
| G1PO0 | - | Set reset cycle |
| G1FS | FSC0 | Set to 0 (waveform generating function) |
| G1FE | IFE0 | Set to 1 (channel operation start) |

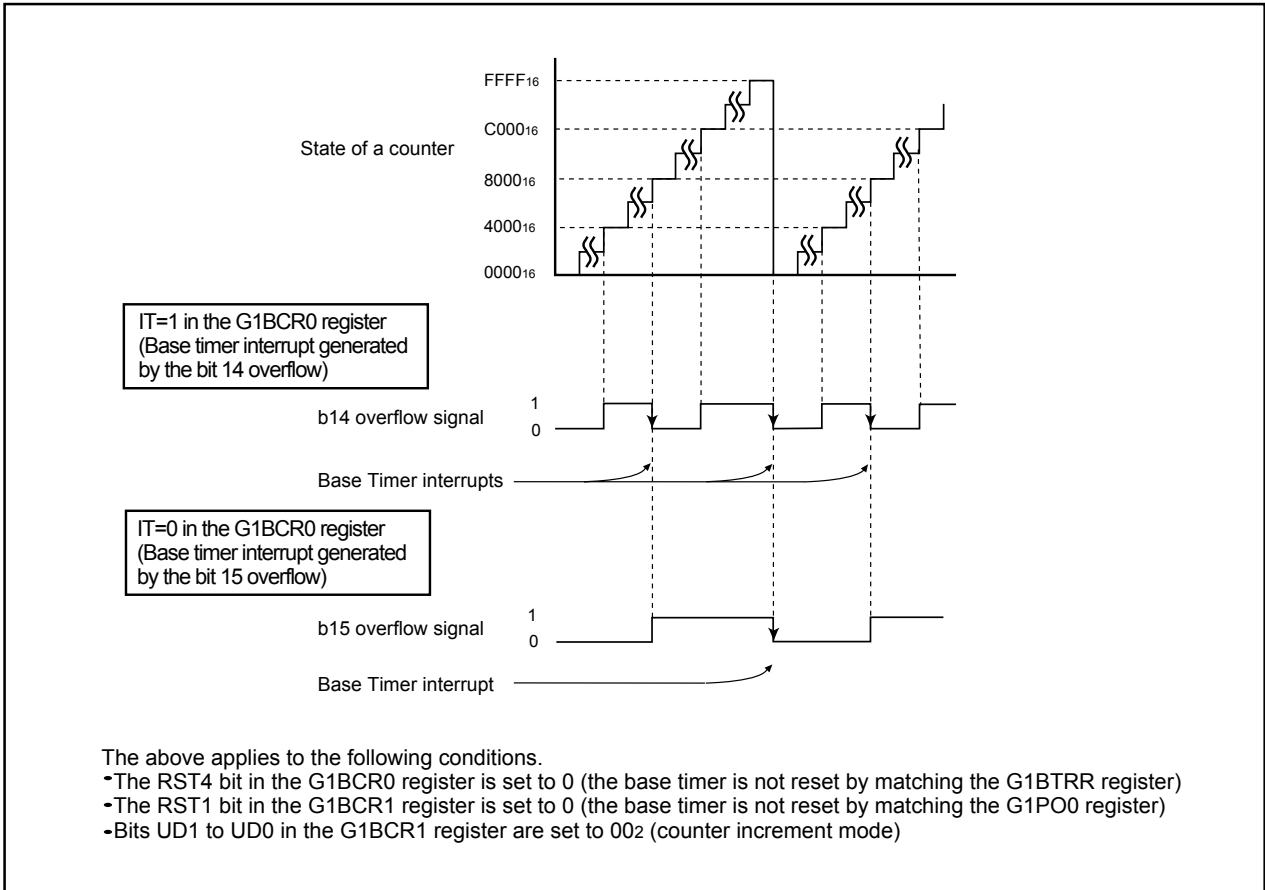


Figure 13.12 Counter Increment Mode

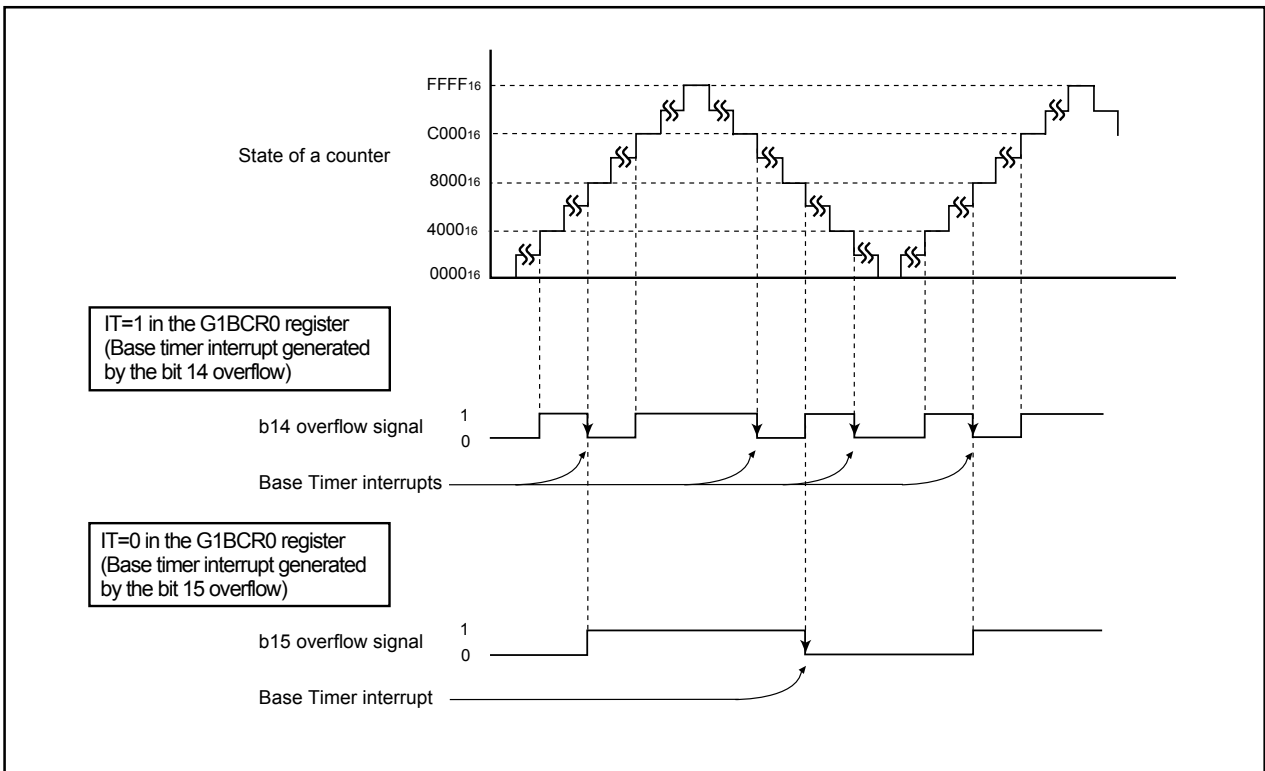


Figure 13.13 Counter Increment/Decrement Mode

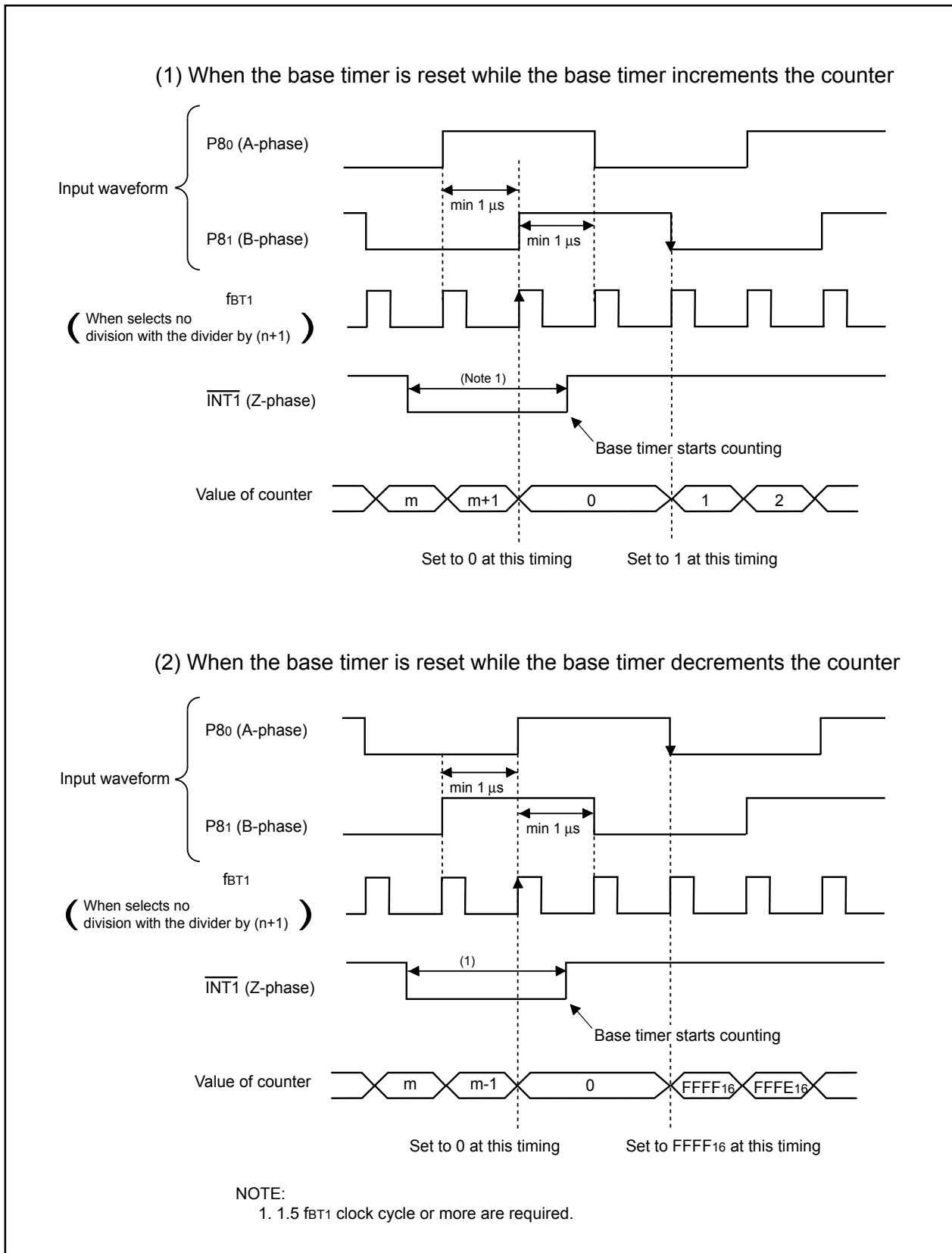


Figure 13.14 Base Timer Operation in Two-phase Pulse Signal Processing Mode

13.1.1 Base Timer Reset Register(G1BTRR)

The G1BTRR register provides the capability to reset the base timer when the base timer count value matches the value stored in the G1BTRR register. The G1BTRR register is enabled by the RST4 bit in the G1BCR0 register. This function is identical in operation to the G1PO0 base timer reset that is enabled by the RST1 bit in the G1BCR0 register. If the free-running operation is not selected, the channel 0 can be used for a waveform generation when the base timer is reset by the G1BTRR register. Do not enable bits RST1 and RST4 simultaneously.

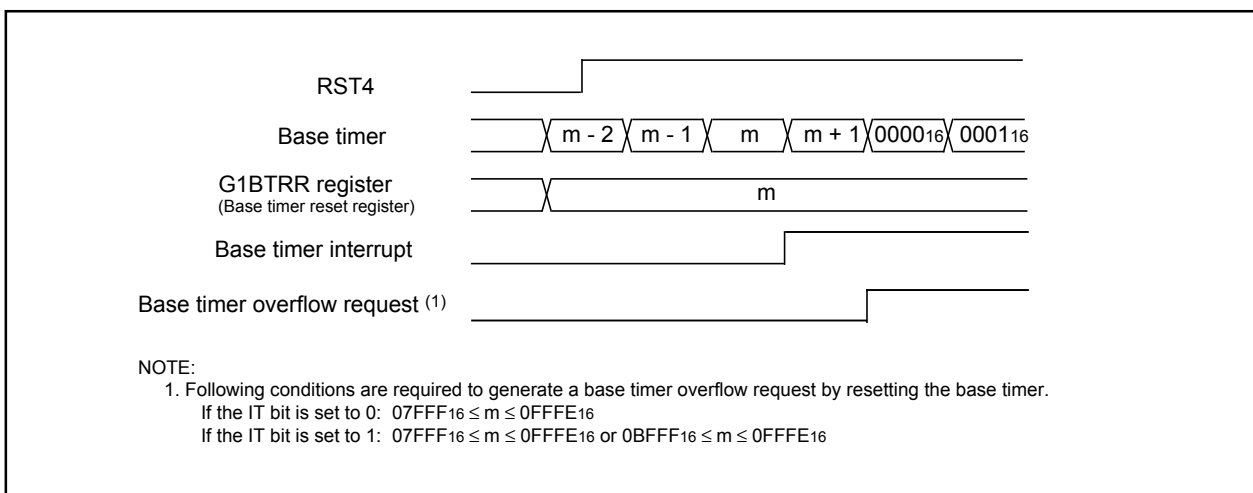


Figure 13.15 Base Timer Reset operation by Base Timer Reset Register

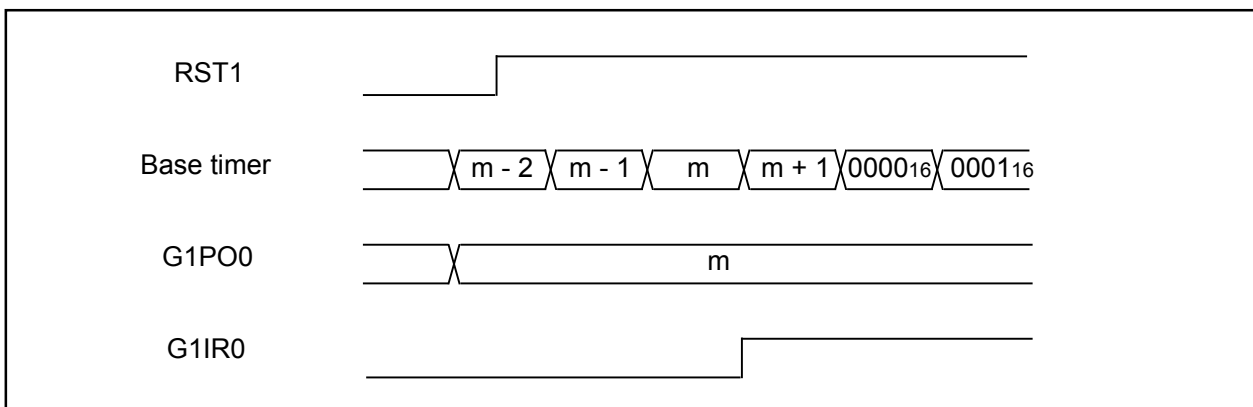


Figure 13.16 Base Timer Reset operation by G1PO0 register

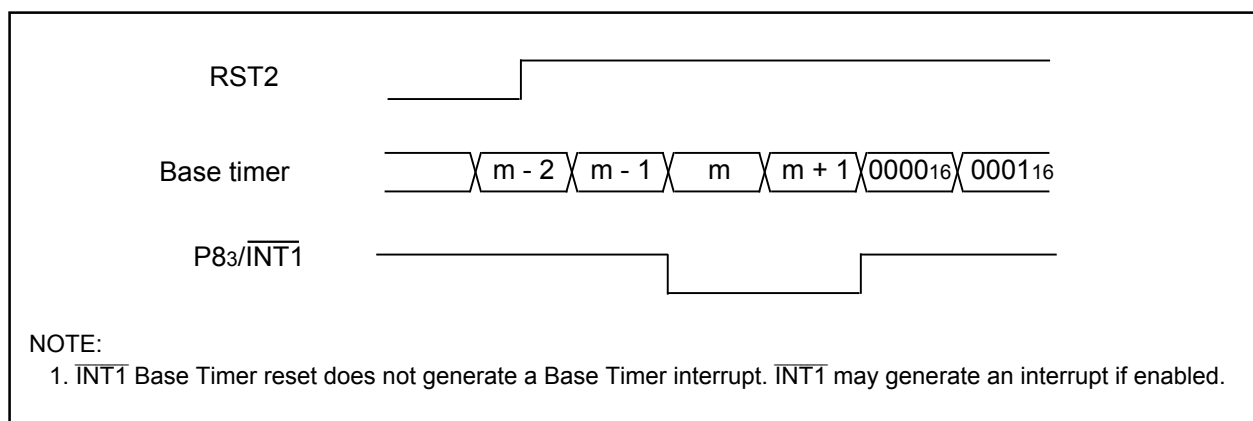


Figure 13.17 Base Timer Reset operation by INT1

13.2 Interrupt Operation

The IC/OC interrupt contains several request causes. **Figure 13.18** shows the IC/OC interrupt block diagram and **Table 13.4** shows the IC/OC interrupt assignment.

When either the base timer reset request or base timer overflow request is generated, the IR bit in the BTIC register corresponding to the IC/OC base timer interrupt is set to 1 (with an interrupt request). Also when an interrupt request in each eight channels (channel *i*) is generated, the bit *i* in the G1IR register is set to 1 (with an interrupt request). At this time, if the bit *i* in the G1IE0 register is 1 (IC/OC interrupt 0 request enabled), the IR bit in the ICOC0IC register corresponding to the IC/OC interrupt 0 is set to 1 (with an interrupt request). And if the bit *i* in the G1IE1 register is 1 (IC/OC interrupt 1 request enabled), the IR bit in the ICOC1IC register corresponding to the IC/OC interrupt 1 is set to 1 (with an interrupt request).

Additionally, because each bit in the G1IR register is not automatically set to 0 even if the interrupt is acknowledged, set to 0 by program. If these bits are left as 1, all IC/OC channel interrupt causes, which are generated after setting the IR bit to 1, will be disabled.

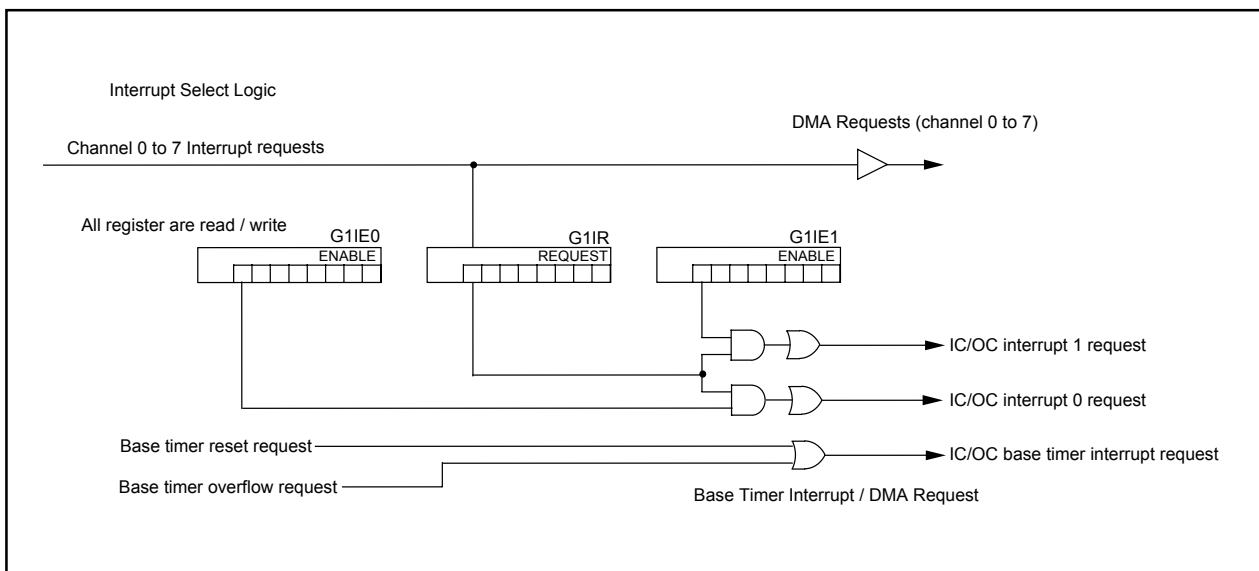


Figure 13.18 IC/OC Interrupt and DMA request generation

Table 13.4 Interrupt Assignment

| Interrupt | Interrupt control register |
|----------------------------|------------------------------|
| IC/OC base timer interrupt | BTIC(0047 ₁₆) |
| IC/OC interrupt 0 | ICOC0IC(0045 ₁₆) |
| IC/OC interrupt 1 | ICOC1IC(0046 ₁₆) |

13.3 DMA Support

Each of the interrupt sources - the eight IC/OC channel interrupts and the one Base Timer interrupt - are capable of generating a DMA request.

13.4 Time Measurement Function

In synchronization with an external trigger input, the value of the base timer is stored into the G1TMj register (j=0 to 7). **Table 13.5** shows specifications of the time measurement function. **Table 13.6** shows register settings associated with the time measurement function. **Figures 13.19** and **13.20** display operational timing of the time measurement function. **Figure 13.21** shows operational timing of the prescaler function and the gate function.

Table 13.5 Time Measurement Function Specifications

| Item | Specification |
|-------------------------------------|--|
| Measurement channel | Channels 0 to 7 |
| Selecting trigger input polarity | Rising edge, falling edge, both edges of the INPC1j pin ⁽¹⁾ |
| Measurement start condition | The IFEj bit in the G1FE register should be set to 1 (channels j function enabled) when the FSCj bit (j=0 to 7) in the G1FS register is set to 1 (time measurement function selected). |
| Measurement stop condition | The IFEj bit should be set to 0 (channel j function disabled) |
| Time measurement timing | <ul style="list-style-type: none"> •No prescaler: every time a trigger signal is applied •Prescaler (for channel 6 and channel 7): every <i>G1TPR_k (k=6,7) register value +1</i> times a trigger signal is applied |
| Interrupt request generation timing | The G1IRi bit (i=0 to 7) in the interrupt request register (See Figure 13.9) is set to 1 at time measurement timing |
| INPC1j pin function ⁽¹⁾ | Trigger input pin |
| Selectable function | <ul style="list-style-type: none"> • Digital filter function The digital filter samples a trigger input signal level every f₁, f₂ or f_{BT1} cycles and passes pulse signal matching trigger input signal level three times • Prescaler function (for channel 6 and channel 7) Time measurement is executed every <i>G1TPR_k register value +1</i> times a trigger signal is applied • Gate function (for channel 6 and channel 7) After time measurement by the first trigger input, trigger input cannot be accepted. However, while the GOC bit in the G1TMCRk register is set to 1 (gate cleared by matching the base timer with the G1POp register (p=4 when k=6, p=5 when k=7)), trigger input can be accepted again by matching the base timer value with the G1POp register setting • Digital Debounce function (for channel7) See 13.6.2 Digital Debounce Function for P17/INT5/INPC17 and 19.6 Digital Debounce Function for details |

NOTE:

1. The INPC10 to INPC17 pins

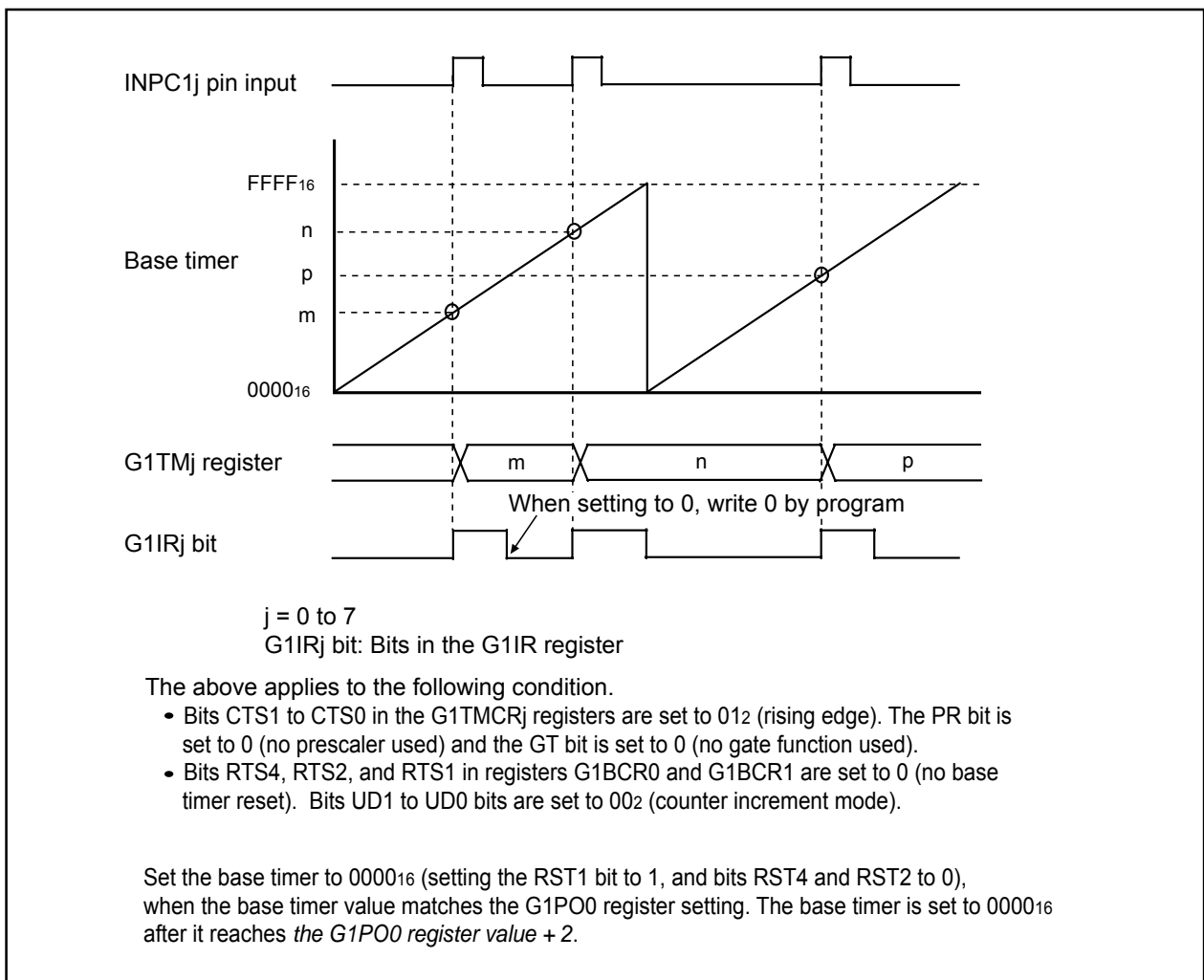
Table 13.6 Register Settings Associated with the Time Measurement Function

| Register | Bit | Function |
|----------|--------------|---------------------------------------|
| G1TMCRj | CTS1 to CTS0 | Select time measurement trigger |
| | DF1 to DF0 | Select the digital filter function |
| | GT, GOC, GSC | Select the gate function |
| | PR | Select the prescaler function |
| G1TPRK | - | Setting value of prescaler |
| G1FS | FSCj | Set to 1 (time measurement function) |
| G1FE | IFEj | Set to 1 (channel j function enabled) |

$j = 0$ to 7 $k = 6, 7$

Bit configurations and function varies with channels used.

Registers associated with the time measurement function must be set after setting registers associated with the base timer.

**Figure 13.19 Time Measurement Function (1)**

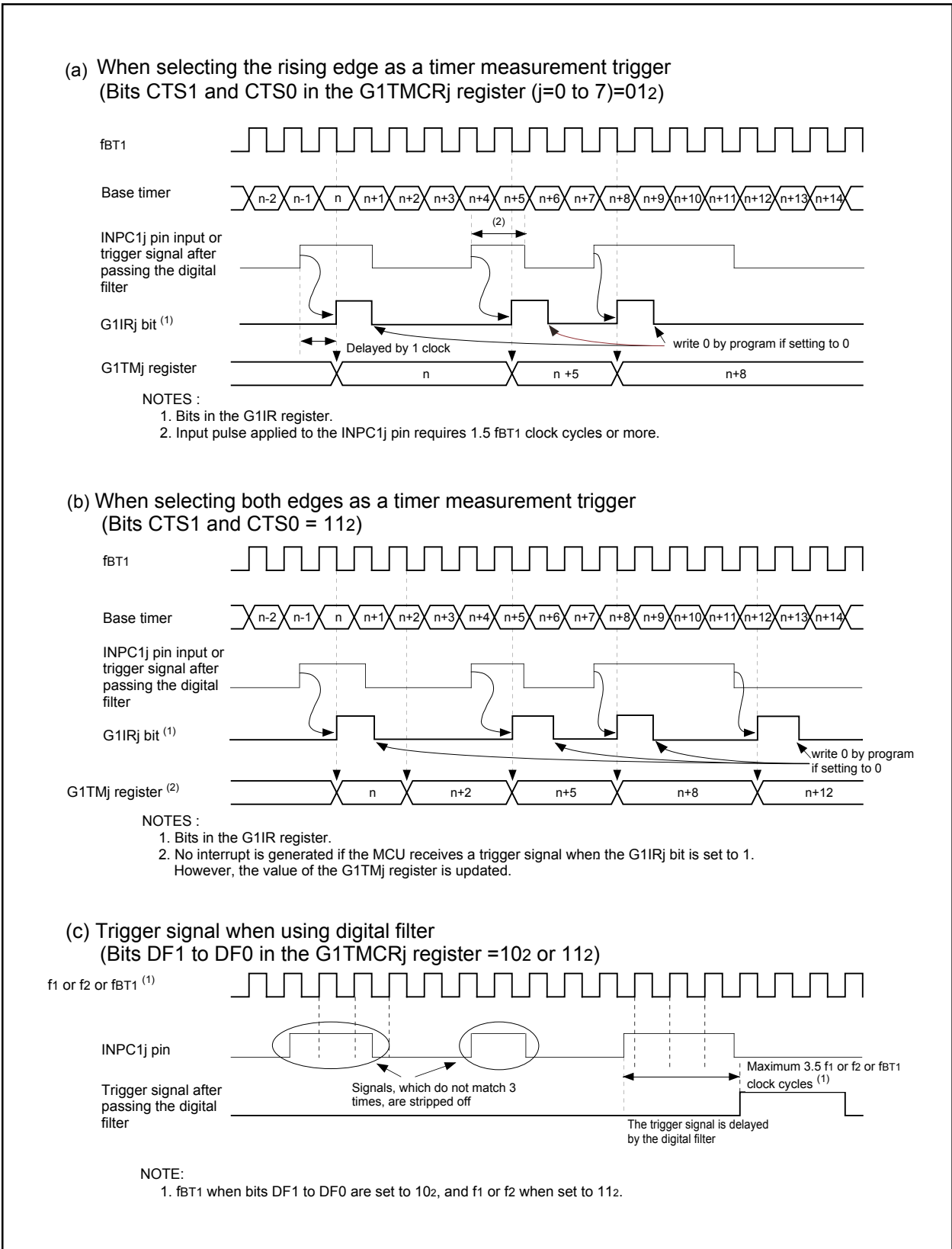


Figure 13.20 Time Measurement Function (2)

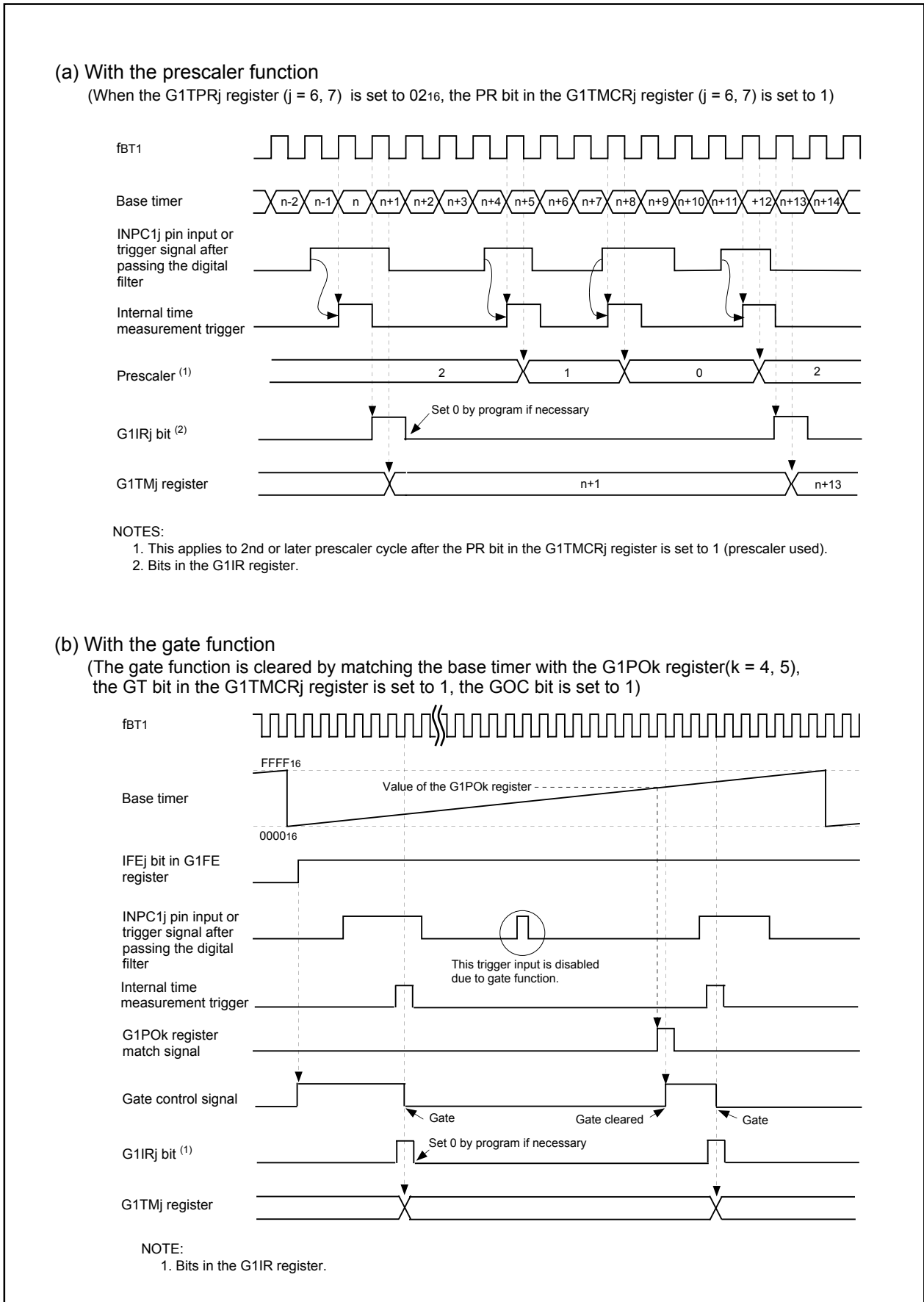


Figure 13.21 Prescaler Function and Gate Function

13.5 Waveform Generating Function

Waveforms are generated when the base timer value matches the G1POj (j=0 to 7) register value.

The waveform generating function has the following three modes :

- Single-phase waveform output mode
- Phase-delayed waveform output mode
- Set/Reset waveform output (SR waveform output) mode

Table 13.7 lists registers associated with the waveform generating function.

Table 13.7 Registers Related to the Waveform Generating Function Settings

| Register | Bit | Function |
|----------|--------------|---|
| G1POCRj | MOD1 to MOD0 | Select output waveform mode |
| | IVL | Select default value |
| | RLD | Select G1POj register value reload timing |
| | INV | Select inverse output |
| G1POj | - | Select timing to output waveform inverted |
| G1FS | FSCj | Set to 0 (waveform generating function) |
| G1FE | IFEj | Set to 1 (enables function on channel j) |

j = 0 to 7

Bit configurations and functions vary with channels used.

Registers associated with the waveform generating function must be set after setting registers associated with the base timer.

13.5.1 Single-Phase Waveform Output Mode

Output signal level of the OUTC1j pin becomes high ("H") when the INV bit in the G1POCRj (j=0 to 7) register is set to 0(output is not reversed) and the base timer value matches the G1POj (j=0 to 7) register value. The "H" signal switches to a low-level ("L") signal when the base timer reaches 0000₁₆. **Table 13.8** lists specifications of single-phase waveform mode. **Figure 13.22** lists an example of single-phase waveform mode operation.

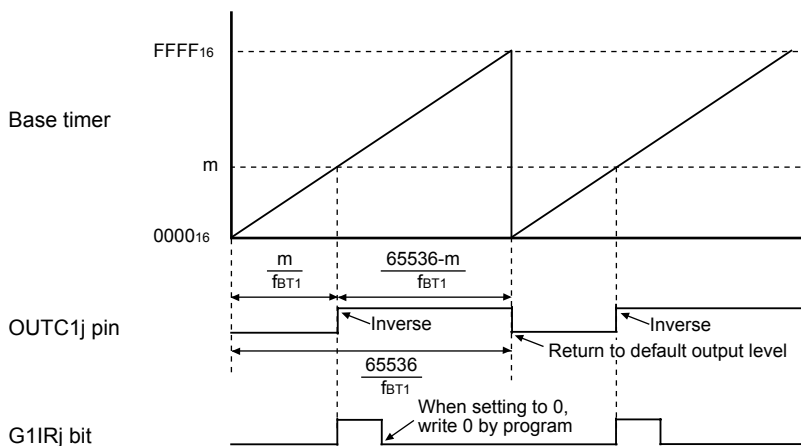
Table 13.8 Single-phase Waveform Output Mode Specifications

| Item | Specification |
|---------------------------------|---|
| Output waveform | <ul style="list-style-type: none"> Free-running operation (bits RST1, RST2, and RST4 of registers G1BCR1 and G1BCR0 are set to 0 (no reset)) <p>Cycle : $\frac{65536}{f_{BT1}}$</p> <p>Default output level width : $\frac{m}{f_{BT1}}$</p> <p>Inverse level width : $\frac{65536-m}{f_{BT1}}$</p> <ul style="list-style-type: none"> The base timer is cleared to 0000₁₆ by matching the base timer with either following register <p>(a) G1PO0 register (enabled by setting RST1 bit to 1, and RST4 and RST2 bits to 0), or</p> <p>(b) G1BTRR register (enabled by setting RST4 bit to 1, and RST2 and RST1 bits to 0)</p> <p>Cycle : $\frac{n+2}{f_{BT1}}$</p> <p>Default output level width : $\frac{m}{f_{BT1}}$</p> <p>Inverse level width : $\frac{n+2-m}{f_{BT1}}$</p> <p>m : setting value of the G1POj register (j=0 to 7), 0001₁₆ to FFFD₁₆ n : setting value of the G1PO0 register or the G1BTRR register, 0001₁₆ to FFFD₁₆ </p> |
| Waveform output start condition | The IFEj bit in the G1FE register is set to 1 (channel j function enabled) |
| Waveform output stop condition | The IFEj bit is set to 0 (channel j function disabled) |
| Interrupt request | The G1IRj bit in the G1IR register is set to 1 when the base timer value matches the G1POj register value (See Figure 13.22) |
| OUTC1j pin ⁽¹⁾ | Pulse signal output pin |
| Selectable function | <ul style="list-style-type: none"> Default value set function: Set starting waveform output level Inverse output function: Waveform output signal is inverted and provided from the OUTC1j pin |

NOTE:

1. Pins OUTC10 to OUTC17.

(1) Free-running operation
 (The RST4, RST2, and RST1 bits in the G1BCR0 and G1BCR1 registers are set to 0)

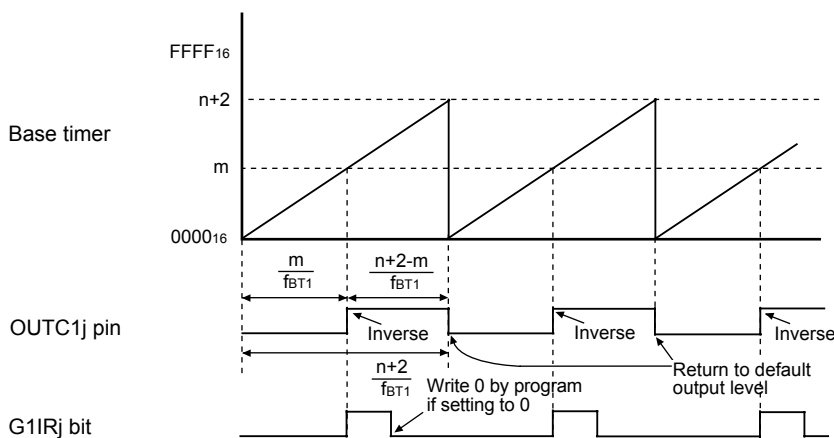


$j=0$ to 7
 m : Setting value of the G1POj register
 G1IRj bit : Bits in the G1IR register

The above applies under the following conditions.

- The IVL bit in the G1POCRj register is set to 0 ("L" output as a default value) and the INV bit is set to 0 (not inverted).
- Bits UD1 to UD0 are set to 00z (counter increment mode).

(2) The base timer is reset when the base timer matches either following register
 (a) G1PO0 (enabled by setting bit RST1 to 1, and bits RST4 and RST2 to 0), or
 (b) G1BTRR (enabled by setting bit RST4 to 1, and bits RST2 and RST1 to 0)



$j=1$ to 7
 m : Setting value of the G1POj register
 n : Setting value of either G1PO0 register or G1BTRR register
 G1IRj bit : Bits in the G1IR register

The above applies under the following conditions.

- The IVL bit in the G1POCRj register is set to 0 ("L" output as a default value) and the INV bit is set to 0 (not inverted).
- Bits UD1 to UD0 are set to 00z (counter increment mode).

Figure 13.22 Single-phase Waveform Output Mode

13.5.2 Phase-Delayed Waveform Output Mode

Output signal level of the OUTC1j pin is inversed every time the base timer value matches the G1POj register value (j=0 to 7). **Table 13.9** lists specifications of phase-delayed waveform mode. **Figure 13.23** shows an example of phase-delayed waveform mode operation.

Table 13.9 Phase-delayed Waveform Output Mode Specifications

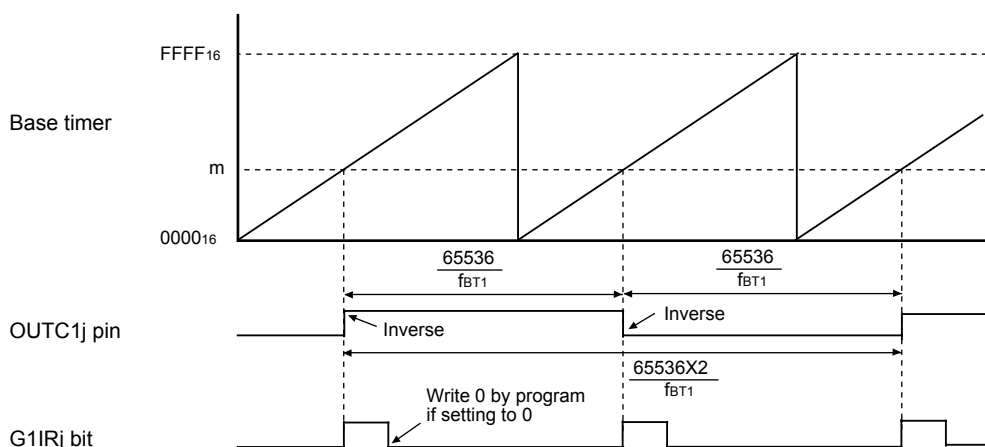
| Item | Specification |
|---------------------------------|--|
| Output waveform | <ul style="list-style-type: none"> Free-running operation (bits RST1, RST2, and RST4 in registers G1BCR1 and G1BCR0 are set to 0 (no reset)) <p>Cycle : $\frac{65536 \times 2}{f_{BT1}}$</p> <p>"H" and "L" width : $\frac{65536}{f_{BT1}}$</p> <ul style="list-style-type: none"> The base timer is cleared to 0000₁₆ by matching the base timer with either following register <p>(a) G1PO0 register (enabled by setting RST1 bit to 1, and bits RST4 and RST2 to 0), or</p> <p>(b) G1BTRR register (enabled by setting RST4 bit to 1, and bits RST2 and RST1 to 0)</p> <p>Cycle : $\frac{2(n+2)}{f_{BT1}}$</p> <p>"H" and "L" width : $\frac{n+2}{f_{BT1}}$</p> <p>n : setting value of either G1PO0 register or G1BTRR register</p> |
| Waveform output start condition | The IFEj bit in the G1FE register is set to 1 (channel j function enabled) |
| Waveform output stop condition | The IFEj bit is set to 0 (channel j function disabled) |
| Interrupt request | The G1IRj bit in the interrupt request register is set to 1 when the base timer value matches the G1POj register value. (See Figure 13.23) |
| OUTC1j pin ⁽¹⁾ | Pulse signal output pin |
| Selectable function | <ul style="list-style-type: none"> Default value set function: Set starting waveform output level Inverse output function : Waveform output signal is inversed and provided from the OUTC1j pin |

NOTE:

1. Pins OUTC10 to OUTC17.

(1) Free-running operation

(Bits RST4, RST2, and RST1 in the registers G1BCR0 and G1BCR1 are set to 0)



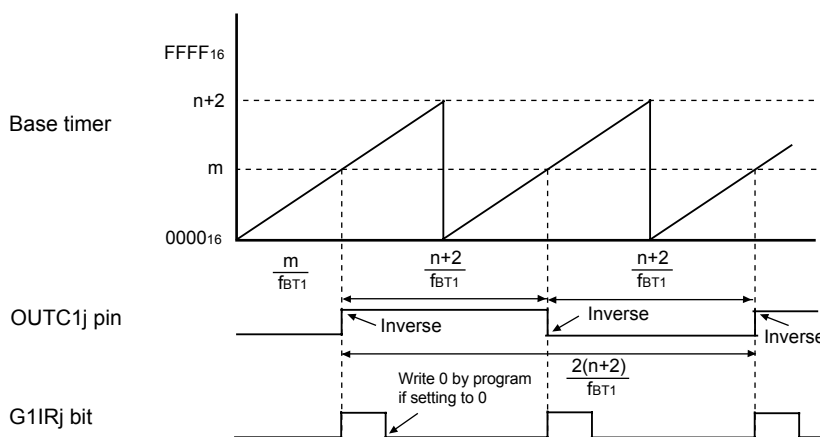
j=0 to 7
 m : Setting value of the G1POj register
 G1IRj bit : Bits in the G1IR register

The above applies under the following conditions.

- The IVL bit in the G1POCRj register is set to 0 (L output as a default value). The INV bit is set to 0 (not inverted).
- Bits UD1 to UD0 are set to 00z (counter increment mode).

(2) Base timer is reset when the base timer matches either following register

- (a) G1PO0 (enabled by setting bit RST1 to 1, and bits RST4 and RST2 to 0), or
- (b) G1BTRR (enabled by setting bit RST4 to 1, and bits RST2 and RST1 to 0)



j=1 to 7
 m : Setting value of the G1POj register n: Setting value of either register G1PO0 or G1BTRR
 G1IRj bit : Bits in the G1IR register

The above applies under the following conditions.

- The IVL bit in the G1POCRj register is set to 0 (L output as a default value). The INV bit is set to 0 (not inverted).
- Bits UD1 to UD0 are set to 00z (counter increment mode).

Figure 13.23 Phase-delayed Waveform Output Mode

13.5.3 Set/Reset Waveform Output (SR Waveform Output) Mode

Output signal level of the OUTC1j pin becomes high ("H") when the INV bit in the G1POCRi (i=0 to 7) is set to 0 (output is not reversed) and the base timer value matches the G1POj register value (j=0, 2, 4, 6). The "H" signal switches to a low-level ("L") signal when the base timer value matches the G1POk (k=j+1) register value. **Table 13.10** lists specifications of SR waveform mode. **Figure 13.24** shows an example of the SR waveform mode operation.

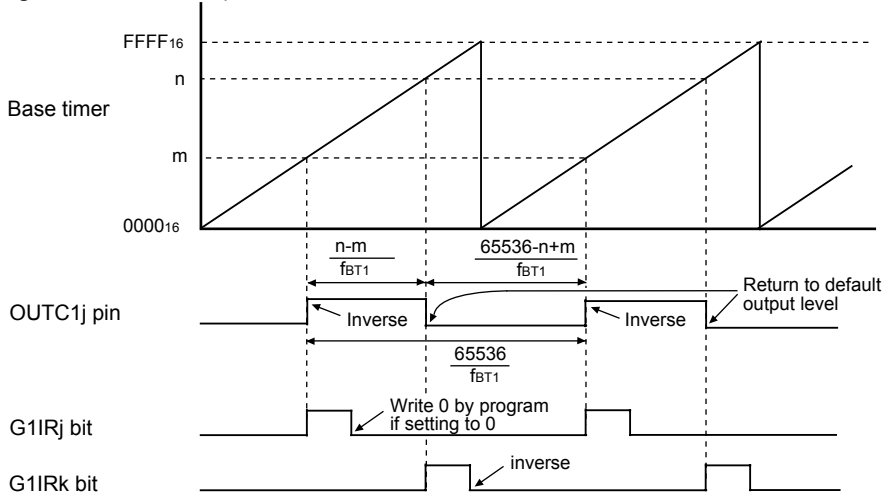
Table 13.10 SR Waveform Output Mode Specifications

| Item | Specification |
|---------------------------------|---|
| Output waveform | <ul style="list-style-type: none"> Free-running operation (the RST1, RTS2, and RST4 bits of the G1BCR1 and G1BCR0 registers are set to 0 (no reset)) <p>Cycle : $\frac{65536}{f_{BT1}}$</p> <p>Inverse level width⁽¹⁾ : $\frac{n-m}{f_{BT1}}$</p> <ul style="list-style-type: none"> The base timer is cleared to 0000₁₆ by matching the base timer with either following register (a) G1PO0 register (enabled by setting RST1 bit to 1, and bits RST4 and RST2 to 0)⁽²⁾, or (b) G1BTRR register (enabled by setting RST4 bit to 1, and bits RST2 and RST1 to 0) <p>Cycle : $\frac{p+2}{f_{BT1}}$</p> <p>Inverse level width⁽¹⁾ : $\frac{n-m}{f_{BT1}}$</p> <p>m : setting value of the G1POj register (j=0, 2, 4, 6)</p> <p>n : setting value of the G1POk register (k=j+1)</p> <p>p : setting value of the G1PO0 register or G1BTRR register</p> <p>value range of m, n, p: 0001₁₆ to FFFD₁₆</p> |
| Waveform output start condition | Bits IFEj and IFEk in the G1FE register is set to 1 (channel j function enabled) |
| Waveform output stop condition | Bits IFEj and IFEk are set to 0 (channel j function disabled) |
| Interrupt request | The G1IRj bit in the G1IR register is set to 1 when the base timer value matches the G1POj register value. The G1IRk bit in the interrupt request register is set to 1 when the base timer value matches the G1POk register value (See Figure 13.24) |
| OUTC1j pin ⁽³⁾ | Pulse signal output pin |
| Selectable function | <ul style="list-style-type: none"> Default value set function : Set starting waveform output level Inverse output function: Waveform output signal is inversed and provided from the OUTC1j pin |

NOTES:

- The odd channel's waveform generating register must have greater value than the even channel's.
- When the G1PO0 register resets the base timer, the channel 0 and channel 1 SR waveform generating functions are not available.
- Pins OUTC10, OUTC12, OUTC14, OUTC16.

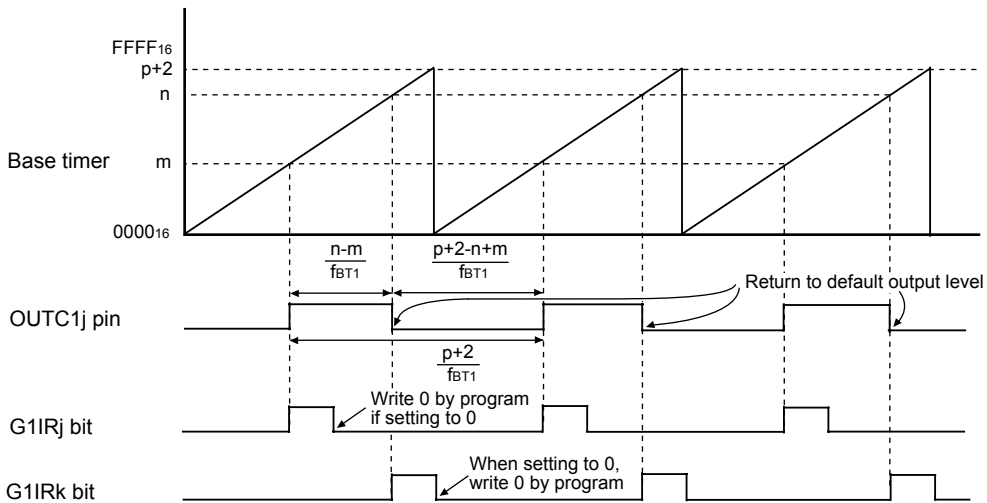
(1) Free-running operation
 (Bits RST2 and RST1 in the G1BCR0 register and the RST4 bit in the G1BCR1 register are set to 0)



$j=0, 2, 4, 6 \quad k=j+1$
 m : Setting value of the G1POj register n : Setting value of the G1POk register
 G1IRj, G1IRk bits: Bits in the G1IR register

- The above applies under the following conditions.
- The IVL bit in the G1POCRj register is set to 0 (L output as a default value). The INV bit is set to 0 (not inverted).
 - Bits UD1 and UD0 are set to 00₂ (counter increment mode).

(2) Base timer is reset when the base timer matches either following register
 (a) G1PO0 (enabled by setting bit RST1 to 1, and bits RST4 and RST2 to 0), or
 (b) G1BTRR (enabled by setting bit RST4 to 1, and bits RST2 and RST1 to 0)



$j=2, 4, 6 \quad k=j+1$
 m : Setting value of the G1POj register n : Setting value of the G1POk register
 p : Setting value of either register G1PO0 or G1BTRR
 G1IRj, G1IRk bits: Bits in the G1IR register

- The above applies under the following conditions.
- The IVL bit in the G1POCRj register is set to 0 (L output as a default value). The INV bit is set to 0 (not inverted).
 - Bits UD1 and UD0 are set to 00₂ (counter increment mode).

Figure 13.24 Set/Reset Waveform Output Mode

13.6 I/O Port Function Select

The value in the G1FE and G1FS registers decides which IC/OC pin to be an input or output pin.

In SR waveform generating mode, two channels, a set of even channel and odd channel, are used every output waveform, however, the waveform is output from an even channel only. In this case, the corresponding pin to the odd channel can be used as an I/O port.

Table 13.11 Pin setting for Time Measurement and Waveform Generating Functions

| Pin | IFE | FSC | MOD1 | MOD0 | Port Direction | Port Data |
|-----------------------|-----|-----|------|------|--|---------------|
| P27/INPC17/ OUTC17 | 0 | X | X | X | Determined by PD27 | P27 |
| | 1 | 1 | X | X | Determined by PD27, Input to INPC17 is always active | P27 or INPC17 |
| | 1 | 0 | 0 | 0 | Single-phase Waveform Output | OUTC17 |
| | 1 | 0 | 0 | 1 | Determined by PD27, SR waveform output mode | P27 |
| | 1 | 0 | 1 | 0 | Phase-delayed Waveform Output | OUTC17 |
| P26/INPC16/ OUTC16 | 0 | X | X | X | Determined by PD26 | P26 |
| | 1 | 1 | X | X | Determined by PD26, Input to INPC16 is always active | P26 or INPC16 |
| | 1 | 0 | 0 | 0 | Single-phase Waveform Output | OUTC16 |
| | 1 | 0 | 0 | 1 | SR Waveform Output | OUTC16 |
| | 1 | 0 | 1 | 0 | Phase-delayed Waveform Output | OUTC16 |
| P25/INPC15/ OUTC15 | 0 | X | X | X | Determined by PD25 | P25 |
| | 1 | 1 | X | X | Determined by PD25, Input to INPC15 is always active | P25 or INPC15 |
| | 1 | 0 | 0 | 0 | Single-phase Waveform Output | OUTC15 |
| | 1 | 0 | 0 | 1 | Determined by PD25, SR Waveform Output mode | P25 |
| | 1 | 0 | 1 | 0 | Phase-delayed Waveform Output | OUTC15 |
| P24/INPC14/ OUTC14 | 0 | X | X | X | Determined by PD24 | P24 |
| | 1 | 1 | X | X | Determined by PD24, Input to INPC14 is always active | P24 or INPC14 |
| | 1 | 0 | 0 | 0 | Single-phase Waveform Output | OUTC14 |
| | 1 | 0 | 0 | 1 | SR Waveform Output | OUTC14 |
| | 1 | 0 | 1 | 0 | Phase-delayed Waveform Output | OUTC14 |
| P23/INPC13/ OUTC13 | 0 | X | X | X | Determined by PD23 | P23 |
| | 1 | 1 | X | X | Determined by PD23, Input to INPC13 is always active | P23 or INPC13 |
| | 1 | 0 | 0 | 0 | Single-phase Waveform Output | OUTC13 |
| | 1 | 0 | 0 | 1 | Determined by PD23, SR waveform output mode | P23 |
| | 1 | 0 | 1 | 0 | Phase-delayed Waveform Output | OUTC13 |
| P22/INPC12/ OUTC12 | 0 | X | X | X | Determined by PD22 | P22 |
| | 1 | 1 | X | X | Determined by PD22, Input to INPC12 is always active | P22 or INPC12 |
| | 1 | 0 | 0 | 0 | Single-phase Waveform Output | OUTC12 |
| | 1 | 0 | 0 | 1 | SR Waveform Output | OUTC12 |
| | 1 | 0 | 1 | 0 | Phase-delayed Waveform Output | OUTC12 |
| P21/INPC11/ OUTC11 | 0 | X | X | X | Determined by PD21 | P21 |
| | 1 | 1 | X | X | Determined by PD21, Input to INPC11 is always active | P21 or INPC11 |
| | 1 | 0 | 0 | 0 | Single-phase Waveform Output | OUTC11 |
| | 1 | 0 | 0 | 1 | Determined by PD21, SR waveform output mode | P21 |
| | 1 | 0 | 1 | 0 | Phase-delayed Waveform Output | OUTC11 |
| P20/INPC10/ OUTC10 | 0 | X | X | X | Determined by PD20 | P20 |
| | 1 | 1 | X | X | Determined by PD20, Input to INPC10 is always active | P20 or INPC10 |
| | 1 | 0 | 0 | 0 | Single-phase Waveform Output | OUTC10 |
| | 1 | 0 | 0 | 1 | SR Waveform Output | OUTC10 |
| | 1 | 0 | 1 | 0 | Phase-delayed Waveform Output | OUTC10 |

IFE: IFE_j (j=0 to 7) bits in the G1FE register.

FSC: FSC_j (j=0 to 7) bits in the G1FS register.

MOD2 to MOD1: Bits in the G1POCR_j (j=0 to 7) register.

13.6.1 INPC17 Alternate Input Pin Selection

The input capture pin for IC/OC channel 7 can be assigned to one of two package pins. The CH7INSEL bit in the G1BCR0 register selects IC/OC INPC17 from P27/OUTC17/INPC17 or P17/ $\overline{\text{INT5}}$ /INPC17/IDU.

13.6.2 Digital Debounce Function for Pin P17/ $\overline{\text{INT5}}$ /INPC17

The $\overline{\text{INT5}}$ /INPC17 input from the P17/ $\overline{\text{INT5}}$ /INPC17/IDU pin has an effective digital debounce function against a noise rejection. Refer to **19.6 Digital Debounce function** for this detail.

14. Serial I/O

Note

The SI/O4 interrupt of peripheral function interrupt is not available in the 64-pin package.

Serial I/O is configured with five channels: UART0 to UART2, SI/O3 and SI/O4.

14.1 UARTi (i=0 to 2)

UARTi each have an exclusive timer to generate a transfer clock, so they operate independently of each other.

Figure 14.1 shows the block diagram of UARTi. **Figures 14.2** and **14.3** shows the block diagram of the UARTi transmit/receive.

UARTi has the following modes:

- Clock synchronous serial I/O mode
- Clock asynchronous serial I/O mode (UART mode).
- Special mode 1 (I²C bus mode): UART2
- Special mode 2: UART2
- Special mode 3 (Bus collision detection function, IEBus mode): UART2
- Special mode 4 (SIM mode): UART2

Figures 14.4 to **14.9** show the UARTi-related registers.

Refer to tables listing each mode for register setting.

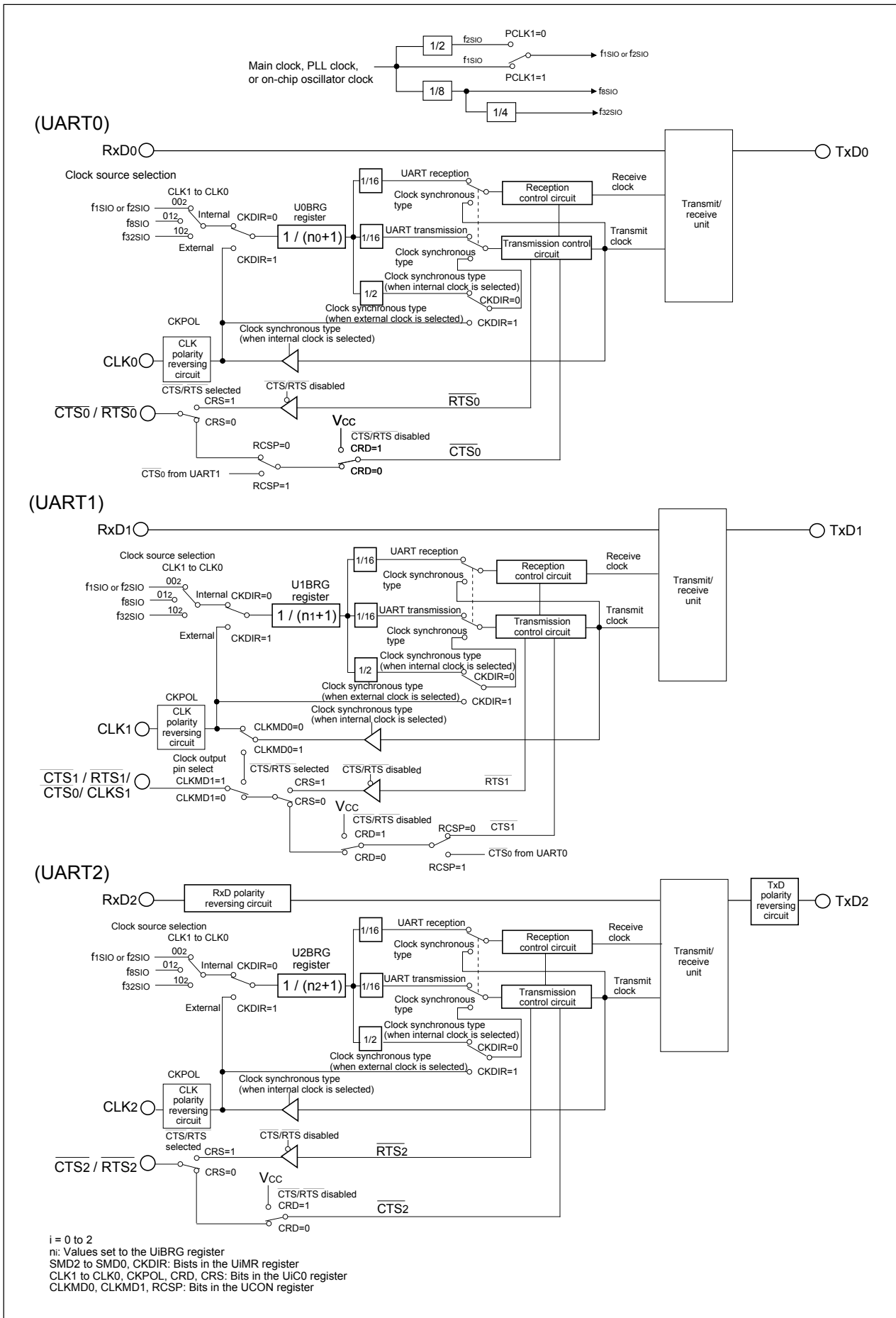


Figure 14.1 Block diagram of UARTi (i = 0 to 2)

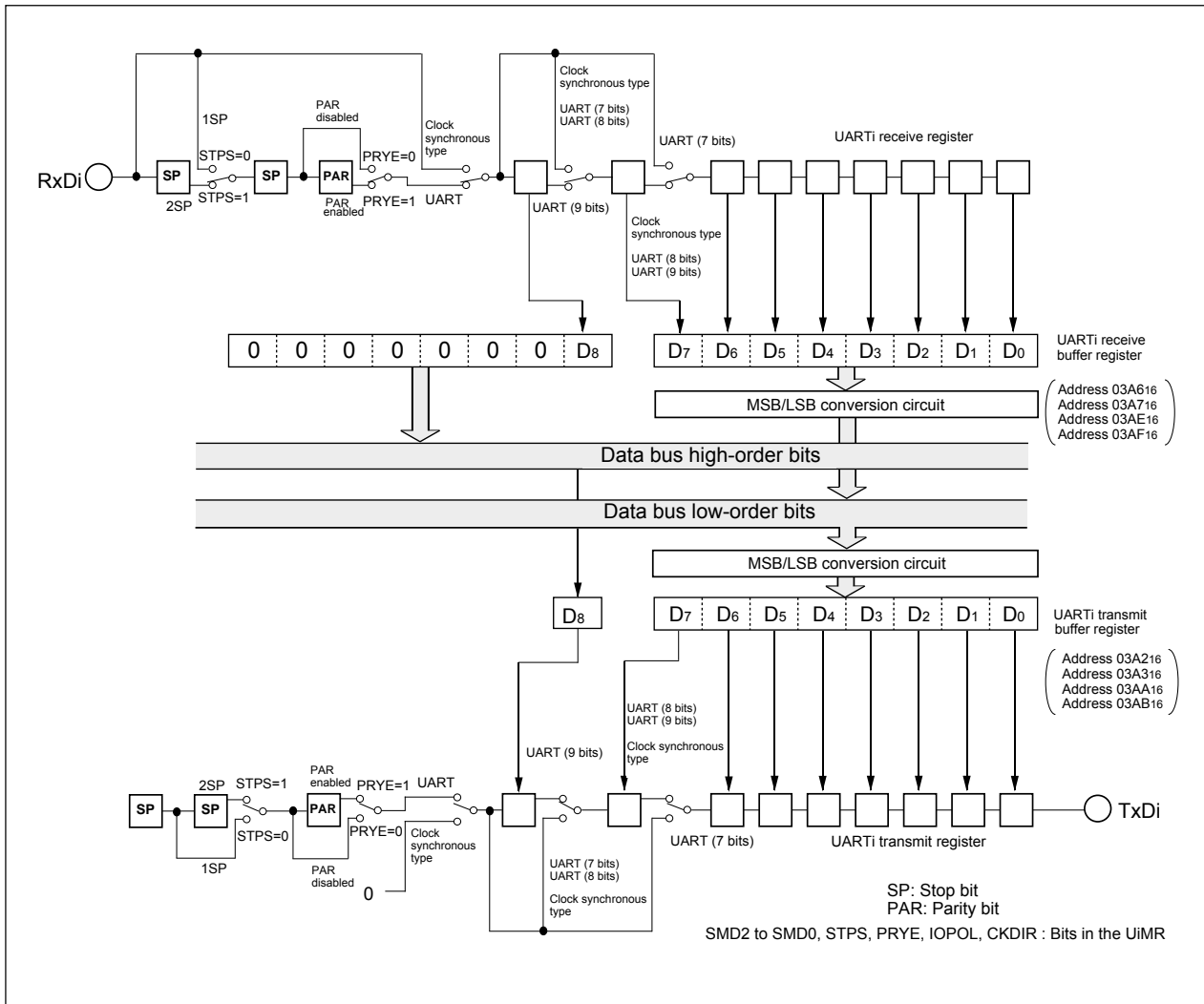


Figure 14.2 Block diagram of UARTi (i = 0, 1) transmit/receive unit

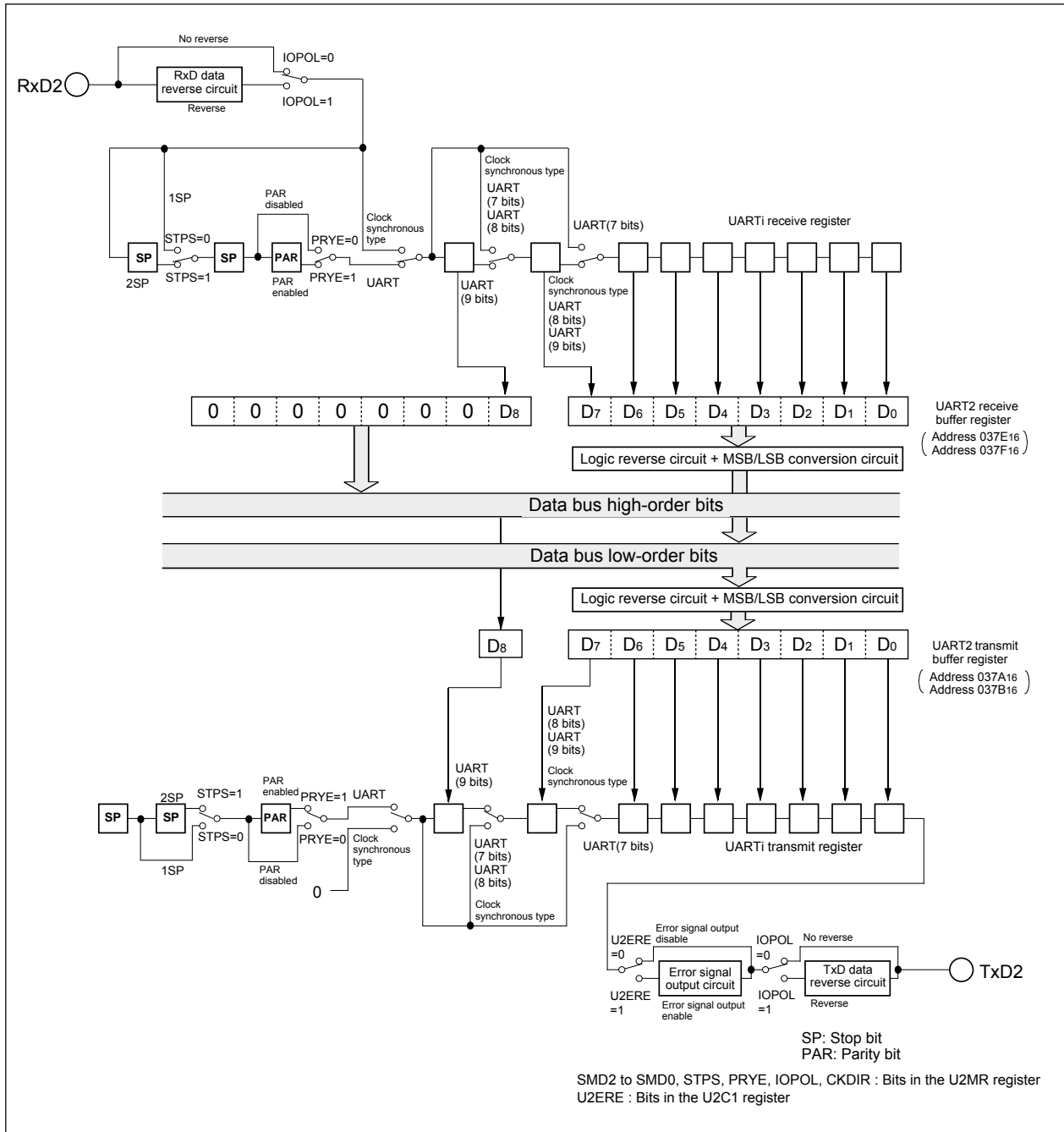


Figure 14.3 Block diagram of UART2 transmit/receive unit

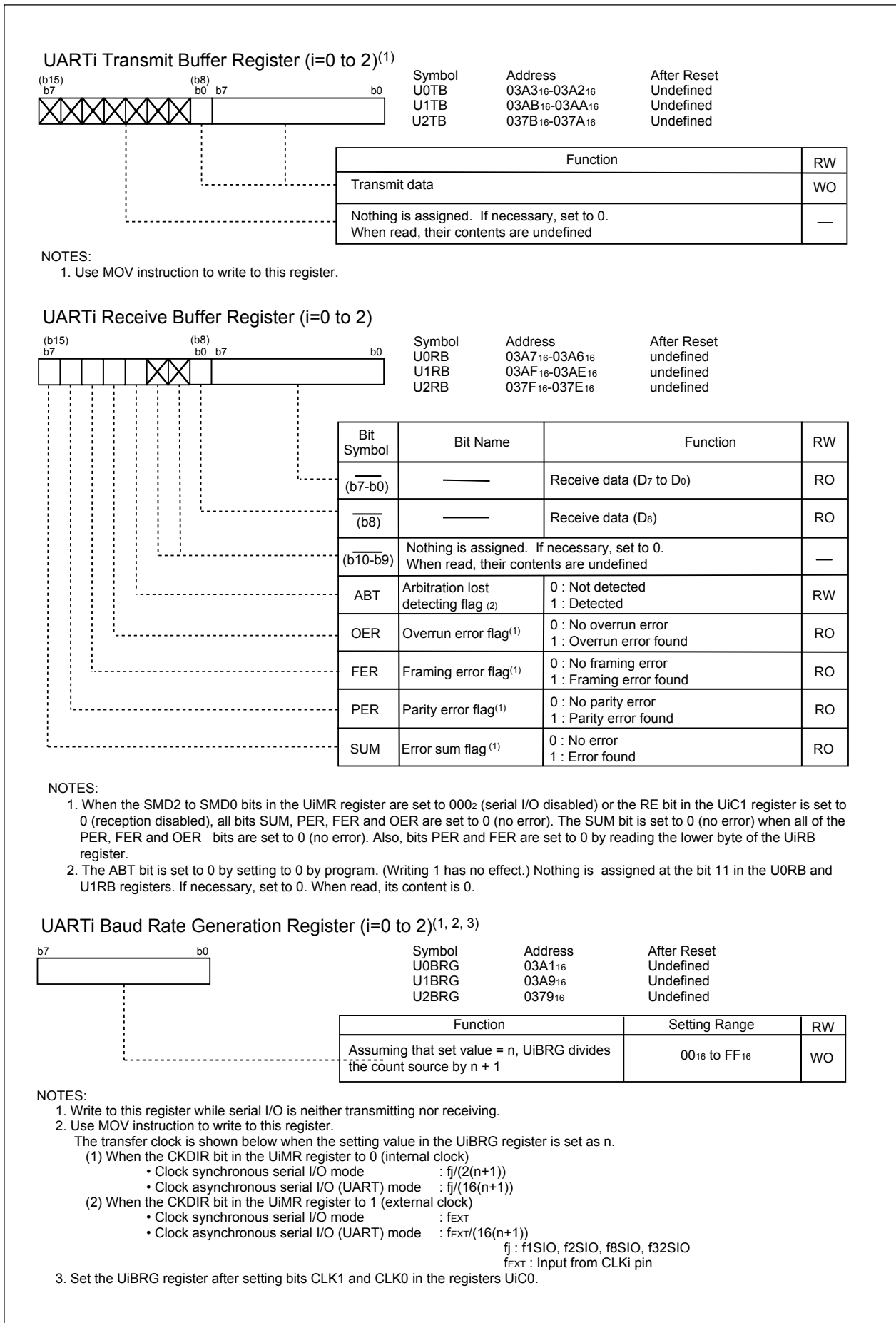


Figure 14.4 U0TB to U2TB, U0RB to U2RB, U0BRG to U2BRG Registers

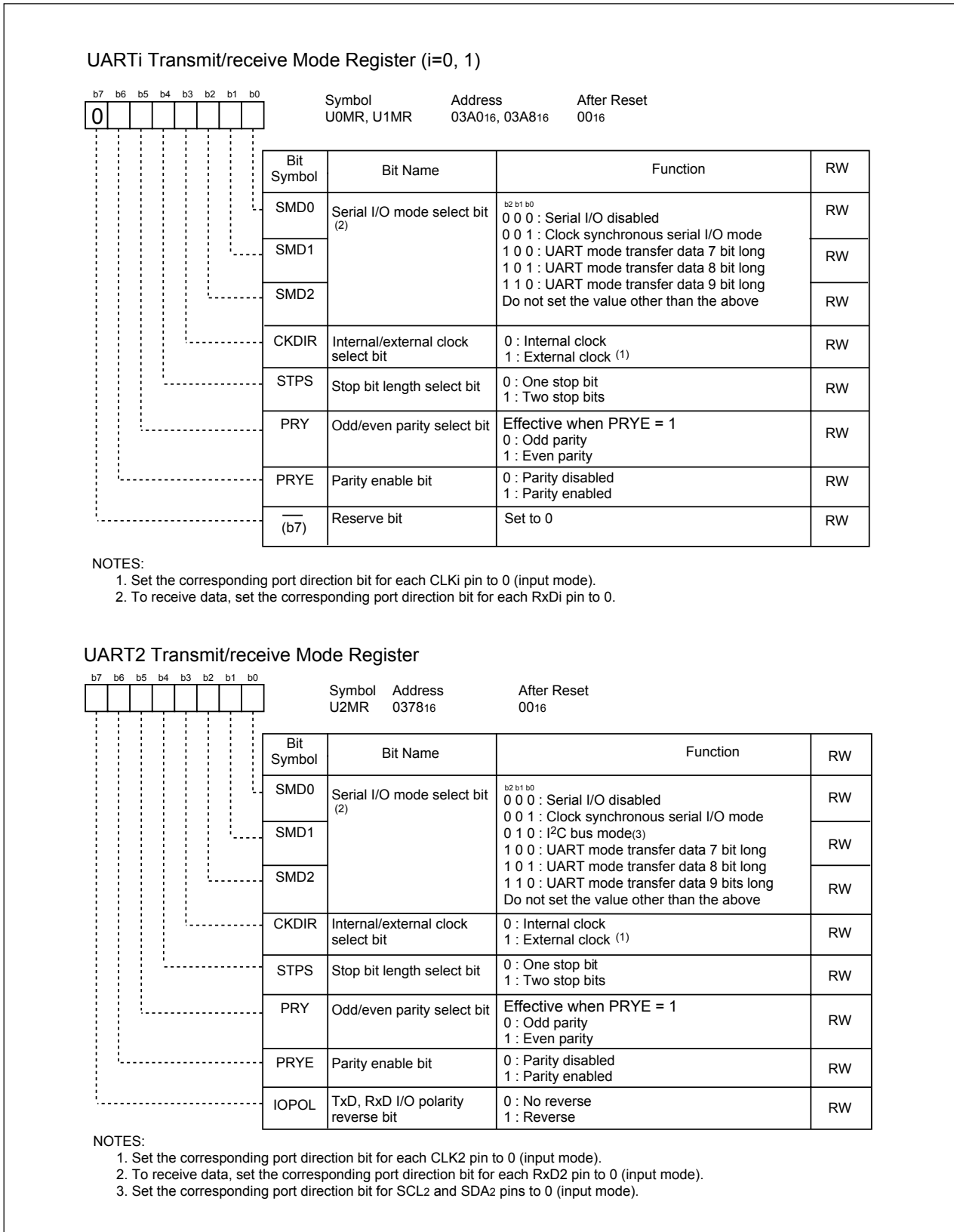


Figure 14.5 U0MR to U2MR Registers

UARTi Transmit/receive Control Register 0 (i=0 to 2)

| Bit | Symbol | Address | After Reset |
|-----|--------|---------|-------------|
| b7 | | | |
| b6 | | | |
| b5 | | | |
| b4 | | | |
| b3 | | | |
| b2 | | | |
| b1 | | | |
| b0 | | | |

| Bit Symbol | Bit Name | Function | RW |
|------------|--|--|----|
| CLK0 | BRG count source select bit ⁽⁷⁾ | b1 b0 0 0 : f1SIO or f2SIO is selected 0 1 : f8SIO is selected 1 0 : f32SIO is selected 1 1 : Do not set | RW |
| CLK1 | | | RW |
| CRS | $\overline{\text{CTS}}/\overline{\text{RTS}}$ function select bit ⁽³⁾ | Effective when CRD is set to 0 0 : $\overline{\text{CTS}}$ function is selected ⁽¹⁾ 1 : RTS function is selected | RW |
| TXEPT | Transmit register empty flag | 0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed) | RO |
| CRD | $\overline{\text{CTS}}/\overline{\text{RTS}}$ disable bit | 0 : $\overline{\text{CTS}}/\overline{\text{RTS}}$ function enabled 1 : $\overline{\text{CTS}}/\overline{\text{RTS}}$ function disabled (P60, P64 and P73 can be used as I/O ports) ⁽⁶⁾ | RW |
| NCH | Data output select bit ⁽⁵⁾ | 0 : Tx/D2/SDA2 and SCLi pins are CMOS output 1 : Tx/D2/SDA2 and SCLi pins are N-channel open-drain output ⁽⁴⁾ | RW |
| CKPOL | CLK polarity select bit | 0 : Transmit data is output at falling edge of transfer clock and receive data is input at rising edge 1 : Transmit data is output at rising edge of transfer clock and receive data is input at falling edge | RW |
| UFORM | Transfer format select bit ⁽²⁾ | 0 : LSB first 1 : MSB first | RW |

NOTES:

- Set the corresponding port direction bit for each $\overline{\text{CTS}}_i$ pin to 0 (input mode).
- Effective when bits SMD2 to SMD0 in the UMR register to 0012 (clock synchronous serial I/O mode) or 0102 (UART mode transfer data 8 bits long). Set the UFORM bit to 1 when bits SMD2 to SMD0 are set to 1012 (I²C bus mode) and 0 when they are set to 1002.
- $\overline{\text{CTS}}_1/\overline{\text{RTS}}_1$ can be used when the CLKMD1 bit in the UCON register is set to 0 (only CLK1 output) and the RCSP bit in the UCON register is set to 0 ($\overline{\text{CTS}}_0/\overline{\text{RTS}}_0$ not separated).
- SDA2 and SCL2 are effective when i = 2.
- When bits SMD2 to SMD in the UIMR register are set to 0002 (serial I/O disable), do not set NCH bit to 1 (Tx/Di/SDA2 and SCL2 pins are N-channel open-drain output).
- When the U1MAP bit in PACR register is 1 (P73 to P70), P70 functions as $\overline{\text{CTS}}/\overline{\text{RTS}}$ pin in UART1.
- When the CLK1 and CLK0 bit settings are changed, set the UIBRG register.

UART Transmit/receive Control Register 2

| Bit | Symbol | Address | After Reset |
|-----|--------|---------|-------------|
| b7 | X | | |
| b6 | | | |
| b5 | | | |
| b4 | | | |
| b3 | | | |
| b2 | | | |
| b1 | | | |
| b0 | | | |

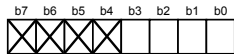
| Bit Symbol | Bit Name | Function | RW |
|------------|--|---|----|
| U0IRS | UART0 transmit interrupt cause select bit | 0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1) | RW |
| U1IRS | UART1 transmit interrupt cause select | 0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1) | RW |
| U0RRM | UART0 continuous receive mode enable bit | 0 : Continuous receive mode disabled 1 : Continuous receive mode enable | RW |
| U1RRM | UART1 continuous receive mode enable bit | 0 : Continuous receive mode disabled 1 : Continuous receive mode enabled | RW |
| CLKMD0 | UART1 CLK/CLKS select bit 0 | Effective when the CLKMD1 bit is set to 1 0 : Clock output from CLK1 1 : Clock output from CLKS1 | RW |
| CLKMD1 | UART1 CLK/CLKS select bit 1 ⁽¹⁾ | 0 : Output from CLK1 only 1 : Transfer clock output from multiple pins function selected | RW |
| RCSP | Separate UART0 $\overline{\text{CTS}}/\overline{\text{RTS}}$ bit | 0 : $\overline{\text{CTS}}/\overline{\text{RTS}}$ shared pin ⁽²⁾ 1 : $\overline{\text{CTS}}/\overline{\text{RTS}}$ separated (P64 pin functions as $\overline{\text{CTS}}_0$ pin) | RW |
| (b7) | Nothing is assigned. If necessary, set to 0. When read, the content is undefined | | — |

NOTES:

- When using multiple transfer clock output pins, make sure the following conditions are met: set the CKDIR bit in the U1MR register to 0 (internal clock)
- When the U1MAP bit in PACR register is set to 1 (P73 to P70), P70 pin functions as $\overline{\text{CTS}}_0$ pin.

Figure 14.6 U0C0 to U2C0 and UCON Registers

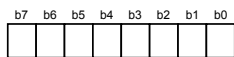
UARTi Transmit/receive Control Register 1 (i=0, 1)



Symbol: U0C1, U1C1
 Address: 03A5₁₆, 03AD₁₆
 After Reset: 00000010₂

| Bit Symbol | Bit Name | Function | RW |
|--------------|---|---|----|
| TE | Transmit enable bit | 0 : Transmission disabled 1 : Transmission enabled | RW |
| TI | Transmit buffer empty flag | 0 : Data present in UiTB register 1 : No data present in UiTB register | RO |
| RE | Receive enable bit | 0 : Reception disabled 1 : Reception enabled | RW |
| RI | Receive complete flag | 0 : No data present in UiRB register 1 : Data present in UiRB register | RO |
| — (b7-b4) | Nothing is assigned. If necessary, set to 0. When read, the content is 0 | | — |

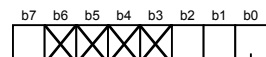
UART2 Transmit/receive Control Register 1



Symbol: U2C1
 Address: 037D₁₆
 After Reset: 00000010₂

| Bit Symbol | Bit Name | Function | RW |
|------------|---|---|----|
| TE | Transmit enable bit | 0 : Transmission disabled 1 : Transmission enabled | RW |
| TI | Transmit buffer empty flag | 0 : Data present in U2TB register 1 : No data present in U2TB register | RO |
| RE | Receive enable bit | 0 : Reception disabled 1 : Reception enabled | RW |
| RI | Receive complete flag | 0 : No data present in U2RB register 1 : Data present in U2RB register | RO |
| U2IRS | UART2 transmit interrupt cause select bit | 0 : Transmit buffer empty (TI = 1) 1 : Transmit is completed (TXEPT = 1) | RW |
| U2RRM | UART2 continuous receive mode enable bit | 0 : Continuous receive mode disabled 1 : Continuous receive mode enabled | RW |
| U2LCH | Data logic select bit | 0 : No reverse 1 : Reverse | RW |
| U2ERE | Error signal output enable bit | 0 : Output disabled 1 : Output enabled | RW |

Pin Assignment Control Register (1)



Symbol: PACR
 Address: 025D₁₆
 After Reset: 00₁₆

| Bit Symbol | Bit Name | Function | RW |
|--------------|-------------------------|---|----|
| PACR0 | Pin enabling bit | 010 : 64 pin 011 : 80 pin All other values are reserved. Do not use. | RW |
| PACR1 | | | RW |
| PACR2 | | | RW |
| — (b6-b3) | Reserved bits | Nothing is assigned. If necessary, set to 0. When read, the content is 0 | — |
| U1MAP | UART1 pin remapping bit | UART1 pins assigned to 0 : P67 to P64 1 : P73 to P70 | RW |

NOTE:
 1. Set the PACR register by the next instruction after setting the PRC2 bit in the PRCR register to 1(write enable).

Figure 14.7 U0C1 to U2C1 Register, and PACR Register

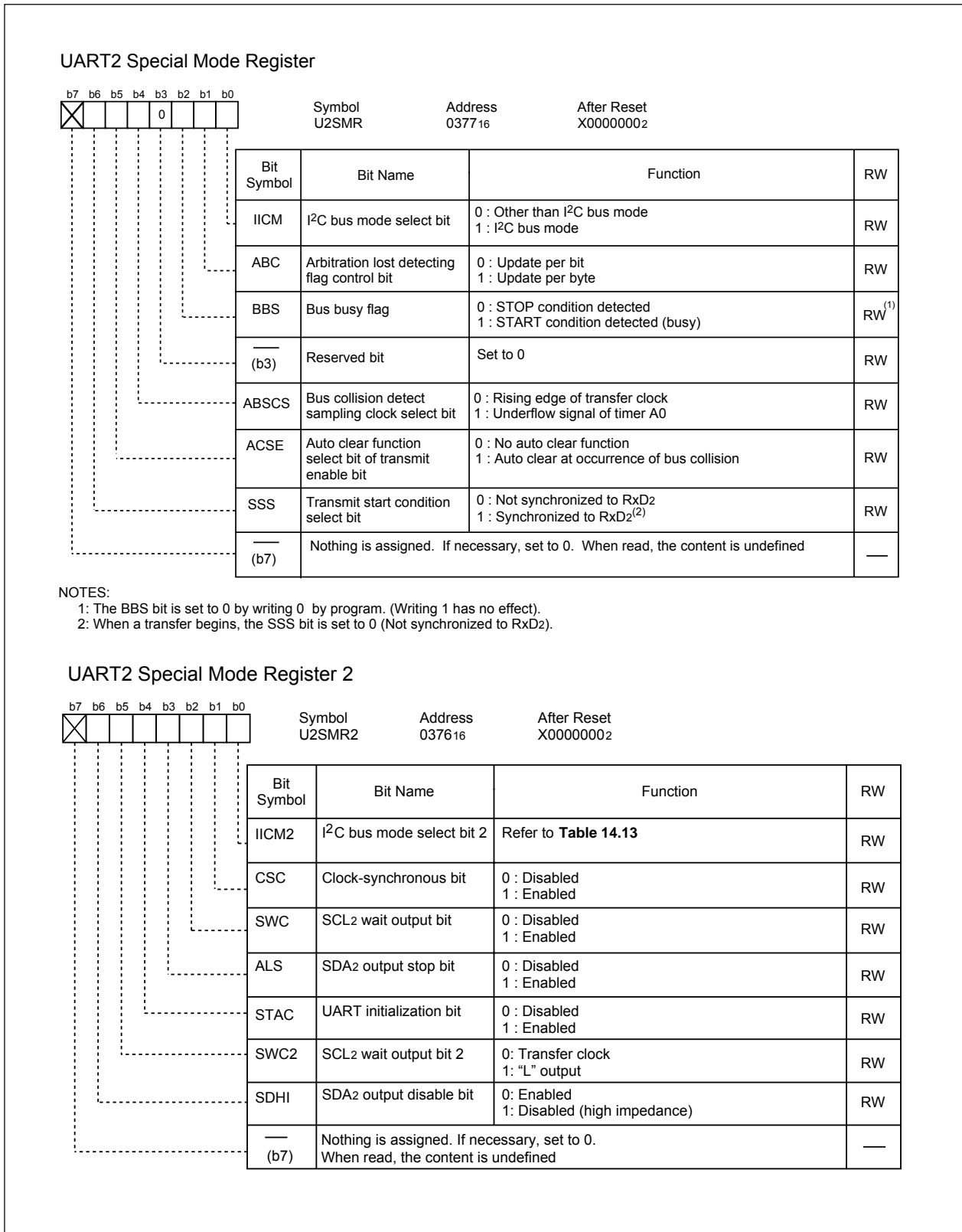


Figure 14.8 U2SMR and U2SMR2 Registers

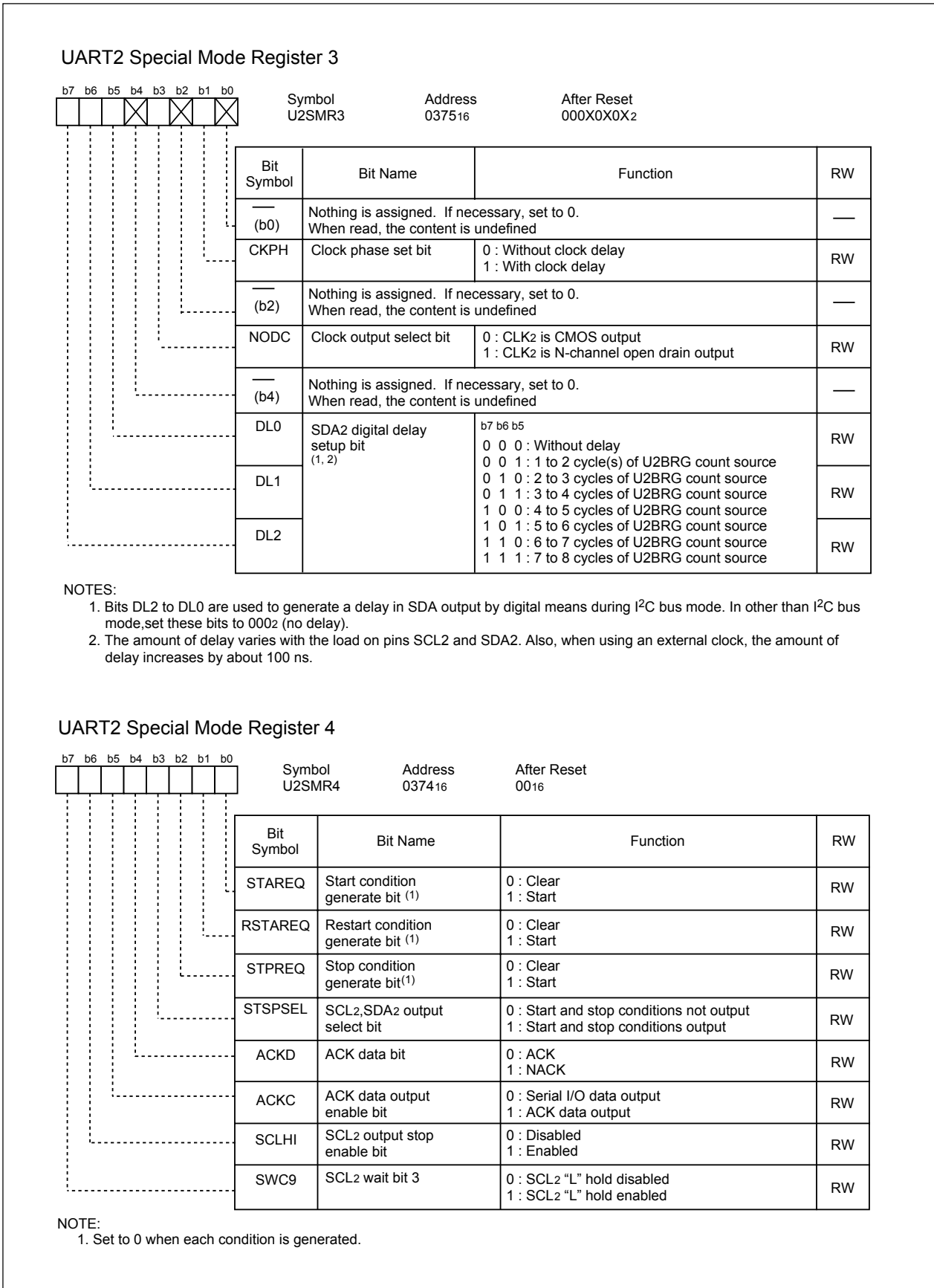


Figure 14.9 U2SMR3 and U2SMR4 Registers

14.1.1 Clock Synchronous serial I/O Mode

The clock synchronous serial I/O mode uses a transfer clock to transmit and receive data. **Table 14.1** lists the specifications of the clock synchronous serial I/O mode. **Table 14.2** lists the registers used in clock synchronous serial I/O mode and the register values set.

Table 14.1 Clock Synchronous Serial I/O Mode Specifications

| Item | Specification |
|-------------------------------------|---|
| Transfer data format | <ul style="list-style-type: none"> Transfer data length: 8 bits |
| Transfer clock | <ul style="list-style-type: none"> The CKDIR bit in the UiMR(i=0 to 2) register is set to 0 (internal clock) : $f_j / (2(n+1))$ $f_j = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}$. n: Setting value of UiBRG register 0016 to FF16 CKDIR bit is set to 1 (external clock) : Input from CLKi pin |
| Transmission, reception control | <ul style="list-style-type: none"> Selectable from CTS function, RTS function or CTS/RTS function disable |
| Transmission start condition | <ul style="list-style-type: none"> Before transmission can start, the following requirements must be met ⁽¹⁾ <ul style="list-style-type: none"> The TE bit in the UiC1 register is set to 1 (transmission enabled) The TI bit in the UiC1 register is set to 0 (data present in UiTB register) If \overline{CTS} function is selected, input on the \overline{CTS}_i pin is set to "L" |
| Reception start condition | <ul style="list-style-type: none"> Before reception can start, the following requirements must be met ⁽¹⁾ <ul style="list-style-type: none"> The RE bit in the UiC1 register is set to 1 (reception enabled) The TE bit in the UiC1 register is set to 1 (transmission enabled) The TI bit in the UiC1 register is set to 0 (data present in the UiTB register) |
| Interrupt request generation timing | <ul style="list-style-type: none"> For transmission, one of the following conditions can be selected <ul style="list-style-type: none"> The UiIRS bit ⁽³⁾ is set to 0 (transmit buffer empty): when transferring data from the UiTB register to the UARTi transmit register (at start of transmission) The UiIRS bit is set to 1 (transfer completed): when the serial I/O finished sending data from the UARTi transmit register For reception <ul style="list-style-type: none"> When transferring data from the UARTi receive register to the UiRB register (at completion of reception) |
| Error detection | <ul style="list-style-type: none"> Overrun error ⁽²⁾ This error occurs if the serial I/O started receiving the next data before reading the UiRB register and received the 7th bit in the the next data |
| Select function | <ul style="list-style-type: none"> CLK polarity selection Transfer data input/output can be chosen to occur synchronously with the rising or the falling edge of the transfer clock LSB first, MSB first selection Whether to start sending/receiving data beginning with bit 0 or beginning with bit 7 can be selected Continuous receive mode selection Reception is enabled immediately by reading the UiRB register Switching serial data logic (UART2) This function reverses the logic value of the transmit/receive data Transfer clock output from multiple pins selection (UART1) The output pin can be selected in a program from two UART1 transfer clock pins that have been set Separate $\overline{CTS}/\overline{RTS}$ pins (UART0) \overline{CTS}_0 and \overline{RTS}_0 are input/output from separate pins UART1 pin remapping selection The UART1 pin can be selected from the P67 to P64 or P73 to P70 |

NOTES:

- When an external clock is selected, the conditions must be met while if the CKPOL bit in the UiC0 register is set to 0 (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock), the external clock is in the high state; if the CKPOL bit in the UiC0 register is set to 1 (transmit data output at the rising edge and the receive data taken in at the falling edge of the transfer clock), the external clock is in the low state.
- If an overrun error occurs, bits 8 to 0 in the UiRB register are undefined. The IR bit in the SiRIC register remains unchanged.
- The U0IRS and U1IRS bits respectively are the bits 0 and 1 in the UCON register; the U2IRS bit is bit 4 in the U2C1 register.

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

| Register | Bit | Function |
|---------------------|---------------------------|---|
| UiTB ⁽³⁾ | 0 to 7 | Set transmission data |
| UiRB ⁽³⁾ | 0 to 7 | Reception data can be read |
| | OER | Overrun error flag |
| UiBRG | 0 to 7 | Set bit rate |
| UiMR ⁽³⁾ | SMD2 to SMD0 | Set to 0012 |
| | CKDIR | Select the internal clock or external clock |
| | IOPOL(i=2) ⁽⁴⁾ | Set to 0 |
| UiC0 | CLK1 to CLK0 | Select the count source for the UiBRG register |
| | CRS | Select \overline{CTS} or \overline{RTS} to use |
| | TXEPT | Transmit register empty flag |
| | CRD | Enable or disable the \overline{CTS} or \overline{RTS} function |
| | NCH | Select TxDi pin output mode |
| | CKPOL | Select the transfer clock polarity |
| | UFORM | Select the LSB first or MSB first |
| UiC1 | TE | Set this bit to 1 to enable transmission/reception |
| | TI | Transmit buffer empty flag |
| | RE | Set this bit to 1 to enable reception |
| | RI | Reception complete flag |
| | U2IRS ⁽¹⁾ | Select the source of UART2 transmit interrupt |
| | U2RRM ⁽¹⁾ | Set this bit to 1 to use UART2 continuous receive mode |
| | U2LCH ⁽³⁾ | Set this bit to 1 to use UART2 inverted data logic |
| | U2ERE ⁽³⁾ | Set to 0 |
| U2SMR | 0 to 7 | Set to 0 |
| U2SMR2 | 0 to 7 | Set to 0 |
| U2SMR3 | 0 to 2 | Set to 0 |
| | NODC | Select clock output mode |
| | 4 to 7 | Set to 0 |
| U2SMR4 | 0 to 7 | Set to 0 |
| UCON | U0IRS, U1IRS | Select the source of UART0/UART1 transmit interrupt |
| | U0RRM, U1RRM | Set this bit to 1 to use continuous receive mode |
| | CLKMD0 | Select the transfer clock output pin when CLKMD1 is set to 1 |
| | CLKMD1 | Set this bit to 1 to output UART1 transfer clock from two pins |
| | RCSP | Set this bit to 1 to accept as input the UART0 \overline{CTS}_0 signal from the P64 pin |
| | 7 | Set to 0 |

NOTES:

1. Set bits 5 and 4 in registers U0C1 and U1C1 to 0. Bits U0IRS, U1IRS, U0RRM, and U1RRM are in the UCON register.
2. Not all register bits are described above. Set those bits to 0 when writing to the registers in clock synchronous serial I/O mode.
3. Set bits 7 and 6 in registers U0C1 and U1C1 to 0.
4. Set the bit 7 in registers U0MR and U1MR to 0.
i=0 to 2

Table 14.3 lists pin functions for the case where the multiple transfer clock output pin select function is deselected. **Table 14.4** lists the P64 pin functions during clock synchronous serial I/O mode.

Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TxDi pin outputs an "H". (If the N-channel open-drain output is selected, this pin is in a high-impedance state.)

Table 14.3 Pin Functions (When Not Select Multiple Transfer Clock Output Pin Function)⁽¹⁾

| Pin Name | Function | Method of Selection |
|---|-----------------------|---|
| TxDi (i = 0 to 2) (P63, P67, P70) | Serial data output | (Outputs dummy data when performing reception only) |
| RxDi (P62, P66, P71) | Serial data input | Set the PD6_2 bit and PD6_6 bit in the PD6 register, and PD7_1 bit in the PD7 register to 0 (Can be used as an input port when performing transmission only) |
| CLKi (P61, P65, P72) | Transfer clock output | Set the CKDIR bit in the UiMR register to 0 |
| | Transfer clock input | Set the CKDIR bit in the UiMR register to 1 Set the PD6_1 bit and PD6_5 bit in the PD6 register, and the PD7_2 bit in the PD7 register to 0 |
| CTS \overline /RTS \overline (P60, P64, P73) | CTS input | Set the CRD bit in the UiC0 register to 0 Set the CRS bit in the UiC0 register to 0 Set the PD6_0 bit and PD6_4 bit in the PD6 register is set to 0, the PD7_3 bit in the PD7 register to 0 |
| | RTS output | Set the CRD bit in the UiC0 register to 0 Set the CRS bit in the UiC0 register to 1 |
| | I/O port | Set the CRD bit in the UiC0 register to 1 |

NOTE:

1: When the U1MAP bit in PACR register is 1 (P73 to P70), UART1 pin is assigned to P73 to P70.

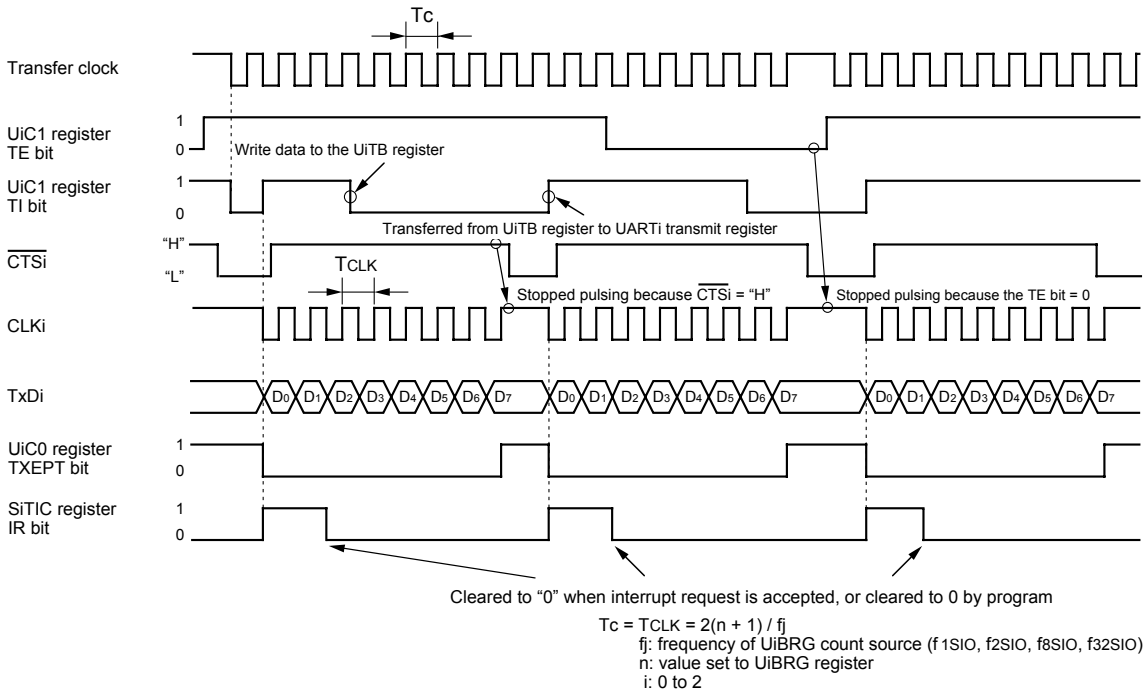
Table 14.4 P64 Pin Functions⁽¹⁾

| Pin Function | Bit Set Value | | | | | |
|---------------------------------|---------------|-----|---------------|------------------|--------|---------------------|
| | U1C0 register | | U0C0 register | | | PD6 register |
| | CRD | CRS | RCSP | CLKMD1 | CLKMD0 | PD6_4 |
| P64 | 1 | — | 0 | 0 | — | Input: 0, Output: 1 |
| \overline CTS ₁ | 0 | 0 | 0 | 0 | — | 0 |
| \overline RTS ₁ | 0 | 1 | 0 | 0 | — | — |
| CTS ₀ ⁽²⁾ | 0 | 0 | 1 | 0 | — | 0 |
| CLKS ₁ | — | — | — | 1 ⁽³⁾ | 1 | — |

NOTES:

- When the U1MAP bit in PACR register is 1 (P73 to P70), this table lists the P70 functions.
- In addition to this, set the CRD bit in the U0C0 register to 0 (CT0 \overline /RT0 \overline enabled) and the CRS bit in the U0C0 register to 1 (\overline RTS₀ selected).
- When the CLKMD1 bit is set to 1 and the CLKMD0 bit is set to 0, the following logic levels are output:
 - High if the CLKPOL bit in the U1C0 register is set to 0
 - Low if the CLKPOL bit in the U1C0 register is set to 1

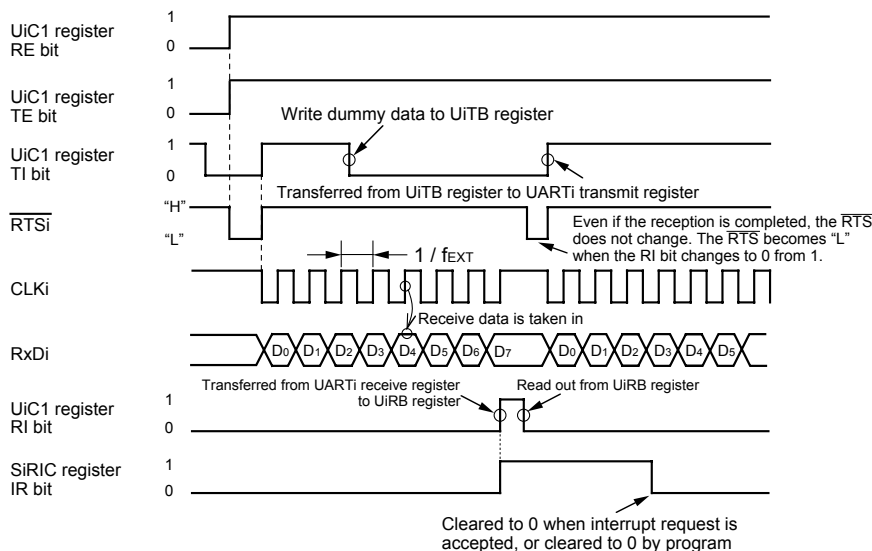
(1) Example of Transmit Timing (Internal clock is selected)



The above timing diagram applies to the case where the register bits are set as follows:

- The CKDIR bit in the UiMR register is set to 0 (internal clock)
- The CRD bit in the UIC0 register is set to 0 (CTS/RTS enabled); CRS bit is set to 0 (CTS selected)
- The CKPOL bit in the UIC0 register is set to 0 (transmit data output at the falling edge and receive data taken in at the rising edge of the transfer clock)
- The UIIRS bit is set to 0 (an interrupt request occurs when the transmit buffer becomes empty): U0IRS bit is the bit 0 in the UCON register U1IRS bit is the bit 1 in the UCON register, and U2IRS bit is the bit 4 in the U2C1 register.

(2) Example of Receive Timing (External clock is selected)



The above timing diagram applies to the case where the register bits are set as follows:

- The CKDIR bit in the UiMR register is set to 1 (external clock)
- The CRD bit in the UIC0 register is set to 0 (CTS/RTS enabled); The CRS bit is set to 1 (RTS selected)
- UIC0 register CKPOL bit is set to 0 (transmit data output at the falling edge and receive data taken in at the rising edge of the transfer clock)

- Make sure the following conditions are met when input to the CLKi pin before receiving data is high:
- UIC0 register TE bit is set to 1 (transmit enabled)
 - UIC0 register RE bit is set to 1 (Receive enabled)
 - Write dummy data to the UiTB register

fEXT: frequency of external clock

Figure 14.10 Typical transmit/receive timings in clock synchronous serial I/O mode

14.1.1.1 Counter Measure for Communication Error Occurs

If a communication error occurs while transmitting or receiving in clock synchronous serial I/O mode, follow the procedures below.

- Resetting the UiRB register (i=0 to 2)

- (1) Set the RE bit in the UiC1 register to 0 (reception disabled)
- (2) Set bits SMD2 to SMD0 in the UiMR register to 0002 (Serial I/O disabled)
- (3) Set bits SMD2 to SMD0 in the UiMR register to 0012 (Clock synchronous serial I/O mode)
- (4) Set the RE bit in the UiC1 register to 1 (reception enabled)

- Resetting the UiTB register (i=0 to 2)

- (1) Set bits SMD2 to SMD0 in the UiMR register to 0002 (Serial I/O disabled)
- (2) Set bits SMD2 to SMD0 in the UiMR register to 0012 (Clock synchronous serial I/O mode)
- (3) 1 is written to TE bit in the UiC1 register (reception enabled), regardless to the TE bit.

14.1.1.2 CLK Polarity Select Function

Use the CKPOL bit in the UiC0 register (i=0 to 2) to select the transfer clock polarity. **Figure 14.11** shows the polarity of the transfer clock.

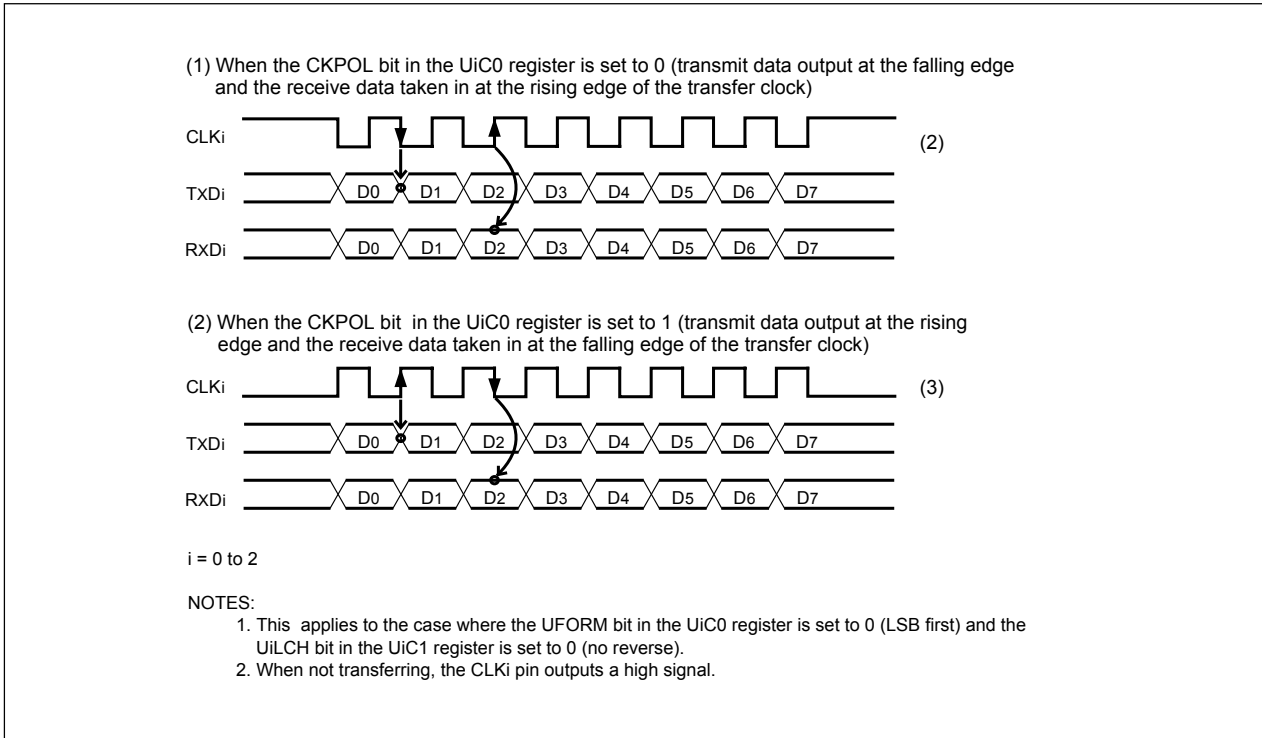


Figure 14.11 Polarity of transfer clock

14.1.1.3 LSB First/MSB First Select Function

Use the UFORM bit in the UiC0 register (i=0 to 2) to select the transfer format. **Figure 14.12** shows the transfer format.

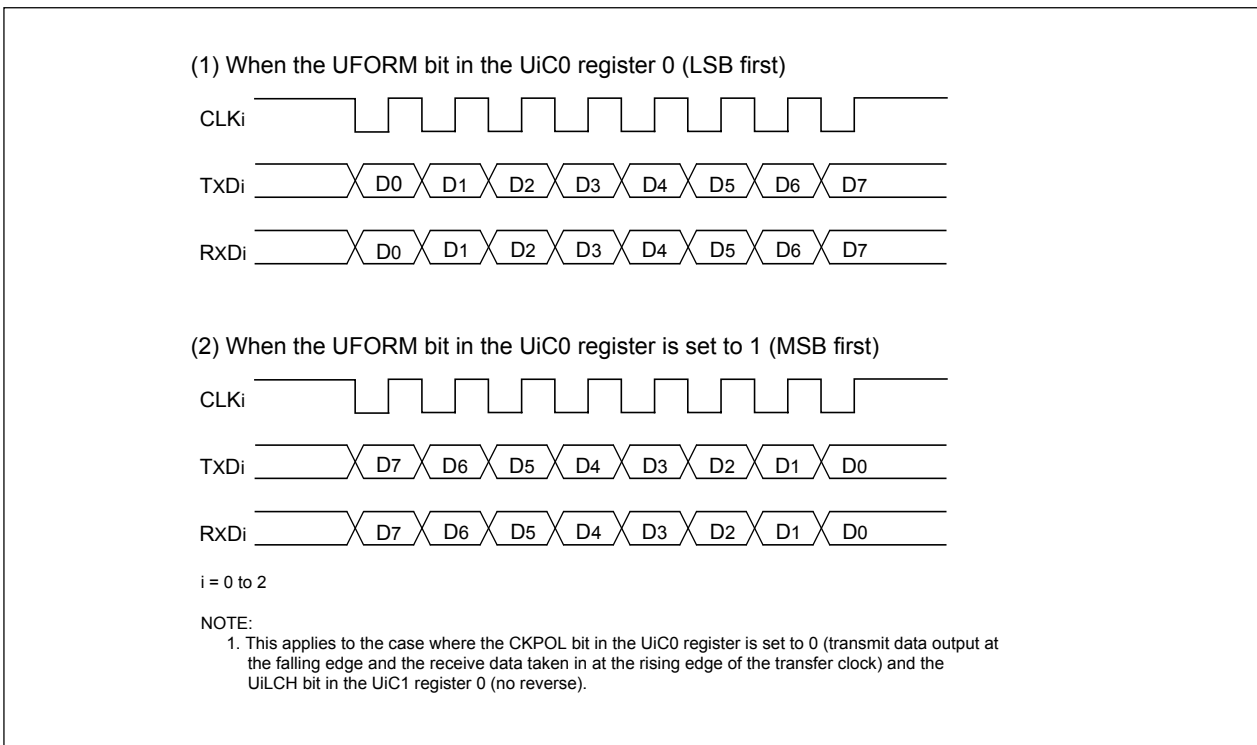


Figure 14.12 Transfer format

14.1.1.4 Continuous receive mode

When the UiRRM bit ($i=0$ to 2) is set to 1 (continuous receive mode), the TI bit in the UiC1 register is set to 0 (data present in the UiTB register) by reading the UiRB register. In this case, i.e., UiRRM bit is set to 1, do not write dummy data to the UiTB register in a program. The U0RRM and U1RRM bits are the bit 2 and bit 3 in the UCON register, respectively, and the U2RRM bit is the bit 5 in the U2C1 register.

14.1.1.5 Serial data logic switch function (UART2)

When the U2LCH bit in the U2C1 register is set to 1 (reverse), the data written to the U2TB register has its logic reversed before being transmitted. Similarly, the received data has its logic reversed when read from the U2RB register. **Figure 14.13** shows serial data logic.

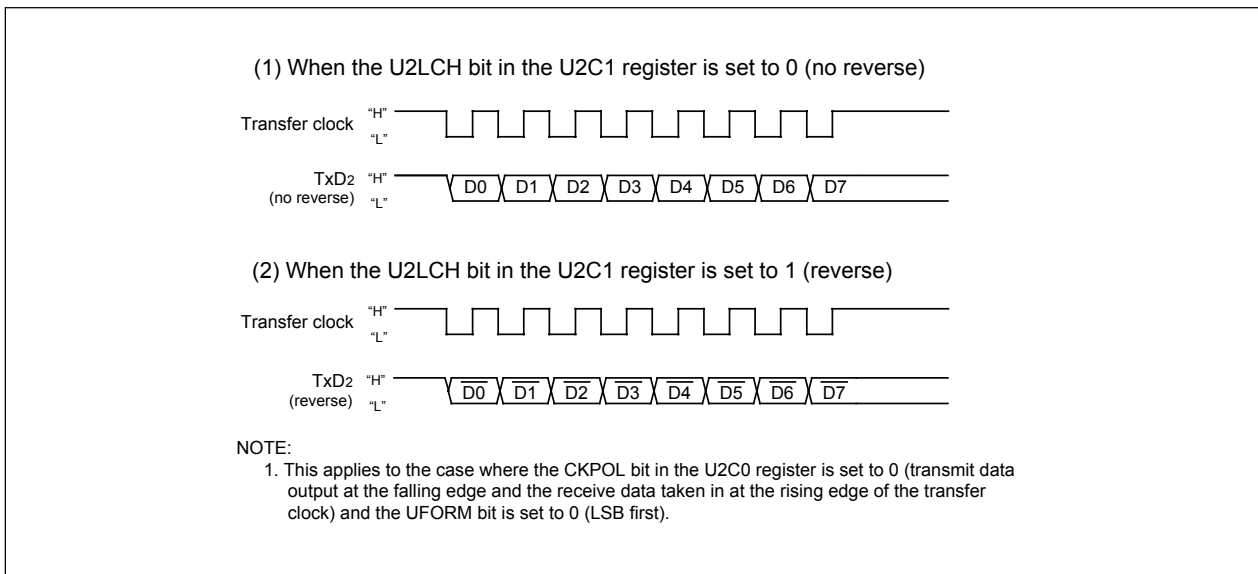


Figure 14.13 Serial data logic switch timing

14.1.1.6 Transfer clock output from multiple pins function (UART1)

The CLKMD1 to CLKMD0 bits in the UCON register can choose one from two transfer clock output pins. (See **Figure 14.14**) This function is valid when the internal clock is selected for UART1.

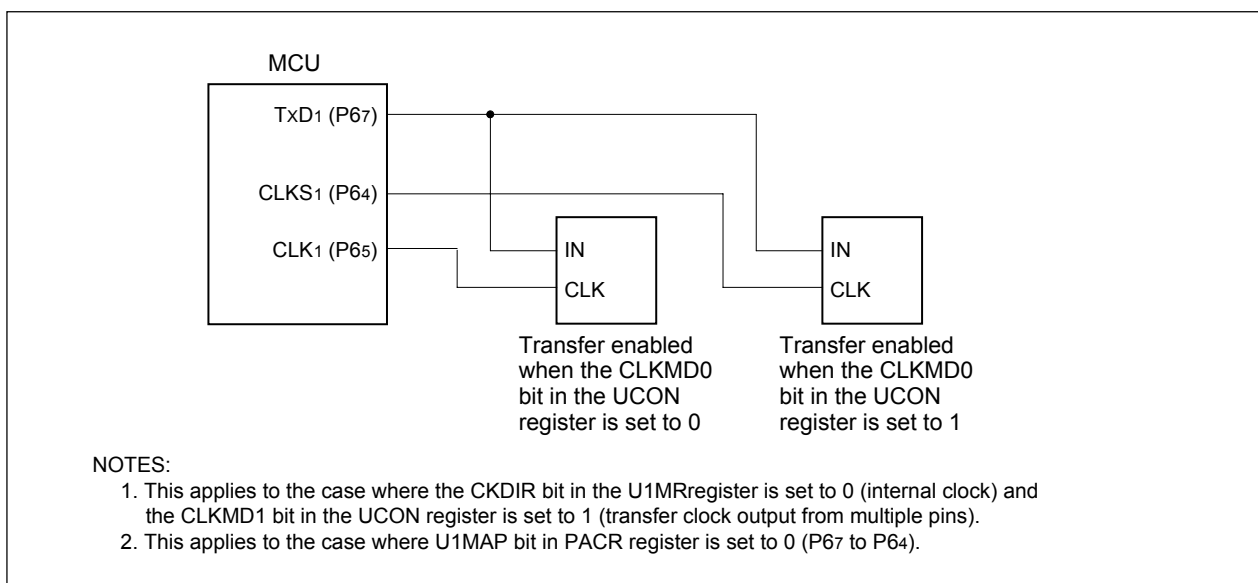


Figure 14.14 Transfer Clock Output From Multiple Pins

14.1.1.7 $\overline{\text{CTS}}/\overline{\text{RTS}}$ separate function (UART0)

This function separates $\overline{\text{CTS}}_0/\overline{\text{RTS}}_0$, outputs $\overline{\text{RTS}}_0$ from the P60 pin, and accepts as input the $\overline{\text{CTS}}_0$ from the P64 pin or P70 pin. To use this function, set the register bits as shown below.

- The CRD bit in the U0C0 register is set to 0 (enables UART0 $\overline{\text{CTS}}/\overline{\text{RTS}}$)
- The CRS bit in the U0C0 register is set to 1 (outputs UART0 $\overline{\text{RTS}}$)
- The CRD bit in the U1C0 register is set to 0 (enables UART1 $\overline{\text{CTS}}/\overline{\text{RTS}}$)
- The CRS bit in the U1C0 register is set to 0 (inputs UART1 $\overline{\text{CTS}}$)
- The RCSP bit in the UCON register is set to 1 (inputs $\overline{\text{CTS}}_0$ from the P64 pin or P70 pin)
- The CLKMD1 bit in the UCON register is set to 0 (CLKS1 not used)

Note that when using the $\overline{\text{CTS}}/\overline{\text{RTS}}$ separate function, UART1 $\overline{\text{CTS}}/\overline{\text{RTS}}$ separate function cannot be used.

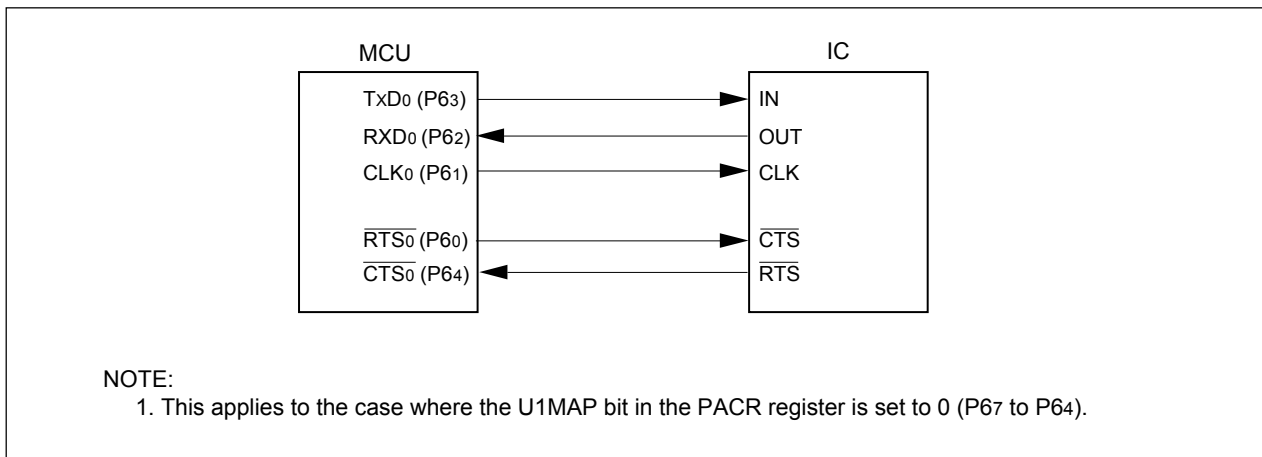


Figure 14.15 CTS/RTS separate function usage

14.1.2 Clock Asynchronous Serial I/O (UART) Mode

The UART mode allows transmitting and receiving data after setting the desired bit rate and transfer data format. **Table 14.5** lists the specifications of the UART mode.

Table 14.5 UART Mode Specifications

| 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"> • The CKDIR bit in the UiMR(i=0 to 2) register is set to 0 (internal clock) : $f_j / (16(n+1))$ $f_j = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}$. n: Setting value of UiBRG register 00₁₆ to FF₁₆ • CKDIR bit is set to 1 (external clock) : $f_{EXT} / 16(n+1)$ f_{EXT}: Input from CLKi pin. n: Setting value of UiBRG register 00₁₆ to FF₁₆ |
| Transmission, reception control | <ul style="list-style-type: none"> • Selectable from CTS function, RTS function or CTS/RTS function disable |
| Transmission start condition | <ul style="list-style-type: none"> • Before transmission can start, the following requirements must be met <ul style="list-style-type: none"> – The TE bit in the UiC1 register is set to 1 (transmission enabled) – The TI bit in the UiC1 register is set to 0 (data present in UiTB register) – If CTS function is selected, input on the CTSi pin is set to "L" |
| Reception start condition | <ul style="list-style-type: none"> • Before reception can start, the following requirements must be met" <ul style="list-style-type: none"> – The RE bit in the UiC1 register is set to 1 (reception enabled) – Start bit detection |
| Interrupt request generation timing | <ul style="list-style-type: none"> • For transmission, one of the following conditions can be selected <ul style="list-style-type: none"> – The UiIRS bit ⁽²⁾ is set to 0 (transmit buffer empty): when transferring data from the UiTB register to the UARTi transmit register (at start of transmission) – The UiIRS bit is set to 1 (transfer completed): when the serial I/O finished sending data from the UARTi transmit register • For reception <ul style="list-style-type: none"> When transferring data from the UARTi receive register to the UiRB register (at completion of reception) |
| Error detection | <ul style="list-style-type: none"> • Overrun error ⁽¹⁾ This error occurs if the serial I/O started receiving the next data before reading the UiRB register and received the bit one before the last stop bit in the the next data • Framing error This error occurs when the number of stop bits set is not detected • Parity error This error occurs when if parity is enabled, the number of 1 in parity and character bits does not match the number of 1 set • Error sum flag This flag is set to 1 when any of the overrun, framing, and parity errors is encountered |
| Select function | <ul style="list-style-type: none"> • LSB first, MSB first selection Whether to start sending/receiving data beginning with bit 0 or beginning with bit 7 can be selected • Serial data logic switch (UART2) This function reverses the logic of the transmit/receive data. The start and stop bits are not reversed. • TxD, RxD I/O polarity switch (UART2) This function reverses the polarities of the TxD pin output and RxD pin input. The logic levels of all I/O data is reversed. • Separate CTS/RTS pins (UART0) CTS₀ and RTS₀ are input/output from separate pins • UART1 pin remapping selection The UART1 pin can be selected from the P67 to P64 or P73 to P70 |

NOTES:

1. If an overrun error occurs, bits 8 to 0 in the UiRB register are undefined. The IR bit in the SiRIC register remains unchanged.

2. Bits U0IRS and U1IRS respectively are the UCON register bits 0 and 1; the U2IRS bit is the U2C1 register bit 4.

Table 14.6 Registers to Be Used and Settings in UART Mode

| Register | Bit | Function |
|----------|---------------------------|--|
| UiTB | 0 to 8 | Set transmission data ⁽¹⁾ |
| UiRB | 0 to 8 | Reception data can be read ⁽¹⁾ |
| | OER, FER, PER, SUM | Error flag |
| UiBRG | 0 to 7 | Set bit rate |
| UiMR | SMD2 to SMD0 | Set these bits to 1002 when transfer data is 7 bits long Set these bits to 1012 when transfer data is 8 bits long Set these bits to 1102 when transfer data is 9 bits long |
| | CKDIR | Select the internal clock or external clock |
| | STPS | Select the stop bit |
| | PRY, PRYE | Select whether parity is included and whether odd or even |
| | IOPOL(i=2) ⁽⁴⁾ | Select the TxD/RxD input/output polarity |
| UiC0 | CLK0, CLK1 | Select the count source for the UiBRG register |
| | CRS | Select $\overline{\text{CTS}}$ or $\overline{\text{RTS}}$ to use |
| | TXEPT | Transmit register empty flag |
| | CRD | Enable or disable the $\overline{\text{CTS}}$ or $\overline{\text{RTS}}$ function |
| | NCH | Select TxDi pin output mode |
| | CKPOL | Set to 0 |
| | UFORM | LSB first or MSB first can be selected when transfer data is 8 bits long. Set this bit to 0 when transfer data is 7 or 9 bits long. |
| UiC1 | TE | Set this bit to 1 to enable transmission |
| | TI | Transmit buffer empty flag |
| | RE | Set this bit to 1 to enable reception |
| | RI | Reception complete flag |
| | U2IRS ⁽²⁾ | Select the source of UART2 transmit interrupt |
| | U2RRM ⁽²⁾ | Set to 0 |
| | UiLCH ⁽³⁾ | Set this bit to 1 to use UART2 inverted data logic |
| | UiERE ⁽³⁾ | Set to 0 |
| UiSMR | 0 to 7 | Set to 0 |
| UiSMR2 | 0 to 7 | Set to 0 |
| UiSMR3 | 0 to 7 | Set to 0 |
| UiSMR4 | 0 to 7 | Set to 0 |
| UCON | U0IRS, U1IRS | Select the source of UART0/UART1 transmit interrupt |
| | U0RRM, U1RRM | Set to 0 |
| | CLKMD0 | Invalid because CLKMD1 is set to 0 |
| | CLKMD1 | Set to 0 |
| | RCSP | Set this bit to 1 to accept as input the UART0 $\overline{\text{CTS}}$ o signal from the P64 pin |
| | 7 | Set to 0 |

NOTES:

1. The bits used for transmit/receive data are as follows: Bit 0 to bit 6 when transfer data is 7 bits long; bits 7 to 0 when transfer data is 8 bits long; bit 0 to bit 8 when transfer data is 9 bits long.
2. Set bits 5 and 4 in registers U0C1 and U1C1 to 0. Bits U0IRS, U1IRS, U0RRM and U1RRM are included in the UCON register.
3. Set bits 7 and 6 in registers U0C1 and U1C1 to 0.
4. Set the bit 7 in registers U0MR and U1MR to 0.

i=0 to 2

Table 14.7 lists the functions of the input/output pins in UART mode. **Table 14.8** lists the P64 pin functions during UART mode. Note that for a period from when the UART_i operation mode is selected to when transfer starts, the TxD_i pin outputs an “H”. (If the N-channel open-drain output is selected, this pin is in a high-impedance state.)

Table 14.7 I/O Pin Functions in UART mode⁽¹⁾

| Pin Name | Function | Method of Selection |
|--|--------------------------------|--|
| TxD _i (i = 0 to 2) (P63, P67, P70) | Serial data output | (Outputs "H" when performing reception only) |
| RxD _i (P62, P66, P71) | Serial data input | PD6_2 bit, PD6_6 bit in the PD6 register and the PD7_1 bit in the PD7 register (Can be used as an input port when performing transmission only) |
| CLK _i (P61, P65, P72) | Input/output port | Set the CKDIR bit in the UiMR register to 0 |
| | Transfer clock input | Set the CKDIR bit in the UiMR register to 1 Set the PD6_1 bit and PD6_5 bit in the PD6 register to 0, PD7_2 bit in the PD7 register to 0 |
| $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$ (P60, P64, P73) | CTS input | Set the CRD bit in the UiC0 register to 0 Set the CRS bit in the UiC0 register to 0 Set the PD6_0 bit and PD6_4 bit in the PD6 register to 0, the PD7_3 bit in the PD7 register to 0 |
| | $\overline{\text{RTS}}$ output | Set the CRD bit in the UiC0 register to 0 Set the CRS bit in the UiC0 register to 1 |
| | Input/output port | Set the CRD bit in the UiC0 register to 1 |

NOTE:

- When the U1MAP bit in PACR register is set to 1 (P73 to P70), UART1 pin is assigned to P73 to P70.

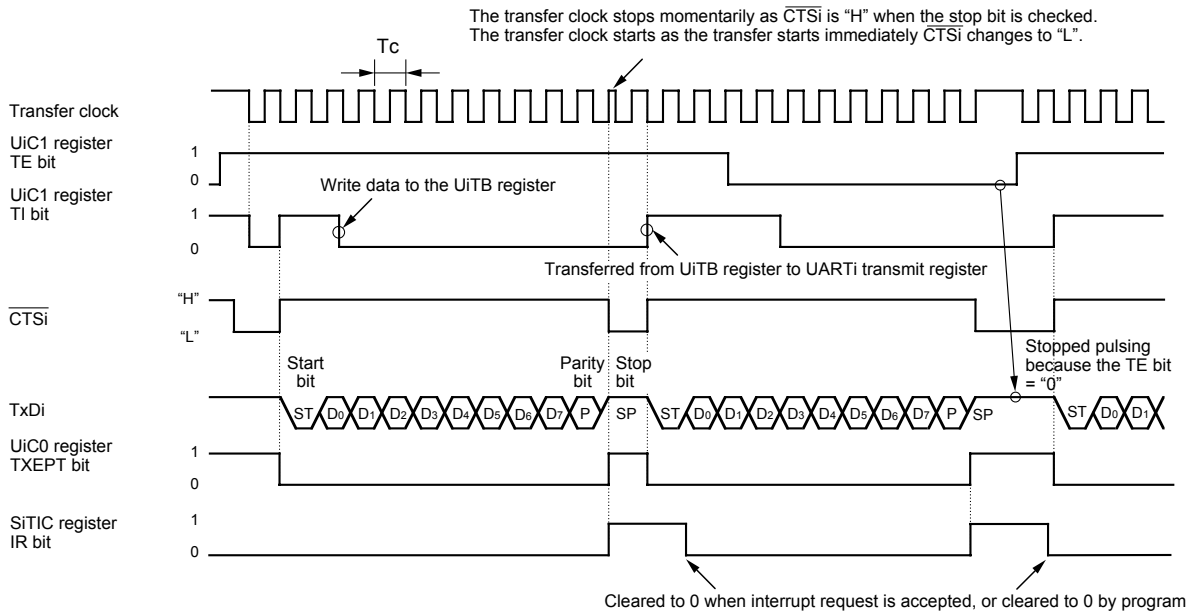
Table 14.8 P64 Pin Functions in UART mode⁽¹⁾

| Pin Function | Bit Set Value | | | | |
|--|---------------|-----|----------------|--------|---------------------|
| | U1C0 register | | U1CON register | | PD6 register |
| | CRD | CRS | RCSP | CLKMD1 | PD6_4 |
| P64 | 1 | — | 0 | 0 | Input: 0, Output: 1 |
| $\overline{\text{CTS}}_1$ | 0 | 0 | 0 | 0 | 0 |
| $\overline{\text{RTS}}_1$ | 0 | 1 | 0 | 0 | — |
| $\overline{\text{CTS}}_0$ ⁽²⁾ | 0 | 0 | 1 | 0 | 0 |

NOTES:

- When the U1MAP bit in PACR register is 1 (P73 to P70), this table lists the P70 functions.
- In addition to this, set the CRD bit in the U0C0 register to 0 ($\overline{\text{CTS}}_0/\overline{\text{RTS}}_0$ enabled) and the CRS bit in the U0C0 register to 1 ($\overline{\text{RTS}}_0$ selected).

• Example of transmit timing when transfer data is 8-bit long (parity enabled, one stop bit)



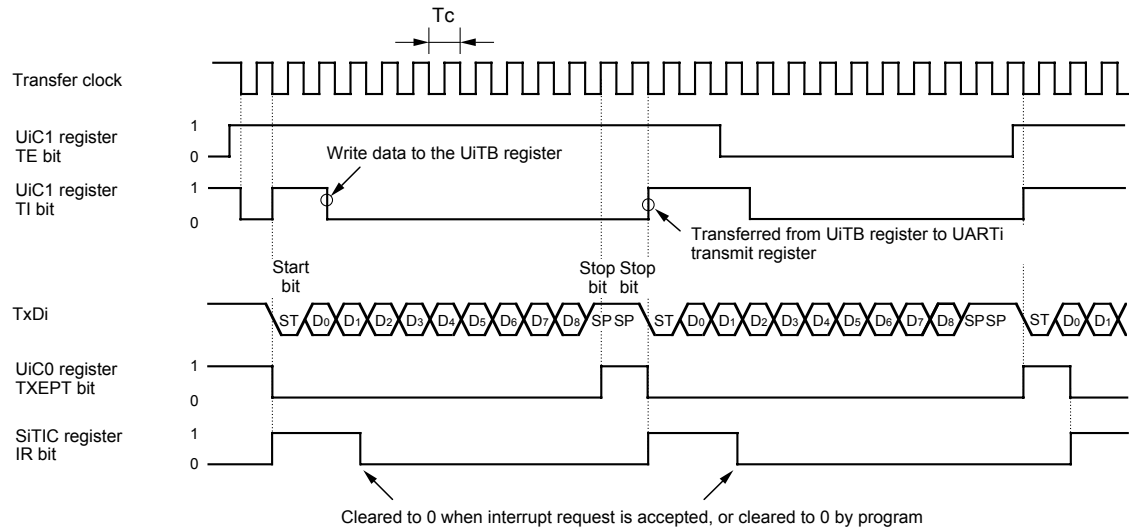
The above timing diagram applies to the case where the register bits are set as follows:

- Set the PRYE bit in the UiMR register to 1 (parity enabled)
- Set the STPS bit in the UiMR register to 0 (1 stop bit)
- Set the CRD bit in the UIC0 register to 0 (CTS/RTS enabled), the CRS bit to 0 (CTS selected).
- Set the UiIRS bit to 1 (an interrupt request occurs when transmit completed):
 U0IRS bit is the UCON register bit 0, U1IRS bit is the UCON register bit 1, and U2IRS bit is the U2C1 register bit 4

$$T_c = 16(n + 1) / f_j \text{ or } 16(n + 1) / f_{EXT}$$

f_j : frequency of UiBRG count source (f1SIO, f2SIO, f8SIO, f32SIO)
 f_{EXT} : frequency of UiBRG count source (external clock)
 n : value set to UiBRG
 $i = 0$ to 2

• Example of transmit timing when transfer data is 9-bit long (parity disabled, two stop bits)



The above timing diagram applies to the case where the register bits are set as follows:

- Set the PRYE bit in the UiMR register to 0 (parity disabled)
- Set the STPS bit in the UiMR register to 1 (2 stop bits)
- Set the CRD bit in the UIC0 register to 1 (CTS/RTS disabled)
- Set the UiIRS bit to 0 (an interrupt request occurs when transmit buffer becomes empty):
 U0IRS bit is the UCON register bit 0, U1IRS bit is the UCON register bit 1, and U2IRS bit is the U2C1 register bit 4

$$T_c = 16(n + 1) / f_j \text{ or } 16(n + 1) / f_{EXT}$$

f_j : frequency of UiBRG count source (f1SIO, f2SIO, f8SIO, f32SIO)
 f_{EXT} : frequency of UiBRG count source (external clock)
 n : value set to UiBRG
 $i = 0$ to 2

Figure 14.16 Typical transmit timing in UART mode (UART0, UART1)

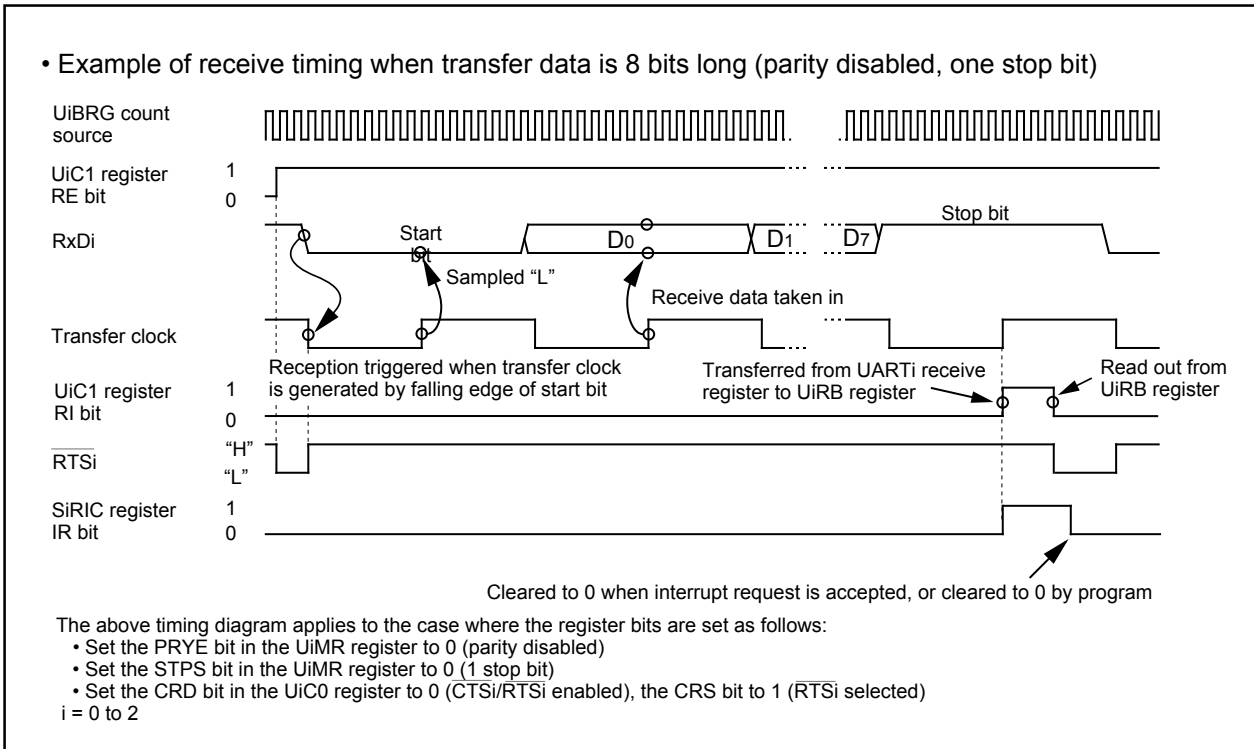


Figure 14.17 Receive Operation

14.1.2.1 Bit Rates

In UART mode, the frequency set by the UiBRG register (i=0 to 2) divided by 16 become the bit rates. Table 14.9 lists example of bit rate and settings.

Table 14.9 Example of Bit Rates and Settings

| Bit Rate (bps) | Count Source of BRG | Peripheral Function Clock : 16MHz | | Peripheral Function Clock : 20MHz | |
|----------------|---------------------|-----------------------------------|-------------------|-----------------------------------|-------------------|
| | | Set Value of BRG : n | Actual Time (bps) | Set Value of BRG : n | Actual Time (bps) |
| 1200 | f8 | 103(67h) | 1202 | 129(81h) | 1202 |
| 2400 | f8 | 51(33h) | 2404 | 64(40h) | 2404 |
| 4800 | f8 | 25(19h) | 4808 | 32(20h) | 4735 |
| 9600 | f1 | 103(67h) | 9615 | 129(81h) | 9615 |
| 14400 | f1 | 68(44h) | 14493 | 86(56h) | 14368 |
| 19200 | f1 | 51(33h) | 19231 | 64(40h) | 19231 |
| 28800 | f1 | 34(22h) | 28571 | 42(2Ah) | 29070 |
| 31250 | f1 | 31(1Fh) | 31250 | 39(27h) | 31250 |
| 38400 | f1 | 25(19h) | 38462 | 32(20h) | 37879 |
| 51200 | f1 | 19(13h) | 50000 | 24(18h) | 50000 |

14.1.2.2 Counter Measure for Communication Error

If a communication error occurs while transmitting or receiving in UART mode, follow the procedure below.

- Resetting the UiRB register (i=0 to 2)
 - (1) Set the RE bit in the UiC1 register to 0 (reception disabled)
 - (2) Set the RE bit in the UiC1 register to 1 (reception enabled)

- Resetting the UiTB register (i=0 to 2)
 - (1) Set bits SMD2 to SMD0 in UiMR register 0002 (Serial I/O disabled)
 - (2) Set bits SMD2 to SMD0 in UiMR register 0012, 1012, 1102
 - (3) 1 is written to TE bit in the UiC1 register (reception enabled), regardless of the TE bit

14.1.2.3 LSB First/MSB First Select Function

As shown in **Figure 14.18**, use the UFORM bit in the UiC0 register to select the transfer format. This function is valid when transfer data is 8 bits long.

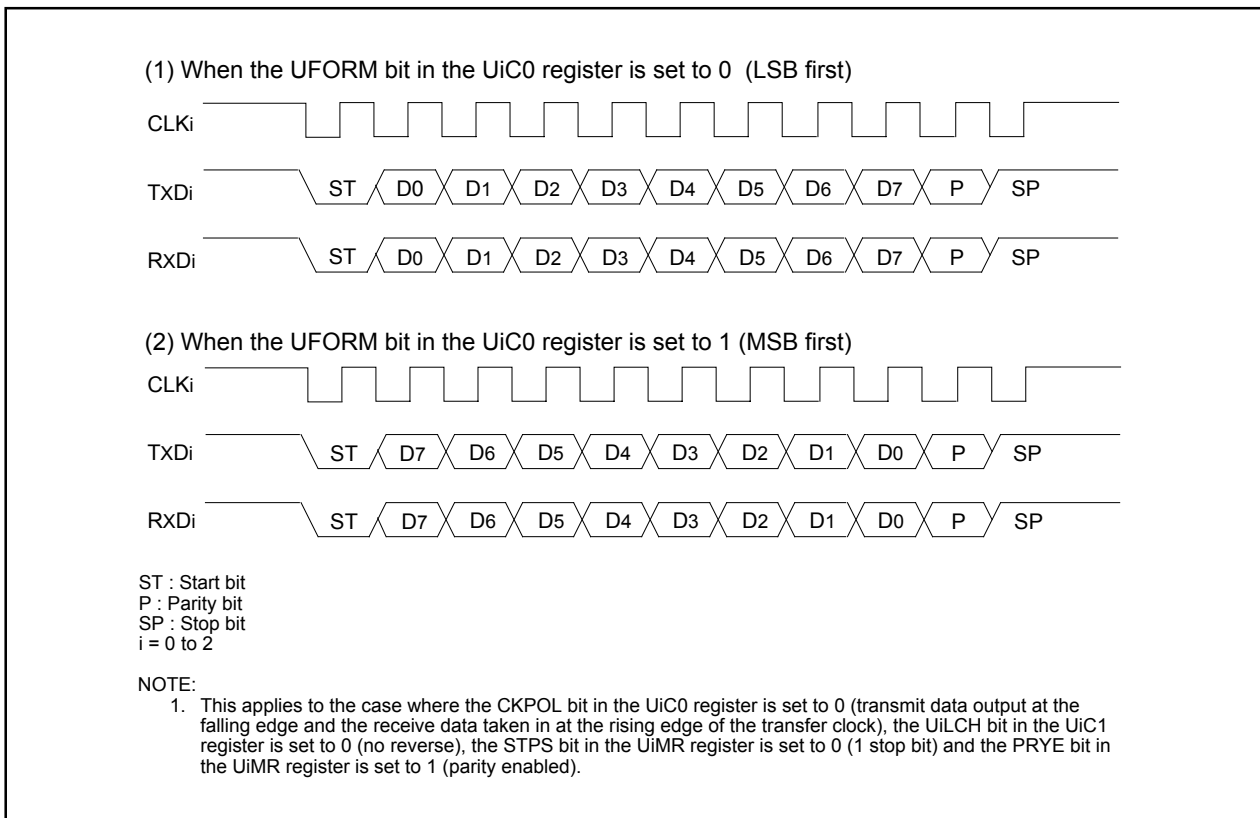


Figure 14.18 Transfer Format

14.1.2.4 Serial Data Logic Switching Function (UART2)

The data written to the U2TB register has its logic reversed before being transmitted. Similarly, the received data has its logic reversed when read from the U2RB register. **Figure 14.19** shows serial data logic.

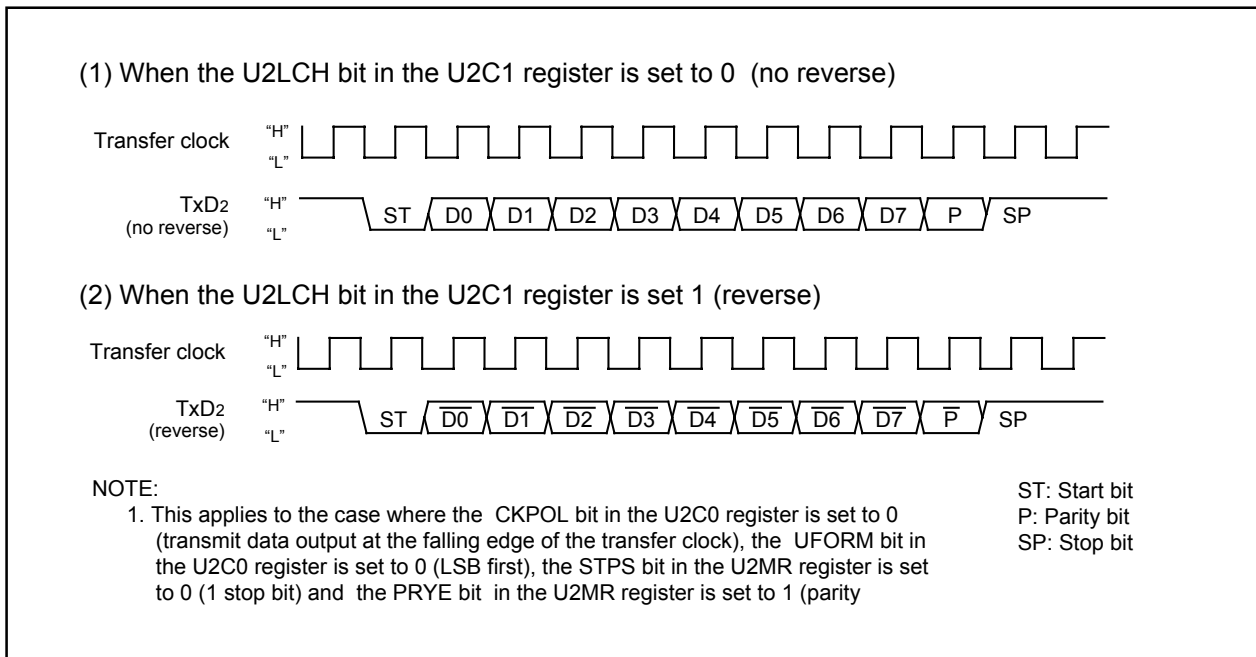


Figure 14.19 Serial Data Logic Switching

14.1.2.5 TxD and RxD I/O Polarity Inverse Function (UART2)

This function inverts the polarities of the TxD2 pin output and RxD2 pin input. The logic levels of all input/output data (including the start, stop and parity bits) are inverted. **Figure 14.20** shows the TxD pin output and RxD pin input polarity inverse.

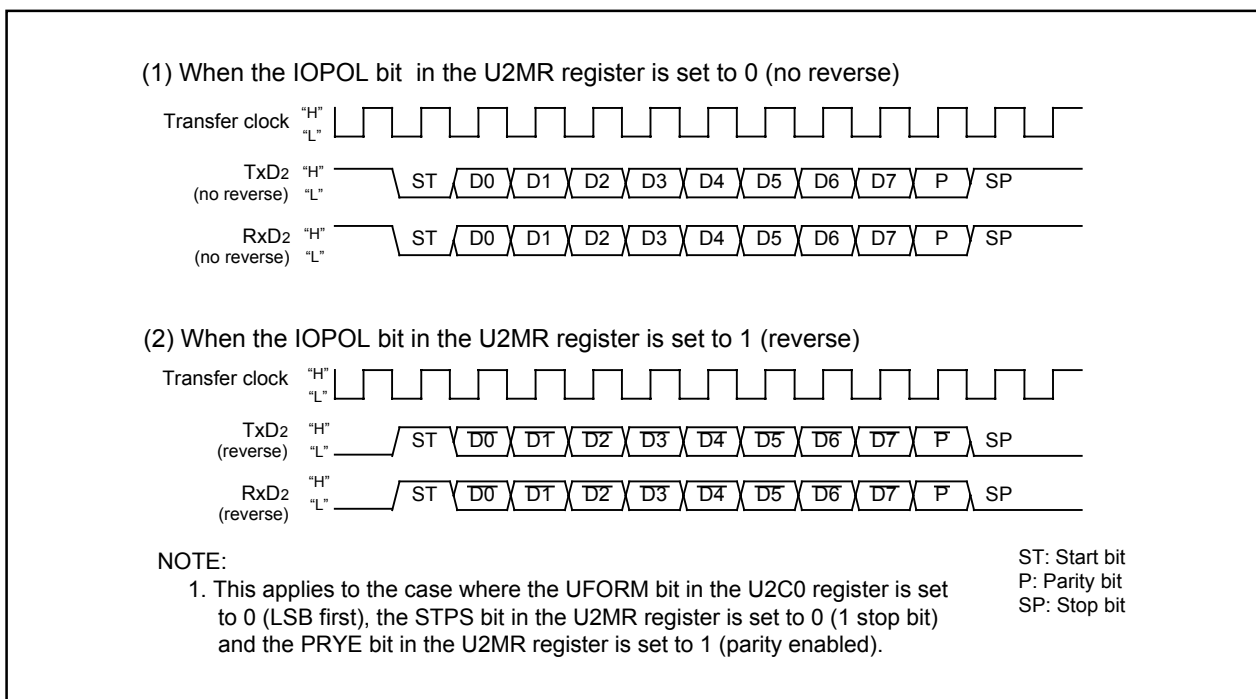


Figure 14.20 TxD and RxD I/O Polarity Inverse

14.1.2.6 CTS/RTS Separate Function (UART0)

This function separates $\overline{\text{CTS}}_0/\overline{\text{RTS}}_0$, outputs $\overline{\text{RTS}}_0$ from the P60 pin, and accepts as input the $\overline{\text{CTS}}_0$ from the P64 pin or P70 pin. To use this function, set the register bits as shown below.

- The CRD bit in the U0C0 register is set to 0 (enables UART0 $\overline{\text{CTS}}/\overline{\text{RTS}}$)
- The CRS bit in the U0C0 register is set to 1 (outputs UART0 $\overline{\text{RTS}}$)
- The CRD bit in the U1C0 register is set to 0 (enables UART1 $\overline{\text{CTS}}/\overline{\text{RTS}}$)
- The CRS bit in the U1C0 register is set to 0 (inputs UART1 $\overline{\text{CTS}}$)
- The RCSP bit in the UCON register is set to 1 (inputs $\overline{\text{CTS}}_0$ from the P64 pin or P70 pin)
- The CLKMD1 bit in the UCON register is set to 0 (CLKS1 not used)

Note that when using the $\overline{\text{CTS}}/\overline{\text{RTS}}$ separate function, UART1 $\overline{\text{CTS}}/\overline{\text{RTS}}$ separate function cannot be used.

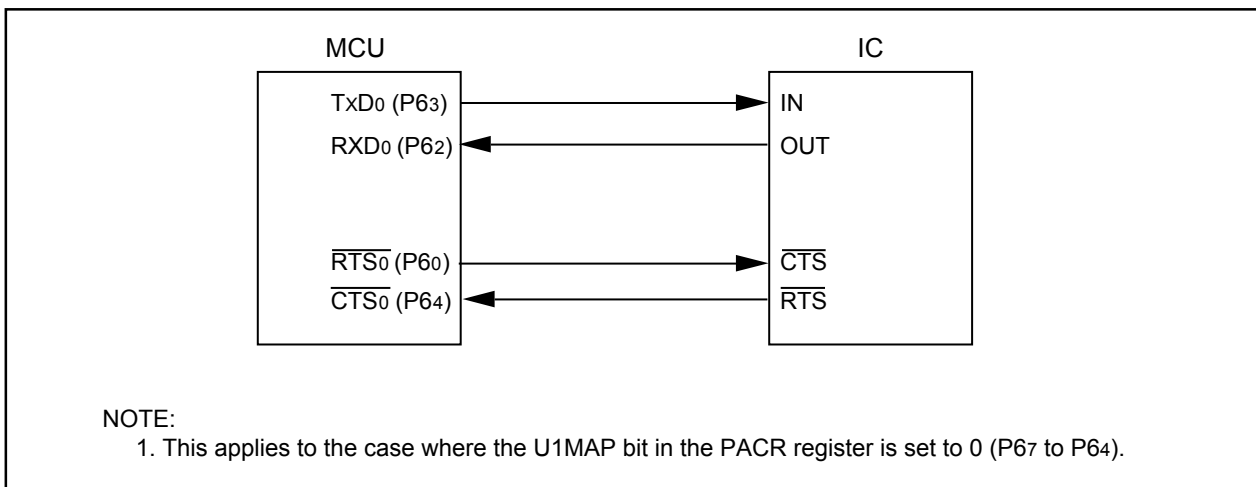


Figure 14.21 $\overline{\text{CTS}}/\overline{\text{RTS}}$ Separate Function

14.1.3 Special Mode 1 (I²C bus mode)(UART2)

I²C bus mode is provided for use as a simplified I²C bus interface compatible mode. **Table 14.10** lists the specifications of the I²C bus mode. **Tables 14.11** and **14.12** list the registers used in the I²C bus mode and the register values set. **Table 14.13** lists the I²C bus mode functions. **Figure 14.22** shows the block diagram for I²C bus mode. **Figure 14.23** shows SCL2 timing.

As shown in **Table 14.13**, the MCU is placed in I²C bus mode by setting bits SMD2 to SMD0 to 0102 and the IICM bit to 1. Because SDA2 transmit output has a delay circuit attached, SDA output does not change state until SCL2 goes low and remains stably low.

Table 14.10 I²C bus mode Specifications

| Item | Specification |
|-------------------------------------|--|
| Transfer data format | • Transfer data length: 8 bits |
| Transfer clock | <ul style="list-style-type: none"> • During master the CKDIR bit in the U2MR register is set to 0 (internal clock) : $f_j / (2(n+1))$ f_j = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}. n: Setting value in the U2BRG register 00₁₆ to FF₁₆ • During slave CKDIR bit is set to 1 (external clock) : Input from SCL2 pin |
| Transmission start condition | <ul style="list-style-type: none"> • Before transmission can start, the following requirements must be met ⁽¹⁾ <ul style="list-style-type: none"> – The TE bit in the U2C1 register is set to 1 (transmission enabled) – The TI bit in the U2C1 register is set to 0 (data present in U2TB register) |
| Reception start condition | <ul style="list-style-type: none"> • Before reception can start, the following requirements must be met ⁽¹⁾ <ul style="list-style-type: none"> – The RE bit in the U2C1 register is set to 1 (reception enabled) – The TE bit in the U2C1 register is set to 1 (transmission enabled) – The TI bit in the U2C1 register is set to 0 (data present in the UiTB register) |
| Interrupt request generation timing | When start or stop condition is detected, acknowledge undetected, and acknowledge detected |
| Error detection | <ul style="list-style-type: none"> • Overrun error ⁽²⁾ This error occurs if the serial I/O started receiving the next data before reading the U2RB register and received the 8th bit in the the next data |
| Select function | <ul style="list-style-type: none"> • Arbitration lost Timing at which the ABT bit in the U2RB register is updated can be selected • SDA digital delay No digital delay or a delay of 2 to 8 U2BRG count source clock cycles selectable • Clock phase setting With or without clock delay selectable |

NOTES:

1. When an external clock is selected, the conditions must be met while the external clock is in the high state.
2. If an overrun error occurs, bits 8 to 0 in the U2RB register are undefined. The IR bit in the S2RIC register remains unchange.

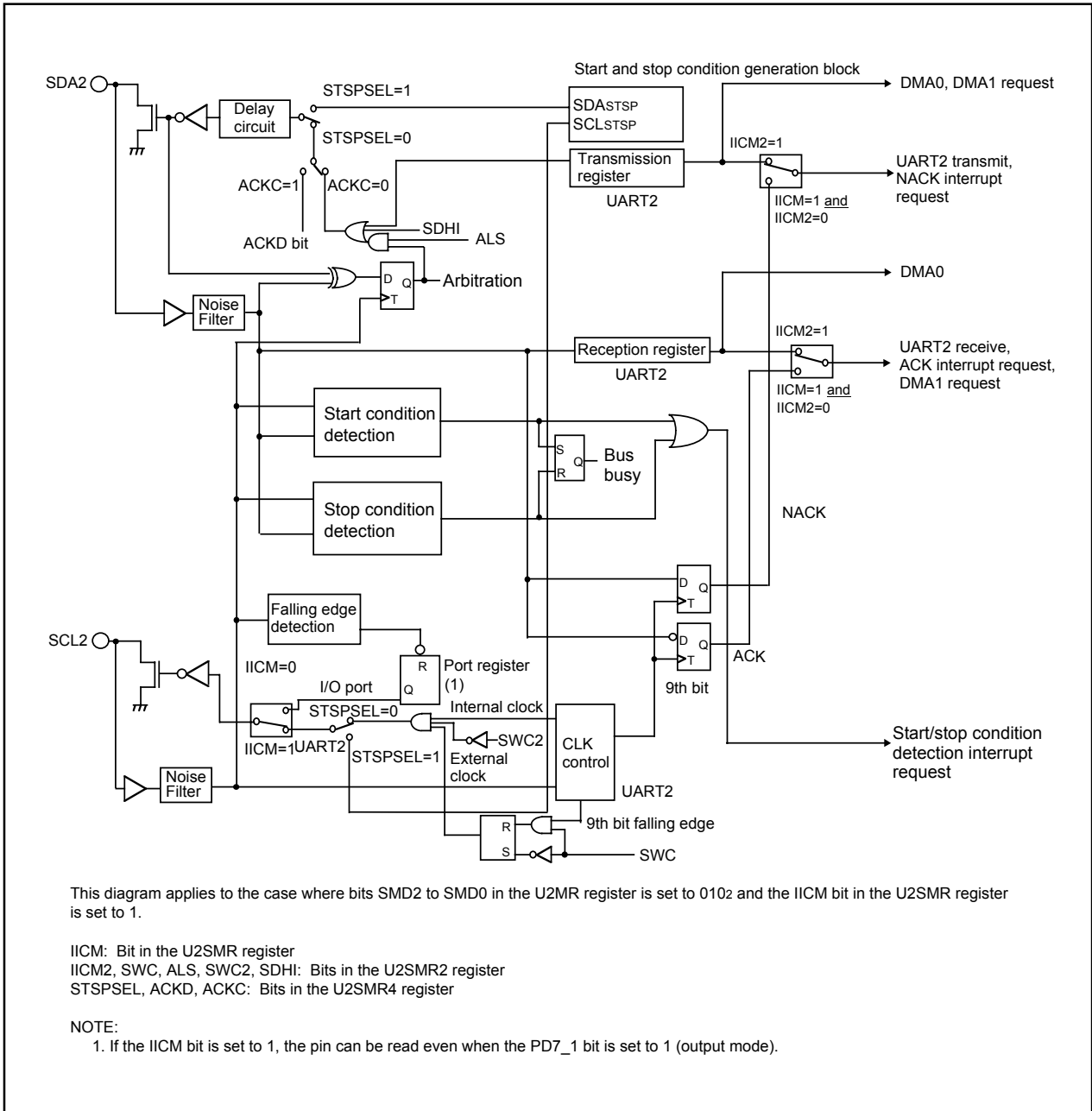


Figure 14.22 I²C bus mode Block Diagram

Table 14.11 Registers to Be Used and Settings in I²C bus mode (1) (Continued)

| Register | Bit | Function | |
|---------------------|---------------------|--|--|
| | | Master | Slave |
| U2TB | 0 to 7 | Set transmission data | Set transmission data |
| U2RB ⁽¹⁾ | 0 to 7 | Reception data can be read | Reception data can be read |
| | 8 | ACK or NACK is set in this bit | ACK or NACK is set in this bit |
| | ABT | Arbitration lost detection flag | Invalid |
| | OER | Overrun error flag | Overrun error flag |
| U2BRG | 0 to 7 | Set bit rate | Invalid |
| U2MR ⁽¹⁾ | SMD2 to SMD0 | Set to 0102 | Set to 0102 |
| | CKDIR | Set to 0 | Set to 1 |
| | IOPOL | Set to 0 | Set to 0 |
| U2C0 | CLK1, CLK0 | Select the count source for the U2BRG register | Invalid |
| | CRS | Invalid because CRD = 1 | Invalid because CRD = 1 |
| | TXEPT | Transmit buffer empty flag | Transmit buffer empty flag |
| | CRD | Set to 1 | Set to 1 |
| | NCH | Set to 1 | Set to 1 |
| | CKPOL | Set to 0 | Set to 0 |
| | UFORM | Set to 1 | Set to 1 |
| U2C1 | TE | Set this bit to 1 to enable transmission | Set this bit to 1 to enable transmission |
| | TI | Transmit buffer empty flag | Transmit buffer empty flag |
| | RE | Set this bit to 1 to enable reception | Set this bit to 1 to enable reception |
| | RI | Reception complete flag | Reception complete flag |
| | U2IRS | Invalid | Invalid |
| | U2RRM, U2LCH, U2ERE | Set to 0 | Set to 0 |
| U2SMR | IICM | Set to 1 | Set to 1 |
| | ABC | Select the timing at which arbitration-lost is detected | Invalid |
| | BBS | Bus busy flag | Bus busy flag |
| | 3 to 7 | Set to 0 | Set to 0 |
| U2SMR2 | IICM2 | Refer to Table 14.13 | Refer to Table 14.13 |
| | CSC | Set this bit to 1 to enable clock synchronization | Set to 0 |
| | SWC | Set this bit to 1 to have SCL2 output fixed to L at the falling edge of the 9th bit of clock | Set this bit to 1 to have SCL2 output fixed to "L" at the falling edge of the 9th bit of clock |
| | ALS | Set this bit to 1 to have SDA2 output stopped when arbitration-lost is detected | Set to 0 |
| | STAC | Set to 0 | Set this bit to 1 to initialize UART2 at start condition detection |
| | SWC2 | Set this bit to 1 to have SCL2 output forcibly pulled low | Set this bit to 1 to have SCL2 output forcibly pulled low |
| | SDHI | Set this bit to 1 to disable SDA2 output | Set this bit to 1 to disable SDA2 output |
| | 7 | Set to 0 | Set to 0 |
| U2SMR3 | 0, 2, 4 and NODC | Set to 0 | Set to 0 |
| | CKPH | Refer to Table 14.13 | Refer to Table 14.13 |
| | DL2 to DL0 | Set the amount of SDA2 digital delay | Set the amount of SDA2 digital delay |

NOTE:

1. Not all bits in the register are described above. Set those bits to 0 when writing to the registers in I²C bus mode.

Table 14.12 Registers to Be Used and Settings in I²C bus Mode (2) (Continued)

| Register | Bit | Function | |
|----------|---------|---|---|
| | | Master | Slave |
| U2SMR4 | STAREQ | Set this bit to 1 to generate start condition | Set to 0 |
| | RSTAREQ | Set this bit to 1 to generate restart condition | Set to 0 |
| | STPREQ | Set this bit to 1 to generate stop condition | Set to 0 |
| | STSPSEL | Set this bit to 1 to output each condition | Set to 0 |
| | ACKD | Select ACK or NACK | Select ACK or NACK |
| | ACKC | Set this bit to 1 to output ACK data | Set this bit to 1 to output ACK data |
| | SCLHI | Set this bit to 1 to have SCL2 output stopped when stop condition is detected | Set to 0 |
| | SWC9 | Set to 0 | Set this bit to 1 to set the SCL2 to "L" hold at the falling edge of the 9th bit of clock |

NOTE:

1: Not all bits in the register are described above. Set those bits to 0 when writing to the registers in I²C bus mode.

Table 14.13 I²C bus mode Functions

| Function | Clock synchronous serial I/O mode (SMD2 to SMD0 = 0012, IICM = 0) | I ² C bus mode (SMD2 to SMD0 = 0102, IICM = 1) | | | |
|--|---|--|--|--|---------------------------|
| | | IICM2 = 0 (NACK/ACK interrupt) | | IICM2 = 1 (UART transmit/ receive interrupt) | |
| | | CKPH = 0 (No clock delay) | CKPH = 1 (Clock delay) | CKPH = 0 (No clock delay) | CKPH = 1 (Clock delay) |
| Factor of interrupt number 10 ⁽¹⁾ (Refer to Fig.14.23) | ————— | Start condition detection or stop condition detection (Refer to Table 14.14) | | | |
| Factor of interrupt number 15 ⁽¹⁾ (Refer to Fig.14.23) | UART2 transmission Transmission started or completed (selected by U2IRS) | No acknowledgment detection (NACK) Rising edge of SCL2 9th bit | UART2 transmission Rising edge of SCL2 9th bit | UART2 transmission Falling edge of SCL2 next to the 9th bit | |
| Factor of interrupt number 16 ⁽¹⁾ (Refer to Fig.14.23) | UART2 reception When 8th bit received CKPOL = 0 (rising edge) CKPOL = 1 (falling edge) | Acknowledgment detection (ACK) Rising edge of SCL2 9th bit | UART2 transmission Falling edge of SCL2 9th bit | | |
| Timing for transferring data from the UART reception shift register to the U2RB register | CKPOL = 0 (rising edge) CKPOL = 1 (falling edge) | Rising edge of SCL2 9th bit | Falling edge of SCL2 9th bit | Falling and rising edges of SCL2 9th bit | |
| UART2 transmission output delay | Not delayed | Delayed | | | |
| Functions of P7 ₀ pin | TxD2 output | SDA2 input/output | | | |
| Functions of P7 ₁ pin | RxD2 input | SCL2 input/output | | | |
| Functions of P7 ₂ pin | CLK2 input or output selected | ————— (Cannot be used in I ² C bus mode) | | | |
| Noise filter width | 15ns | 200ns | | | |
| Read RxD2 and SCL2 pin levels | Possible when the corresponding port direction bit = 0 | Always possible no matter how the corresponding port direction bit is set | | | |
| Initial value of TxD2 and SDA2 outputs | CKPOL = 0 (H) CKPOL = 1 (L) | The value set in the port register before setting I ² C bus mode ⁽²⁾ | | | |
| Initial and end values of SCL2 | ————— | H | L | H | |
| DMA1 factor (Refer to Fig. 14.23) | UART2 reception | Acknowledgment detection (ACK) | UART2 reception Falling edge of SCL2 9th bit | | |
| Store received data | 1st to 8th bits are stored in bits bit 7 to 0 in the U2RB register | 1st to 8th bits are stored in bits bit 7 to 0 in the U2RB register | 1st to 7th bits are stored into the bit 6 to bit 0 in the U2RB register, with 8th bit stored in the bit 8 in the U2RB register | | |
| | | | 1st to 8th bits are stored in U2RB register bit 7 to bit 0 ⁽³⁾ | | |
| Read received data | U2RB register status is read directly as is | | | Read U2RB register Bit 6 to bit 0 as bit 7 to bit 1, and bit 8 as bit 0 ⁽⁴⁾ | |

NOTES:

1. If the source or cause of any interrupt is changed, the IR bit in the interrupt control register for the changed interrupt may inadvertently be set to 1 (interrupt requested). (Refer to "Notes on interrupts" in Precautions.)
If one of the bits shown below is changed, the interrupt source, the interrupt timing, etc. change. Therefore, always be sure to clear the IR bit to 0 (interrupt not requested) after changing those bits
Bits SMD2 to the SMD0 in the U2MR register, the IICM bit in the U2SMR register,
the IICM2 bit in the U2SMR2 register, the CKPH bit in the U2SMR3 register
2. Set the initial value of SDA2 output while bits SMD2 to SMD0 in the U2MR register is set to 0002 (serial I/O disabled).
3. Second data transfer to U2RB register (Rising edge of SCL2 9th bit)
4. First data transfer to U2RB register (Falling edge of SCL2 9th bit)

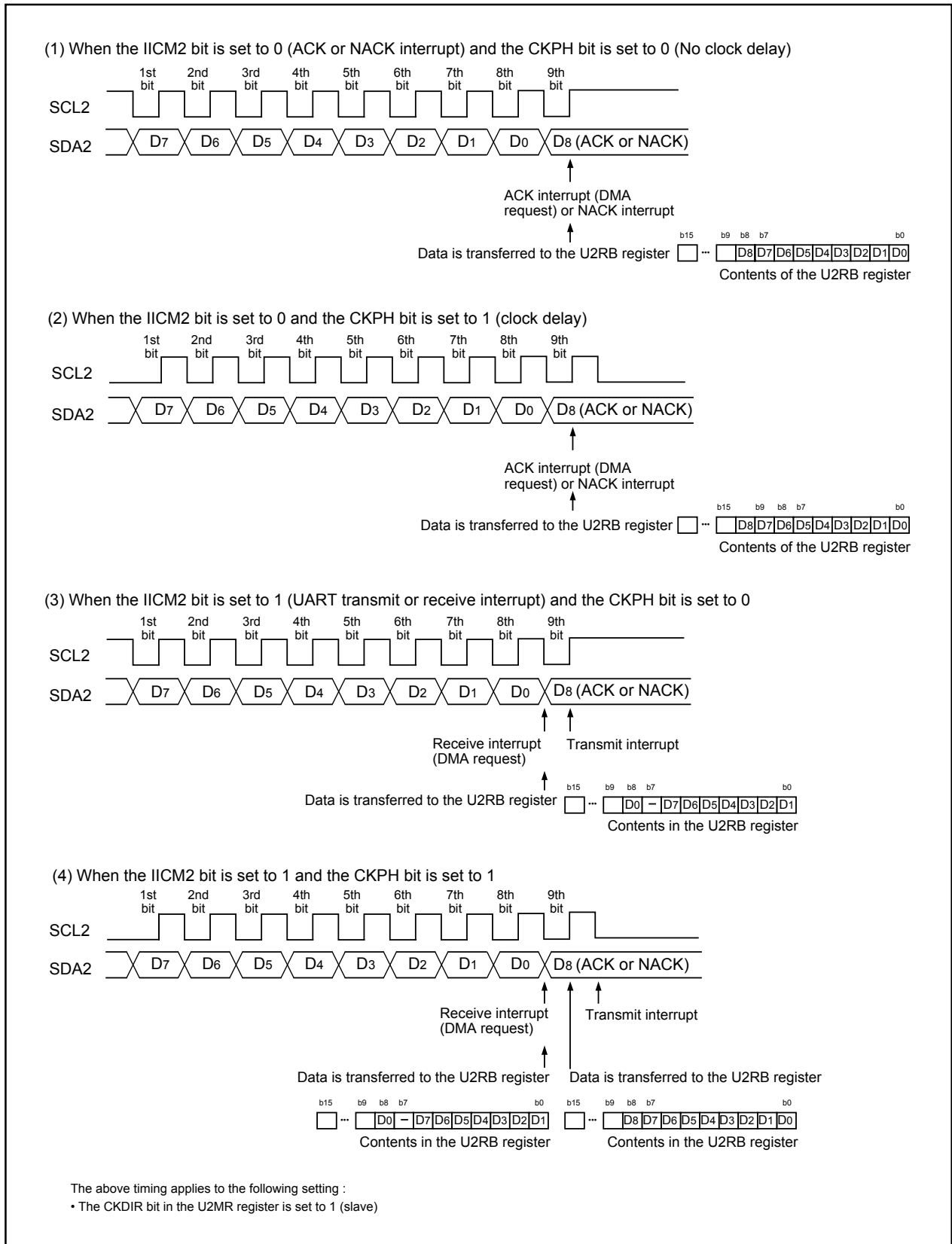


Figure 14.23 Transfer to U2RB Register and Interrupt Timing

14.1.3.1 Detection of Start and Stop Condition

Whether a start or a stop condition has been detected is determined.

A start condition-detected interrupt request is generated when the SDA2 pin changes state from high to low while the SCL2 pin is in the high state. A stop condition-detected interrupt request is generated when the SDA2 pin changes state from low to high while the SCL2 pin is in the high state.

Because the start and stop condition-detected interrupts share the interrupt control register and vector, check the BBS bit in the U2SMR register to determine which interrupt source is requesting the interrupt.

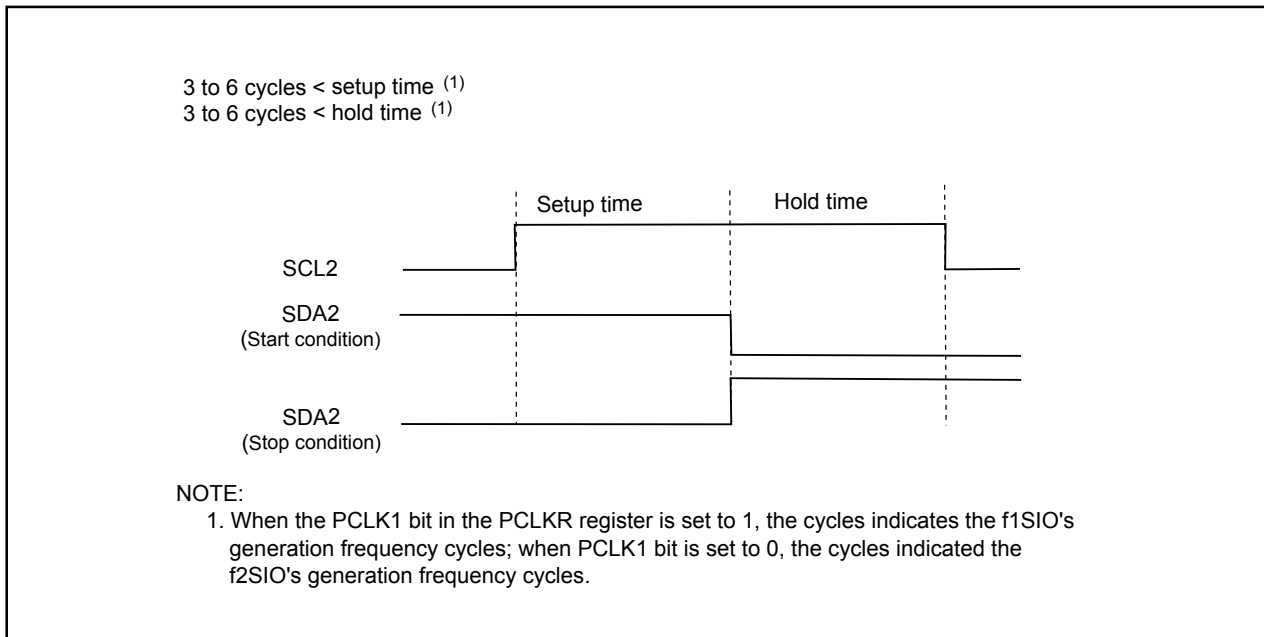


Figure 14.24 Detection of Start and Stop Condition

14.1.3.2 Output of Start and Stop Condition

A start condition is generated by setting the STAREQ bit in the U2SMR4 register to 1 (start).

A restart condition is generated by setting the RSTAREQ bit in the U2SMR4 register to 1 (start).

A stop condition is generated by setting the STPREQ bit in the U2SMR4 register to 1 (start).

The output procedure is described below.

(1) Set the STAREQ bit, RSTAREQ bit or STPREQ bit to 1 (start).

(2) Set the STSPSEL bit in the U2SMR4 register to 1 (output).

Make sure that no interrupts or DMA transfers will occur between (1) and (2).

The function of the STSPSEL bit is shown in **Table 14.14** and **Figure 14.25**.

Table 14.14 STSPSEL Bit Functions

| Function | STSPSEL = 0 | STSPSEL = 1 |
|--|---|--|
| Output of SCL2 and SDA2 pins | Output transfer clock and data/Program with a port determines how the start condition or stop condition is output | The STAREQ, RSTAREQ and STPREQ bit determine how the start condition or stop condition is output |
| Start/stop condition interrupt request generation timing | Start/stop condition are detected | Start/stop condition generation are completed |

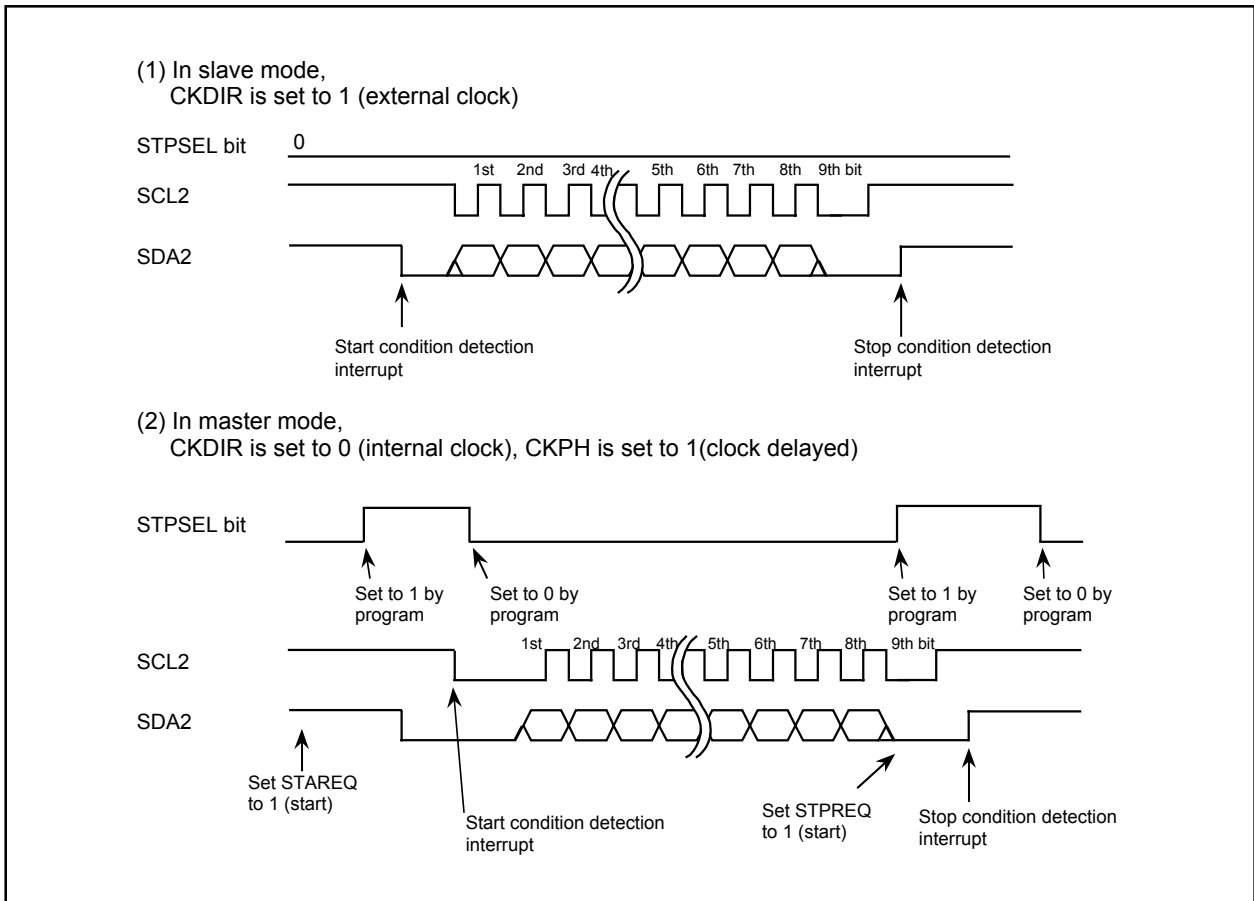


Figure 14.25 STSPSEL Bit Functions

14.1.3.3 Arbitration

Unmatching of the transmit data and SDA2 pin input data is checked synchronously with the rising edge of SCL2. Use the ABC bit in the U2SMR register to select the timing at which the ABT bit in the U2RB register is updated. If the ABC bit is set to 0 (updated bitwise), the ABT bit is set to 1 at the same time unmatching is detected during check, and is cleared to 0 when not detected. In cases when the ABC bit is set to 1, if unmatching is detected even once during check, the ABT bit is set to 1 (unmatching detected) at the falling edge of the clock pulse of 9th bit. If the ABT bit needs to be updated bitwise, clear the ABT bit to 0 (undetected) after detecting acknowledge in the first byte, before transferring the next byte.

Setting the ALS bit in the U2SMR2 register to 1 (SDA2 output stop enabled) causes arbitration-lost to occur, in which case the SDA2 pin is placed in the high-impedance state at the same time the ABT bit is set to 1 (unmatching detected).

14.1.3.4 Transfer Clock

Data is transmitted/received using a transfer clock like the one shown in **Figure 14.25**.

The CSC bit in the U2SMR2 register is used to synchronize the internally generated clock (internal SCL2) and an external clock supplied to the SCL2 pin. In cases when the CSC bit is set to 1 (clock synchronization enabled), if a falling edge on the SCL2 pin is detected while the internal SCL2 is high, the internal SCL2 goes low, at which time the U2BRG register value is reloaded with and starts counting in the low-level interval. If the internal SCL2 changes state from low to high while the SCL2 pin is low, counting stops, and when the SCL2 pin goes high, counting restarts.

In this way, the UART2 transfer clock is comprised of the logical product of the internal SCL2 and SCL2 pin signal. The transfer clock works from a half period before the falling edge of the internal SCL2 1st bit to the rising edge of the 9th bit. To use this function, select an internal clock for the transfer clock. The SWC bit in the U2SMR2 register allows to select whether the SCL2 pin should be fixed to or freed from low-level output at the falling edge of the 9th clock pulse.

If the SCLHI bit in the U2SMR4 register is set to 1 (enabled), SCL2 output is turned off (placed in the high-impedance state) when a stop condition is detected.

Setting the SWC2 bit in the U2SMR2 register is set to 1 (0 output) makes it possible to forcibly output a low-level signal from the SCL2 pin even while sending or receiving data. Clearing the SWC2 bit to 0 (transfer clock) allows the transfer clock to be output from or supplied to the SCL2 pin, instead of outputting a low-level signal.

If the SWC9 bit in the U2SMR4 register is set to 1 (SCL2 hold low enabled) when the CKPH bit in the U2SMR3 register is set to 1, the SCL2 pin is fixed to low-level output at the falling edge of the clock pulse next to the ninth. Setting the SWC9 bit to 0 (SCL2 hold low disabled) frees the SCL2 pin from low-level output.

14.1.3.5 SDA Output

The data written to the bit 7 to bit 0 (D7 to D0) in the U2TB register is sequentially output beginning with D7. The ninth bit (D8) is ACK or NACK.

The initial value of SDA2 transmit output can only be set when IICM is set to 1 (I²C bus mode) and bits SMD2 to SMD0 in the U2MR register is set to 0002 (serial I/O disabled).

Bits DL2 to DL0 in the U2SMR3 register allow to add no delays or a delay of 2 to 8 U2BRG count source clock cycles to SDA2 output.

Setting the SDHI bit in the U2SMR2 register to 1 (SDA2 output disabled) forcibly places the SDA2 pin in the high-impedance state. Do not write to the SDHI bit synchronously with the rising edge of the UART2 transfer clock. This is because the ABT bit may inadvertently be set to 1 (detected).

14.1.3.6 SDA Input

When the IICM2 bit is set to 0, the 1st to 8th bits (D7 to D0) in the received data are stored in bits 7 to 0 in the U2RB register. The 9th bit (D8) is ACK or NACK.

When the IICM2 bit is set to 1, the 1st to 7th bits (D7 to D1) in the received data are stored in the bit 6 to bit 0 in the U2RB register and the 8th bit (D0) is stored in the bit 8 in the U2RB register. Even when the IICM2 bit is set to 1, providing the CKPH bit is set to 1, the same data as when the IICM2 bit is set to 0 can be read out by reading the U2RB register after the rising edge of the corresponding clock pulse of 9th bit.

14.1.3.7 ACK and NACK

If the STSPSEL bit in the U2SMR4 register is set to 0 (start and stop conditions not generated) and the ACKC bit in the U2SMR4 register is set to 1 (ACK data output), the value of the ACKD bit in the U2SMR4 register is output from the SDA2 pin.

If the IICM2 bit is set to 0, a NACK interrupt request is generated if the SDA2 pin remains high at the rising edge of the 9th bit of transmit clock pulse. An ACK interrupt request is generated if the SDA2 pin is low at the rising edge of the 9th bit of transmit clock pulse.

If ACK2 is selected for the cause of DMA1 request, a DMA transfer can be activated by detection of an acknowledge.

14.1.3.8 Initialization of Transmission/Reception

If a start condition is detected while the STAC bit is set to 1 (UART2 initialization enabled), the serial I/O operates as described below.

- The transmit shift register is initialized, and the content of the U2TB register is transferred to the transmit shift register. In this way, the serial I/O starts sending data synchronously with the next clock pulse applied. However, the UART2 output value does not change state and remains the same as when a start condition was detected until the first bit in the data is output synchronously with the input clock.
- The receive shift register is initialized, and the serial I/O starts receiving data synchronously with the next clock pulse applied.
- The SWC bit is set to 1 (SCL2 wait output enabled). Consequently, the SCL2 pin is pulled low at the falling edge of the ninth clock pulse.

Note that when UART2 transmission/reception is started using this function, the TI does not change state. Note also that when using this function, the selected transfer clock should be an external clock.

14.1.4 Special Mode 2 (UART2)

Multiple slaves can be serially communicated from one master. Transfer clock polarity and phase are selectable. **Table 14.15** lists the specifications of Special Mode 2. **Table 14.16** lists the registers used in Special Mode 2 and the register values set. **Figure 14.26** shows communication control example for Special Mode 2.

Table 14.15 Special Mode 2 Specifications

| Item | Specification |
|-------------------------------------|--|
| Transfer data format | <ul style="list-style-type: none"> Transfer data length: 8 bits |
| Transfer clock | <ul style="list-style-type: none"> Master mode the CKDIR bit in the U2MR register is set to 0 (internal clock) : $f_j / (2(n+1))$ $f_j = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}$. n: Setting value in the U2BRG register 00₁₆ to FF₁₆ Slave mode CKDIR bit is set to 1 (external clock selected) : Input from CLK2 pin |
| Transmit/receive control | Controlled by input/output ports |
| Transmission start condition | <ul style="list-style-type: none"> Before transmission can start, the following requirements must be met ⁽¹⁾ <ul style="list-style-type: none"> The TE bit in the U2C1 register is set to 1 (transmission enabled) The TI bit in the U2C1 register is set to 0 (data present in U2TB register) |
| Reception start condition | <ul style="list-style-type: none"> Before reception can start, the following requirements must be met ⁽¹⁾ <ul style="list-style-type: none"> The RE bit in the U2C1 register is set to 1 (reception enabled) The TE bit in the U2C1 register is set to 1 (transmission enabled) The TI bit in the U2C1 register is set to 0 (data present in the U2TB register) |
| Interrupt request generation timing | <ul style="list-style-type: none"> For transmission, one of the following conditions can be selected <ul style="list-style-type: none"> The U2IRS bit in the U2C1 register is set to 0 (transmit buffer empty): when transferring data from the U2TB register to the UART2 transmit register (at start of transmission) The U2IRS bit is set to 1 (transfer completed): when the serial I/O finished sending data from the UART2 transmit register For reception When transferring data from the UART2 receive register to the U2RB register (at completion of reception) |
| Error detection | <ul style="list-style-type: none"> Overrun error ⁽²⁾ This error occurs if the serial I/O started receiving the next data before reading the U2RB register and received the 7th bit in the the next data |
| Select function | <ul style="list-style-type: none"> Clock phase setting Selectable from four combinations of transfer clock polarities and phases |

NOTES:

- When an external clock is selected, the conditions must be met while if the CKPOL bit in the U2C0 register is set to 0 (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock), the external clock is in the high state; if the CKPOL bit in the U2C0 register is set to 1 (transmit data output at the rising edge and the receive data taken in at the falling edge of the transfer clock), the external clock is in the low state.
- If an overrun error occurs, bits 8 to 0 in the U2RB register are undefined. The IR bit in the S2RIC register remains unchanged.

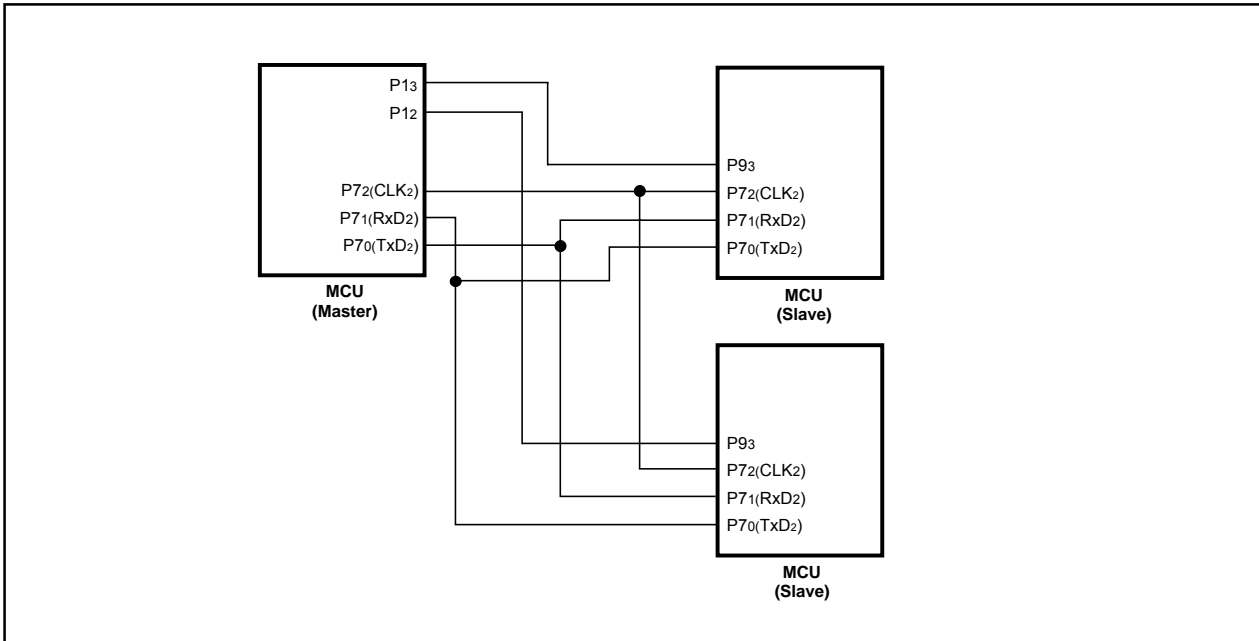


Figure 14.26 Serial Bus Communication Control Example (UART2)

Table 14.16 Registers to Be Used and Settings in Special Mode 2

| Register | Bit | Function |
|---------------------|---------------------|---|
| U2TB ⁽¹⁾ | 0 to 7 | Set transmission data |
| U2RB ⁽¹⁾ | 0 to 7 | Reception data can be read |
| | OER | Overrun error flag |
| U2BRG | 0 to 7 | Set bit rate |
| U2MR ⁽¹⁾ | SMD2 to SMD0 | Set to 0012 |
| | CKDIR | Set this bit to 0 for master mode or 1 for slave mode |
| | IOPOL | Set to 0 |
| U2C0 | CLK1, CLK0 | Select the count source for the U2BRG register |
| | CRS | Invalid because CRD is set to 1 |
| | TXEPT | Transmit register empty flag |
| | CRD | Set to 1 |
| | NCH | Select TxD2 pin output format |
| | CKPOL | Clock phases can be set in combination with the CKPH bit in the U2SMR3 register |
| U2C1 | UFORM | Select the LSB first or MSB first |
| | TE | Set this bit to 1 to enable transmission |
| | TI | Transmit buffer empty flag |
| | RE | Set this bit to 1 to enable reception |
| | RI | Reception complete flag |
| | U2IRS | Select UART2 transmit interrupt cause |
| U2SMR | U2RRM, U2LCH, U2ERE | Set to 0 |
| | 0 to 7 | Set to 0 |
| | 0 to 7 | Set to 0 |
| U2SMR3 | CKPH | Clock phases can be set in combination with the CKPOL bit in the U2C0 register |
| | NODC | Set to 0 |
| | 0, 2, 4 to 7 | Set to 0 |
| U2SMR4 | 0 to 7 | Set to 0 |

NOTE:

1. Not all bits in the registers are described above. Set those bits to 0 when writing to the registers in Special Mode 2.

14.1.4.1 Clock Phase Setting Function

One of four combinations of transfer clock phases and polarities can be selected using the CKPH bit in the U2SMR3 register and the CKPOL bit in the U2C0 register.

Make sure the transfer clock polarity and phase are the same for the master and slave to communicate.

14.1.4.1.1 Master (Internal Clock)

Figure 14.27 shows the transmission and reception timing in master (internal clock).

14.1.4.1.2 Slave (External Clock)

Figure 14.28 shows the transmission and reception timing (CKPH=0) in slave (external clock) while Figure 14.29 shows the transmission and reception timing (CKPH=1) in slave (external clock).

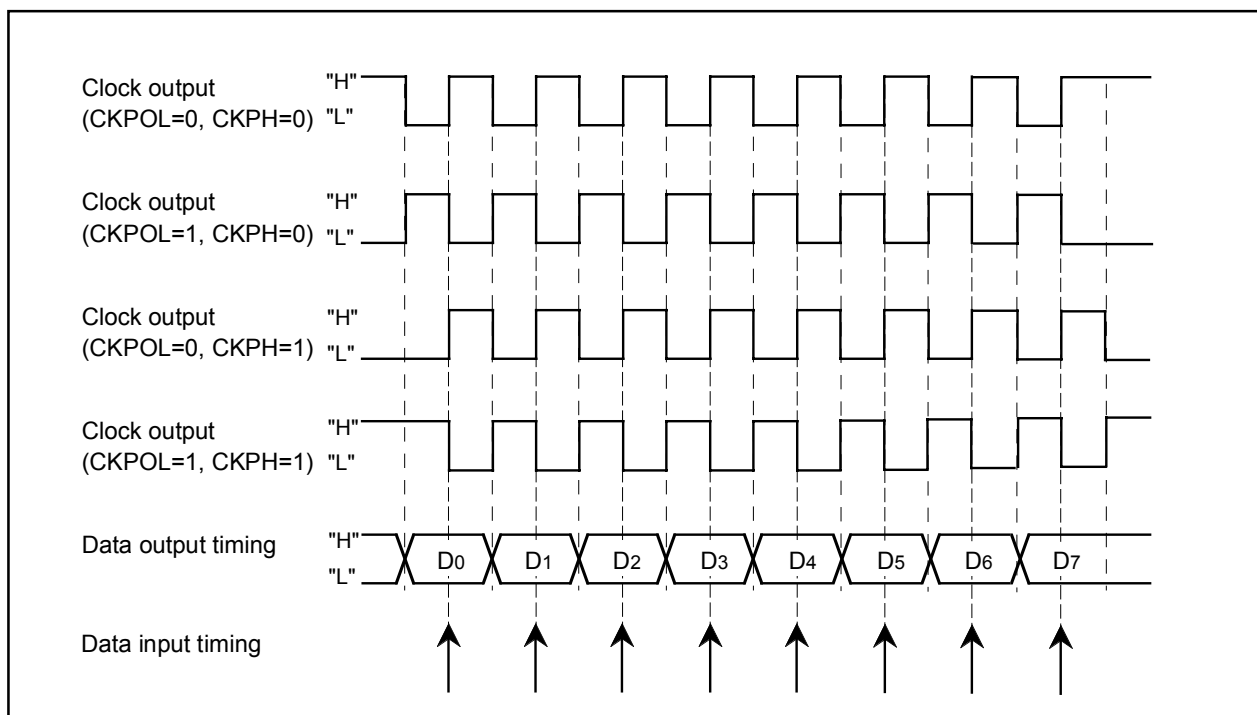


Figure 14.27 Transmission and Reception Timing in Master Mode (Internal Clock)

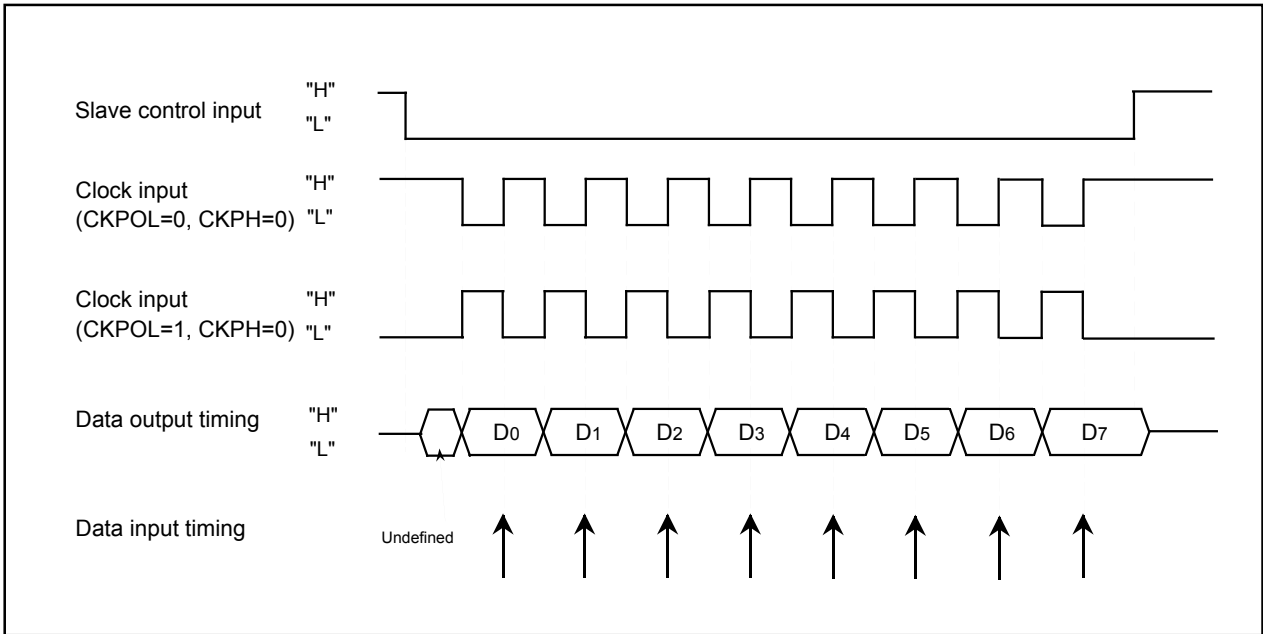


Figure 14.28 Transmission and Reception Timing (CKPH=0) in Slave Mode (External Clock)

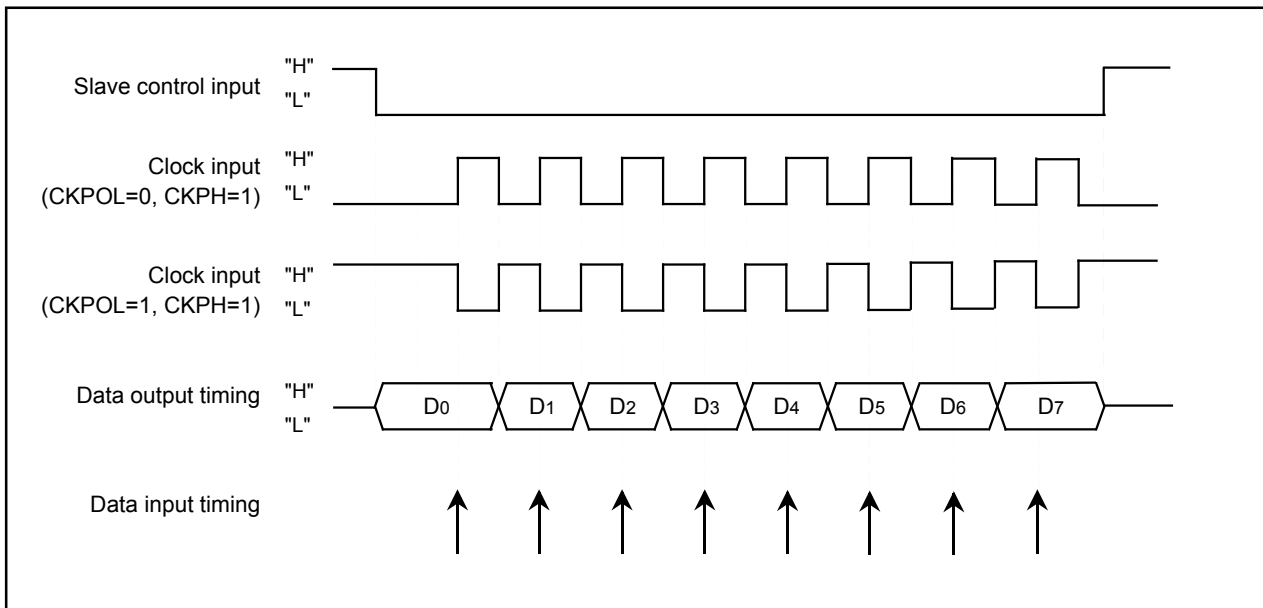


Figure 14.29 Transmission and Reception Timing (CKPH=1) in Slave Mode (External Clock)

14.1.5 Special Mode 3 (IEBus mode)(UART2)

In this mode, one bit in the IEBus is approximated with one byte of UART mode waveform.

Table 14.17 lists the registers used in IEBus mode and the register values set. **Figure 14.30** shows the functions of bus collision detect function related bits.

If the TxD2 pin output level and RxD2 pin input level do not match, a UART2 bus collision detect interrupt request is generated.

Table 14.17 Registers to Be Used and Settings in IEBus Mode

| Register | Bit | Function |
|---------------------|------------------------|---|
| U2TB | 0 to 8 | Set transmission data |
| U2RB ⁽¹⁾ | 0 to 8 | Reception data can be read |
| | OER,FER,PER,SUM | Error flag |
| U2BRG | 0 to 7 | Set bit rate |
| U2MR | SMD2 to SMD0 | Set to 1102 |
| | CKDIR | Select the internal clock or external clock |
| | STPS | Set to 0 |
| | PRY | Invalid because PRYE is set to 0 |
| | PRYE | Set to 0 |
| | IOPOL | Select the TxD/RxD input/output polarity |
| U2C0 | CLK1, CLK0 | Select the count source for the U2BRG register |
| | CRS | Invalid because CRDis set to 1 |
| | TXEPT | Transmit register empty flag |
| | CRD | Set to 1 |
| | NCH | Select TxD2 pin output mode |
| | CKPOL | Set to 0 |
| | UFORM | Set to 0 |
| U2C1 | TE | Set this bit to 1 to enable transmission |
| | TI | Transmit buffer empty flag |
| | RE | Set this bit to 1 to enable reception |
| | RI | Reception complete flag |
| | U2IRS | Select the source of UART2 transmit interrupt |
| | U2RRM, U2LCH, U2ERE | Set to 0 |
| U2SMR | 0 to 3, 7 | Set to 0 |
| | ABSCS | Select the sampling timing at which to detect a bus collision |
| | ACSE | Set this bit to 1 to use the auto clear function of transmit enable bit |
| | SSS | Select the transmit start condition |
| U2SMR2 | 0 to 7 | Set to 0 |
| U2SMR3 | 0 to 7 | Set to 0 |
| U2SMR4 | 0 to 7 | Set to 0 |

NOTE:

1. Not all register bits are described above. Set those bits to 0 when writing to the registers in IEBus mode.

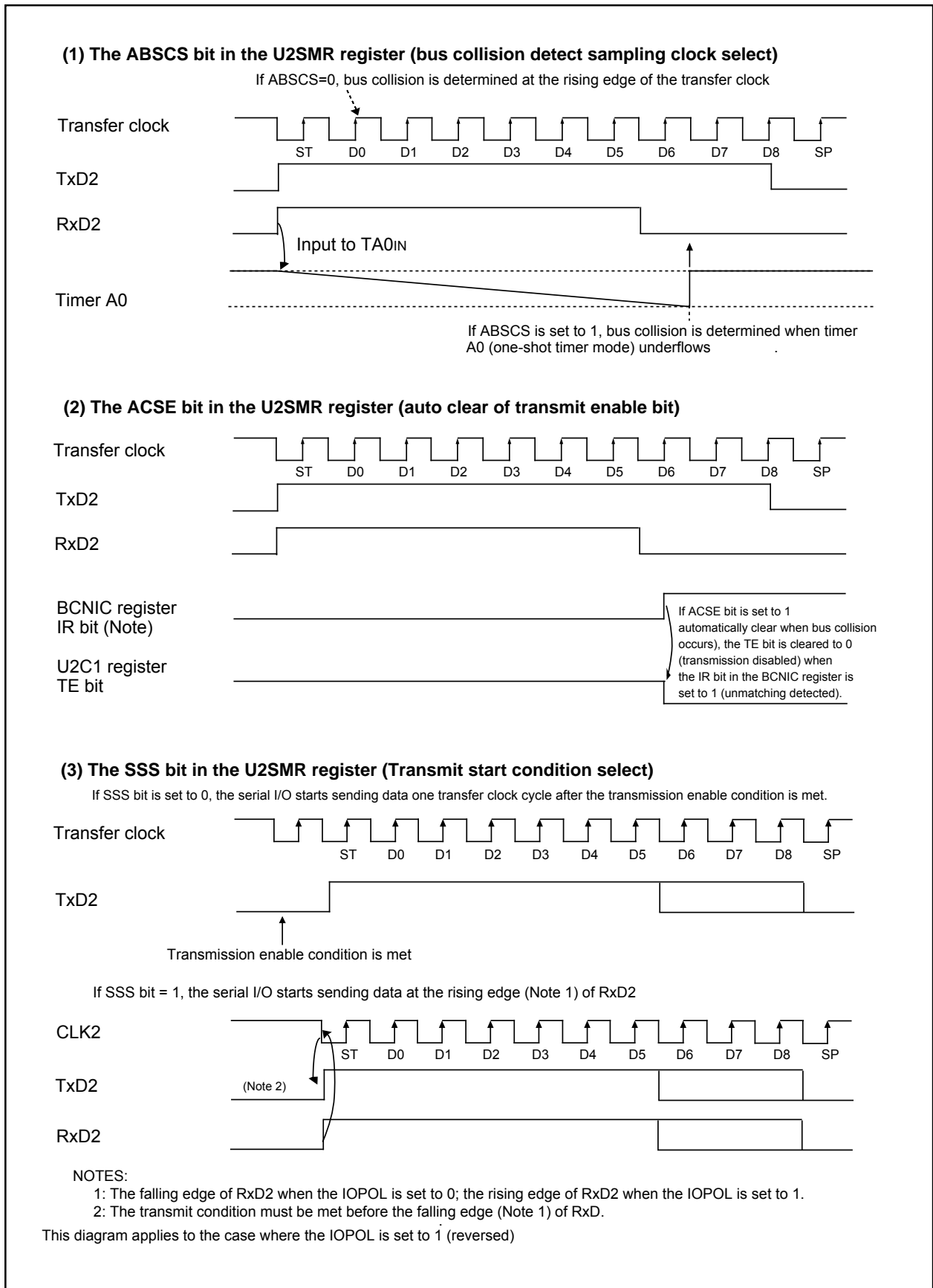


Figure 14.30 Bus Collision Detect Function-Related Bits

14.1.6 Special Mode 4 (SIM Mode) (UART2)

Based on UART mode, this is an SIM interface compatible mode. Direct and inverse formats can be implemented, and this mode allows output of a low from the TxD2 pin when a parity error is detected.

Table 14.18 lists the specifications of SIM mode. **Table 14.19** lists the registers used in the SIM mode and the register values set.

Table 14.18 SIM Mode Specifications

| Item | Specification |
|--|---|
| Transfer data format | <ul style="list-style-type: none"> • Direct format • Inverse format |
| Transfer clock | <ul style="list-style-type: none"> • The CKDIR bit in the U2MR register is set to 0 (internal clock) : $f_i / (16(n+1))$ $f_i = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}$. n: Setting value of U2BRG register 00₁₆ to FF₁₆ • The CKDIR bit is set to 1 (external clock) : $f_{EXT} / 16(n+1)$ f_{EXT}: Input from CLK2 pin. n: Setting value of U2BRG register 00₁₆ to FF₁₆ |
| Transmission start condition | <ul style="list-style-type: none"> • Before transmission can start, the following requirements must be met <ul style="list-style-type: none"> – The TE bit in the U2C1 register is set to 1 (transmission enabled) – The TI bit in the U2C1 register is set to 0 (data present in U2TB register) |
| Reception start condition | <ul style="list-style-type: none"> • Before reception can start, the following requirements must be met <ul style="list-style-type: none"> – The RE bit in the U2C1 register is set to 1 (reception enabled) – Start bit detection |
| Interrupt request generation timing ⁽²⁾ | <ul style="list-style-type: none"> • For transmission When the serial I/O finished sending data from the U2TB transfer register (U2IRS bit =1) • For reception When transferring data from the UART2 receive register to the U2RB register (at completion of reception) |
| Error detection | <ul style="list-style-type: none"> • Overrun error ⁽¹⁾ This error occurs if the serial I/O started receiving the next data before reading the U2RB register and received the bit one before the last stop bit in the the next data • Framing error This error occurs when the number of stop bits set is not detected • Parity error During reception, if a parity error is detected, parity error signal is output from the TxD2 pin. During transmission, a parity error is detected by the level of input to the RxD2 pin when a transmission interrupt occurs • Error sum flag This flag is set to 1 when any of the overrun, framing, and parity errors is encountered |

NOTES:

1. If an overrun error occurs, bits 8 to 0 in the U2RB register are undefined. The IR bit in the S2RIC register remains unchanged.
2. A transmit interrupt request is generated by setting the U2IRS bit in the U2C1 register to 1 (transmission complete) and U2ERE bit to 1 (error signal output) after reset. Therefore, when using SIM mode, be sure to clear the IR bit to 0 (no interrupt request) after setting these bits.

Table 14.19 Registers to Be Used and Settings in SIM Mode

| Register | Bit | Function |
|----------------------|-----------------|---|
| U2TB ⁽¹⁾ | 0 to 7 | Set transmission data |
| U2RB ⁽¹⁾ | 0 to 7 | Reception data can be read |
| | OER,FER,PER,SUM | Error flag |
| U2BRG | 0 to 7 | Set bit rate |
| U2MR | SMD2 to SMD0 | Set to 1012 |
| | CKDIR | Select the internal clock or external clock |
| | STPS | Set to 0 |
| | PRY | Set this bit to 1 for direct format or 0 for inverse format |
| | PRYE | Set to 1 |
| | IOPOL | Set to 0 |
| U2C0 | CLK1, CLK0 | Select the count source for the U2BRG register |
| | CRS | Invalid because CRDis set to 1 |
| | TXEPT | Transmit register empty flag |
| | CRD | Set to 1 |
| | NCH | Set to 0 |
| | CKPOL | Set to 0 |
| | UFORM | Set this bit to 0 for direct format or 1 for inverse format |
| U2C1 | TE | Set this bit to 1 to enable transmission |
| | TI | Transmit buffer empty flag |
| | RE | Set this bit to 1 to enable reception |
| | RI | Reception complete flag |
| | U2IRS | Set to 1 |
| | U2RRM | Set to 0 |
| | U2LCH | Set this bit to 0 for direct format or 1 for inverse format |
| | U2ERE | Set to 1 |
| U2SMR ⁽¹⁾ | 0 to 3 | Set to 0 |
| U2SMR2 | 0 to 7 | Set to 0 |
| U2SMR3 | 0 to 7 | Set to 0 |
| U2SMR4 | 0 to 7 | Set to 0 |

NOTE:

1. Not all register bits are described above. Set those bits to 0 when writing to the registers in SIM mode.

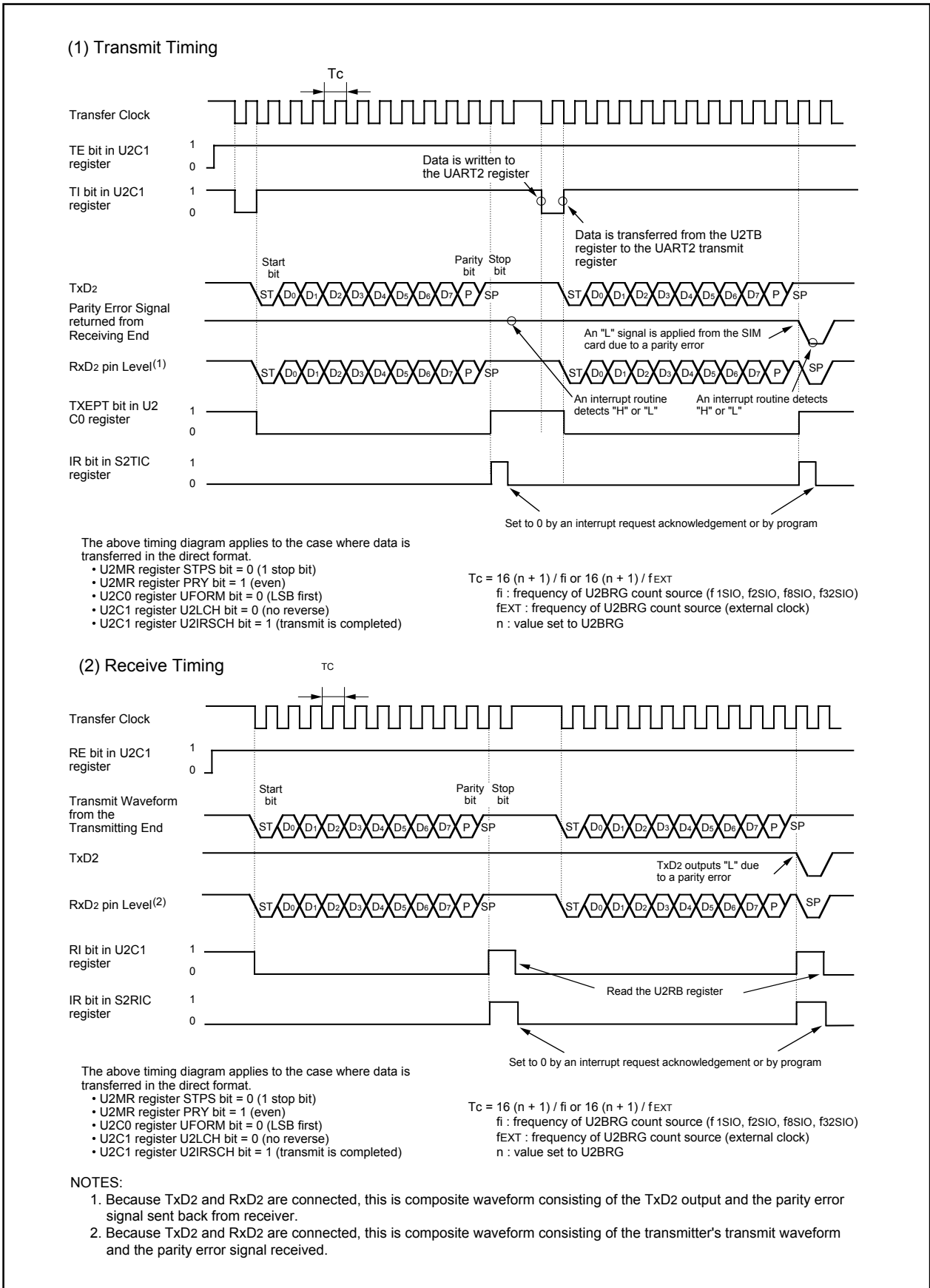


Figure 14.31 Transmit and Receive Timing in SIM Mode

Figure 14.32 shows the example of connecting the SIM interface. Connect TXD2 and RXD2 and apply pull-up.

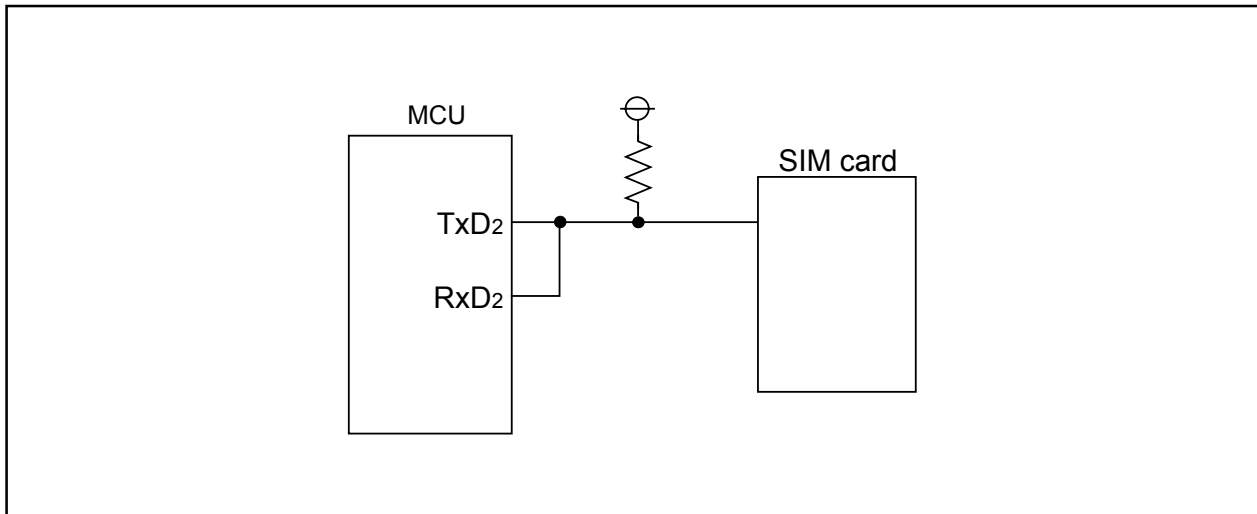


Figure 14.32 SIM Interface Connection

14.1.6.1 Parity Error Signal Output

The parity error signal is enabled by setting the U2ERE bit in the U2C1 register to 1.

- When receiving

The parity error signal is output when a parity error is detected while receiving data. This is achieved by pulling the TXD2 output low with the timing shown in **Figure 14.33**. If the R2RB register is read while outputting a parity error signal, the PER bit is cleared to 0 and at the same time the TXD2 output is returned high.

- When transmitting

A transmission-finished interrupt request is generated at the falling edge of the transfer clock pulse that immediately follows the stop bit. Therefore, whether a parity signal has been returned can be determined by reading the port that shares the RXD2 pin in a transmission-finished interrupt service routine.

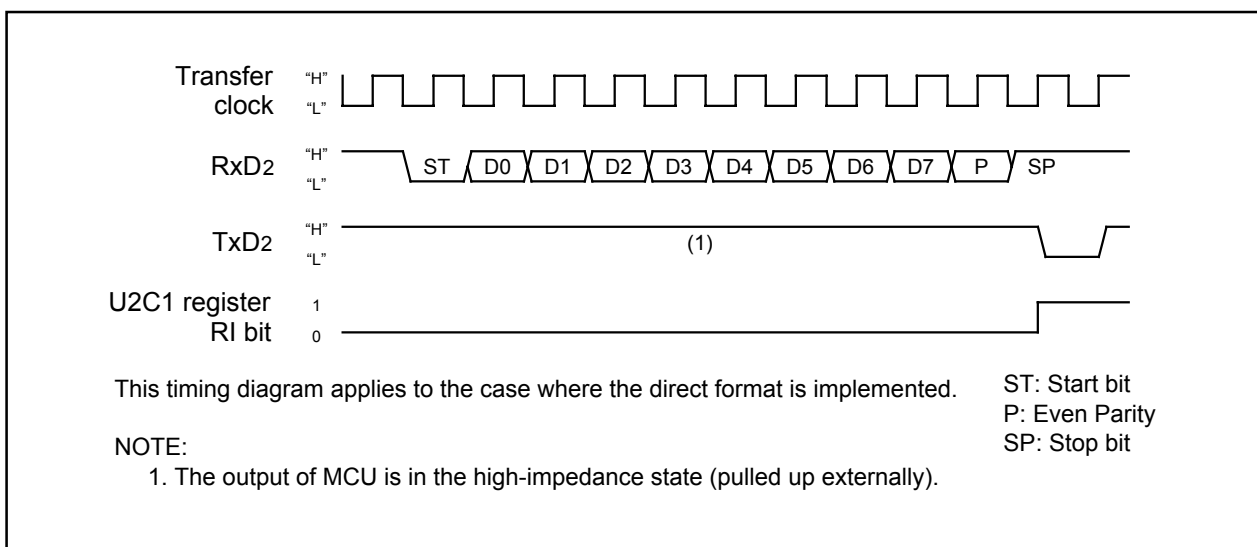


Figure 14.33 Parity Error Signal Output Timing

14.1.6.2 Format

- Direct Format

Set the PRY bit in the U2MR register to 1, the UFORM bit in U2C0 register to 0 and the U2LCH bit in U2C1 register to 0.

- Inverse Format

Set the PRY bit to 0, UFORM bit to 1 and U2LCH bit to 1.

Figure 14.34 shows the SIM interface format.

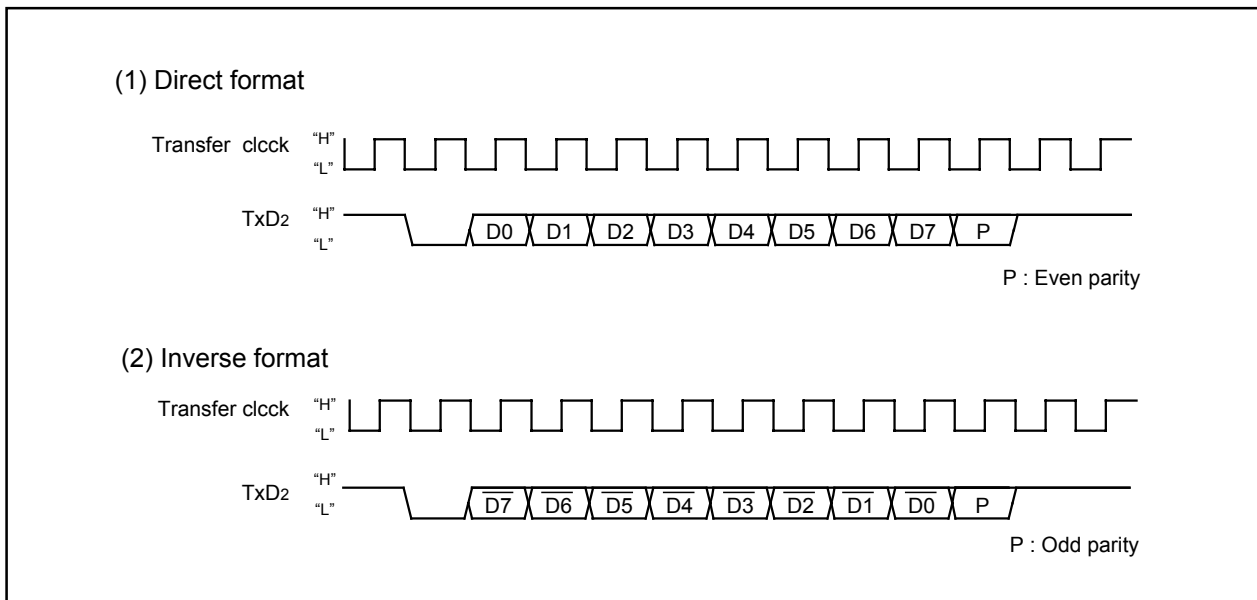


Figure 14.34 SIM Interface Format

14.2 SI/O3 and SI/O4

Note

The SI/O4 interrupt of peripheral function interrupt is not available in the 64-pin package.

SI/O3 and SI/O4 are exclusive clock-synchronous serial I/Os.

Figure 14.35 shows the block diagram of SI/O3 and SI/O4, and **Figure 14.36** shows the SI/O3 and SI/O4-related registers.

Table 14.20 shows the specifications of SI/O3 and SI/O4.

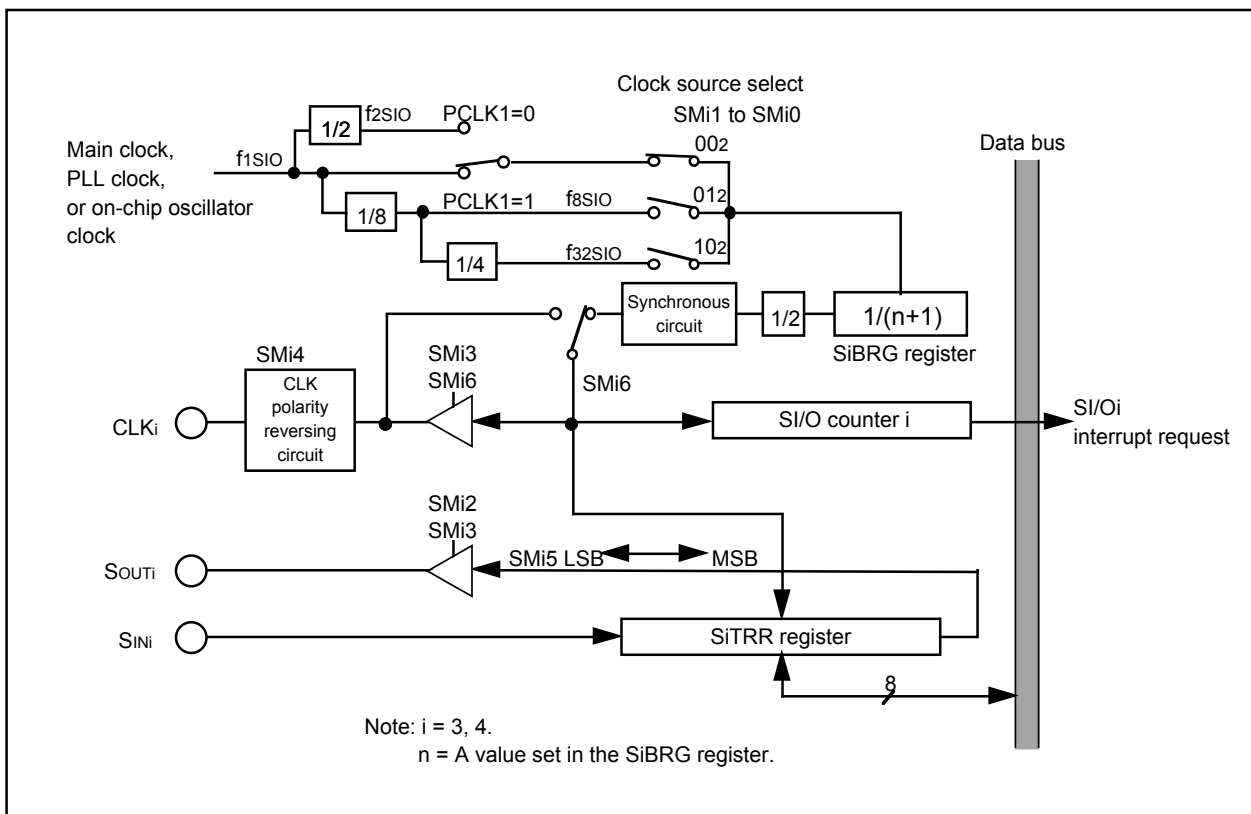


Figure 14.35 SI/O3 and SI/O4 Block Diagram

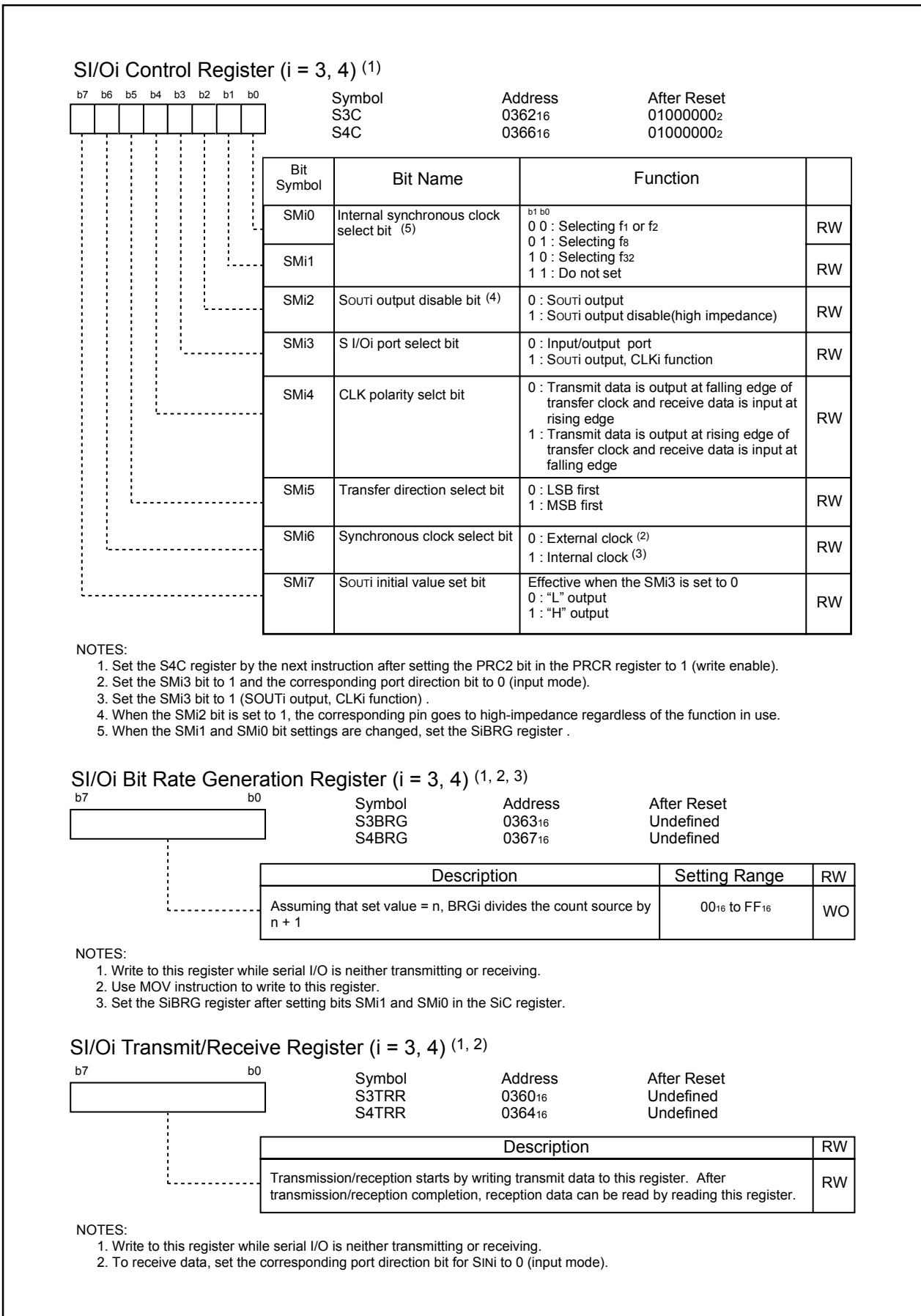


Figure 14.36 S3C and S4C Registers, S3BRG and S4BRG Registers, and S3TRR and S4TRR Registers

Table 14.20 SI/O3 and SI/O4 Specifications

| Item | Specification |
|--|--|
| Transfer data format | • Transfer data length: 8 bits |
| Transfer clock | • The SMi6 bit in the SiC (i=3, 4) register is set to 1 (internal clock) : $f_j / (2(n+1))$ $f_j = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}$. n=Setting value of SiBRG register 0016 to FF16. • SMi6 bit is set to 0 (external clock) : Input from CLKi pin ⁽¹⁾ |
| Transmission/reception start condition | • Before transmission/reception can start, the following requirements must be met Write transmit data to the SiTRR register ^(2, 3) |
| Interrupt request generation timing | • When the SMi4 bit in the SiC register is set to 0 The rising edge of the last transfer clock pulse ⁽⁴⁾ • When SMi4 is set to 1 The falling edge of the last transfer clock pulse ⁽⁴⁾ |
| CLKi pin function | I/O port, transfer clock input, transfer clock output |
| SOUTi pin function | I/O port, transmit data output, high-impedance |
| SINi pin function | I/O port, receive data input |
| Select function | • LSB first or MSB first selection Whether to start sending/receiving data beginning with bit 0 or beginning with bit 7 can be selected • Function for setting an SOUTi initial value set function When the SMi6 bit in the SiC register is set to 0 (external clock), the SOUTi pin output level while not transmitting can be selected. • CLK polarity selection Whether transmit data is output/input timing at the rising edge or falling edge of transfer clock can be selected. |

NOTE:

- To set the SMi6 bit in the SiC register to 0 (external clock), follow the procedure described below.
 - If the SMi4 bit in the SiC register is set to 0, write transmit data to the SiTRR register while input on the CLKi pin is high. The same applies when rewriting the SMi7 bit in the SiC register.
 - If the SMi4 bit is set to 1, write transmit data to the SiTRR register while input on the CLKi pin is low. The same applies when rewriting the SMi7 bit.
 - Because shift operation continues as long as the transfer clock is supplied to the SI/Oi circuit, stop the transfer clock. Unlike UART0 to UART2, SI/Oi (i = 3 to 4) is not separated between the transfer register and buffer. Therefore, do not write the next transmit data to the SiTRR register during transmission.
- When the SMi6 bit in the SiC register is set to 1 (internal clock), SOUTi retains the last data for a 1/2 transfer clock period after completion of transfer and, thereafter, goes to a high-impedance state. However, if transmit data is written to the SiTRR register during this period, SOUTi immediately goes to a high-impedance state, with the data hold time thereby reduced.
- When the SMi6 bit in the SiC register is set to 1 (internal clock), the transfer clock stops in the high state if the SMi4 bit is set to 0, or stops in the low state if the SMi4 bit is set to 1.

14.2.1 SI/Oi Operation Timing

Figure 14.37 shows the SI/Oi operation timing

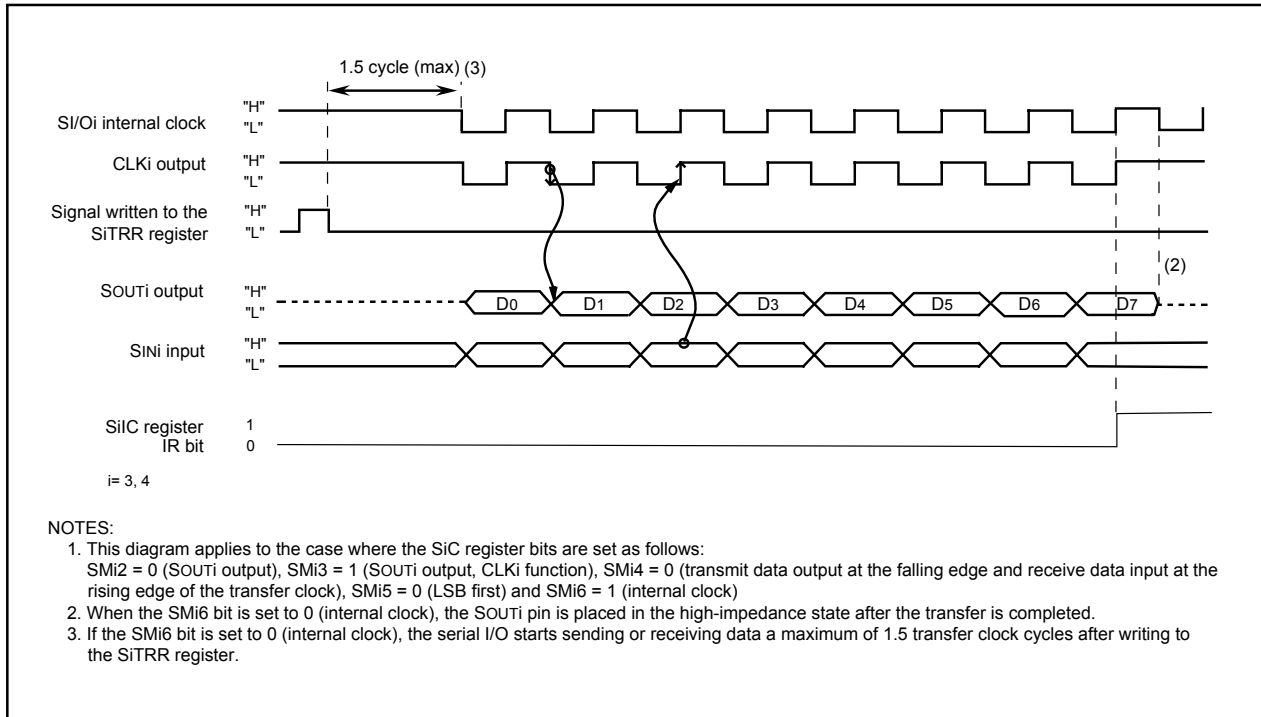


Figure 14.37 SI/Oi Operation Timing

14.2.2 CLK Polarity Selection

The the SMI4 bit in the SiC register allows selection of the polarity of the transfer clock. Figure 14.38 shows the polarity of the transfer clock.

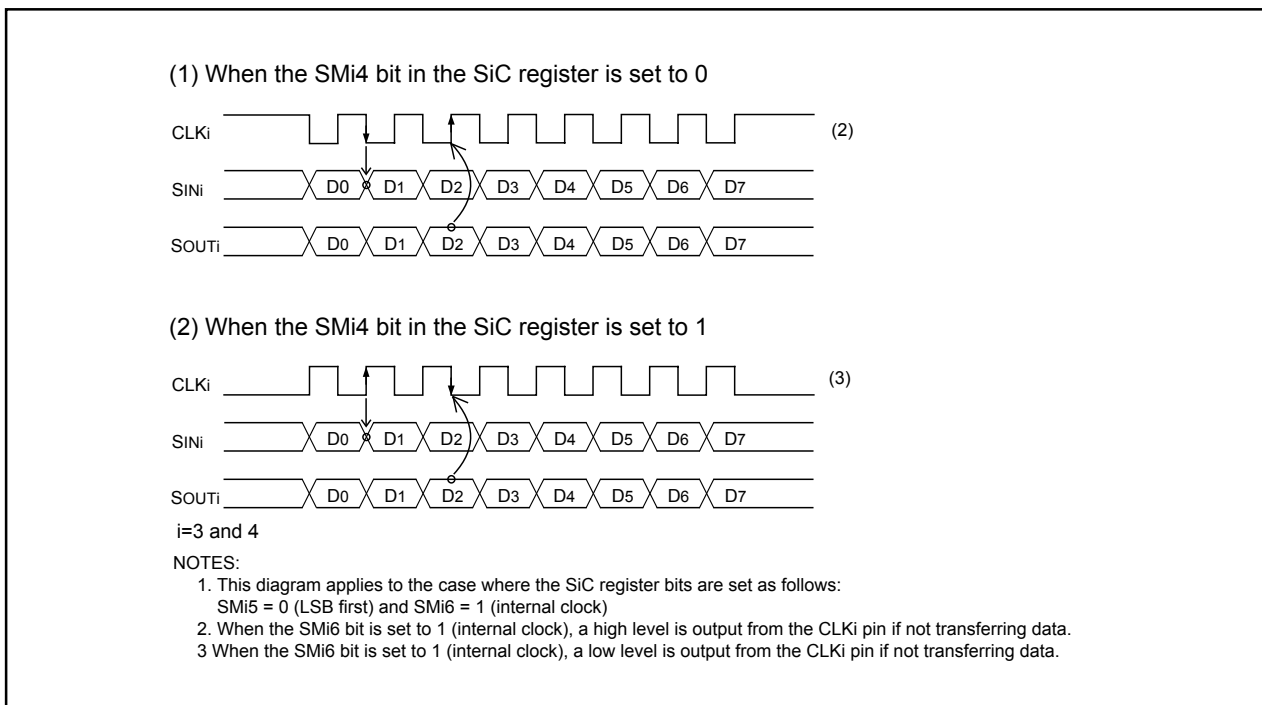


Figure 14.38 Polarity of Transfer Clock

14.2.3 Functions for Setting an SOUTi Initial Value

If the SMi6 bit in SiC register is set to 0 (external clock), the SOUTi pin output level can be fixed high or low when not transferring data. However, when transmitting data consecutively, the last bit (bit 0) value of the last transmitted data is retained between the successive data transmissions. **Figure 14.39** shows the timing chart for setting an SOUTi initial value and how to set it.

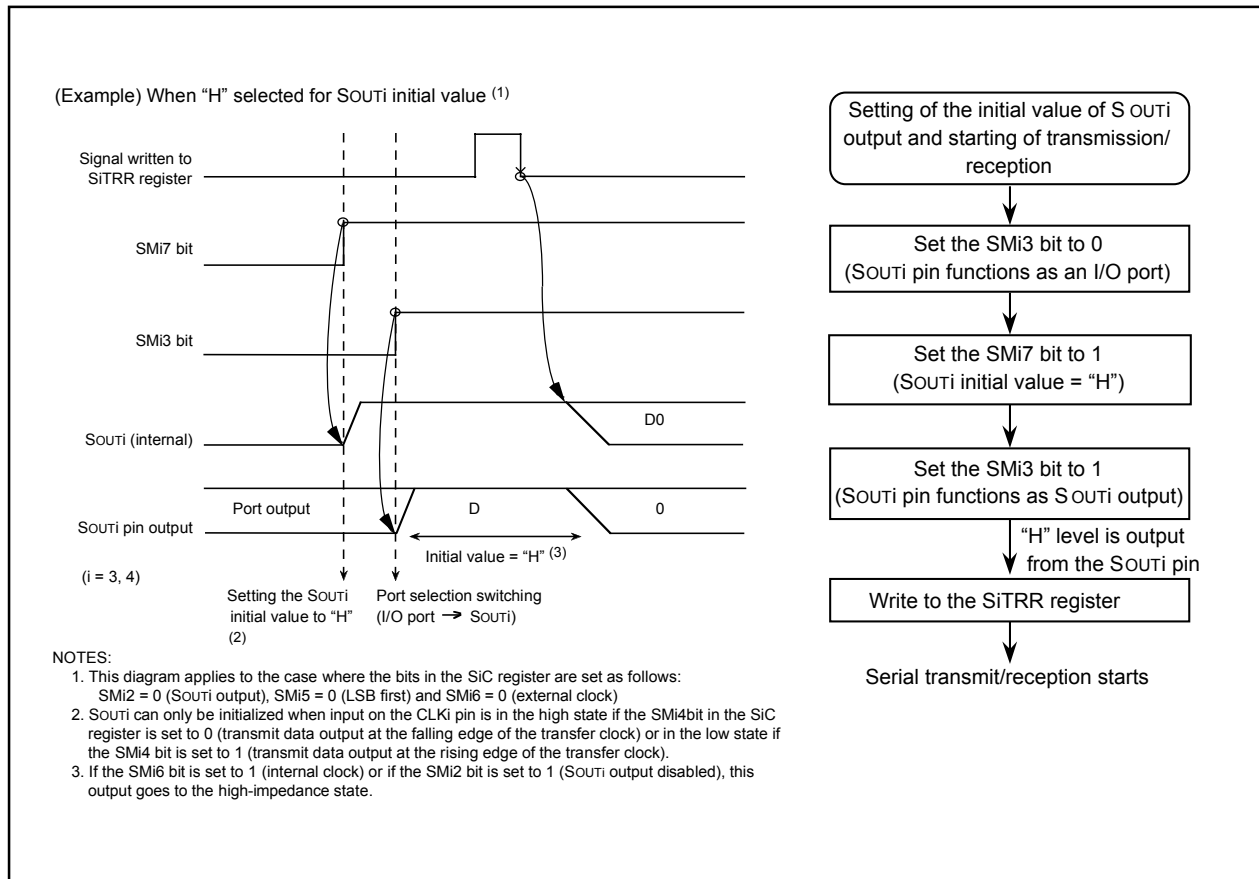


Figure 14.39 SOUTi Initial Value Setting

15. A/D Converter

Note

Ports P04 to P07(AN04 to AN07), P10 to P13(AN20 to AN23) and P95 to P97(AN25 to AN27) are not available in 64-pin package. Do not use port P04 to P07(AN04 to AN07), P10 to P13(AN20 to AN23) and P95 to P97(AN25 to AN27) as analog input pins in 64-pin package.

The MCU contains one A/D converter circuit based on 10-bit successive approximation method configured with a capacitive-coupling amplifier. The analog inputs share the pins with P100 to P107 (AN0 to AN7), P00 to P07 (AN00 to AN07), and P10 to P13, P93, P95 to P97 (AN20 to AN27), and P90 to P92 (AN30 to AN32). Similarly, $\overline{\text{ADTRG}}$ input shares the pin with P15. Therefore, when using these inputs, make sure the corresponding port direction bits are set to 0 (input mode).

When not using the A/D converter, set the VCUT bit 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 A/D conversion result is stored in the ADi register bits for ANi, AN0i, AN2i (i = 0 to 7), and AN3i pins (i = 0 to 2). **Table 15.1** shows the A/D converter performance. **Figure 15.1** shows the A/D converter block diagram and **Figures 15.2 to 15.4** show the A/D converter associated with registers.

Table 15.1 A/D Converter Performance

| Item | Performance |
|---|---|
| A/D Conversion Method | Successive approximation (capacitive coupling amplifier) |
| Analog Input Voltage ⁽¹⁾ | 0V to AVCC (VCC) |
| Operating Clock ϕ_{AD} ⁽²⁾ | $f_{\text{AD}}/\text{divided-by-2}$ or $f_{\text{AD}}/\text{divided-by-3}$ or $f_{\text{AD}}/\text{divided-by-4}$ or $f_{\text{AD}}/\text{divided-by-6}$ or $f_{\text{AD}}/\text{divided-by-12}$ or f_{AD} |
| Resolution | 8-bit or 10-bit (selectable) |
| Integral Nonlinearity Error | When AVCC = Vref = 5V <ul style="list-style-type: none"> • With 8-bit resolution: $\pm 2\text{LSB}$ • With 10-bit resolution: $\pm 3\text{LSB}$ When AVCC = Vref = 3.3V <ul style="list-style-type: none"> • With 8-bit resolution: $\pm 2\text{LSB}$ • With 10-bit resolution: $\pm 5\text{LSB}$ |
| Operating Modes | One-shot mode, repeat mode, single sweep mode, repeat sweep mode 0, repeat sweep mode 1, simultaneous sample sweep mode and delayed trigger mode 0,1 |
| Analog Input Pins | 8 pins (AN0 to AN7) + 8 pins (AN00 to AN07) + 8 pins (AN20 to AN27) + 3 pins (AN30 to AN32) (80-pin package) 8 pins (AN0 to AN7) + 4 pins (AN00 to AN03) + 1 pin (AN24) + 3 pins (AN30 to AN32) (64-pin package) |
| Conversion Speed Per Pin | <ul style="list-style-type: none"> • Without sample and hold function 8-bit resolution: 49 ϕ_{AD} cycles, 10-bit resolution: 59 ϕ_{AD} cycles • With sample and hold function 8-bit resolution: 28 ϕ_{AD} cycles, 10-bit resolution: 33 ϕ_{AD} cycles |

NOTES:

1. Not dependent on use of sample and hold function.
2. Set the ϕ_{AD} frequency to 10 MHz or less.
 Without sample-and-hold function, set the ϕ_{AD} frequency to 250kHz or more.
 With the sample and hold function, set the ϕ_{AD} frequency to 1MHz or more.

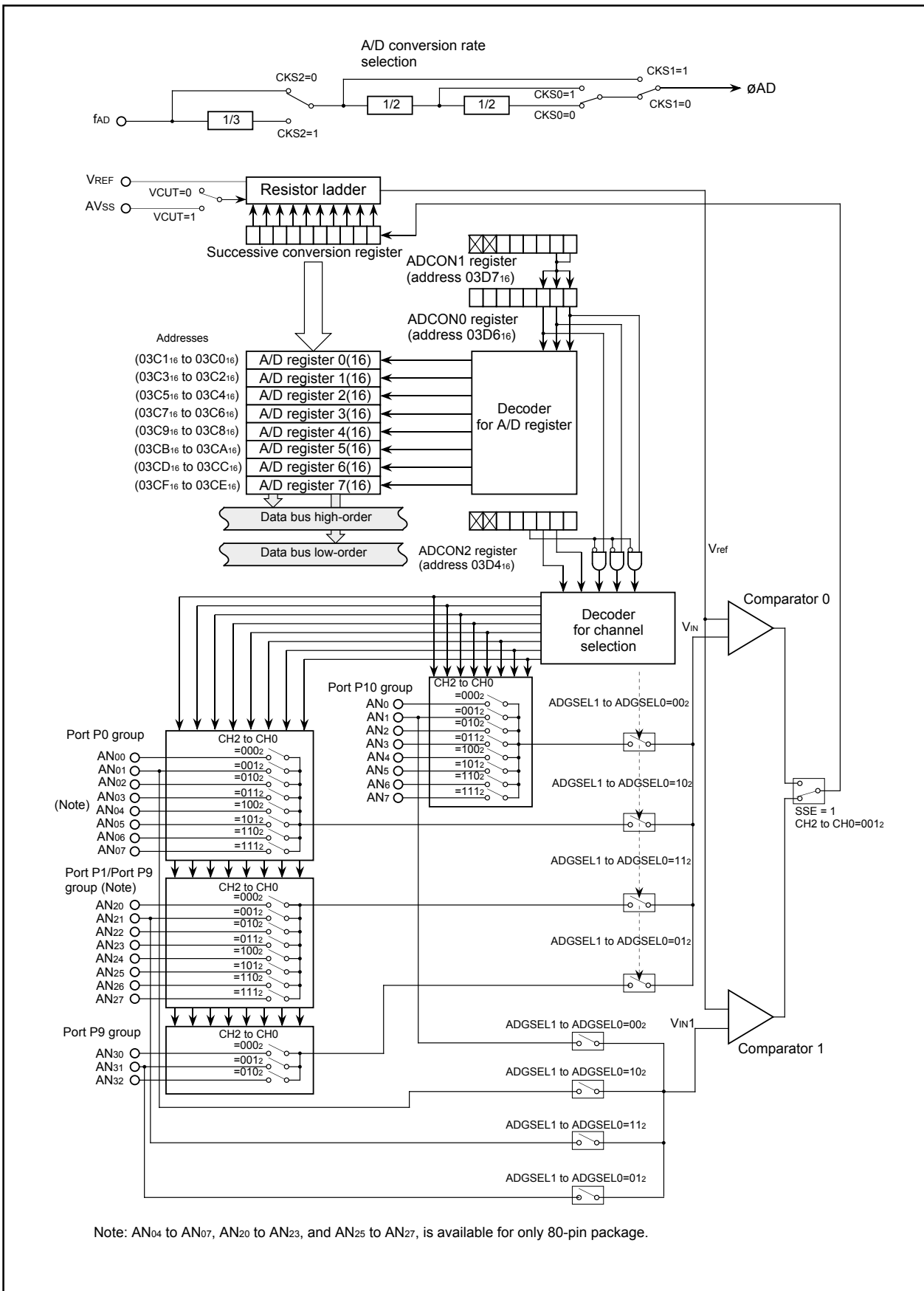
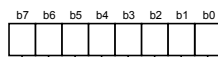


Figure 15.1 A/D Converter Block Diagram

A/D Control Register 0 ⁽¹⁾



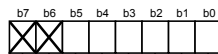
Symbol: ADCON0
Address: 03D6₁₆
After Reset: 00000XXX₂

| Bit Symbol | Bit Name | Function | RW |
|------------|---------------------------------|---|----|
| CH0 | Analog input pin select bit | Function varies with each operation mode | RW |
| CH1 | | | RW |
| CH2 | | | RW |
| MD0 | A/D operation mode select bit 0 | ^{b4 b3} 0 0: One-shot mode or Delayed trigger mode 0,1 0 1: Repeat mode 1 0: Single sweep mode or Simultaneous sample sweep mode 1 1: Repeat sweep mode 0 or Repeat sweep mode 1 | RW |
| MD1 | | | RW |
| TRG | Trigger select bit | 0: Software trigger 1: Hardware trigger | RW |
| ADST | A/D conversion start flag | 0: A/D conversion disabled 1: A/D conversion started | RW |
| CKS0 | Frequency select bit 0 | See Table 15.2 | RW |

NOTE:

1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be undefined.

A/D Control Register 1 ⁽¹⁾



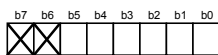
Symbol: ADCON1
Address: 03D7₁₆
After Reset: 00₁₆

| Bit Symbol | Bit Name | Function | RW |
|------------|--|---|----|
| SCAN0 | A/D sweep pin select bit | Function varies with each operation mode | RW |
| SCAN1 | | | RW |
| MD2 | A/D operation mode select bit 1 | 0 : Other than repeat sweep mode 1 1 : Repeat sweep mode 1 | RW |
| BITS | 8/10-bit mode select bit | 0 : 8-bit mode 1 : 10-bit mode | RW |
| CKS1 | Frequency select bit 1 | See Table 15.2 | RW |
| VCUT | Vref connect Bit ⁽²⁾ | 0 : Vref not connected 1 : Vref connected | RW |
| (b7-b6) | Nothing is assigned. If necessary, set to 0. When read, its content is 0 | | — |

NOTES:

1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be undefined.
2. If the VCUT bit is reset from 0 (VREF unconnected) to 1 (VREF connected), wait for 1 μs or more before starting A/D conversion.

A/D Control Register 2 ⁽¹⁾



Symbol: ADCON2
Address: 03D4₁₆
After Reset: 00₁₆

| Bit Symbol | Bit Name | Function | RW |
|------------|--|--|----|
| SMP | A/D conversion method select bit | 0: Without sample and hold 1: With sample and hold | RW |
| ADGSEL0 | A/D input group select bit | ^{b2 b1} 0 0: Select port P10 group 0 1: Select port P9 group 1 0: Select port P0 group 1 1: Select port P1/P9 group | RW |
| ADGSEL1 | | | RW |
| (b3) | Reserved bit | Set to 0 | RW |
| CKS2 | Frequency select bit 2 | See Table 15.2 | RW |
| TRG1 | Trigger select bit | Function varies with each operation mode | RW |
| (b7-b6) | Nothing is assigned. If necessary, set to 0. When read, the content is 0 | | — |

NOTS:

1. If the ADCON2 register is rewritten during A/D conversion, the conversion result will be undefined.

Figure 15.2 ADCON0 to ADCON2 Registers

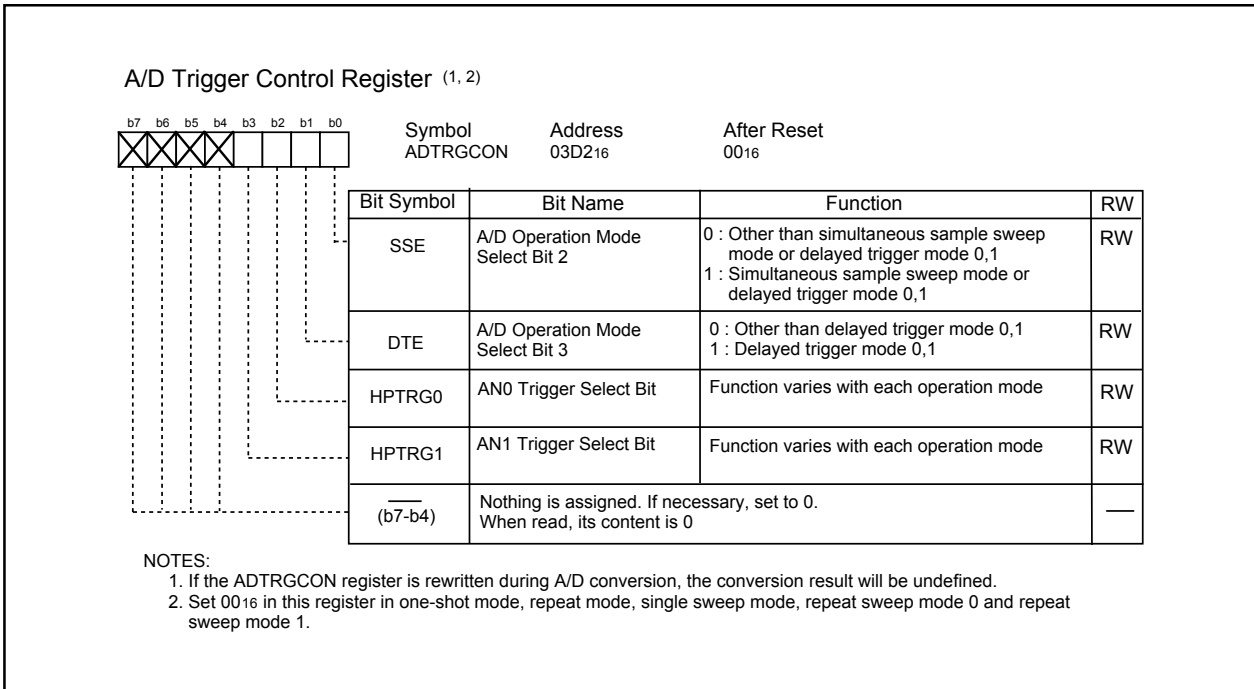


Figure 15.3 ADTRGCON Register

Table 15.2 A/D Conversion Frequency Select

| CKS2 | CKS1 | CKS0 | \emptyset AD |
|------|------|------|-------------------|
| 0 | 0 | 0 | fAD divided by 4 |
| 0 | 0 | 1 | fAD divided by 2 |
| 0 | 1 | 0 | fAD |
| 0 | 1 | 1 | |
| 1 | 0 | 0 | fAD divided by 12 |
| 1 | 0 | 1 | fAD divided by 6 |
| 1 | 1 | 0 | fAD divided by 3 |
| 1 | 1 | 1 | |

NOTE:

1. \emptyset AD frequency must be under 10 MHz. Combination of the CKS0 bit in the ADCON0 register, the CKS1 bit in the ADCON1 register, and the CKS2 bit in the ADCON2 register selects \emptyset AD.

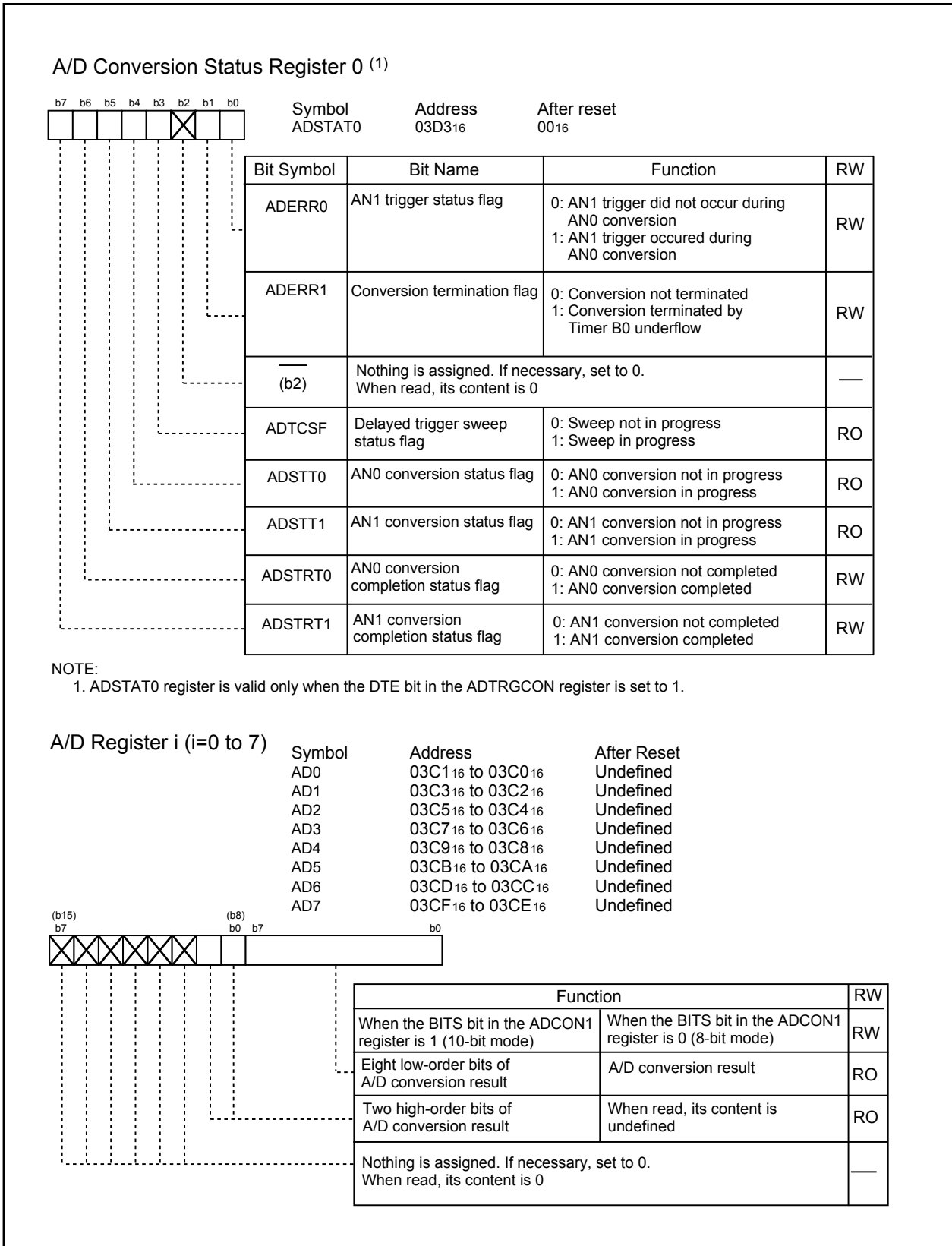


Figure 15.4 ADSTAT0 Register and AD0 to AD7 Registers

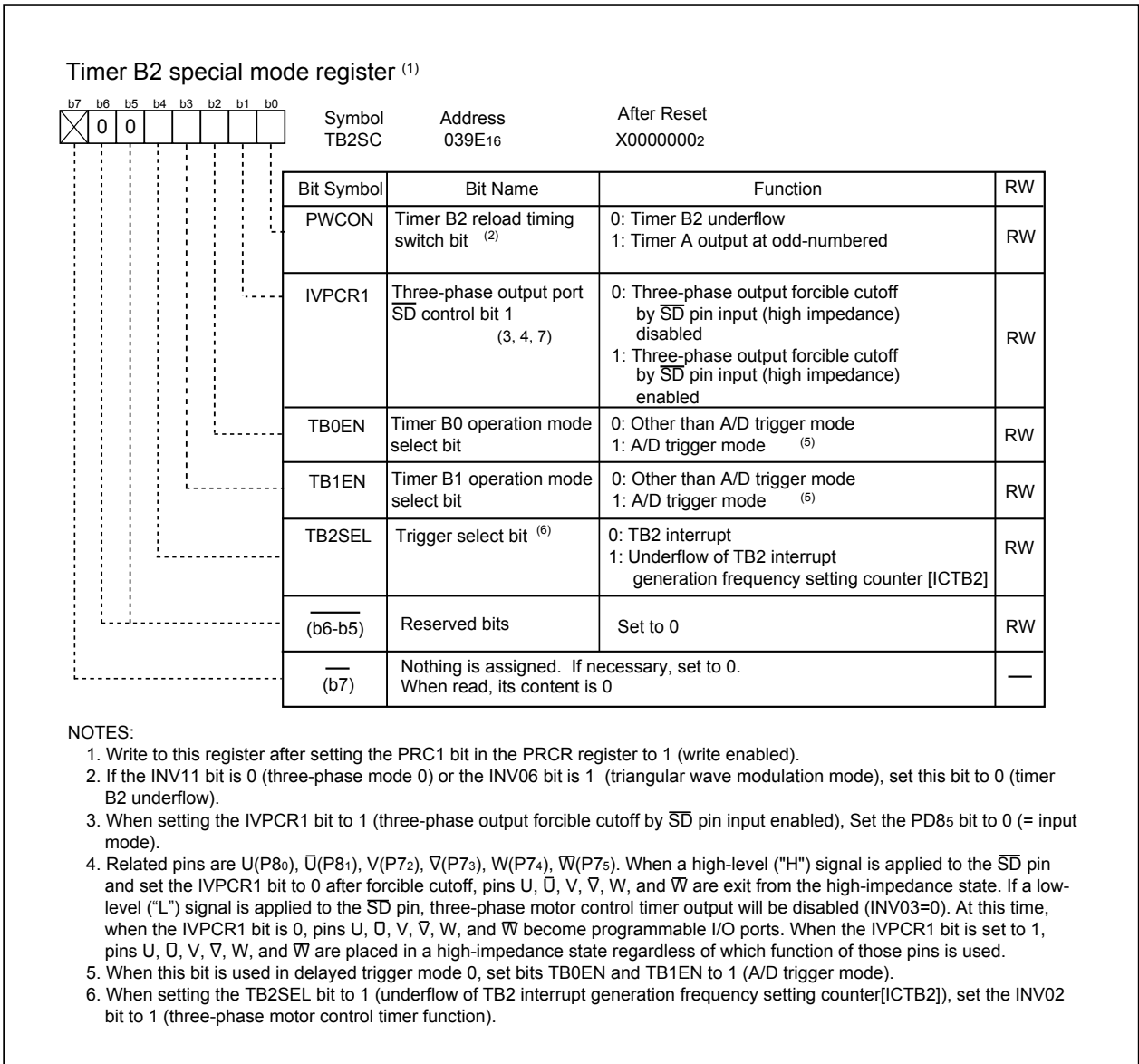


Figure 15.5 TB2SC Register

15.1 Operating Modes

15.1.1 One-Shot Mode

In one-shot mode, analog voltage applied to a selected pin is once converted to a digital code. **Table 15.3** shows the one-shot mode specifications. **Figure 15.6** shows the operation example in one-shot mode. **Figure 15.7** shows registers ADCON0 to ADCON2 in one-shot mode.

Table 15.3 One-shot Mode Specifications

| Item | Specification |
|-------------------------------------|---|
| Function | Bits CH2 to CH0 in the ADCON0 register and registers ADGSEL1 and ADGSEL0 in the ADCON2 register select pins. Analog voltage applied to a selected pin is once converted to a digital code |
| A/D Conversion Start Condition | <ul style="list-style-type: none"> When the TRG bit in the ADCON0 register is 0 (software trigger) Set the ADST bit in the ADCON0 register to 1 (A/D conversion started) When the TRG bit in the ADCON0 register is 1 (hardware trigger) The \overline{ADTRG} pin input changes state from "H" to "L" after setting the ADST bit to 1 (A/D conversion started) |
| A/D Conversion Stop Condition | <ul style="list-style-type: none"> A/D conversion completed (If a software trigger is selected, the ADST bit is set to 0 (A/D conversion halted)). Set the ADST bit to 0 |
| Interrupt Request Generation Timing | A/D conversion completed |
| Analog Input Pin | Select one pin from AN0 to AN7, AN00 to AN07, AN20 to AN27, AN30 to AN32 |
| Readout of A/D Conversion Result | Readout one of registers AD0 to AD7 that corresponds to the selected pin |

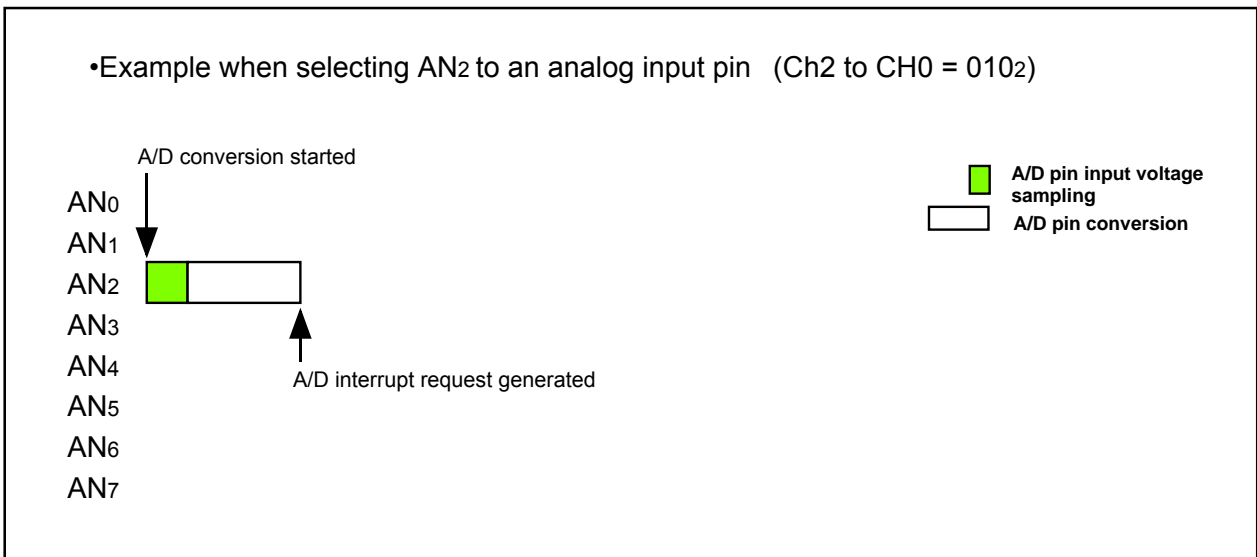
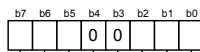


Figure 15.6 Operation Example in One-Shot Mode

A/D Control Register 0 ⁽¹⁾



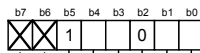
Symbol: ADCON0
Address: 03D6₁₆
After Reset: 00000XXX₂

| Bit Symbol | Bit Name | Function | RW |
|------------|--|--|----|
| CH0 | Analog input pin select bit (2, 3) | b2:b1:00 0 0 0: Select AN ₀ 0 0 1: Select AN ₁ 0 1 0: Select AN ₂ 0 1 1: Select AN ₃ 1 0 0: Select AN ₄ 1 0 1: Select AN ₅ 1 1 0: Select AN ₆ 1 1 1: Select AN ₇ | RW |
| CH1 | | | RW |
| CH2 | | | RW |
| MD0 | A/D operation mode select bit 0 ⁽³⁾ | b4:b3 0 0: One-shot mode or delayed trigger mode 0, 1 | RW |
| MD1 | | | RW |
| TRG | Trigger select bit | 0: Software trigger 1: Hardware trigger (AD _{TRG} trigger) | RW |
| ADST | A/D conversion start flag | 0: A/D conversion disabled 1: A/D conversion started | RW |
| CKS0 | Frequency select bit 0 | See Table 15.2 | RW |

NOTES:

1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be undefined.
2. AN₀ to AN₇, AN₂₀ to AN₂₇, and AN₃₀ to AN₃₂ can be used in the same way as AN₀ to AN₇. Use bits ADGSEL1 and ADGSEL0 in the ADCON2 register to select the desired pin.
3. After rewriting bits MD1 and MD0, set bits CH2 to CH0 over again using another instruction.

A/D Control Register 1 ⁽¹⁾



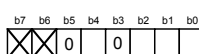
Symbol: ADCON1
Address: 03D7₁₆
After Reset: 00₁₆

| Bit Symbol | Bit Name | Function | RW |
|------------|---|---|----|
| SCAN0 | A/D Sweep Pin Select Bit | Invalid in one-shot mode | RW |
| SCAN1 | | | RW |
| MD2 | A/D Operation Mode Select Bit 1 | 0 : Any mode other than repeat sweep mode 1 | RW |
| BITS | 8/10-Bit Mode Select Bit | 0 : 8-bit mode 1 : 10-bit mode | RW |
| CKS1 | Frequency Select Bit 1 | Refer to Table 15.2 | RW |
| VCUT | Vref Connect Bit ⁽²⁾ | 1 : Vref connected | RW |
| (b7-b6) | Nothing is assigned. If necessary, set to 0. When read, the contents are 0. | | — |

NOTES:

1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be undefined.
2. If the VCUT bit is reset from 0 (Vref unconnected) to 1 (Vref connected), wait for 1 μs or more before starting A/D conversion.

A/D Control Register 2 ⁽¹⁾



Symbol: ADCON2
Address: 03D4₁₆
After Reset: 00₁₆

| Bit Symbol | Bit Name | Function | RW |
|------------|---|---|----|
| SMP | A/D conversion method select bit | 0: Without sample and hold 1: With sample and hold | RW |
| ADGSEL0 | A/D input group select bit | b2:b1 0 0: Select port P10 group 0 1: Select port P9 group 1 0: Select port P0 group 1 1: Select port P1/P9 group | RW |
| ADGSEL1 | | | RW |
| (b3) | Reserved bit | Set to 0 | RW |
| CKS2 | Frequency select bit 2 | See Table 15.2 | RW |
| TRG1 | Trigger select bit 1 | Set to 0 in one-shot mode | RW |
| (b7-b6) | Nothing is assigned. If necessary, set to 0. When read, the content is 0. | | — |

NOTE:

1. If the ADCON2 register is rewritten during A/D conversion, the conversion result will be undefined.

Figure 15.7 ADCON0 to ADCON2 Registers in One-Shot Mode

15.1.2 Repeat mode

In repeat mode, analog voltage applied to a selected pin is repeatedly converted to a digital code. **Table 15.4** shows the repeat mode specifications. **Figure 15.8** shows the operation example in repeat mode. **Figure 15.9** shows the ADCON0 to ADCON2 registers in repeat mode.

Table 15.4 Repeat Mode Specifications

| Item | Specification |
|-------------------------------------|--|
| Function | Bits CH2 to CH0 in the ADCON0 register and the ADGSEL1 to ADGSEL0 bits in the ADCON2 register select pins. Analog voltage applied to a selected pin is repeatedly converted to a digital code |
| A/D Conversion Start Condition | <ul style="list-style-type: none"> When the TRG bit in the ADCON0 register is 0 (software trigger) Set the ADST bit in the ADCON0 register to 1 (A/D conversion started) When the TRG bit in the ADCON0 register is 1 (hardware trigger) The $\overline{\text{ADTRG}}$ pin input changes state from "H" to "L" after setting the ADST bit to 1 (A/D conversion started) |
| A/D Conversion Stop Condition | Set the ADST bit to 0 (A/D conversion halted) |
| Interrupt Request Generation Timing | None generated |
| Analog Input Pin | Select one pin from AN0 to AN7, AN00 to AN07, AN20 to AN27, and AN30 to AN32 |
| Readout of A/D Conversion Result | Readout one of the AD0 to AD7 registers that corresponds to the selected pin |

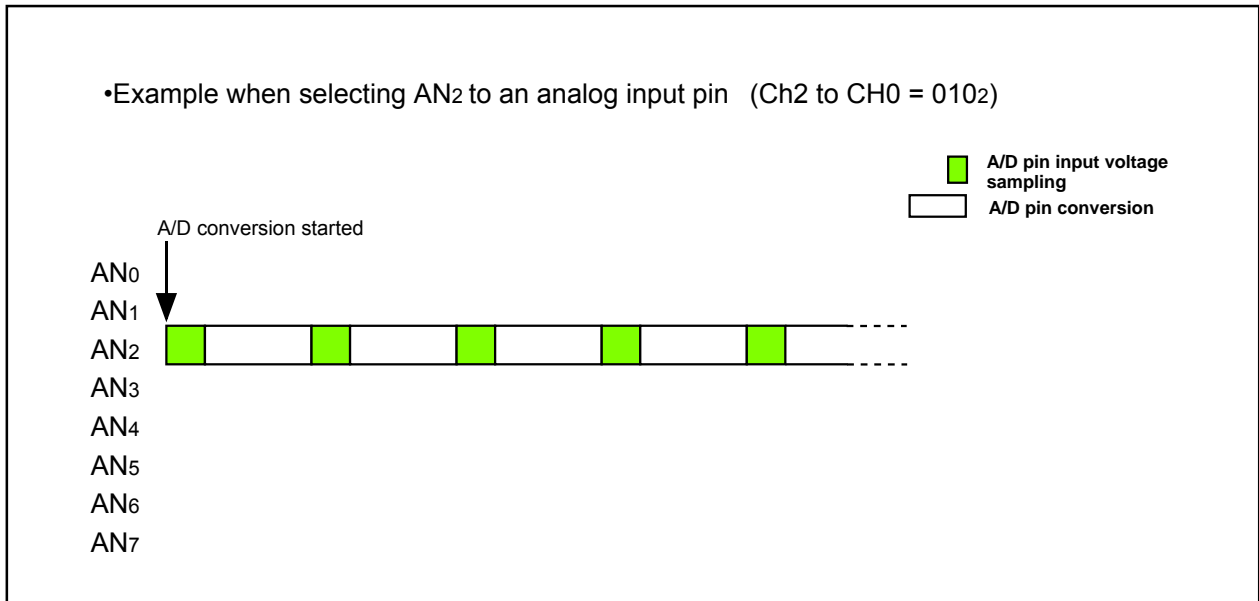


Figure 15.8 Operation Example in Repeat Mode

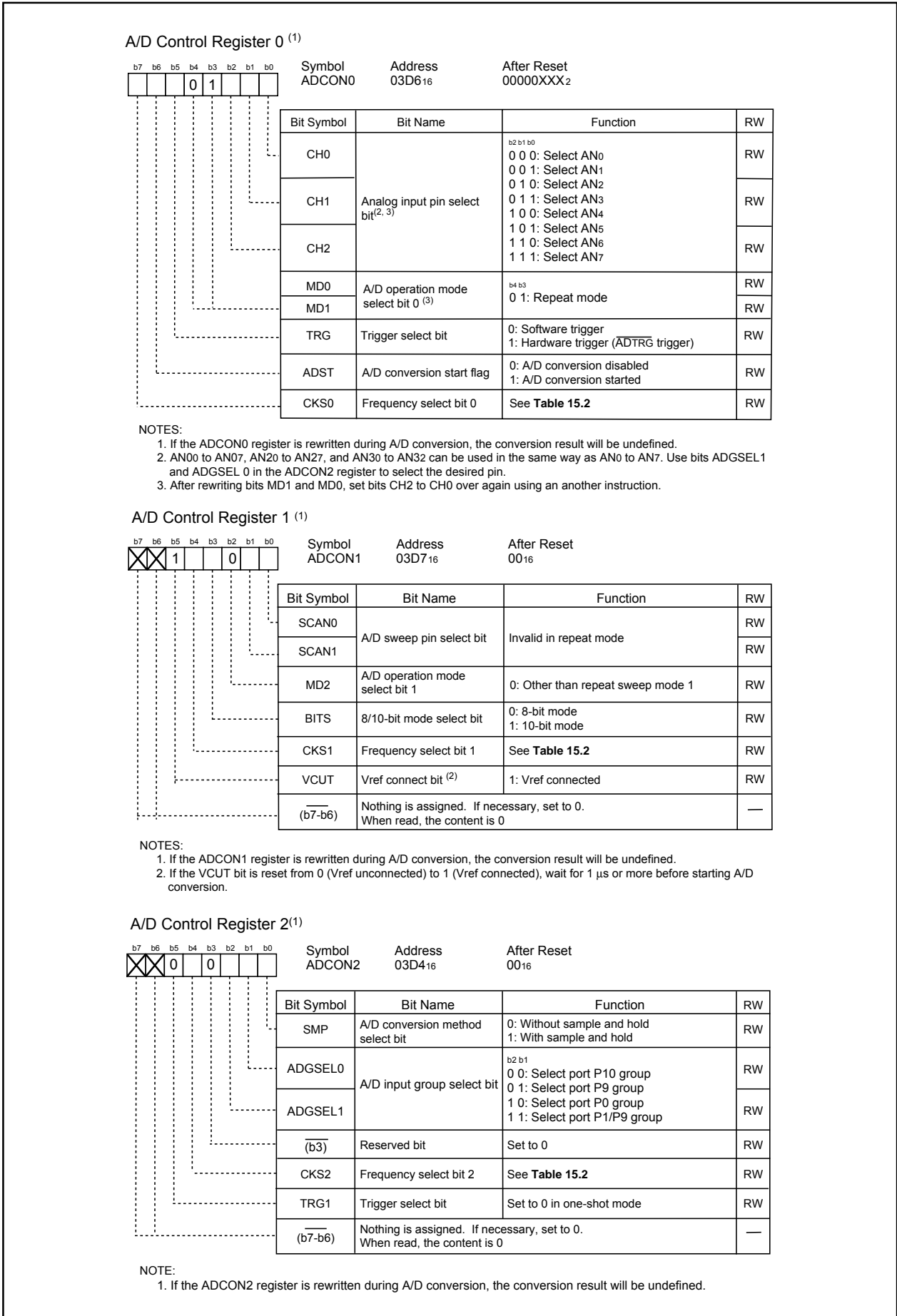


Figure 15.9 ADCON0 to ADCON2 Registers in Repeat Mode

15.1.3 Single Sweep Mode

In single sweep mode, analog voltages applied to the selected pins are converted one-by-one to a digital code. **Table 15.5** shows the single sweep mode specifications. **Figure 15.10** shows the operation example in single sweep mode. **Figure 15.11** shows the ADCON0 to ADCON2 registers in single sweep mode.

Table 15.5 Single Sweep Mode Specifications

| Item | Specification |
|-------------------------------------|--|
| Function | Bits SCAN1 to SCAN0 in the ADCON1 register and bits ADGSEL1 and ADGSEL0 in the ADCON2 register select pins. Analog voltage applied to the selected pins is converted one-by-one to a digital code |
| A/D Conversion Start Condition | <ul style="list-style-type: none"> When the TRG bit in the ADCON0 register is 0 (software trigger) Set the ADST bit in the ADCON0 register to 1 (A/D conversion started) When the TRG bit in the ADCON0 register is 1 (hardware trigger) The $\overline{\text{ADTRG}}$ pin input changes state from "H" to "L" after setting the ADST bit to 1 (A/D conversion started) |
| A/D Conversion Stop Condition | <ul style="list-style-type: none"> A/D conversion completed(When selecting a software trigger, the ADST bit is set to 0 (A/D conversion halted)). Set the ADST bit to 0 |
| Interrupt Request Generation Timing | A/D conversion completed |
| Analog Input Pin | Select from AN0 to AN1 (2 pins), AN0 to AN3 (4 pins), AN0 to AN5 (6 pins), AN0 to AN7 (8 pins) ⁽¹⁾ |
| Readout of A/D Conversion Result | Readout one of registers AD0 to AD7 that corresponds to the selected pin |

NOTE:

- AN0₀ to AN0₇, AN₂ to AN₂₇, and AN₃₀ to AN₃₂ can be used in the same way as AN0 to AN7. However, all input pins need to belong to the same group.

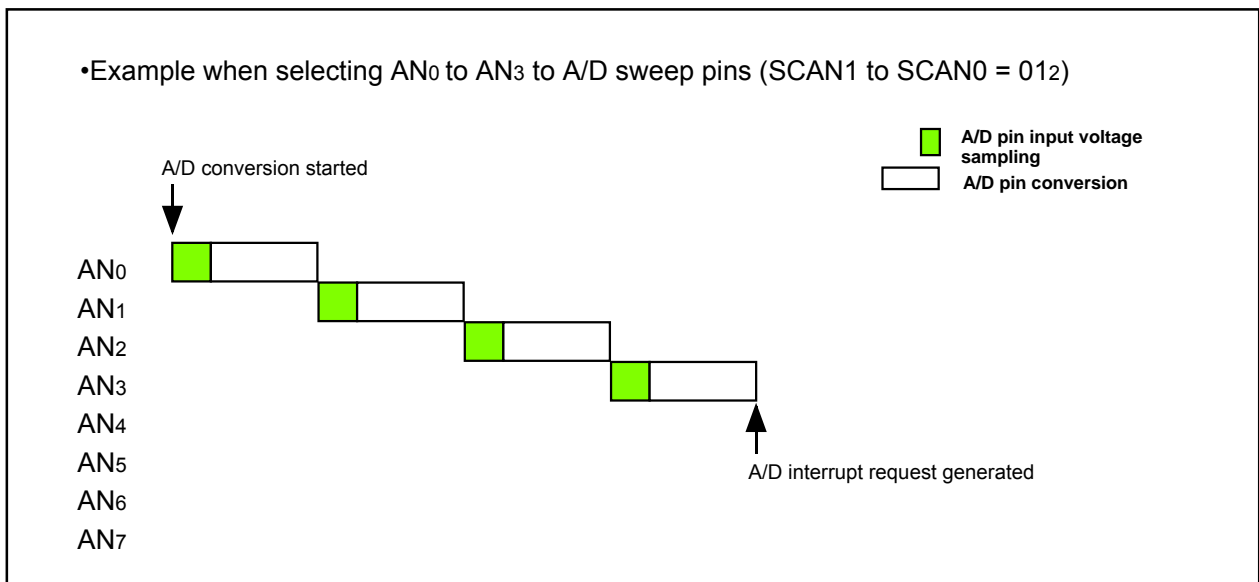


Figure 15.10 Operation Example in Single Sweep Mode

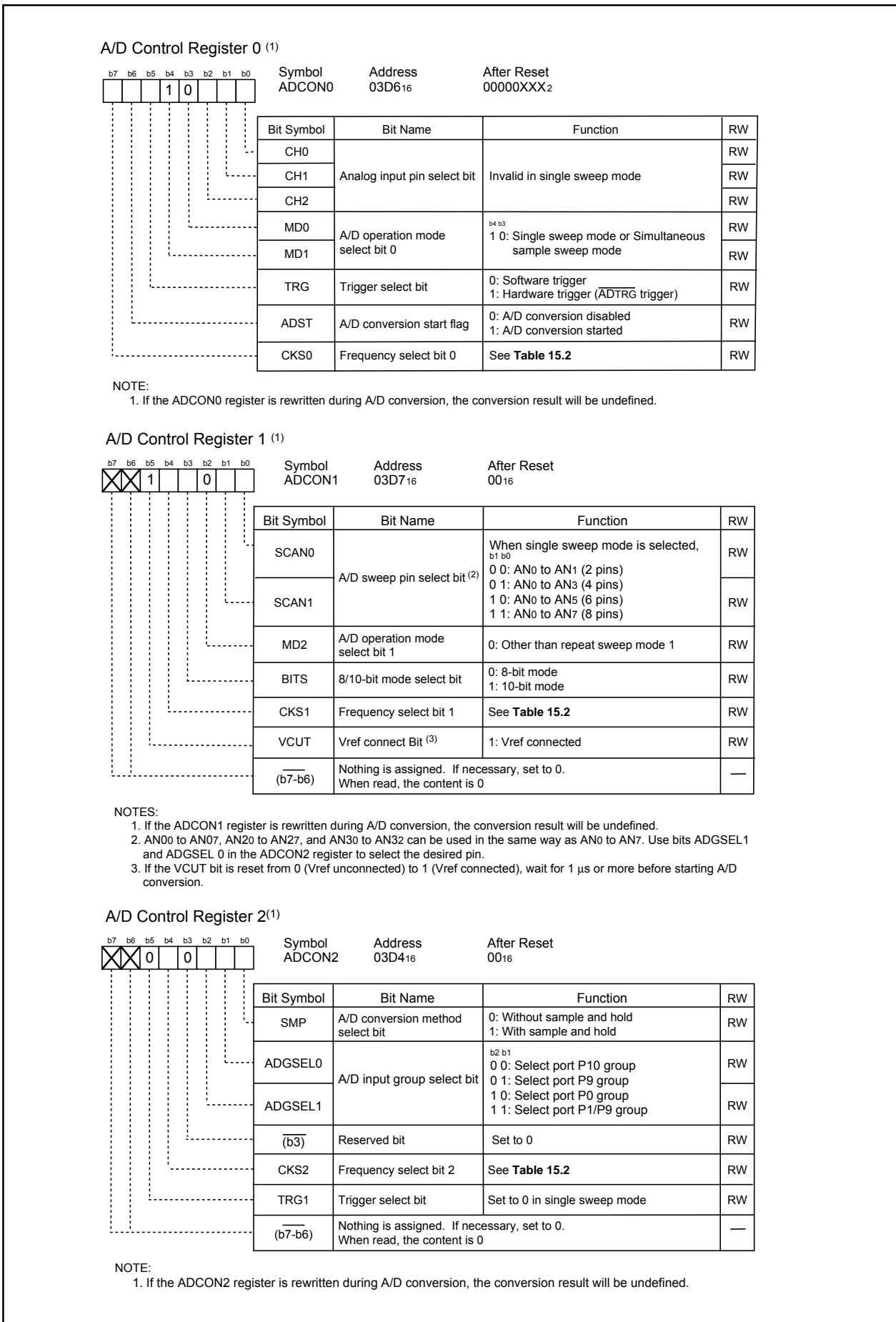


Figure 15.11 ADCON0 to ADCON2 Registers in Single Sweep Mode

15.1.4 Repeat Sweep Mode 0

In repeat sweep mode 0, analog voltages applied to the selected pins are repeatedly converted to a digital code. **Table 15.6** shows the repeat sweep mode 0 specifications. **Figure 15.12** shows the operation example in repeat sweep mode 0. **Figure 15.13** shows the ADCON0 to ADCON2 registers in repeat sweep mode 0.

Table 15.6 Repeat Sweep Mode 0 Specifications

| Item | Specification |
|-------------------------------------|--|
| Function | Bits SCAN1 and SCAN0 in the ADCON1 register and bits ADGSEL1 and ADGSEL0 in the ADCON2 register select pins. Analog voltage applied to the selected pins is repeatedly converted to a digital code |
| A/D Conversion Start Condition | <ul style="list-style-type: none"> When the TRG bit in the ADCON0 register is 0 (software trigger) Set the ADST bit in the ADCON0 register to 1 (A/D conversion started) When the TRG bit in the ADCON0 register is 1 (Hardware trigger) The $\overline{\text{ADTRG}}$ pin input changes state from “H” to “L” after setting the ADST bit to 1 (A/D conversion started) |
| A/D Conversion Stop Condition | Set the ADST bit to 0 (A/D conversion halted) |
| Interrupt Request Generation Timing | None generated |
| Analog Input Pin | Select from AN0 to AN1 (2 pins), AN0 to AN3 (4 pins), AN0 to AN5 (6 pins), AN0 to AN7 (8 pins) ⁽¹⁾ |
| Readout of A/D Conversion Result | Readout one of registers AD0 to AD7 that corresponds to the selected pin |

NOTES:

- AN00 to AN07, AN 20 to AN27, and AN30 to AN32 can be used in the same way as AN0 to AN7. However, all input pins need to belong to the same group.

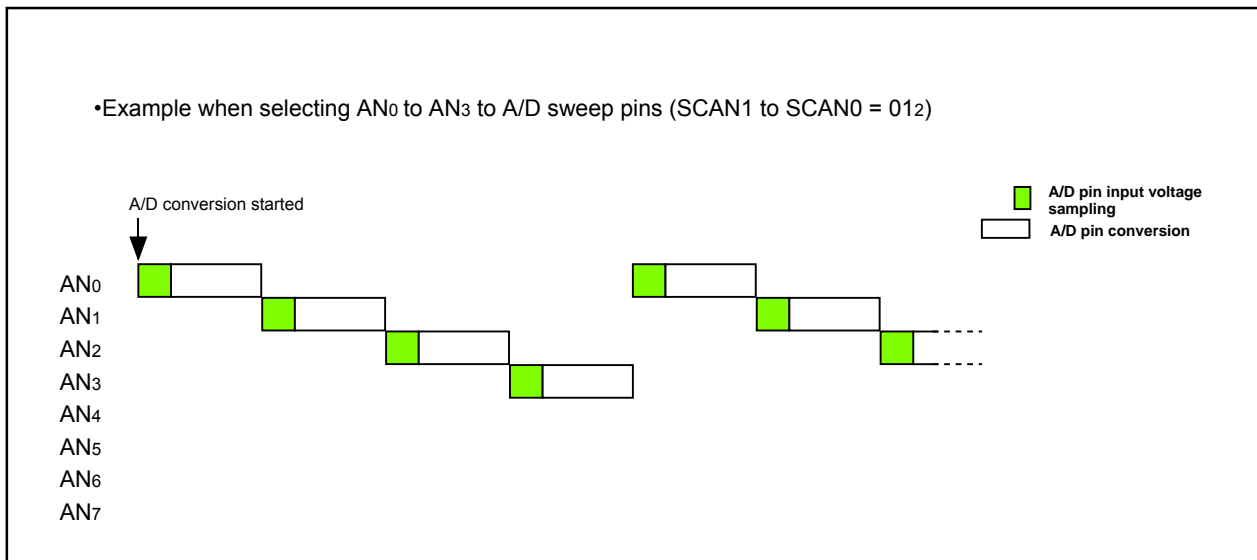
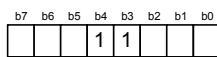


Figure 15.12 Operation Example in Repeat Sweep Mode 0

A/D Control Register 0 ⁽¹⁾



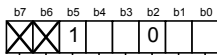
Symbol: ADCON0
Address: 03D6₁₆
After Reset: 0000XXX₂

| Bit Symbol | Bit Name | Function | RW |
|------------|---------------------------------|--|----|
| CH0 | Analog input pin select bit | Invalid in repeat sweep mode 0 | RW |
| CH1 | | | RW |
| CH2 | | | RW |
| MD0 | A/D operation mode select bit 0 | b ₄ b ₃ 1 1: Repeat sweep mode 0 or repeat sweep mode 1 | RW |
| MD1 | | | RW |
| TRG | Trigger select bit | 0: Software trigger 1: Hardware trigger (ADTRG trigger) | RW |
| ADST | A/D conversion start flag | 0: A/D conversion disabled 1: A/D conversion started | RW |
| CKS0 | Frequency select bit 0 | See Table 15.2 | RW |

NOTE:

1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be undefined.

A/D Control Register 1 ⁽¹⁾



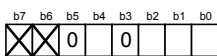
Symbol: ADCON1
Address: 03D7₁₆
After Reset: 00₁₆

| Bit Symbol | Bit Name | Function | RW |
|-----------------------------------|--|--|----|
| SCAN0 | A/D sweep pin select bit ⁽²⁾ | When repeat sweep mode 0 is selected, b ₁ b ₀ 0 0: AN ₀ to AN ₁ (2 pins) 0 1: AN ₀ to AN ₃ (4 pins) 1 0: AN ₀ to AN ₅ (6 pins) 1 1: AN ₀ to AN ₇ (8 pins) | RW |
| SCAN1 | | | RW |
| MD2 | A/D operation mode select bit 1 | 0: Other than repeat sweep mode 1 | RW |
| BITS | 8/10-bit mode select bit | 0: 8-bit mode 1: 10-bit mode | RW |
| CKS1 | Frequency select bit 1 | See Table 15.2 | RW |
| VCUT | Vref connect Bit ⁽³⁾ | 1: Vref connected | RW |
| (b ₇ -b ₆) | Nothing is assigned. If necessary, set to 0. When read, the content is 0 | | — |

NOTES:

1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be undefined.
2. AN₀ to AN₇, AN₂₀ to AN₂₇, and AN₃₀ to AN₃₂ can be used in the same way as AN₀ to AN₇. Use bits ADGSEL1 and ADGSEL0 in the ADCON2 register to select the desired pin.
3. If the VCUT bit is reset from 0 (Vref unconnected) to 1 (Vref connected), wait for 1 μs or more before starting A/D conversion.

A/D Control Register 2 ⁽¹⁾



Symbol: ADCON2
Address: 03D4₁₆
After Reset: 00₁₆

| Bit Symbol | Bit Name | Function | RW |
|-----------------------------------|--|---|----|
| SMP | A/D conversion method select bit | 0: Without sample and hold 1: With sample and hold | RW |
| ADGSEL0 | A/D input group select bit | b ₂ b ₁ 0 0: Select port P10 group 0 1: Select port P9 group 1 0: Select port P0 group 1 1: Select port P1/P9 group | RW |
| ADGSEL1 | | | RW |
| (b ₃) | Reserved bit | Set to 0 | RW |
| CKS2 | Frequency select bit 2 | See Table 15.2 | RW |
| TRG1 | Trigger select bit | Set to 0 in repeat sweep mode 0 | RW |
| (b ₇ -b ₆) | Nothing is assigned. If necessary, set to 0. When read, the content is 0 | | — |

NOTE:

1. If the ADCON2 register is rewritten during A/D conversion, the conversion result will be undefined.

Figure 15.13 ADCON0 to ADCON2 Registers in Repeat Sweep Mode 0

15.1.5 Repeat Sweep Mode 1

In repeat sweep mode 1, analog voltage is applied to the all selected pins are converted to a digital code, with mainly used in the selected pins. **Table 15.7** shows the repeat sweep mode 1 specifications. **Figure 15.14** shows the operation example in repeat sweep mode 1. **Figure 15.15** shows registers ADCON0 to ADCON2 in repeat sweep mode 1.

Table 15.7 Repeat Sweep Mode 1 Specifications

| Item | Specification |
|--|---|
| Function | Bits SCAN1 and SCAN0 in the ADCON1 register and bits ADGSEL1 and ADGSEL0 in the ADCON2 register mainly select pins. Analog voltage applied to the all selected pins is repeatedly converted to a digital code Example : When selecting AN0 Analog voltage is converted to a digital code in the following order AN0 → AN1 → AN0 → AN2 → AN0 → AN3, and so on. |
| A/D Conversion Start Condition | <ul style="list-style-type: none"> When the TRG bit in the ADCON0 register is 0 (software trigger) Set the ADST bit in the ADCON0 register to 1 (A/D conversion started) When the TRG bit in the ADCON0 register is 1 (hardware trigger) The ADTRG pin input changes state from "H" to "L" after setting the ADST bit to 1 (A/D conversion started) |
| A/D Conversion Stop Condition | Set the ADST bit to 0 (A/D conversion halted) |
| Interrupt Request Generation Timing | None generated |
| Analog Input Pins Mainly Used in A/D Conversions | Select from AN0 (1 pins), AN0 to AN1 (2 pins), AN0 to AN2 (3 pins), AN0 to AN3 (4 pins) ⁽¹⁾ |
| Readout of A/D Conversion Result | Readout one of registers AD0 to AD7 that corresponds to the selected pin |

NOTES:

1. AN00 to AN07, AN 20 to AN27, and AN30 to AN32 can be used in the same way as AN0 to AN7. However, all input pins need to belong to the same group.

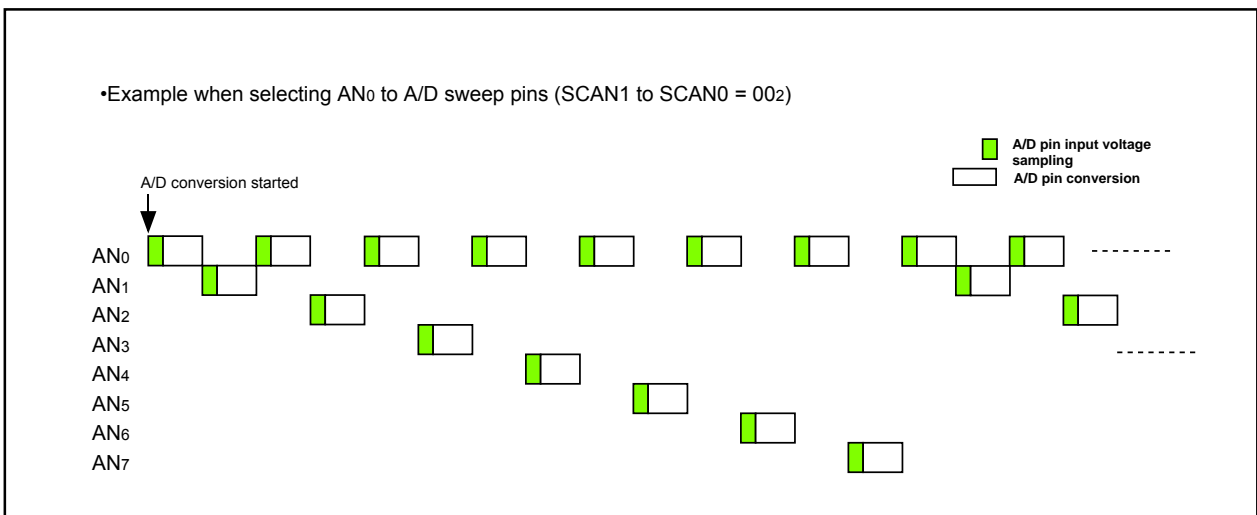


Figure 15.14 Operation Example in Repeat Sweep Mode 1

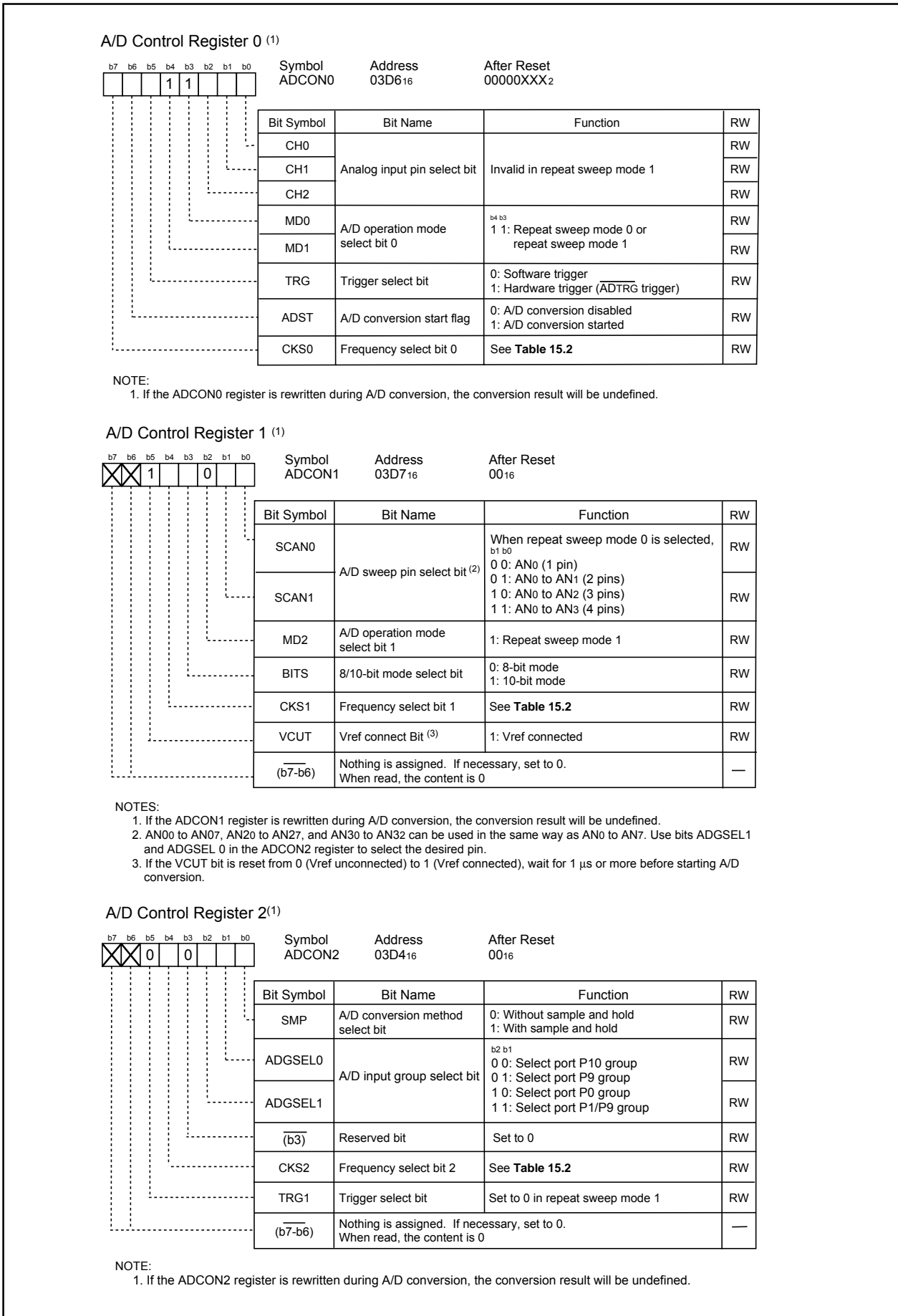


Figure 15.15 ADCON0 to ADCON2 Registers in Repeat Sweep Mode 1

15.1.6 Simultaneous Sample Sweep Mode

In simultaneous sample sweep mode, analog voltages applied to the selected pins are converted one-by-one to a digital code. The input voltages of AN0 and AN1 are sampled simultaneously using two circuits of sample and hold circuit. **Table 15.8** shows the simultaneous sample sweep mode specifications. **Figure 15.16** shows the operation example in simultaneous sample sweep mode. **Figure 15.17** shows registers ADCON0 to ADCON2 and **Figure 15.18** shows ADTRGCON registers in simultaneous sample sweep mode. **Table 15.9** shows the trigger select bit setting in simultaneous sample sweep mode. In simultaneous sample sweep mode, Timer B0 underflow can be selected as a trigger by combining software trigger, $\overline{\text{ADTRG}}$ trigger, Timer B2 underflow, Timer B2 interrupt generation frequency setting counter underflow or A/D trigger mode of Timer B.

Table 15.8 Simultaneous Sample Sweep Mode Specifications

| Item | Specification |
|----------------------------------|---|
| Function | Bits SCAN1 and SCAN0 in the ADCON1 register and bits ADGSEL1 and ADGSEL0 in the ADCON2 register select pins. Analog voltage applied to the selected pins is converted one-by-one to a digital code. At this time, the input voltage of AN0 and AN1 are sampled simultaneously. |
| A/D Conversion Start Condition | When the TRG bit in the ADCON0 register is 0 (software trigger) Set the ADST bit in the ADCON0 register to 1 (A/D conversion started) When the TRG bit in the ADCON0 register is 1 (hardware trigger) The trigger is selected by bits TRG1 and HPTRG0 (See Table 15.9) The $\overline{\text{ADTRG}}$ pin input changes state from “H” to “L” after setting the ADST bit to 1 (A/D conversion started) Timer B0, B2 or Timer B2 interrupt generation frequency setting counter underflow after setting the ADST bit to 1 (A/D conversion started) |
| A/D Conversion Stop Condition | A/D conversion completed (If selecting software trigger, the ADST bit is automatically set to 0). Set the ADST bit to 0 (A/D conversion halted) |
| Interrupt Generation Timing | A/D conversion completed |
| Analog Input Pin | Select from AN0 to AN1 (2 pins), AN0 to AN3 (4 pins), AN0 to AN5 (6 pins), or AN0 to AN7 (8 pins) ⁽¹⁾ |
| Readout of A/D conversion result | Readout one of registers AN0 to AN7 that corresponds to the selected pin |

NOTE:

1. AN0 to AN07, AN 2 to AN27, and AN30 to AN32 can be used in the same way as AN0 to AN7. However, all input pins need to belong to the same group.

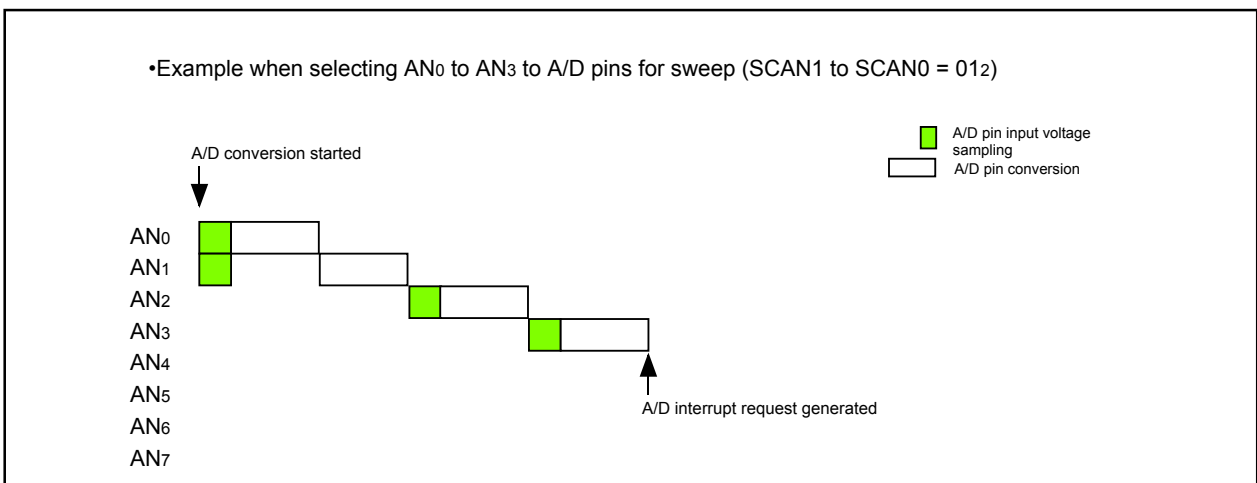


Figure 15.16 Operation Example in Simultaneous Sample Sweep Mode

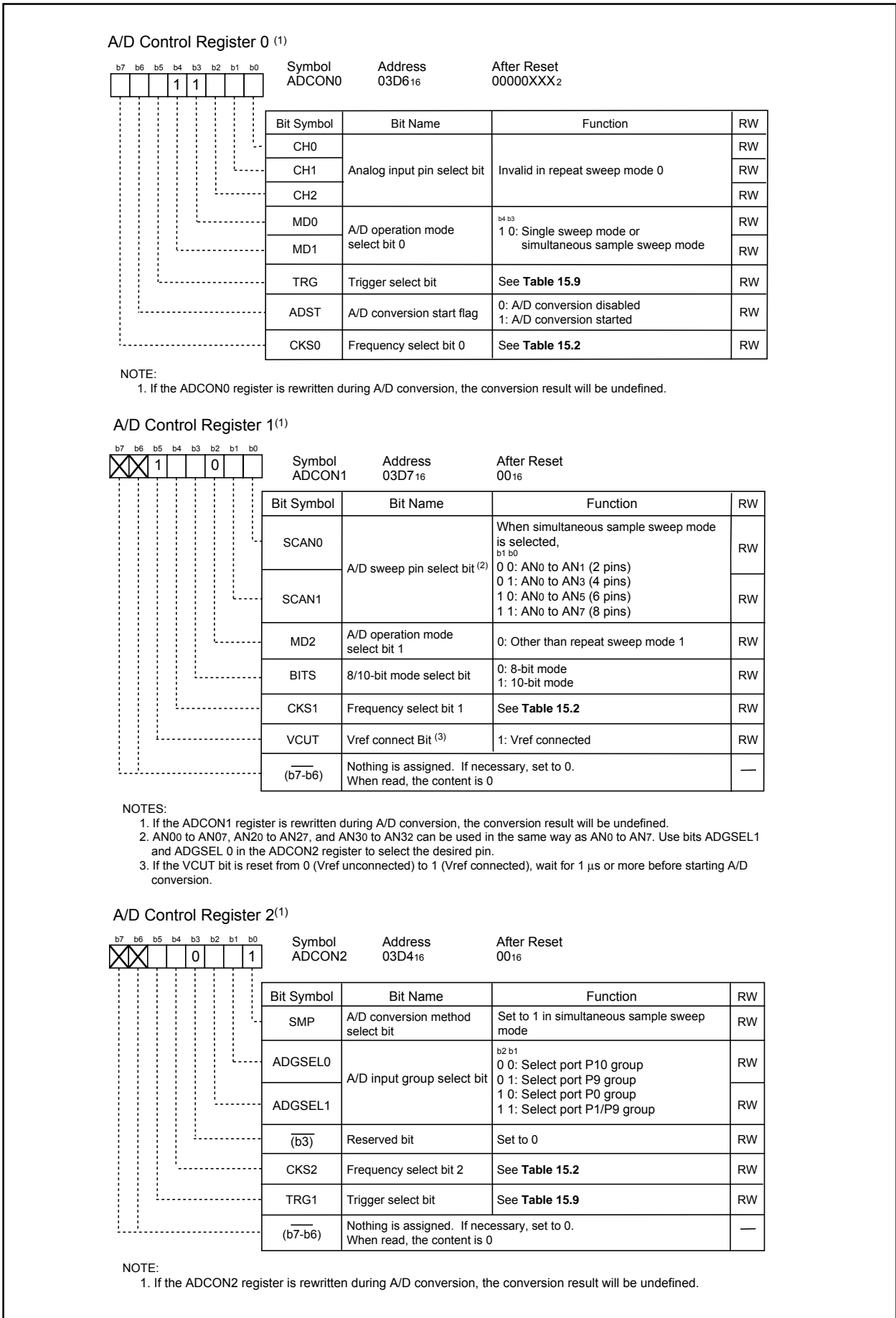


Figure 15.17 ADCON0 to ADCON2 Registers in Simultaneous Sample Sweep Mode

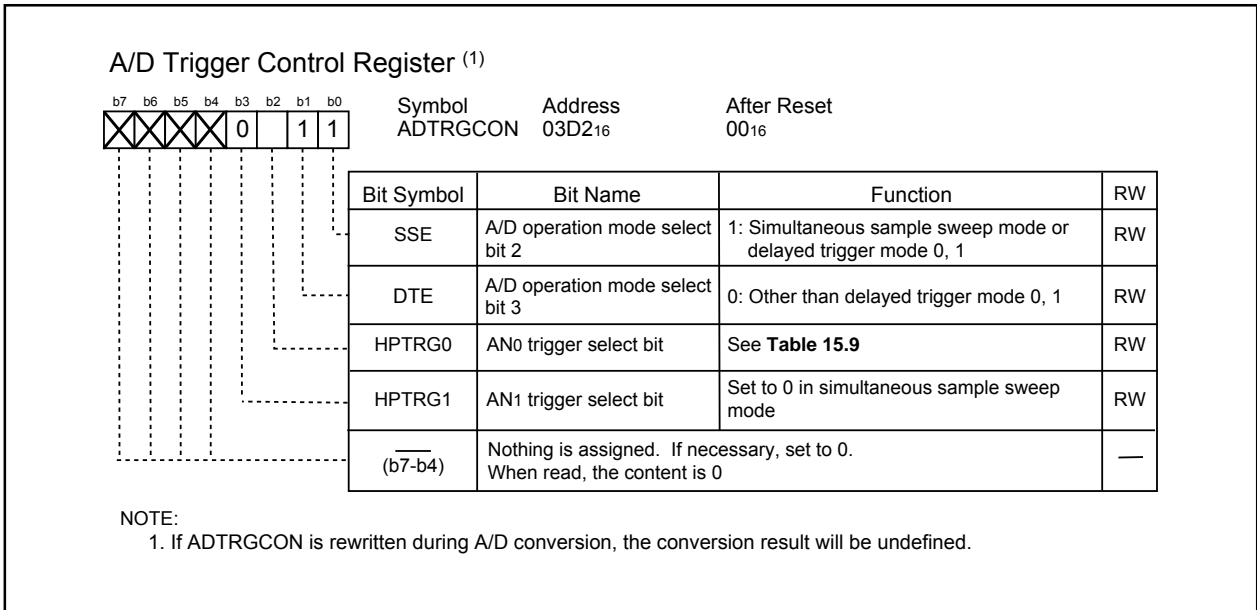


Figure 15.18 ADTRGCON Register in Simultaneous Sample Sweep Mode

Table 15.9 Trigger Select Bit Setting in Simultaneous Sample Sweep Mode

| TRG | TRG1 | HPTRG0 | TRIGGER |
|-----|------|--------|---|
| 0 | - | - | Software trigger |
| 1 | - | 1 | Timer B0 underflow (1) |
| 1 | 0 | 0 | $\overline{\text{ADTRG}}$ |
| 1 | 1 | 0 | Timer B2 or Timer B2 interrupt generation frequency setting counter underflow (2) |

NOTES:

1. A count can be started for Timer B2, Timer B2 interrupt generation frequency setting counter underflow or the INT5 pin falling edge as count start conditions of Timer B0.
2. Select Timer B2 or Timer B2 interrupt generation frequency setting counter using the TB2SEL bit in the TB2SC register.

15.1.7 Delayed Trigger Mode 0

In delayed trigger mode 0, analog voltages applied to the selected pins are converted one-by-one to a digital code. The delayed trigger mode 0 used in combination with A/D trigger mode of Timer B. The Timer B0 underflow starts a single sweep conversion. After completing the AN0 pin conversion, the AN1 pin is not sampled and converted until the Timer B1 underflow is generated. When the Timer B1 underflow is generated, the single sweep conversion is restarted with the AN1 pin. **Table 15.10** shows the delayed trigger mode 0 specifications. **Figure 15.19** shows the operation example in delayed trigger mode 0. **Figures 15.20** and **15.21** show each flag operation in the ADSTAT0 register that corresponds to the operation example. **Figure 15.22** shows registers ADCON0 to ADCON2 in delayed trigger mode 0. **Figure 15.23** shows the ADTRGCON register in delayed trigger mode 0 and **Table 15.11** shows the trigger select bit setting in delayed trigger mode 0.

Table 15.10 Delayed Trigger Mode 0 Specifications

| Item | Specification |
|-------------------------------------|---|
| Function | Bits SCAN1 and SCAN0 in the ADCON1 register and bits ADGSEL1 and ADGSEL0 in the ADCON2 register select pins. Analog voltage applied to the input voltage of the selected pins are converted one-by-one to the digital code. At this time, timer B0 underflow generation starts AN0 pin conversion. Timer B1 underflow generation starts conversion after the AN1 pin. ⁽¹⁾ |
| A/D Conversion Start | AN0 pin conversion start condition <ul style="list-style-type: none"> •When Timer B0 underflow is generated if Timer B0 underflow is generated again before Timer B1 underflow is generated , the conversion is not affected •When Timer B0 underflow is generated during A/D conversion of pins after the AN1 pin, conversion is halted and the sweep is restarted from the AN0 pin again AN1 pin conversion start condition <ul style="list-style-type: none"> •When Timer B1 underflow is generated during A/D conversion of the AN0 pin, the input voltage of the AN1 pin is sampled. The AN1 conversion and the rest of the sweep start when AN0 conversion is completed. |
| A/D Conversion Stop Condition | <ul style="list-style-type: none"> •When single sweep conversion from the AN0 pin is completed •Set the ADST bit to 0 (A/D conversion halted)⁽²⁾ |
| Interrupt request generation timing | A/D conversion completed |
| Analog input pin | Select from AN0 to AN1 (2 pins), AN0 to AN3 (4 pins), AN0 to AN5 (6 pins) and AN0 to AN7 (8 pins) ⁽³⁾ |
| Readout of A/D conversion result | Readout one of registers AN0 to AN7 that corresponds to the selected pins |

NOTES:

1. Set the larger value than the value of the timer B0 register to the timer B1 register. The count source for timer B0 and timer B1 must be the same.
2. Do not write 1 (A/D conversion started) to the ADST bit in delayed trigger mode 0. When write 1, unexpected interrupts may be generated.
3. AN0 to AN07, AN 2 to AN27, and AN30 to AN32 can be used in the same way as AN0 to AN7. However, all input pins need to belong to the same group.

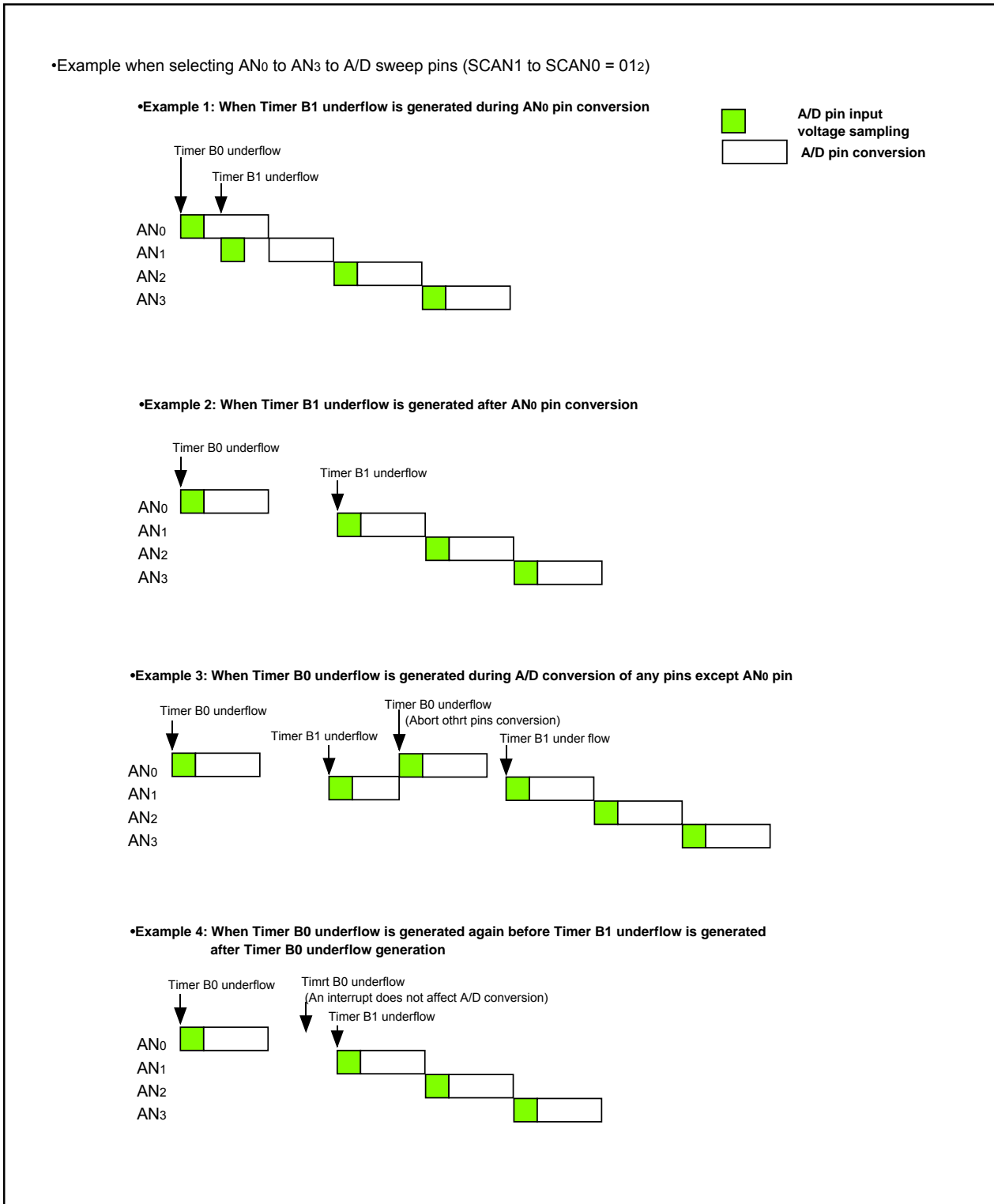
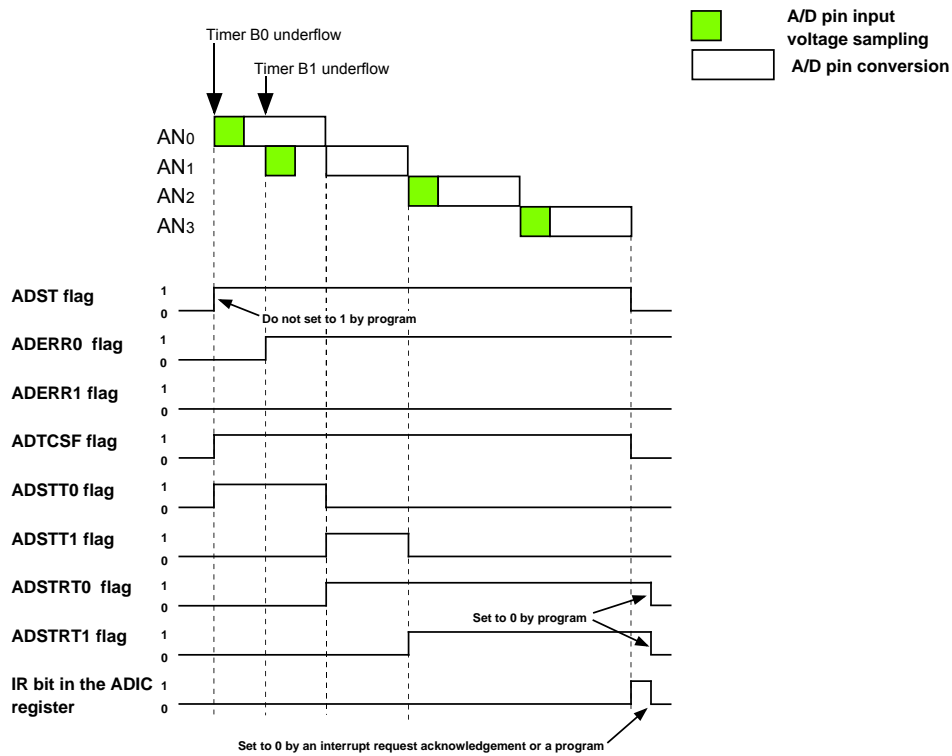


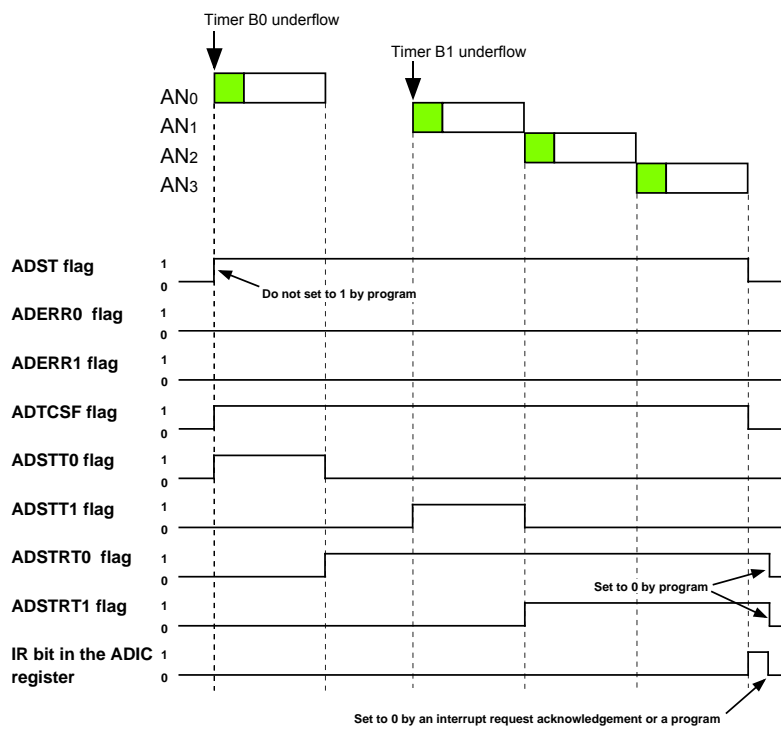
Figure 15.19 Operation Example in Delayed Trigger Mode 0

•Example when selecting AN0 to AN3 to A/D sweep pins (SCAN1 to SCAN0 = 012)

•Example 1: When Timer B1 underflow is generated during AN0 pin conversion



•Example 2: When Timer B1 underflow is generated after AN0 pin conversion



ADST flag: Bit 6 in the ADCON0 register
 ADERR0, ADERR1, ADTCSF, ADSTT0, ADSTT1, ADSTR0 and ADSTR1 flag: bits 0, 1, 3, 4, 5, 6 and 7 in the ADSTAT0 register

Figure 15.20 Each Flag Operation in ADSTAT0 Register Associated with the Operation Example in Delayed Trigger Mode 0 (1)

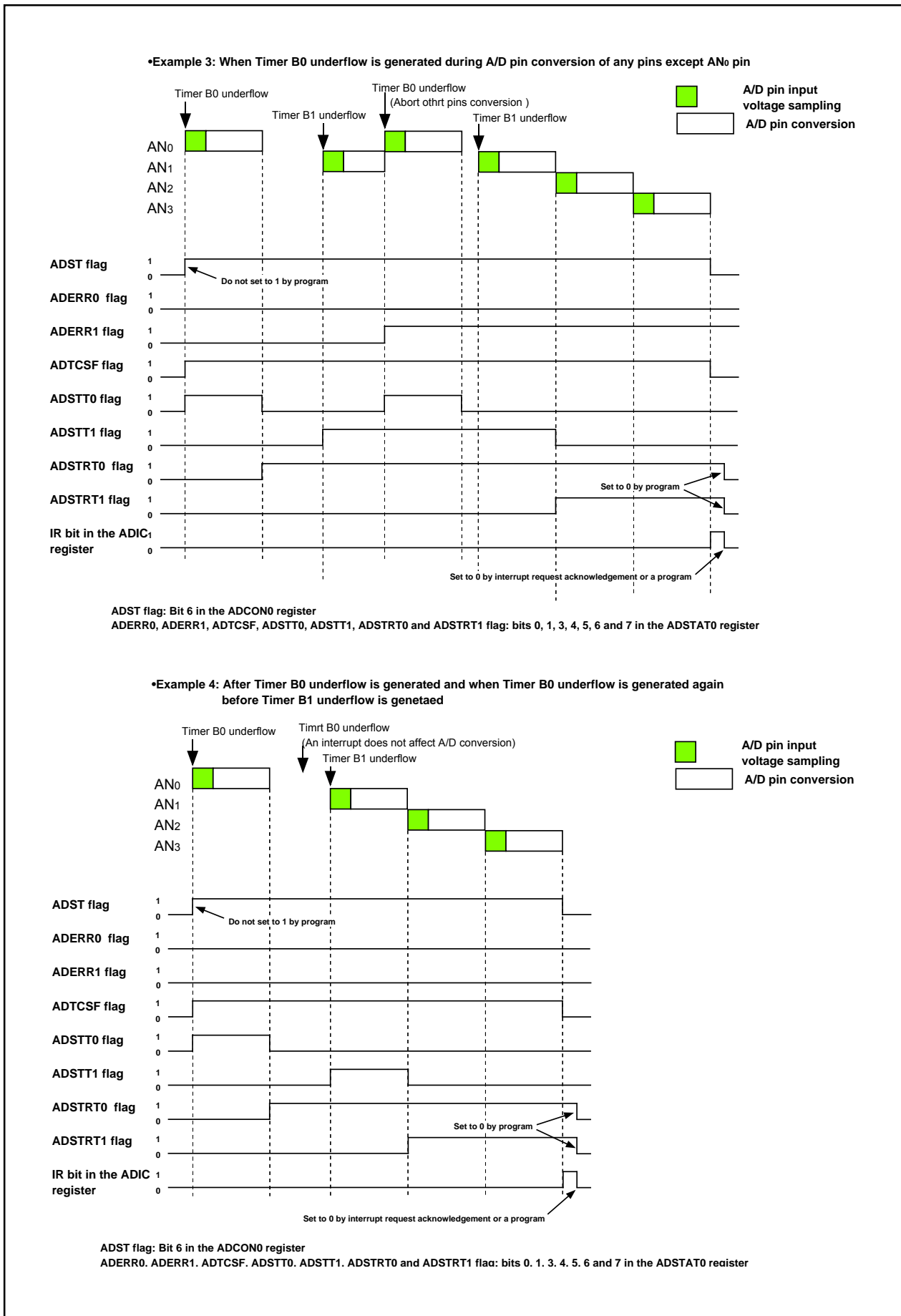
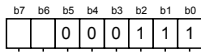


Figure 15.21 Each Flag Operation in ADSTAT0 Register Associated with the Operation Example in Delayed Trigger Mode 0 (2)

A/D Control Register 0 ⁽¹⁾



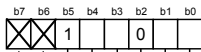
Symbol: ADCON0
Address: 03D6₁₆
After Reset: 00000XXX₂

| Bit Symbol | Bit Name | Function | RW |
|------------|--|--|----|
| CH0 | Analog input pin select bit | b2:b1:b0 1 1 1: Set to 111b in delayed trigger mode 0 | RW |
| CH1 | | | RW |
| CH2 | | | RW |
| MD0 | A/D operation mode select bit 0 | b4:b3 0 0: One-shot mode or delayed trigger mode | RW |
| MD1 | | | RW |
| TRG | Trigger select bit | Refer to Table 15.11 | RW |
| ADST | A/D conversion start flag ⁽²⁾ | 0: A/D conversion disabled 1: A/D conversion started | RW |
| CKS0 | Frequency select bit 0 | Refer to Table 15.2 | RW |

NOTES:

1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be undefined.
2. Do not write 1 in delayed trigger mode 0. When write, set to 0.

A/D Control Register 1 ⁽¹⁾



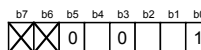
Symbol: ADCON1
Address: 03D7₁₆
After Reset: 00₁₆

| Bit Symbol | Bit Name | Function | RW |
|------------|--|--|----|
| SCAN0 | A/D sweep pin select bit ⁽²⁾ | When selecting delayed trigger sweep mode 0 b1:b0 0 0: AN0 to AN1 (2 pins) 0 1: AN0 to AN3 (4 pins) 1 0: AN0 to AN5 (6 pins) 1 1: AN0 to AN7 (8 pins) | RW |
| SCAN1 | | | RW |
| MD2 | A/D operation mode select bit 1 | 0: Any mode other than repeat sweep mode 1 | RW |
| BITS | 8/10-bit mode select bit | 0: 8-bit mode 1: 10-bit mode | RW |
| CKS1 | Frequency select bit 1 | Refer to Table 15.2 | RW |
| VCUT | Vref connect bit ⁽³⁾ | 1: Vref connected | RW |
| (b7-b6) | Nothing is assigned. If necessary, set to 0. When read, its content is 0 | | — |

NOTES:

1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be undefined.
2. AN0₀ to AN0₇, AN2₀ to AN2₇, and AN3₀ to AN3₂ can be used in the same way as AN₀ to AN₇. Use bits ADGSEL1 and ADGSEL0 in the ADCON2 register to select the desired pin.
3. If the VCUT bit is reset from 0 (Vref unconnected) to 1 (Vref connected), wait for 1 μs or more before starting A/D conversion.

A/D Control Register 2 ⁽¹⁾



Symbol: ADCON2
Address: 03D4₁₆
After Reset: 00₁₆

| Bit Symbol | Bit Name | Function | RW |
|------------|--|---|----|
| SMP | A/D conversion method select bit ⁽²⁾ | 1: With sample and hold | RW |
| ADGSEL0 | A/D input group select bit | b2:b1 0 0: Select port P10 group 0 1: Select port P9 group 1 0: Select port P0 group 1 1: Select port P1/P9 group | RW |
| ADGSEL1 | | | RW |
| (b3) | Reserved bit | | RW |
| CKS2 | Frequency select bit 2 | Refer to Table 15.2 | RW |
| TRG1 | Trigger select bit 1 | Refer to Table 15.11 | RW |
| (b7-b6) | Nothing is assigned. If necessary, set to 0. When read, its content is 0 | | — |

NOTES:

1. If the ADCON2 register is rewritten during A/D conversion, the conversion result will be undefined.
2. Set to 1 in delayed trigger mode 0.

Figure 15.22 ADCON0 to ADCON2 Registers in Delayed Trigger Mode 0

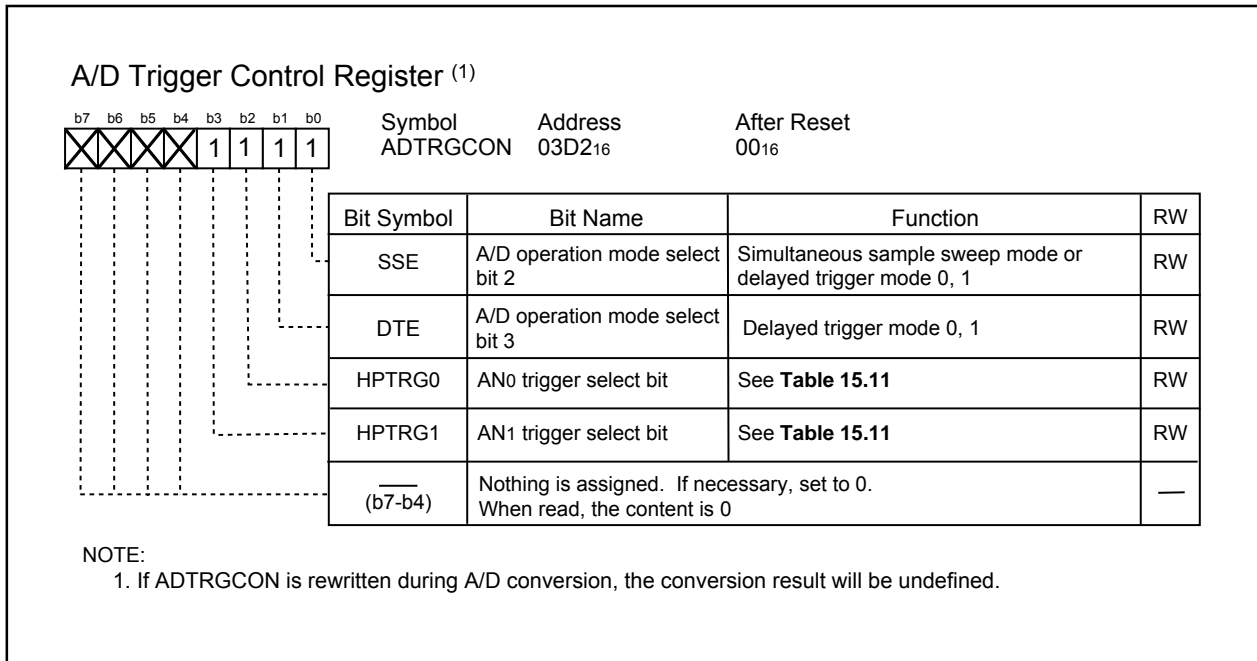


Figure 15.23 ADTRGCON Register in Delayed Trigger Mode 0

Table 15.11 Trigger Select Bit Setting in Delayed Trigger Mode 0

| TRG | TRG1 | HPTRG0 | HPTRG1 | Trigger |
|-----|------|--------|--------|------------------------|
| 0 | 0 | 1 | 1 | Timer B0, B1 underflow |

15.1.8 Delayed Trigger Mode 1

In delayed trigger mode 1, analog voltages applied to the selected pins are converted one-by-one to a digital code. When the input of the $\overline{\text{ADTRG}}$ pin (falling edge) changes state from “H” to “L”, a single sweep conversion is started. After completing the AN0 pin conversion, the AN1 pin is not sampled and converted until the second $\overline{\text{ADTRG}}$ pin falling edge is generated. When the second $\overline{\text{ADTRG}}$ falling edge is generated, the single sweep conversion of the pins after the AN1 pin is restarted. **Table 15.12** shows the delayed trigger mode 1 specifications. **Figure 15.24** shows the operation example of delayed trigger mode 1. **Figure 15.25** and **15.26** show each flag operation in the ADSTAT0 register that corresponds to the operation example. **Figure 15.27** shows registers ADCON0 to ADCON2 in delayed trigger mode 1. **Figure 15.28** shows the ADTRGCON register in delayed trigger mode 1. **Table 15.13** shows the trigger select bit setting in delayed trigger mode 1.

Table 15.12 Delayed Trigger Mode 1 Specifications

| Item | Specification |
|-------------------------------------|---|
| Function | Bits SCAN1 and SCAN0 in the ADCON1 register and bits ADGSEL1 and ADGSEL0 in the ADCON2 register select pins. Analog voltages applied to the selected pins are converted one-by-one to a digital code. At this time, the $\overline{\text{ADTRG}}$ pin falling edge starts AN0 pin conversion and the second $\overline{\text{ADTRG}}$ pin falling edge starts conversion of the pins after AN1 pin |
| A/D Conversion Start Condition | AN0 pin conversion start condition The $\overline{\text{ADTRG}}$ pin input changes state from “H” to “L” (falling edge) ⁽¹⁾ AN1 pin conversion start condition ⁽²⁾ The $\overline{\text{ADTRG}}$ pin input changes state from “H” to “L” (falling edge) •When the second $\overline{\text{ADTRG}}$ pin falling edge is generated during A/D conversion of the AN0 pin, input voltage of AN1 pin is sampled or after at the time of $\overline{\text{ADTRG}}$ falling edge. The conversion of AN1 and the rest of the sweep starts when AN0 conversion is completed. •When the $\overline{\text{ADTRG}}$ pin falling edge is generated again during single sweep conversion of pins after the AN1 pin, the conversion is not affected |
| A/D Conversion Stop Condition | •A/D conversion completed •Set the ADST bit to 0 (A/D conversion halted) ⁽³⁾ |
| Interrupt Request Generation Timing | Single sweep conversion completed |
| Analog Input Pin | Select from AN0 to AN1 (2 pins), AN0 to AN3 (4 pins), AN0 to AN5 (6 pins) and AN0 to AN7 (8 pins) ⁽⁴⁾ |
| Readout of A/D Conversion Result | Readout one of registers AN0 to AN7 that corresponds to the selected pins |

NOTES:

- Do not generate the next $\overline{\text{ADTRG}}$ pin falling edge after the AN1 pin conversion is started until all selected pins complete A/D conversion. When an $\overline{\text{ADTRG}}$ pin falling edge is generated again during A/D conversion, its trigger is ignored. The falling edge of $\overline{\text{ADTRG}}$ pin, which was input after all selected pins complete A/D conversion, is considered to be the next AN0 pin conversion start condition.
- The $\overline{\text{ADTRG}}$ pin falling edge is detected synchronized with the operation clock fAD. Therefore, when the $\overline{\text{ADTRG}}$ pin falling edge is generated in shorter periods than fAD, the second $\overline{\text{ADTRG}}$ pin falling edge may not be detected. Do not generate the $\overline{\text{ADTRG}}$ pin falling edge in shorter periods than fAD.
- Do not write 1 (A/D conversion started) to the ADST bit in delayed trigger mode 1. When write 1, unexpected interrupts may be generated.
- AN0 to AN07, AN 2 to AN27, and AN30 to AN32 can be used in the same way as AN0 to AN7. However, all input pins need to belong to the same group.

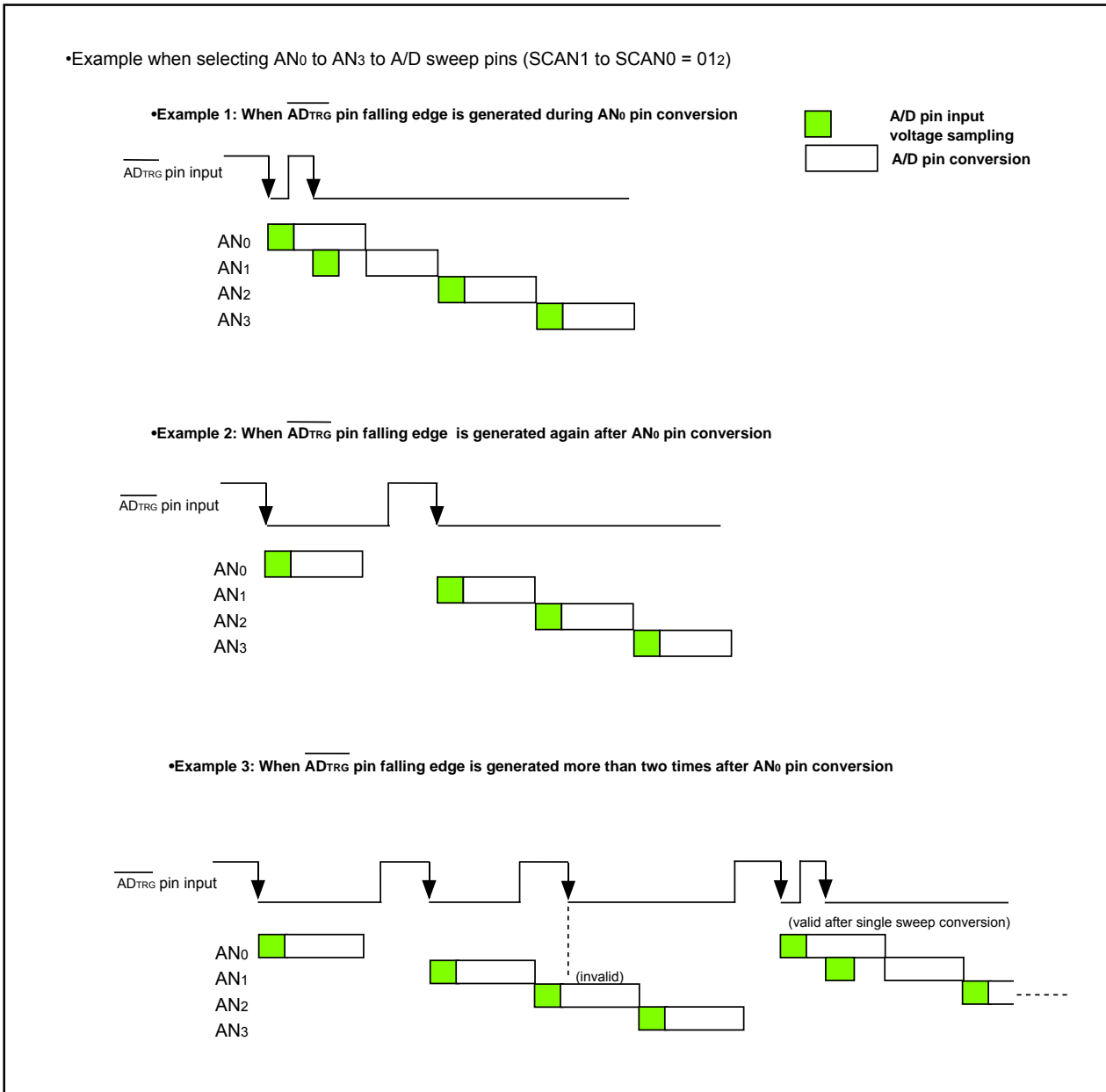
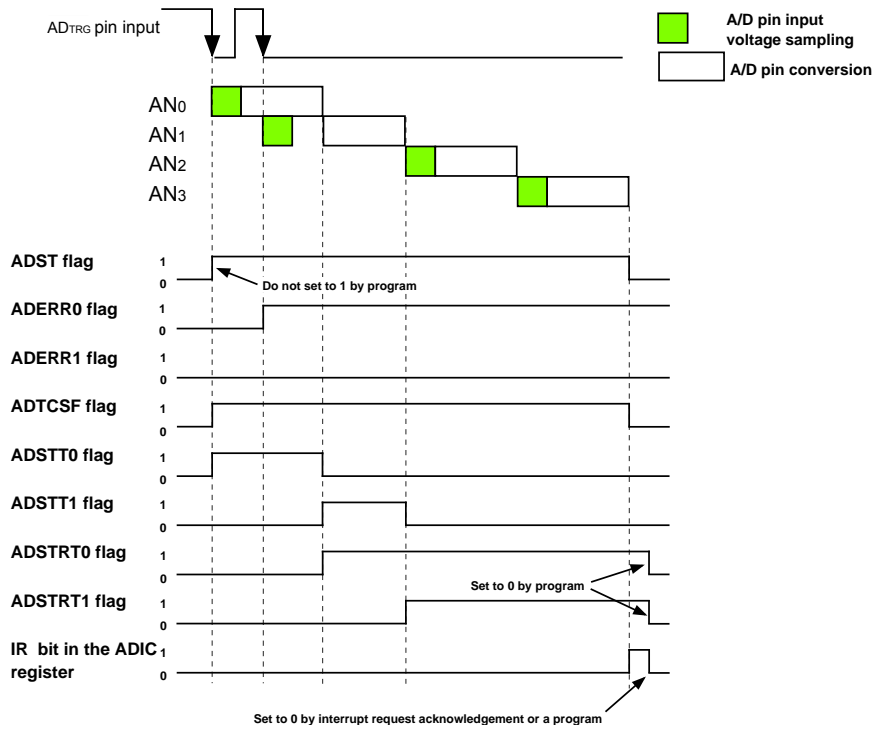


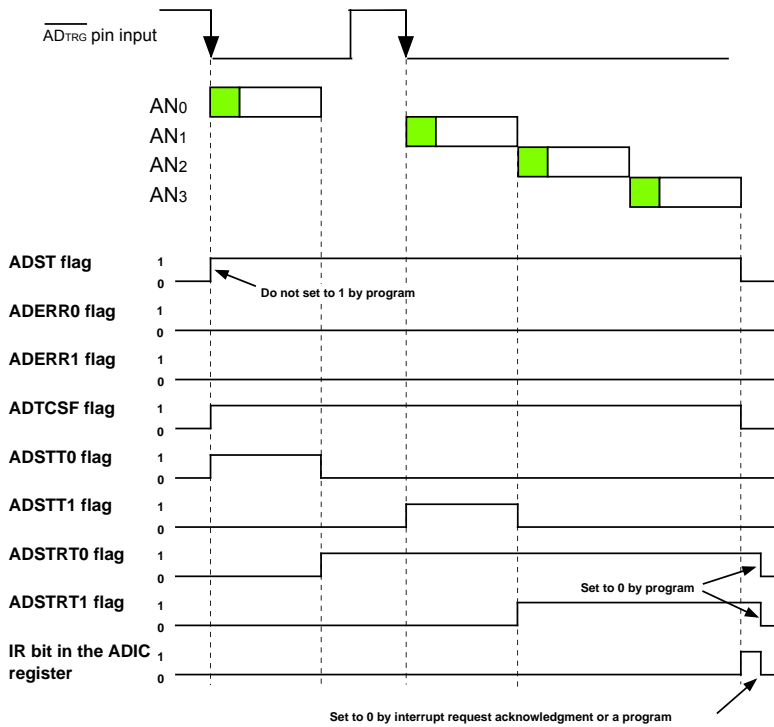
Figure 15.24 Operation Example in Delayed Trigger Mode1

•Example when selecting AN0 to AN3 to A/D sweep pins (SCAN1 to SCAN0 = 012)

•Example 1: When ADTRG pin falling edge is generated during AN0 pin conversion



•Example 2: When ADTRG pin falling edge is generated again after AN0 pin conversion



ADST flag: Bit 6 in the ADCON0 register
 ADERR0, ADERR1, ADTCSF, ADSTT0, ADSTT1, ADSTR0 and ADSTR1 flag: bits 0, 1, 3, 4, 5, 6 and 7 in the ADSTAT0 register

Figure 15.25 Each Flag Operation in ADSTAT0 Register Associated with the Operation Example in Delayed Trigger Mode 1 (1)

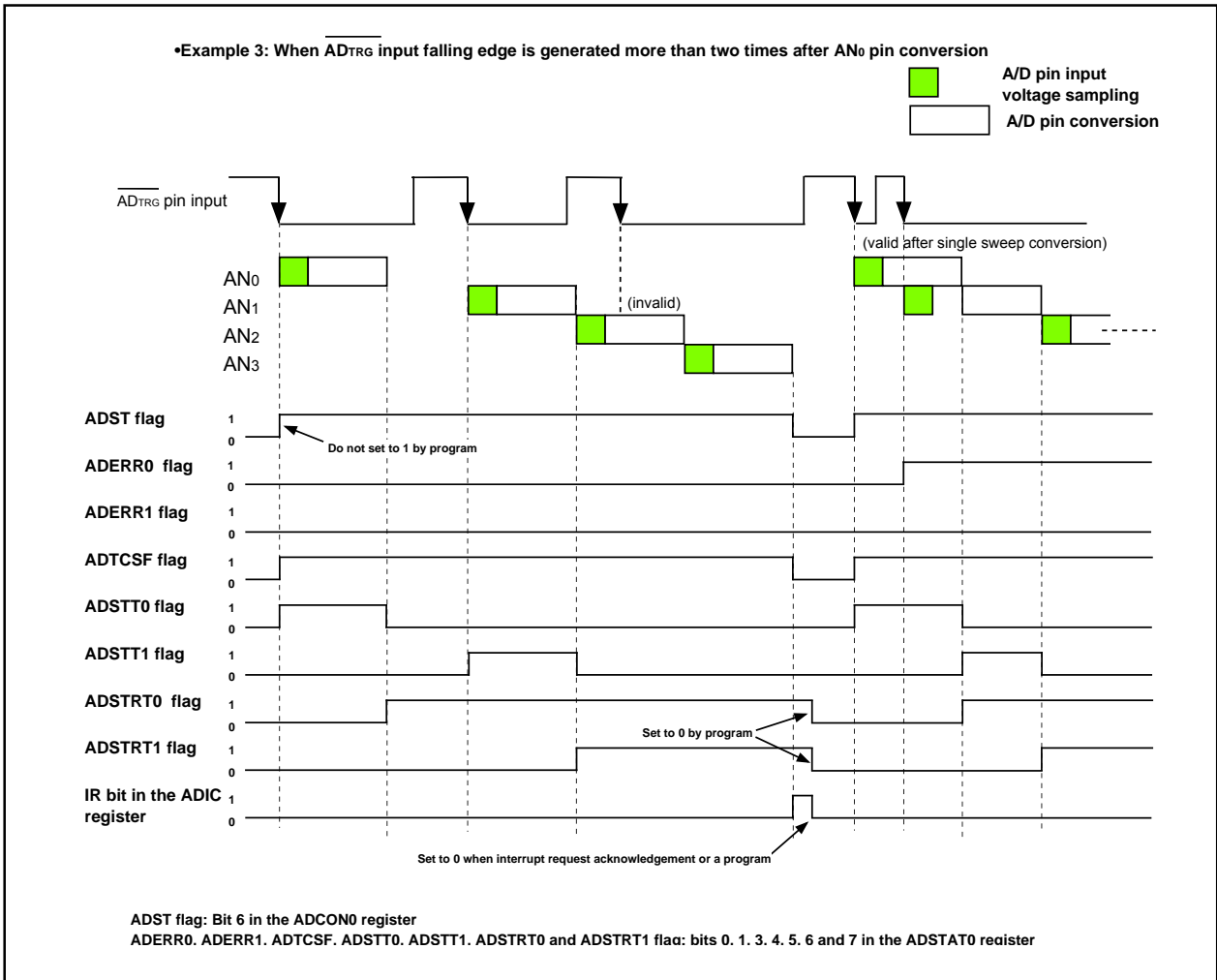


Figure 15.26 Each Flag Operation in ADSTAT0 Register Associated with the Operation Example in Delayed Trigger Mode 1 (2)

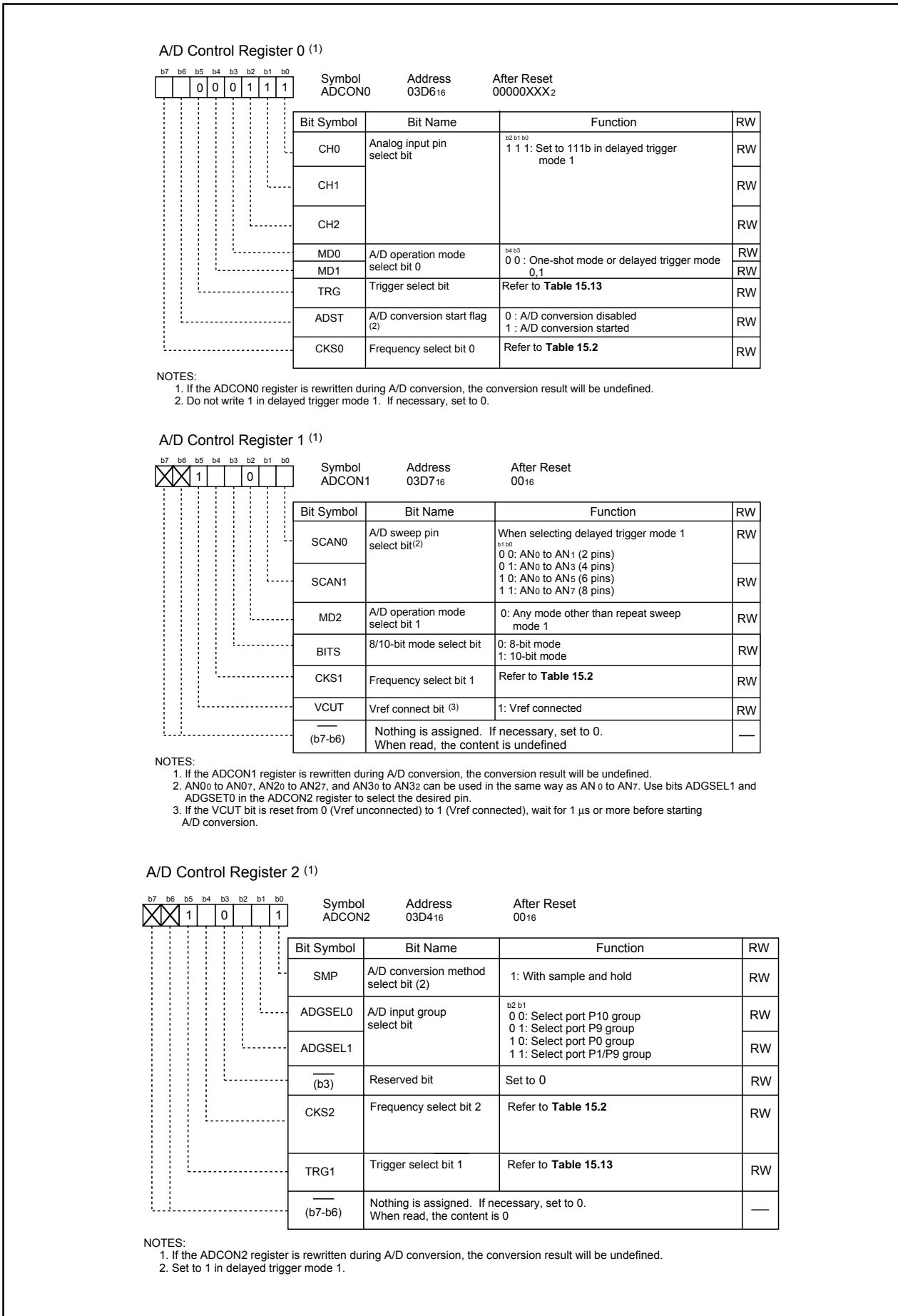


Figure 15.27 ADCON0 to ADCON2 Registers in Delayed Trigger Mode 1

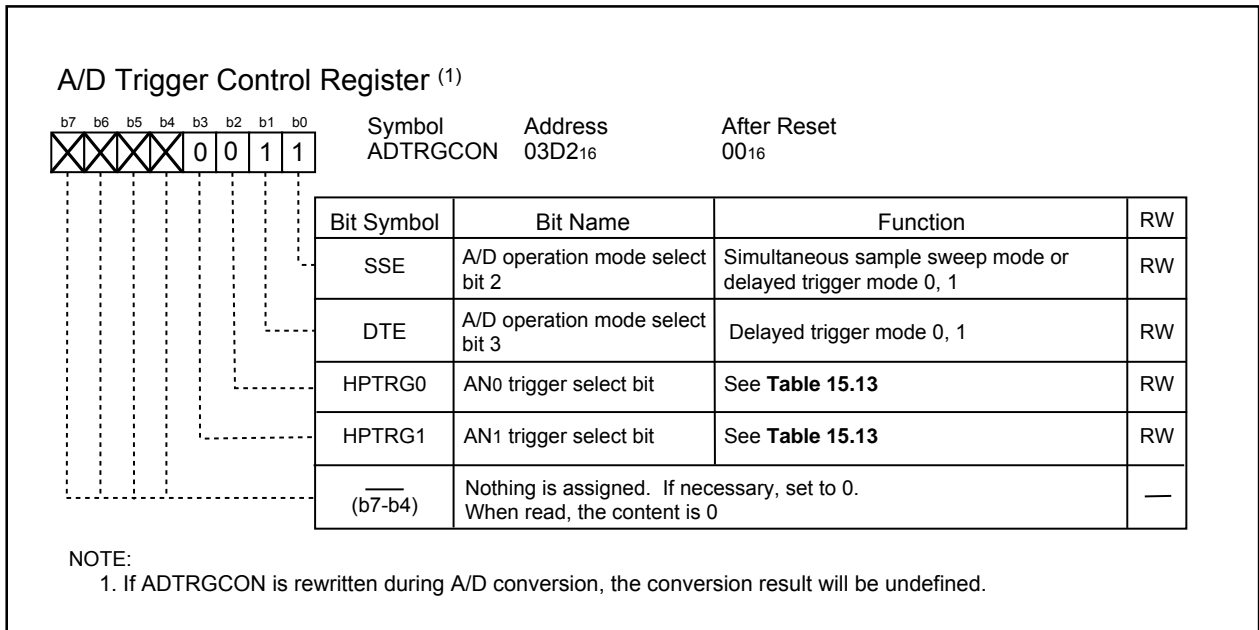


Figure 15.28 ADTRGCON Register in Delayed Trigger Mode 1

Table 15.13 Trigger Select Bit Setting in Delayed Trigger Mode 1

| TRG | TRG1 | HPTRG0 | HPTRG1 | Trigger |
|-----|------|--------|--------|--------------|
| 0 | 1 | 0 | 0 | <u>ADTRG</u> |

15.2 Resolution Select Function

The BITS bit in the ADCON1 register determines the resolution. When the BITS bit is set to 1 (10-bit precision), the A/D conversion result is stored into bits 0 to 9 in the ADi register (i=0 to 7). When the BITS bit is set to 0 (8-bit precision), the A/D conversion result is stored into bits 7 to 0 in the ADi register.

15.3 Sample and Hold

When the SMP bit in the ADCON 2 register is set to 1 (with the sample and hold function), A/D conversion rate per pin increases to 28 ϕ AD cycles for 8-bit resolution or 33 ϕ AD cycles for 10-bit resolution. The sample and hold function is available in one-shot mode, repeat mode, single sweep mode, repeat sweep mode 0 and repeat sweep mode 1. In these modes, start A/D conversion after selecting whether the sample and hold circuit is to be used or not. In simultaneous sample sweep mode, delayed trigger mode 0 or delayed trigger mode, set to use the Sample and Hold function before starting A/D conversion.

15.4 Power Consumption Reducing Function

When the A/D converter is not used, the VCUT bit in the ADCON1 register isolates the resistor ladder of the A/D converter from the reference voltage input pin (VREF). Power consumption is reduced by shutting off any current flow into the resistor ladder from the VREF pin.

When using the A/D converter, set the VCUT bit to 1 (Vref connected) before setting the ADST bit in the ADCON0 register to 1 (A/D conversion started). Do not set the ADST bit and VCUT bit to 1 simultaneously, nor set the VCUT bit to 0 (Vref unconnected) during A/D conversion.

15.5 Output Impedance of Sensor under A/D Conversion

To carry out A/D conversion properly, charging the internal capacitor C shown in **Figure 15.29** has to be completed within a specified period of time. T (sampling time) as the specified time. Let output impedance of sensor equivalent circuit be R0, MCU's internal resistance be R, precision (error) of the A/D converter be X, and the A/D converter's resolution be Y (Y is 1024 in the 10-bit mode, and 256 in the 8-bit mode).

$$VC \text{ is generally } VC = VIN \left\{ 1 - e^{-\frac{1}{C(R0+R)} t} \right\}$$

$$\text{And when } t = T, \quad VC = VIN - \frac{X}{Y} VIN = VIN \left(1 - \frac{X}{Y} \right)$$

$$e^{-\frac{1}{C(R0+R)} T} = \frac{X}{Y}$$

$$-\frac{1}{C(R0+R)} T = \ln \frac{X}{Y}$$

$$\text{Hence, } R0 = -\frac{T}{C \cdot \ln \frac{X}{Y}} - R$$

Figure 15.29 shows analog input pin and external sensor equivalent circuit. When the difference between VIN and VC becomes 0.1 LSB, we find impedance R0 when voltage between pins. VC changes from 0 to VIN-(0.1/1024) VIN in timer T. (0.1/1024) means that A/D precision drop due to insufficient capacitor charge is held to 0.1LSB at time of A/D conversion in the 10-bit mode. Actual error however is the value of absolute precision added to 0.1LSB. When f(XIN) = 10MHz, T=0.3µs in the A/D conversion mode with sample & hold. Output impedance R0 for sufficiently charging capacitor C within time T is determined as follows.

T = 0.3µs, R = 7.8kΩ, C = 1.5pF, X = 0.1, and Y = 1024. Hence,

$$R0 = -\frac{0.3 \times 10^{-6}}{1.5 \times 10^{-12} \cdot \ln \frac{0.1}{1024}} - 7.8 \times 10^3 \approx 13.9 \times 10^3$$

Thus, the allowable output impedance of the sensor circuit capable of thoroughly driving the A/D converter turns out of be approximately 13.9kΩ.

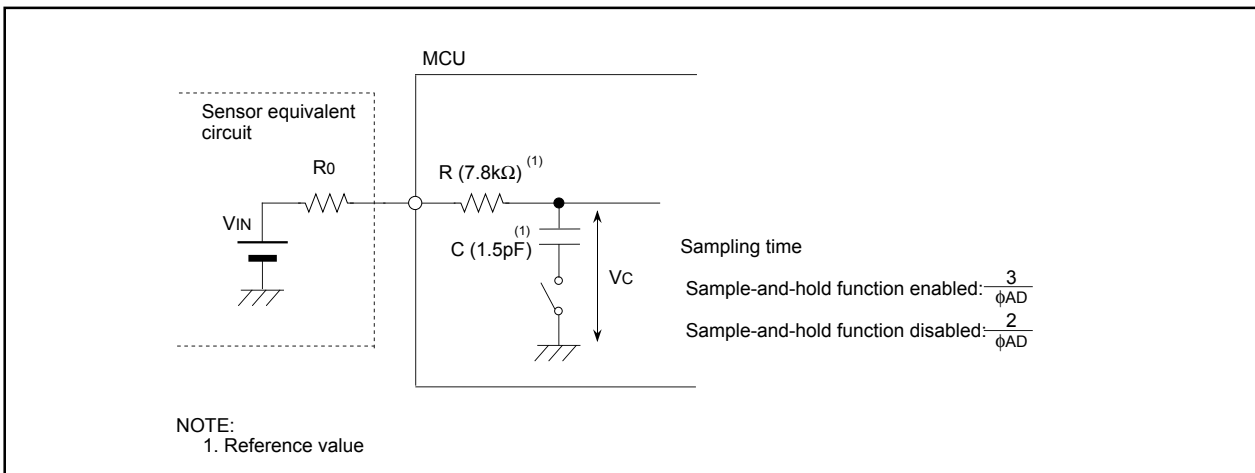


Figure 15.29 Analog Input Pin and External Sensor Equivalent Circuit

16. Multi-master I²C bus Interface

The multi-master I²C bus interface is a serial communication circuit based on Philips I²C bus data transfer format, equipped with arbitration lost detection and synchronous functions. **Figure 16.1** shows a block diagram of the multi-master I²C bus interface and **Table 16.1** lists the multi-master I²C bus interface functions.

The multi-master I²C bus interface consists of the S0D0 register, the S00 register, the S20 register, the S3D0 register, the S4D0 register, the S10 register, the S2D0 register and other control circuits.

Figures 16.2 to 16.8 show the registers associated with the multi-master I²C bus.

Table 16.1 Multi-master I²C bus interface functions

| Item | Function |
|---------------------|--|
| Format | Based on Philips I ² C bus standard: 7-bit addressing format High-speed clock mode Standard clock mode |
| Communication mode | Based on Philips I ² C bus standard: Master transmit Master receive Slave transmit Slave receive |
| SCL clock frequency | 16.1kHz to 400kHz (at V _{IIC} ⁽¹⁾ = 4MHz) |
| I/O pin | Serial data line SDAMM(SDA) Serial clock line SDLMM(SCL) |

NOTE:

1. V_{IIC}=I²C system clock

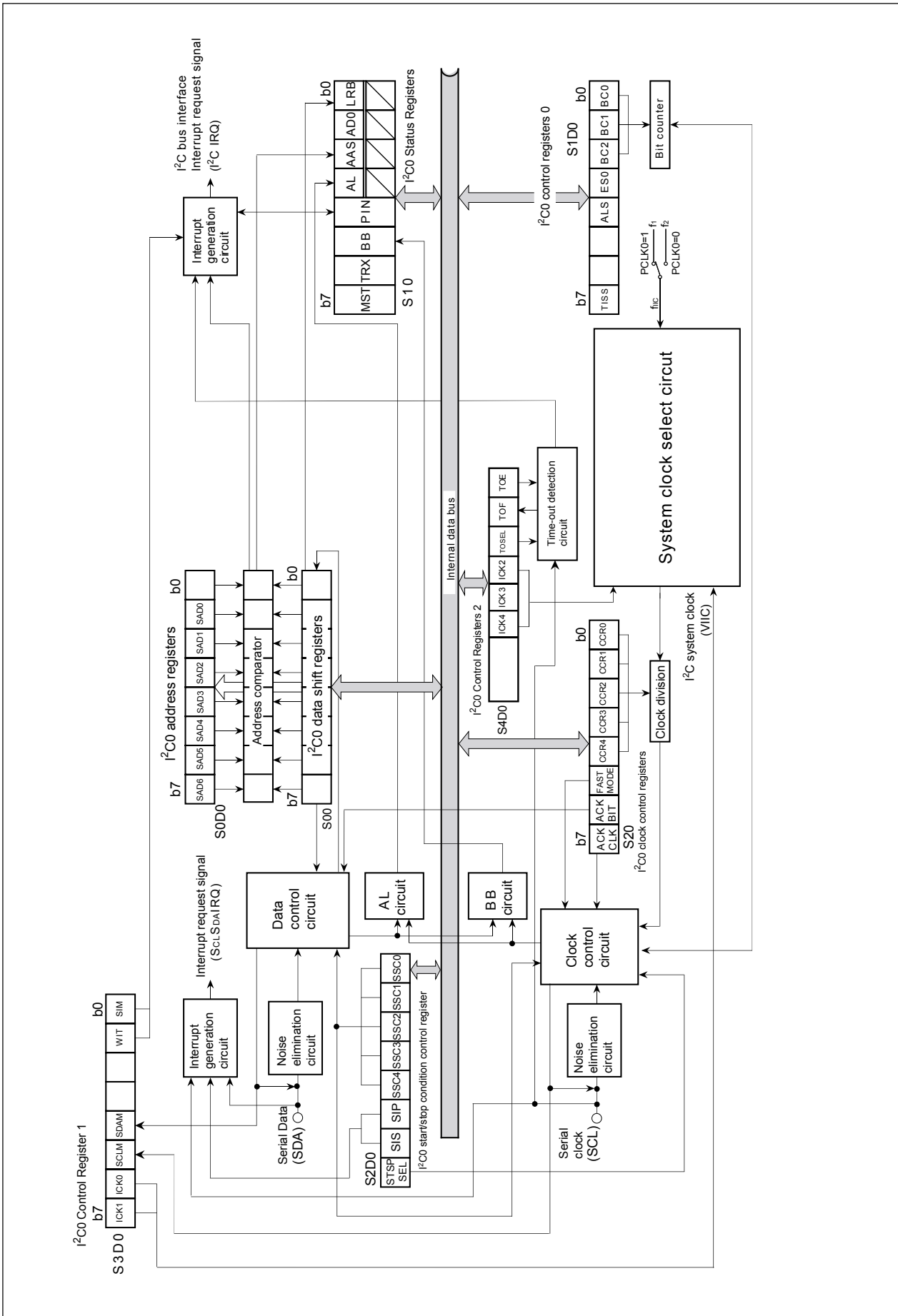


Figure 16.1 Block diagram of multi-master I²C bus interface

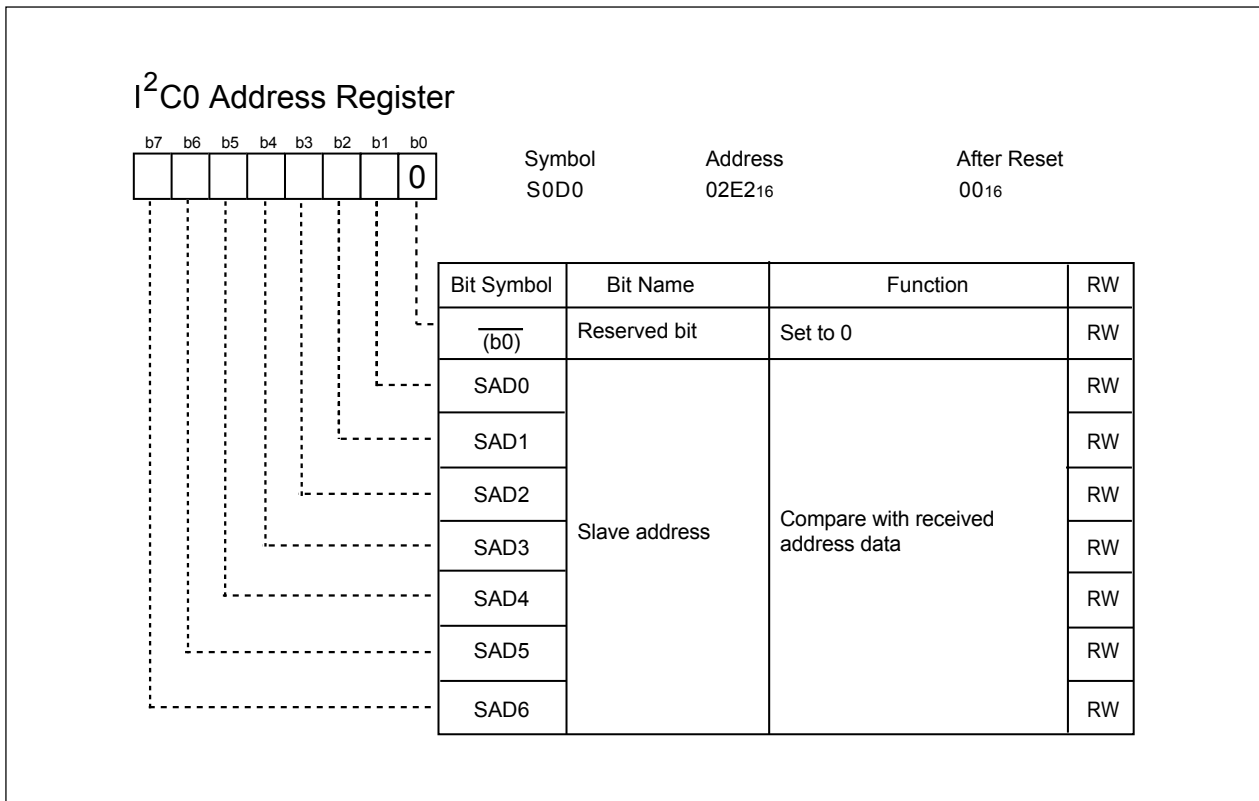


Figure 16.2 S0D0 Register

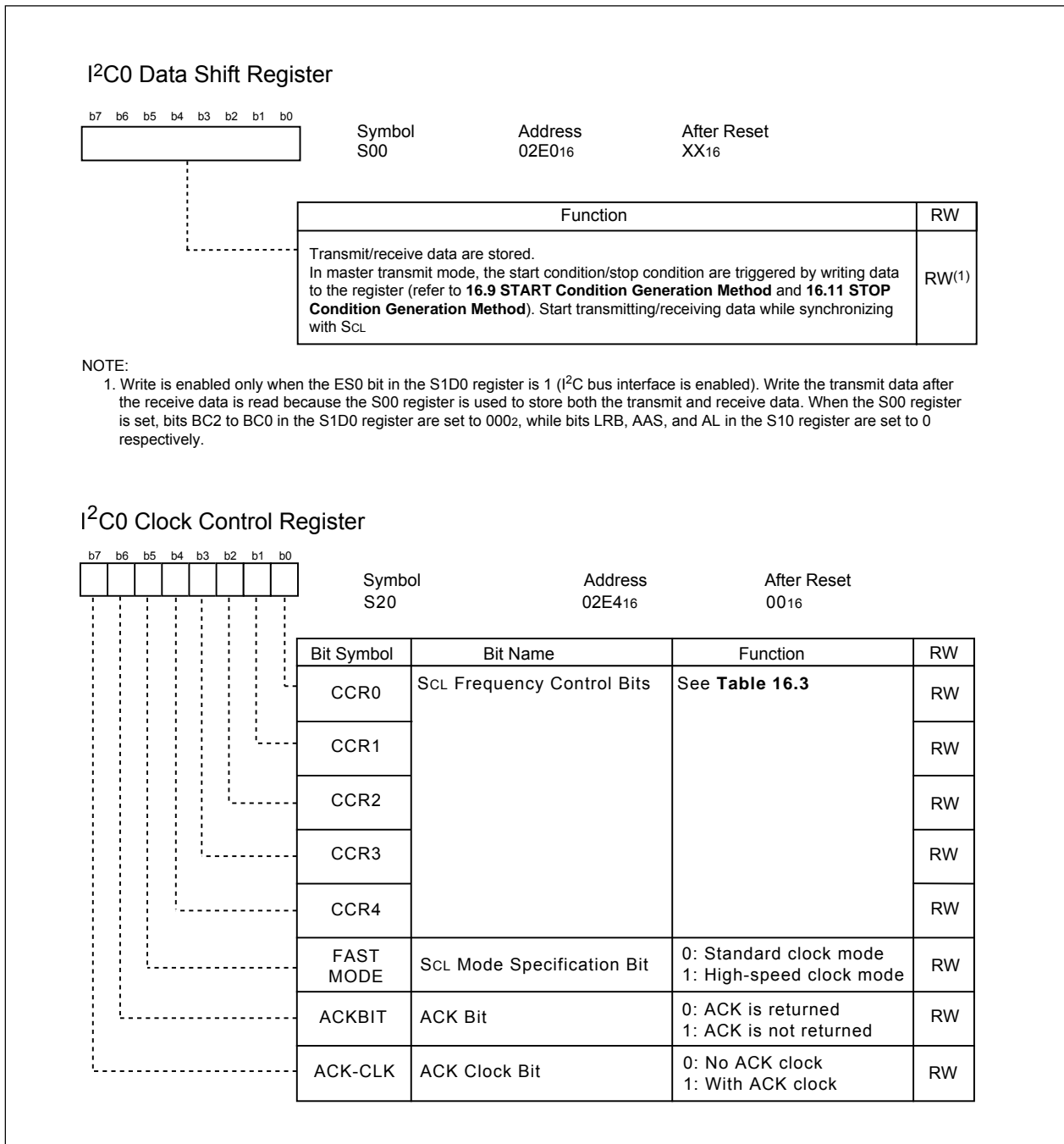


Figure 16.3 S00 and S20 Registers

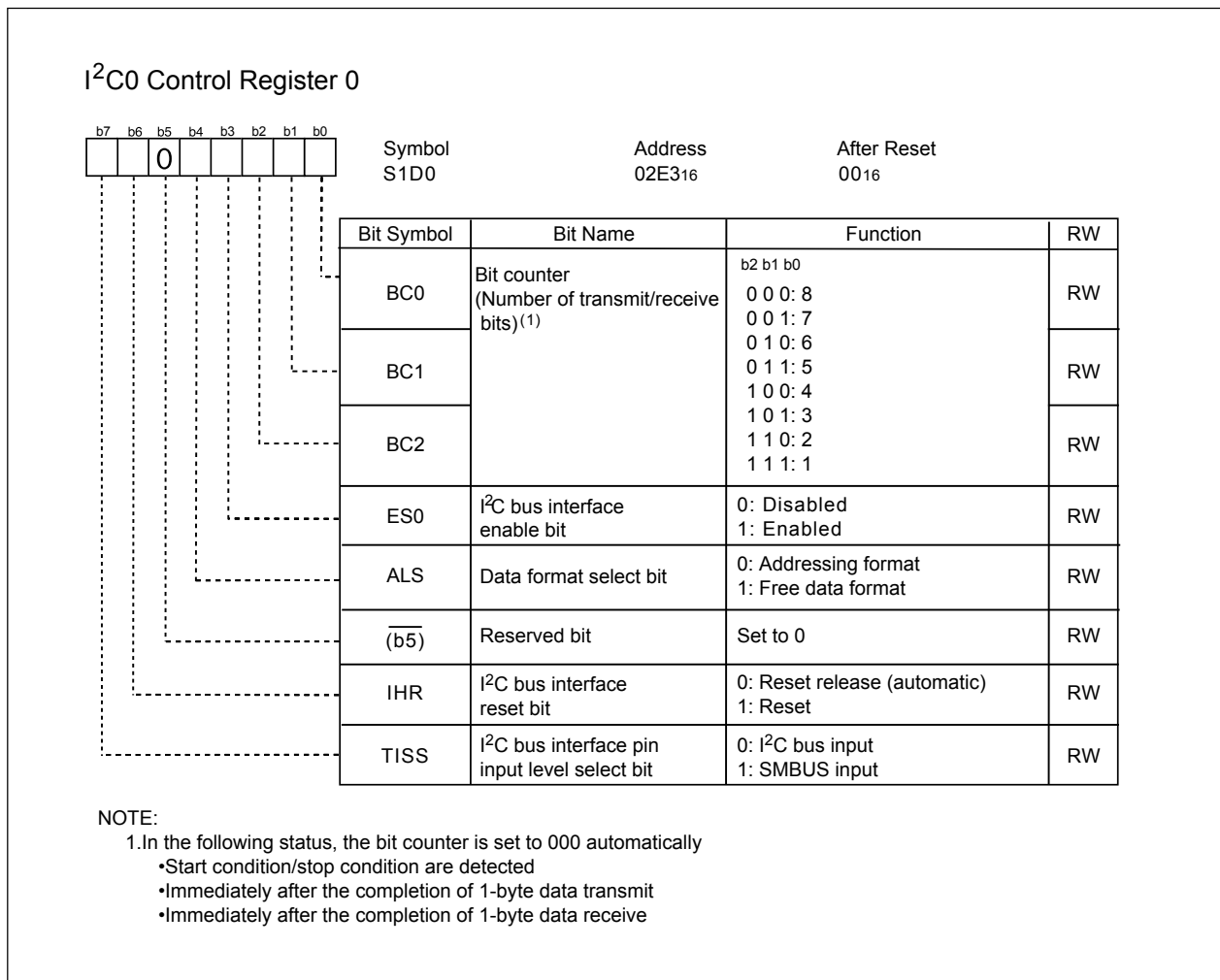


Figure 16.4 S1D0 Register

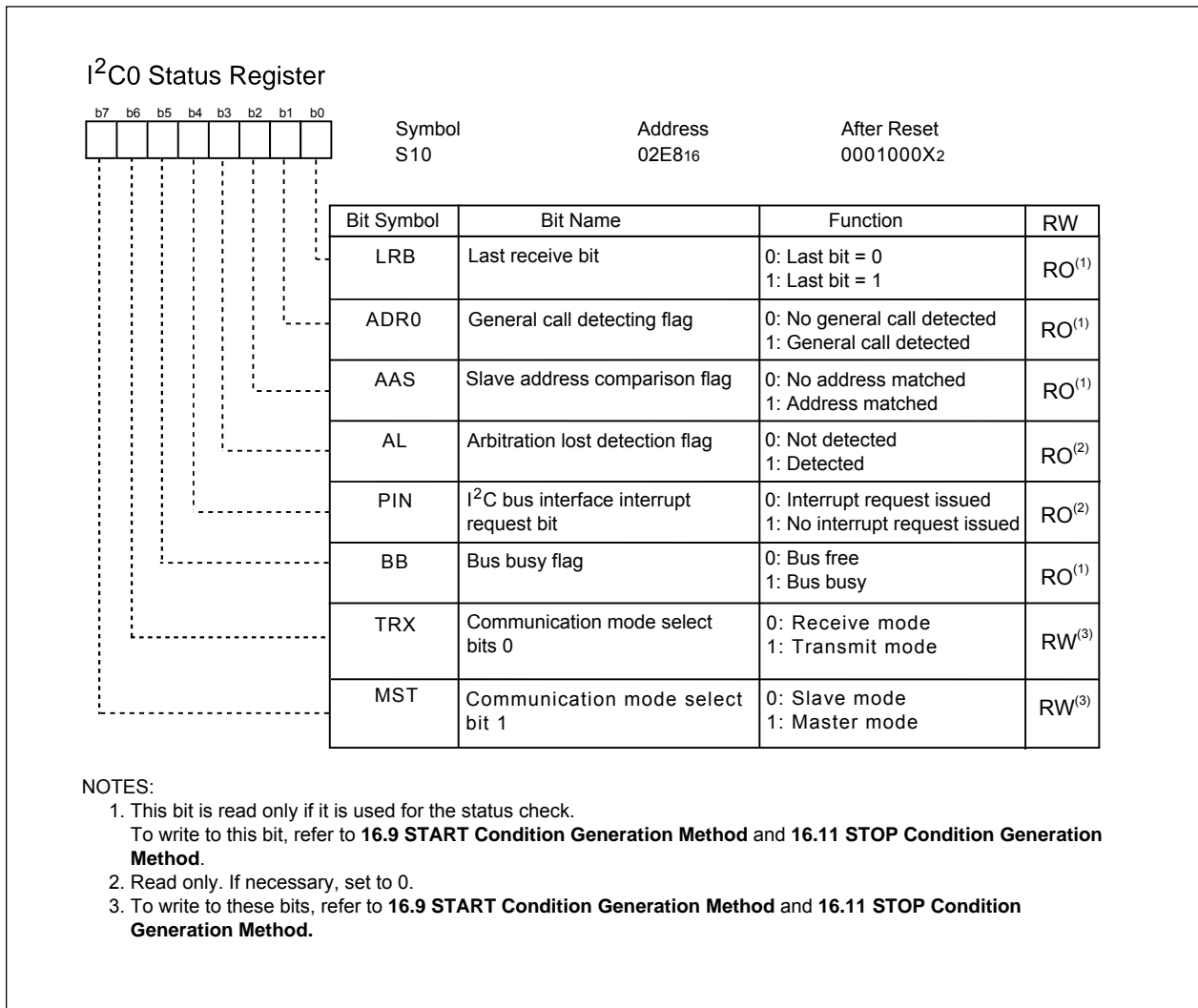


Figure 16.5 S10 Register

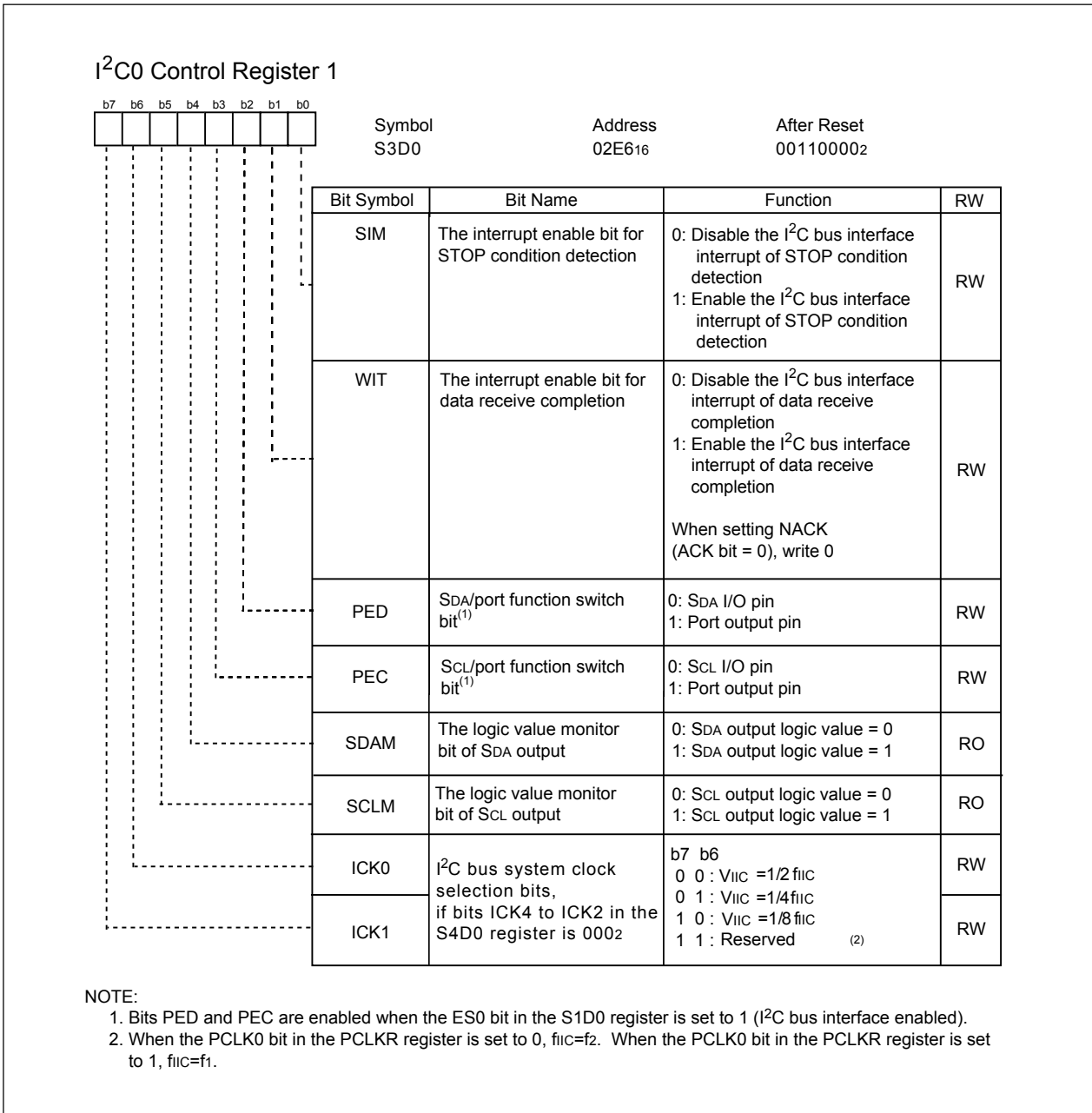


Figure 16.6 S3D0 Register

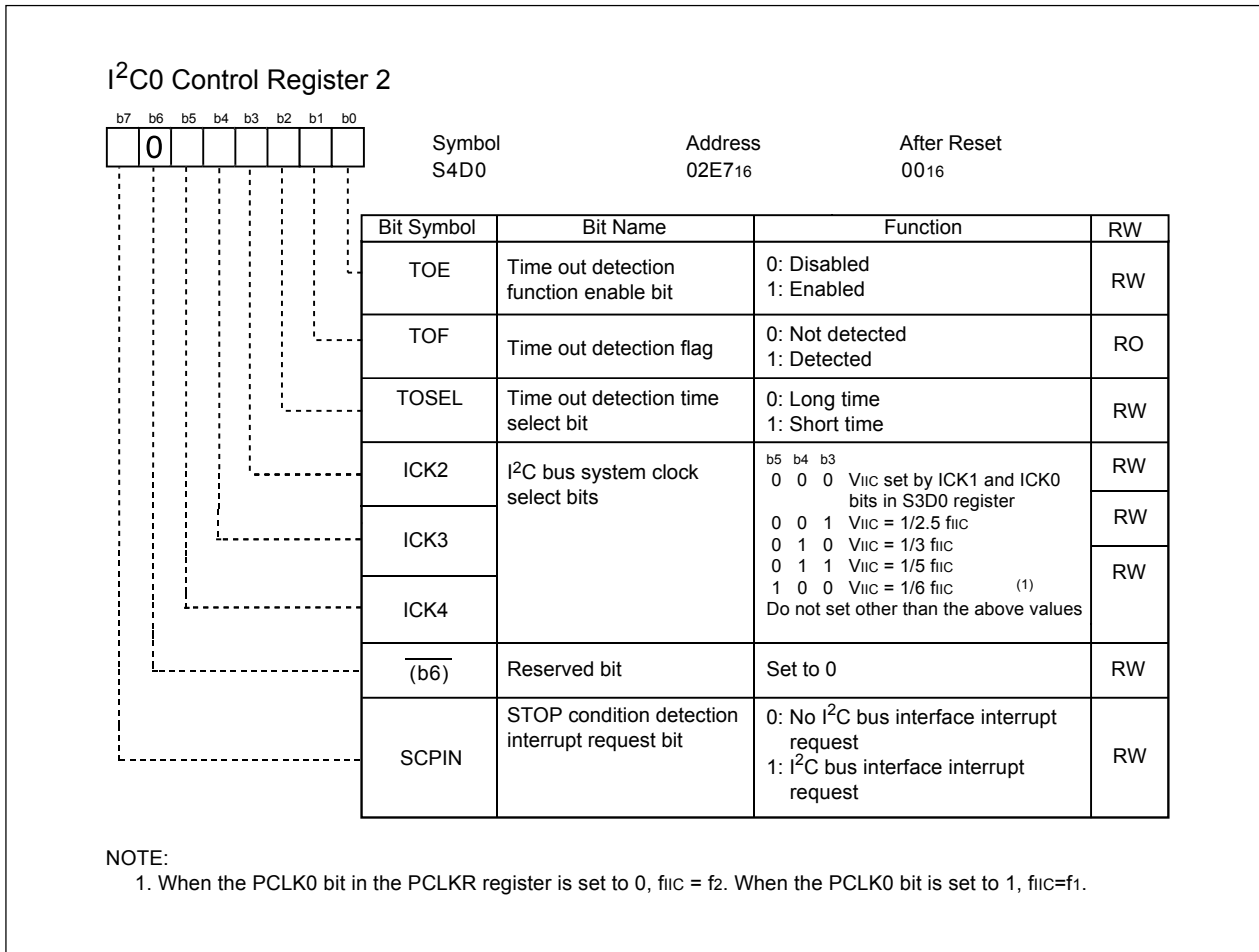


Figure 16.7 S4D0 Register

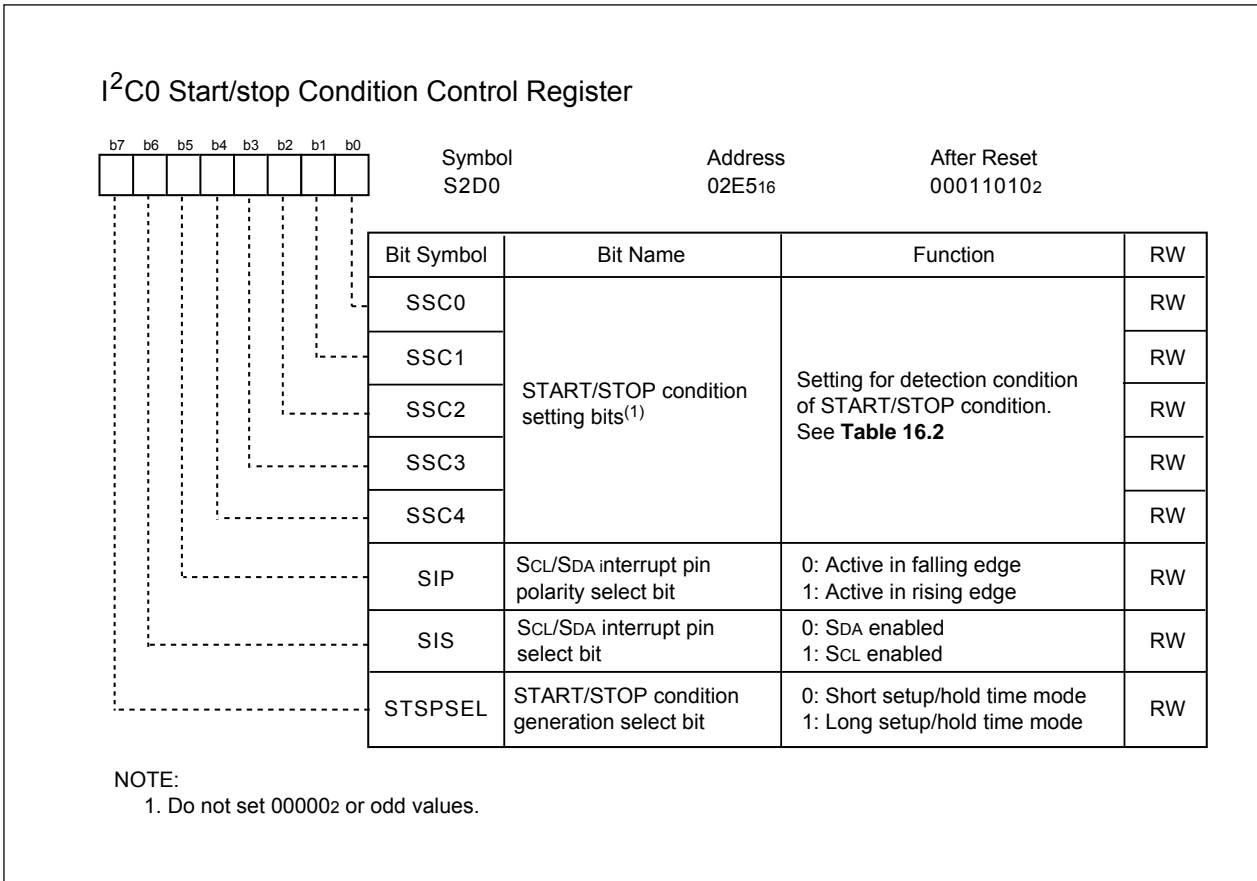


Figure 16.8 S2D0 Register

Table 16.2 Recommended setting (SSC4-SSC0) start/stop condition at each oscillation frequency

| Oscillation f ₁ (MHz) | I ² C bus system clock select | I ² C bus system clock(MHz) | SSC4-SSC0 ⁽¹⁾ | SCL release time (cycle) | Setup time (cycle) | Hold time (cycle) |
|----------------------------------|--|--|--------------------------|--------------------------|--------------------|-------------------|
| 10 | 1 / 2f ₁ ⁽²⁾ | 5 | XXX11110 | 6.2 μs (31) | 3.2 μs (16) | 3.0 μs (15) |
| 8 | 1 / 2f ₁ ⁽²⁾ | 4 | XXX11010 | 6.75 μs(27) | 3.5 μs (14) | 3.25 μs(13) |
| | | | XXX11000 | 6.25 μs(25) | 3.25 μs (13) | 3.0 μs (12) |
| 8 | 1 / 8f ₁ ⁽²⁾ | 1 | XXX00100 | 5.0 μs (5) | 3.0 μs (3) | 2.0 μs (2) |
| 4 | 1 / 2f ₁ ⁽²⁾ | 2 | XXX01100 | 6.5 μs (13) | 3.5 μs (7) | 3.0 μs (6) |
| | | | XXX01010 | 5.5 μs (11) | 3.0 μs (6) | 2.5 μs (5) |
| 2 | 1 / 2f ₁ ⁽²⁾ | 1 | XXX00100 | 5.0 μs (5) | 3.0 μs (3) | 2.0 μs (2) |

NOTES:

1. Do not set odd values or 00000₂ to START/STOP condition setting bits (SSC4 to SSC0)
2. When the PCLK0 bit in the PCLKR register is set to 1.

16.1 I²C0 Data Shift Register (S00 register)

The S00 register is an 8-bit data shift register to store a received data and to write a transmit data. When a transmit data is written to the S00 register, the transmit data is synchronized with a SCL clock and the data is transferred from bit 7. Then, every one bit of the data is transmitted, the register's content is shifted for one bit to the left. When the SCL clock and the data is imported into the S00 register from bit 0. Every one bit of the data is imported, the register's content is shifted for one bit to the left. **Figure 16.9** shows the timing to store the receive data to the S00 register.

The S00 register can be written when the ES0 bit in the S1D0 register is set to 1 (I²C0 bus interface enabled). If the S00 register is written when the ES0 bit is set to 1 and the MST bit in the S10 register is set to 1 (master mode), the bit counter is reset and the SCL clock is output. Write to the S00 register when the START condition is generated or when an "L" signal is applied to the SCL pin. The S00 register can be read anytime regardless of the ES0 bit value.

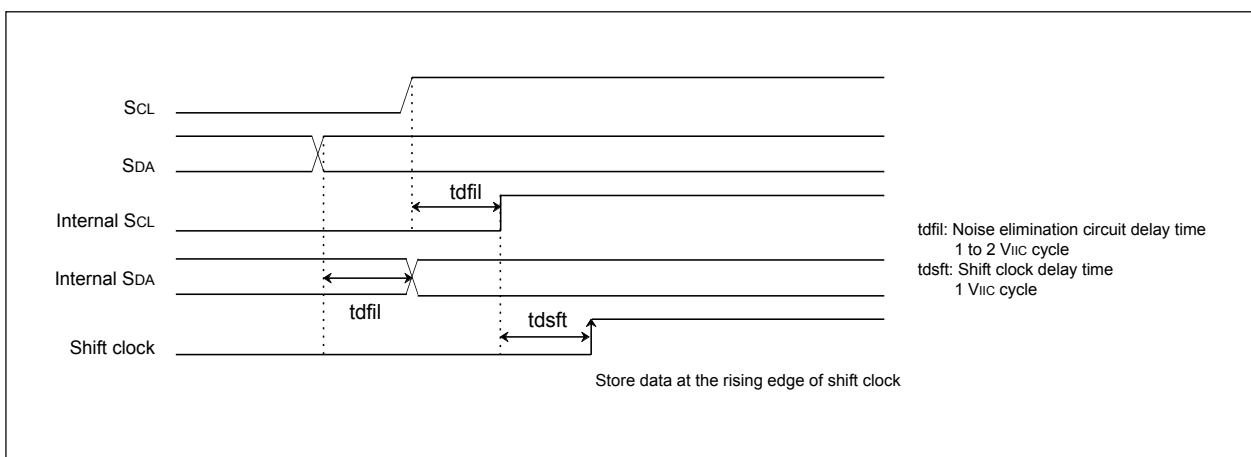


Figure 16.9 The Receive Data Storing Timing of S00 Register

16.2 I²C0 Address Register (S0D0 register)

The S0D0 register consists of bits SAD6 to SAD0, total of 7. At the addressing is formatted, slave address is detected automatically and the 7-bit received address data is compared with the contents of bits SAD6 to SAD0.

16.3 I²C0 Clock Control Register (S20 register)

The S20 register is used to set the ACK control, SCL mode and the SCL frequency.

16.3.1 Bits 0 to 4: SCL Frequency Control Bits (CCR0–CCR4)

These bits control the SCL frequency. See Table 16.3 .

16.3.2 Bit 5: SCL Mode Specification Bit (FAST MODE)

The FAST MODE bit selects SCL mode. When the FAST MODE bit is set to 0, standard clock mode is entered. When it is set to 1, high-speed clock mode is entered.

When using the high-speed clock mode I²C bus standard (400 kbit/s maximum) to connect buses, set the FAST MODE bit to 1 (select SCL mode as high-speed clock mode) and use the I²C bus system clock (V_{IIC}) at 4 MHz or more frequency.

16.3.3 Bit 6: ACK Bit (ACKBIT)

The ACKBIT bit sets the SDA status when an ACK clock⁽¹⁾ is generated. When the ACKBIT bit is set to 0, ACK is returned and the clock applied to SDA becomes "L" when ACK clock is generated. When it is set to 1, ACK is not returned and the clock applied to SDA maintains "H" at ACK clock generation.

When the ACKBIT bit is set to 0, the address data is received. When the slave address matches with the address data, SDA becomes "L" automatically (ACK is returned). When the slave address and the address data are not matched, SDA becomes "H" (ACK is not returned).

NOTE:

1. ACK clock: Clock for acknowledgment

16.3.4 Bit 7: ACK Clock Bit (ACK-CLK)

The ACK-CLK bit sets a clock for data transfer acknowledgement. When the ACK-CLK bit is set to 0, ACK clock is not generated after data is transferred. When it is set to 1, a master generates ACK clock every one-bit data transfer is completed. The device, which transmits address data and control data, leave SDA pin open (apply "H" signal to SDA) when ACK clock is generated. The device which receives data, receives the generated ACKBIT bit.

NOTE:

1. Do not rewrite the S20 register, other than the ACKBIT bit during data transfer. If data is written to other than the ACKBIT bit during transfer, the I²C bus clock circuit is reset and the data may not be transferred successfully.

Table 16.3 Setting values of S20 register and SCL frequency

| Setting value of CCR4 to CCR0 | | | | | SCL frequency (at V _{IIC} =4MHz, unit : kHz) (1) | |
|-------------------------------|------|------|------|------|---|-----------------------|
| CCR4 | CCR3 | CCR2 | CCR1 | CCR0 | Standard clock mode | High-speed clock mode |
| 0 | 0 | 0 | 0 | 0 | Setting disabled | Setting disabled |
| 0 | 0 | 0 | 0 | 1 | Setting disabled | Setting disabled |
| 0 | 0 | 0 | 1 | 0 | Setting disabled | Setting disabled |
| 0 | 0 | 0 | 1 | 1 | - (2) | 333 |
| 0 | 0 | 1 | 0 | 0 | - (2) | 250 |
| 0 | 0 | 1 | 0 | 1 | 100 | 400 (3) |
| 0 | 0 | 1 | 1 | 0 | 83.3 | 166 |
| ↓ | ↓ | ↓ | ↓ | ↓ | 500 / CCR value (3) | 1000 / CCR value (3) |
| 1 | 1 | 1 | 0 | 1 | 17.2 | 34.5 |
| 1 | 1 | 1 | 1 | 0 | 16.6 | 33.3 |
| 1 | 1 | 1 | 1 | 1 | 16.1 | 32.3 |

NOTES:

- The duty of the SCL clock output is 50 %. The duty becomes 35 to 45 % only when high-speed clock mode is selected and the CCR value = 5 (400 kHz, at V_{IIC} = 4 MHz). "H" duration of the clock fluctuates from -4 to +2 I²C system clock cycles in standard clock mode, and fluctuates from -2 to +2 I²C system clock cycles in high-speed clock mode. In the case of negative fluctuation, the frequency does not increase because the "L" is extended instead of "H" reduction. These are the values when the SCL clock synchronization by the synchronous function is not performed. The CCR value is the decimal notation value of the CCR4 to CCR0 bits.
- Each value of the SCL frequency exceeds the limit at V_{IIC} = 4 MHz or more. When using these setting values, use V_{IIC} = 4 MHz or less. Refer to **Figure 16.6**.
- The data formula of SCL frequency is described below:
 - V_{IIC}/(8 x CCR value) Standard clock mode
 - V_{IIC}/(4 x CCR value) High-speed clock mode (CCR value ≠ 5)
 - V_{IIC}/(2 x CCR value) High-speed clock mode (CCR value = 5)
 Do not set 0 to 2 as the CCR value regardless of the V_{IIC} frequency. Set 100 kHz (max.) in standard clock mode and 400 kHz (max.) in high-speed clock mode to the SCL frequency by setting the CCR4 to CCR0 bits.

16.4 I²C0 Control Register 0 (S1D0)

The S1D0 register controls data communication format.

16.4.1 Bits 0 to 2: Bit Counter (BC0–BC2)

Bits BC2 to BC0 decide how many bits are in one byte data transferred next. After the selected numbers of bits are transferred successfully, I²C bus interface interrupt request is generated and bits BC2 to BC0 are reset to 000₂. At this time, if the ACK-CLK bit in the S20 register is set to 1 (with ACK clock), one bit for ACK clock is added to the numbers of bits selected by the BC2 to BC0 bits.

In addition, bits BC2 to BC0 become 000₂ even though the START condition is detected and the address data is transferred in 8 bits.

16.4.2 Bit 3: I²C Interface Enable Bit (ES0)

The ES0 bit enables to use the multi-master I²C bus interface. When the ES0 bit is set to 0, I²C bus interface is disabled and the SDA and SCL pins are placed in a high-impedance state. When the ES0 bit is set to 1, the interface is enabled.

When the ES0 bit is set to 0, the process is followed.

- 1)The bits in the S10 register are set as MST = 0, TRX = 0, PIN = 1, BB = 0, AL = 0, AAS = 0, ADR0 = 0
- 2)The S00 register cannot be written.
- 3)The TOF bit in the S4D0 register is set to 0 (time-out detection flag is not detected)
- 4)The I²C system clock (V_{IIC}) stops counting while the internal counter and flags are reset.

16.4.3 Bit 4: Data Format Select Bit (ALS)

The ALS bit determines whether the slave address is recognized. When the ALS bit is set to 0, an addressing format is selected and an address data is recognized. Only if the comparison is matched between the slave address stored into the S0D0 register and the received address data or if the general call is received, the data is transferred. When the ALS bit is set to 1, the free data format is selected and the slave address is not recognized.

16.4.4 Bit 6: I²C bus Interface Reset Bit (IHR)

The IHR bit is used to reset the I²C bus interface circuit when the error communication occurs.

When the ES0 bit in the S1D0 register is set to 1 (I²C bus interface is enabled), the hardware is reset by writing 1 to the IHR bit. Flags are processed as follows:

- 1)The bits in the S10 register are set as MST = 0, TRX = 0, PIN to 1, BB = 0, AL = 0, AAS = 0, and ADR0 = 0
- 2)The TOF bit in the S4D0 register is set to 0 (time-out detection flag is not detected)
- 3)The internal counter and flags are reset.

The I²C bus interface circuit is reset after 2.5 V_{IIC} cycles or less, and the IHR bit becomes 0 automatically by writing 1 to the IHR bit. **Figure 16.10** shows the reset timing.

16.4.5 Bit 7: I²C bus Interface Pin Input Level Select Bit (TISS)

The TISS bit selects the input level of the SCL and SDA pins for the multi-master I²C bus interface. When the TISS bit is set to 1, the P20 and P21 become the SMBus input level.

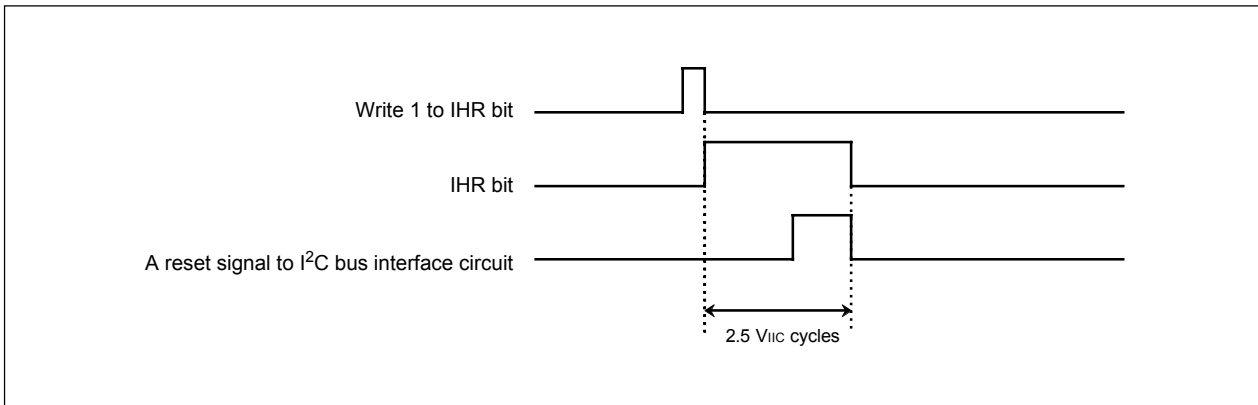


Figure 16.10 The timing of reset to the I²C bus interface circuit

16.5 I²C0 Status Register (S10 register)

The S10 register monitors the I²C bus interface status. When using the S10 register to check the status, use the 6 low-order bits for read only.

16.5.1 Bit 0: Last Receive Bit (LRB)

The LRB bit stores the last bit value of received data. It can also be used to confirm whether ACK is received. If the ACK-CLK bit in the S20 register is set to 1 (with ACK clock) and ACK is returned when the ACK clock is generated, the LRB bit is set to 0. If ACK is not returned, the LRB bit is set to 1. When the ACK-CLK bit is set to 0 (no ACK clock), the last bit value of received data is input. When writing data to the S00 register, the LRB bit is set to 0.

16.5.2 Bit 1: General Call Detection Flag (ADR0)

When the ALS bit in the S1D0 register is set to 0 (addressing format), this ADR0 flag is set to 1 by receiving the general calls⁽¹⁾, whose address data are all 0, in slave mode.

The ADR0 flag is set to 0 when STOP or START conditions is detected or when the IHR bit in the S1D0 register is set to 1 (reset).

NOTE:

1. General call: A master device transmits the general call address 00₁₆ to all slaves. When the master device transmits the general call, all slave devices receive the controlled data after general call.

16.5.3 Bit 2: Slave Address Comparison Flag (AAS)

The AAS flag indicates a comparison result of the slave address data after enabled by setting the ALS bit in the S1D0 register to 0 (addressing format).

The AAS flag is set to 1 when the 7 bits of the address data are matched with the slave address stored into the S0D0 register, or when a general call is received, in slave receive mode. The AAS flag is set to 0 by writing data to the S00 register. When the ES0 bit in the S1D0 register is set to 0 (I²C bus interface disabled) or when the IHR bit in the S1D0 register is set to 1 (reset), the AAS flag is also set to 0.

16.5.4 Bit 3: Arbitration Lost Detection Flag (AL)⁽¹⁾

In master transmit mode, if an "L" signal is applied to the SDA pin by other than the MCU, the AL flag is set to 1 by determining that the arbitration is los and the TRX bit in the S10 register is set to 0 (receive mode) at the same time. The MST bit in the S10 register is set to 0 (slave mode) after transferring the bytes which lost the arbitration.

The arbitration lost can be detected only in master transmit mode. When writing data to the S00 register, the AL flag is set to 0. When the ES0 bit in the S1D0 register is set to 0 (I²C bus interface disabled) or when the IHR bit in the S1D0 register is set to 1 (reset), the AL flag is set to 0.

NOTE:

1. Arbitration lost: communication disabled as a master

16.5.5 Bit 4: I²C bus Interface Interrupt Request Bit (PIN)

The PIN bit generates an I²C bus interface interrupt request signal. Every one byte data is transferred, the PIN bit is changed from 1 to 0. At the same time, an I²C bus interface interrupt request is generated. The PIN bit is synchronized with the last clock of the internal transfer clock (when ACK-CLK=1, the last clock is the ACK clock: when the ACK-CLK=0, the last clock is the 8th clock) and it becomes 0. The interrupt request is generated on the falling edge of the PIN bit. When the PIN bit is set to 0, the clock applied to SCL maintains "L" and further clock generation is disabled. When the ACK-CLK bit is set to 1 and the WIT bit in the S3D0 register is set to 1 (enable the I²C bus interface interrupt of data receive completion). The PIN bit is synchronized with the last clock and the falling edge of the ACK clock. Then, the PIN bit is set to 0 and I²C bus interface interrupt request is generated. **Figure 16.11** shows the timing of the I²C bus interface interrupt request generation.

The PIN bit is set to 1 in one of the following conditions:

- When data is written to the S00 register
- When data is written to the S20 register (when the WIT bit is set to 1 and the internal WAIT flag is set to 1)
- When the ES0 bit in the S1D0 register is set to 0 (I²C bus interface disabled)
- When the IHR bit in the S1D0 register is set to 1(reset)

The PIN bit is set to 0 in one of the following conditions:

- With completion of 1-byte data transmit (including a case when arbitration lost is detected)
- With completion of 1-byte data receive
- When the ALS bit in the S1D0 register is set to 0 (addressing format) and slave address is matched or general call address is received successfully in slave receive mode
- When the ALS bit is set to 1 (free format) and the address data is received successfully in slave receive mode

16.5.6 Bit 5: Bus Busy Flag (BB)

The BB flag indicates the operating conditions of the bus system. When the BB flag is set to 0, a bus system is not in use and a START condition can be generated. The BB flag is set and reset based on an input signal of the SCL and SDA pins either in master mode or in slave mode. When the START condition is detected, the BB flag is set to 1. On the other hand, when the STOP condition is detected, the BB flag is set to 0. Bits SSC4 to SSC0 in the S2D0 register decide to detect between the START condition and the STOP condition. When the ES0 bit in the S1D0 register is set to 0 (I²C bus interface disabled) or when the IHR bit in the S1D0 register is set to 1 (reset), the BB flag is set to 0. Refer to **16.9 START Condition Generation Method** and **16.11 STOP Condition Generation Method**.

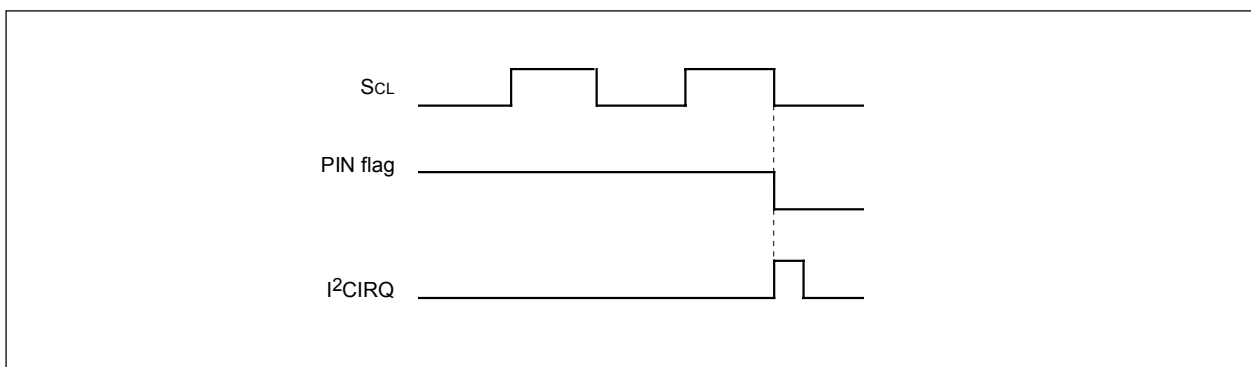


Figure 16.11 Interrupt request signal generation timing

16.5.7 Bit 6: Communication Mode Select Bit (Transfer Direction Select Bit: TRX)

This TRX bit decides a transfer direction for data communication. When the TRX bit is set to 0, receive mode is entered and data is received from a transmit device. When the TRX bit is set to 1, transmit mode is entered, and address data and control data are output to the SDAMM, synchronized with a clock generated in the SCLMM.

The TRX bit is set to 1 automatically in the following condition:

- In slave mode, when the ALS in the S1D0 register to 0(addressing format), the AAS flag is set to 1 (address match) after the address data is received, and the received R/W bit is set to 1

The TRX bit is set to 0 in one of the following conditions:

- When an arbitration lost is detected
- When a STOP condition is detected
- When a START condition is detected
- When a START condition is disabled by the START condition duplicate protect function ⁽¹⁾
- When the MST bit in the S10 register is set to 0(slave mode) and a start condition is detected
- When the MST bit is set to 0 and the ACK non-return is detected
- When the ES0 bit is set to 0(I²C bus interface disabled)
- When the IHR bit in the S1D0 register is set to 1(reset)

16.5.8 Bit 7: Communication mode select bit (master/slave select bit: MST)

The MST bit selects either master mode or slave mode for data communication. When the MST bit is set to 0, slave mode is entered and the START/STOP condition generated by a master device are received. The data communication is synchronized with the clock generated by the master. When the MST bit is set to 1, master mode is entered and the START/STOP condition is generated.

Additionally, clocks required for the data communication are generated on the SCLMM.

The MST bit is set to 0 in one of the following conditions.

- After 1-byte data of a master whose arbitration is lost if arbitration lost is detected
- When a STOP condition is detected
- When a START condition is detected
- When a start condition is disabled by the START condition duplicate protect function ⁽¹⁾
- When the IHR bit in the S1D0 register is set to 1(reset)
- When the ES0 bit is set to 0(I²C bus interface disabled)

NOTE:

1. START condition duplicate protect function:

When the START condition is generated, after confirming that the BB flag in the S1D0 register is set to 0 (bus free), all the MST, TRX and BB flags are set to 1 at the same time. However, if the BB flag is set to 1 immediately after the BB flag setting is confirmed because a START condition is generated by other master device, bits MST and TRX cannot be written. The duplicate protect function is valid from the rising edge of the BB flag until slave address is received. Refer to **16.9 START Condition Generation Method** for details.

16.6 I²C0 Control Register 1 (S3D0 register)

The S3D0 register controls the I²C bus interface circuit.

16.6.1 Bit 0 : Interrupt Enable Bit by STOP Condition (SIM)

The SIM bit enables the I²C bus interface interrupt request by detecting a STOP condition. If the SIM bit is set to 1, the I²C bus interface interrupt request is generated by the STOP condition detect (no need to change in the PIN flag).

16.6.2 Bit 1: Interrupt Enable Bit at the Completion of Data Receive (WIT)

If the WIT bit is set to 1 while the ACK-CLK bit in the S20 register is set to 1 (ACK clock), the I²C bus interface interrupt request is generated and the PIN bit is set to 1 at the falling edge of the last data bit clock. Then an "L" signal is applied to the SCLMM and the ACK clock generation is controlled. **Table 16.4** and **Figure 16.12** show the interrupt generation timing and the procedure of communication restart. After the communication is restarted, the PIN bit is set to 0 again, synchronized with the falling edge of the ACK clock, and the I²C bus interface interrupt request is generated.

Table 16.4 Timing of Interrupt Generation in Data Receive Mode

| I ² C bus Interface Interrupt Generation Timing | Procedure of Communication Restart |
|--|---|
| 1) Synchronized with the falling edge of the last data bit clock | Set the ACK bit in the S20 register. Set the PIN bit to 1. (Do not write to the S00 register. The ACK clock operation may be unstable.) |
| 2) Synchronized with the falling edge of the ACK clock | Set the S00 register |

The internal WAIT flag can be read by reading the WIT bit. The internal WAIT flag is set to 1 after writing data to the S00 register and it is set to 0 after writing to the S20 register.

Consequently, the I²C bus interface interrupt request generated by the timing 1) or 2) can be determined. (See **Figure 16.12**)

When the data is transmitted and the address data is received immediately after the START condition, the WAIT flag remains 0 regardless of the WIT bit setting, and the I²C bus interface interrupt request is only generated at the falling edge of the ACK clock. Set the WIT bit to 0 when the ACK-CLK bit in the S20 register is set to 0 (no ACK clock).

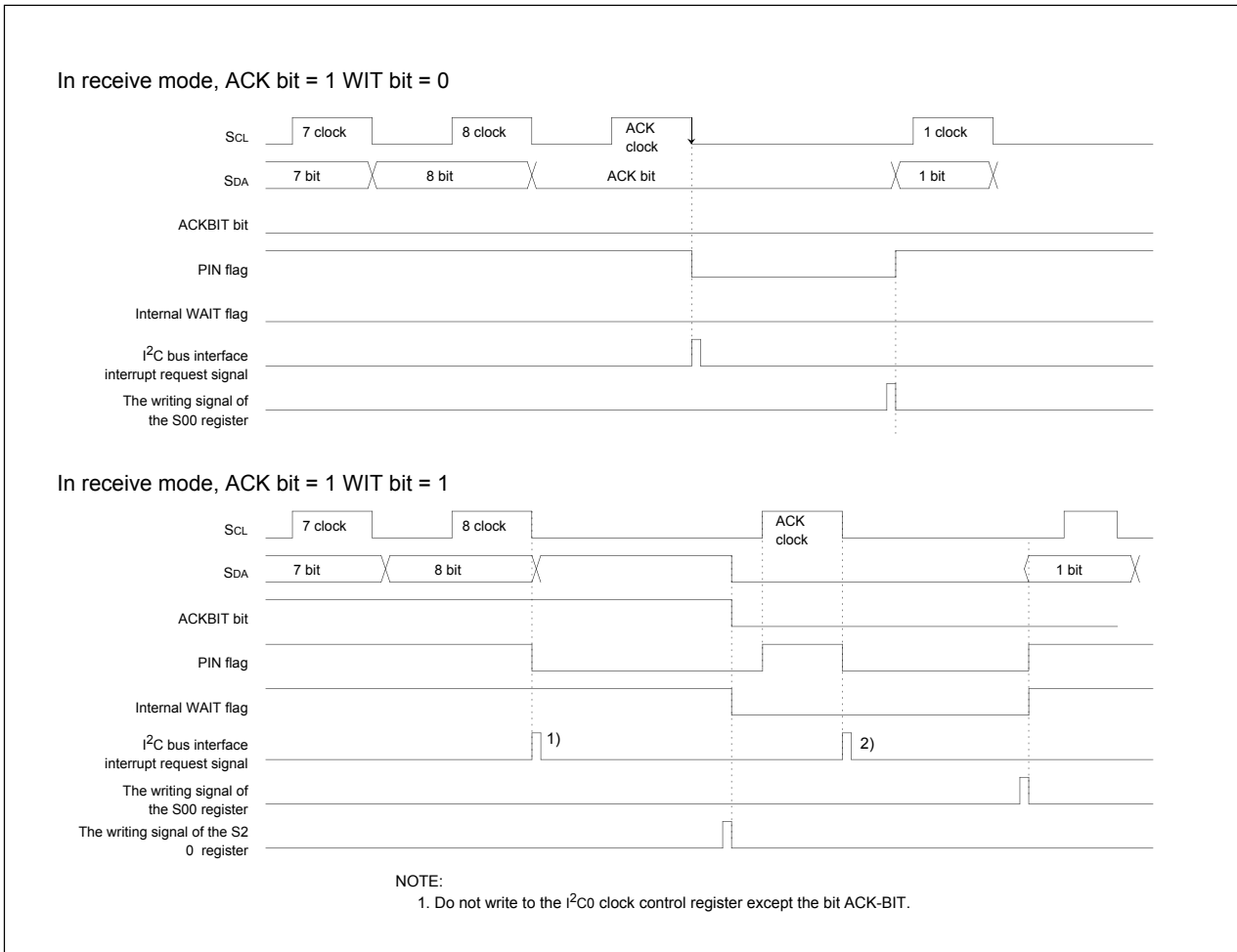


Figure 16.12 The timing of the interrupt generation at the completion of the data receive

16.6.3 Bits 2,3 : Port Function Select Bits PED, PEC

If the ES0 bit in the S1D0 register is set to 1 (I²C bus interface enabled), the SDAMM functions as an output port. When the PED bit is set to 1 and the SCLMM functions as an output port when the PEC bit is set to 1. Then the setting values of bits P2_0 and P2_1 in the port P2 register are output to the I²C bus, regardless of the internal SCL/SDA output signals. (SCL/SDA pins are connected to I²C bus interface circuit)

The bus data can be read by reading the port pi direction register in input mode, regardless of the setting values of the PED and PEC bits. **Table 16.5** shows the port specification.

Table 16.5 Port specifications

| Pin Name | ES9 Bit | PED Bit | P20 Port Direction Register | Function |
|----------|---------|---------|-----------------------------|--|
| P20 | 0 | - | 0/1 | Port I/O function |
| | 1 | 0 | - | SDA I/O function |
| | 1 | 1 | - | SDA input function, port output function |
| Pin Name | ES0 Bit | PEC Bit | P21 Port Direction Register | Function |
| P21 | 0 | - | 0/1 | Port I/O function |
| | 1 | 0 | - | SCL I/O function |
| | 1 | 1 | - | SCL input function, port output function |

16.6.4 Bits 4,5 : SDA/SCL Logic Output Value Monitor Bits SDAM/SCLM

Bits SDAM/SCLM can monitor the logic value of the SDA and SCL output signals from the I²C bus interface circuit. The SDAM bit monitors the SDA output logic value. The SCLM bit monitors the SCL output logic value. The SDAM and SCLM bits are read-only. If necessary, set them to 0.

16.6.5 Bits 6,7 : I²C System Clock Select Bits ICK0, ICK1

The ICK1 bit, ICK0 bit, bits ICK4 to ICK2 in the S4D0 register, and the PCLK0 bit in the PCLKR register can select the system clock (V_{IIC}) of the I²C bus interface circuit.

The I²C bus system clock V_{IIC} can be selected among 1/2 f_{IIC}, 1/2.5 f_{IIC}, 1/3 f_{IIC}, 1/4 f_{IIC}, 1/5 f_{IIC}, 1/6 f_{IIC} and 1/8 f_{IIC}. f_{IIC} can be selected between f₁ and f₂ by the PCLK0 bit setting.

Table 16.6 I²C system clock select bits

| I3CK4[S4D0] | ICK3[S4D0] | ICK2[S4D0] | ICK1[S3D0] | ICK0[S3D0] | I ² C system clock |
|-------------|------------|------------|------------|------------|---|
| 0 | 0 | 0 | 0 | 0 | V _{IIC} = 1/2 f _{IIC} |
| 0 | 0 | 0 | 0 | 1 | V _{IIC} = 1/4 f _{IIC} |
| 0 | 0 | 0 | 1 | 0 | V _{IIC} = 1/8 f _{IIC} |
| 0 | 0 | 1 | X | X | V _{IIC} = 1/2.5 f _{IIC} |
| 0 | 1 | 0 | X | X | V _{IIC} = 1/3 f _{IIC} |
| 0 | 1 | 1 | X | X | V _{IIC} = 1/5 f _{IIC} |
| 1 | 0 | 0 | X | X | V _{IIC} = 1/6 f _{IIC} |

(Do not set the combination other than the above)

16.6.6 Address Receive in STOP/WAIT Mode

When WAIT mode is entered after the CM02 bit in the CM0 register is set to 0 (do not stop the peripheral function clock in wait mode), the I²C bus interface circuit can receive address data in WAIT mode. However, the I²C bus interface circuit is not operated in STOP mode or in low power consumption mode, because the I²C bus system clock V_{IIC} is not supplied.

16.7 I²C0 Control Register 2 (S4D0 Register)

The S4D0 register controls the error communication detection.

If the SCL clock is stopped counting during data transfer, each device is stopped, staying online. To avoid the situation, the I²C bus interface circuit has a function to detect the time-out when the SCL clock is stopped in high-level ("H") state for a specific period, and to generate an I²C bus interface interrupt request. See **Figure 16.13**.

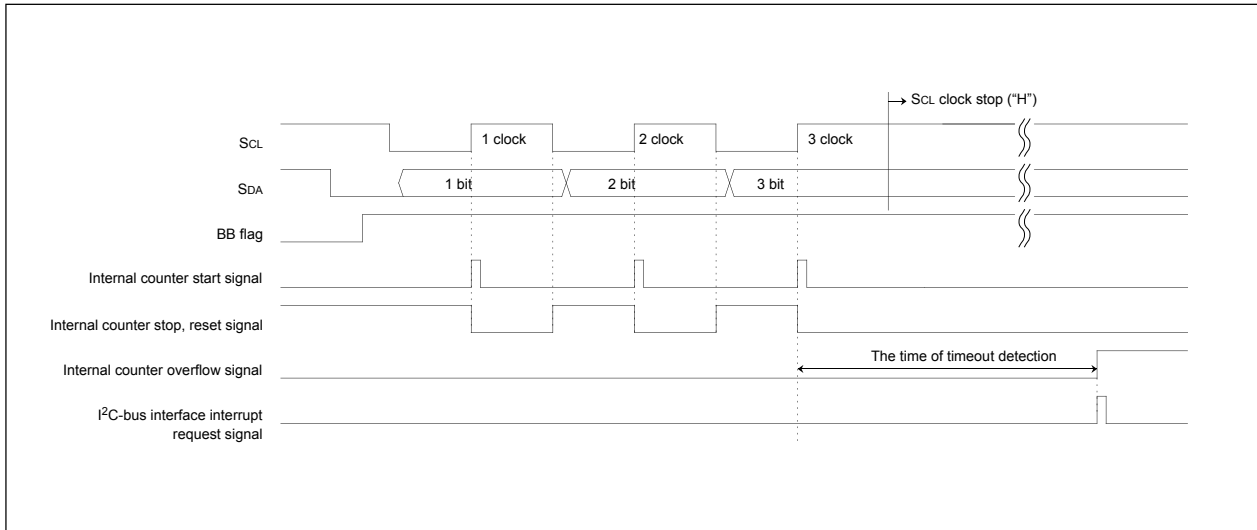


Figure 16.13 The timing of time-out detection

16.7.1 Bit0: Time-Out Detection Function Enable Bit (TOE)

The TOE bit enables the time-out detection function. When the TOE bit is set to 1, time-out is detected and the I²C bus interface interrupt request is generated when the following conditions are met.

- 1) the BB flag in the S10 register is set to 1 (bus busy)
- 2) the SCL clock stops for time-out detection period while high-level ("H") signal is maintained (see **Table 16.7**)

The internal counter measures the time-out detection time and the TOSEL bit selects between two modes, long time and short time. When time-out is detected, set the ES0 bit to 0 (I²C bus interface disabled) and reset the counter.

16.7.2 Bit1: Time-Out Detection Flag (TOF)

The TOF flag indicates the time-out detection. If the internal counter which measures the time-out period overflows, the TOF flag is set to 1 and the I²C bus interface interrupt request is generated at the same time.

16.7.3 Bit2: Time-Out Detection Period Select Bit (TOSEL)

The TOSEL bit selects time-out detection period from long time mode and short time mode. When the TOSEL bit is set to 0, long time mode is selected. When it is set to 1, short time mode is selected, respectively. The internal counter increments as a 16-bit counter in long time mode, while the counter increments as a 14-bit counter in short time mode, based on the I²C system clock (V_{IIC}) as a counter source. **Table 16.7** shows examples of time-out detection period.

Table 16.7 Examples of Time-out Detection Period (Unit: ms)

| V _{IIC} (MHz) | Long time mode | Short time mode |
|------------------------|----------------|-----------------|
| 4 | 16.4 | 4.1 |
| 2 | 32.8 | 8.2 |
| 1 | 65.6 | 16.4 |

16.7.4 Bits 3,4,5: I²C System Clock Select Bits (ICK2-4)

Bits ICK4 to ICK2, and bits ICK1 and ICK0 in the S3D0 register, and the PCLK0 bit in the PCLKR register select the system clock (V_{IIC}) of the I²C bus interface circuit. See **Table 16.6** for the setting values.

16.7.5 Bit7: STOP Condition Detection Interrupt Request Bit (SCPIN)

The SCPIN bit monitors the stop condition detection interrupt. The SCPIN bit is set to 1 when the I²C bus interface interrupt is generated by detecting the STOP condition. When this bit is set to 0 by program, it becomes 0. However, no change occurs even if it is set to 1.

16.8 I²C0 START/STOP Condition Control Register (S2D0 Register)

The S2D0 register controls the START/STOP condition detections.

16.8.1 Bit0-Bit4: START/STOP Condition Setting Bits (SSC0-SSC4)

The SCL release time and the set-up and hold times are measured on the base of the I²C bus system clock (V_{IIC}). Therefore, the detection conditions changes, depending on the oscillation frequency (X_{IN}) and the I²C bus system clock select bits. It is necessary to set bits SSC4 to SSC0 to the appropriate value to set the SCL release time, the set-up and hold times by the system clock frequency (See **Table 16.10**). Do not set odd numbers or 000002 to bits SSC4 to SSC0. **Table 16.2** shows the reference value to bits SSC4 to SSC0 at each oscillation frequency in standard clock mode. The detection of START/STOP conditions starts immediately after the ES0 bit in the S1D0 register is set to 1 (I²C bus interface enabled).

16.8.2 Bit5: SCL/SDA Interrupt Pin Polarity Select Bit (SIP)

The The SIP bit detect the rising edge or the falling edge of the SCLMM or SDAMM to generate SCL/SDA interrupts. The SIP bit selects the polarity of the SCLMM or the SDAMM for interrupt.

16.8.3 Bit6 : SCL/SDA Interrupt Pin Select Bit (SIS)

The SIS bit selects a pin to enable SCL/SDA interrupt.

NOTE:

1. The SCL/SDA interrupt request may be set when changing the SIP, SIS and ES0 bit settings in the S1D0 register. When using the SCL/SDA interrupt, set the above bits, while the SCL/SDA interrupt is disabled. Then, enable the SCL/SDA interrupt after setting the SCL/SDA bit in the IR register to 0.

16.8.4 Bit7: START/STOP Condition Generation Select Bit (STSPSEL)

The STSPSEL bit selects the set-up/hold times, based on the I²C system clock cycles, when the START/STOP condition is generated (See **Table 16.8**). Set the STSPSEL bit to 1 if the I²C bus system clock frequency is over 4MHz.

16.9 START Condition Generation Method

Set the MST bit, TRX bit and BB flags in the S10 register to 1 and set the PIN bit and 4 low-order bits in the S10 register to 0 simultaneously, to enter START condition standby mode, when the ES0 bit in the S1D0 register is set to 1 (I²C bus interface enabled) and the BB flag is set to 0 (bus free). When the slave address is written to the S00 register next, START condition is generated and the bit counter is reset to 0002 and 1-byte SCL signal is output. The START condition generation timing varies between standard clock mode and high-speed clock mode. See **Figure 16.16** and **Table 16.8**.

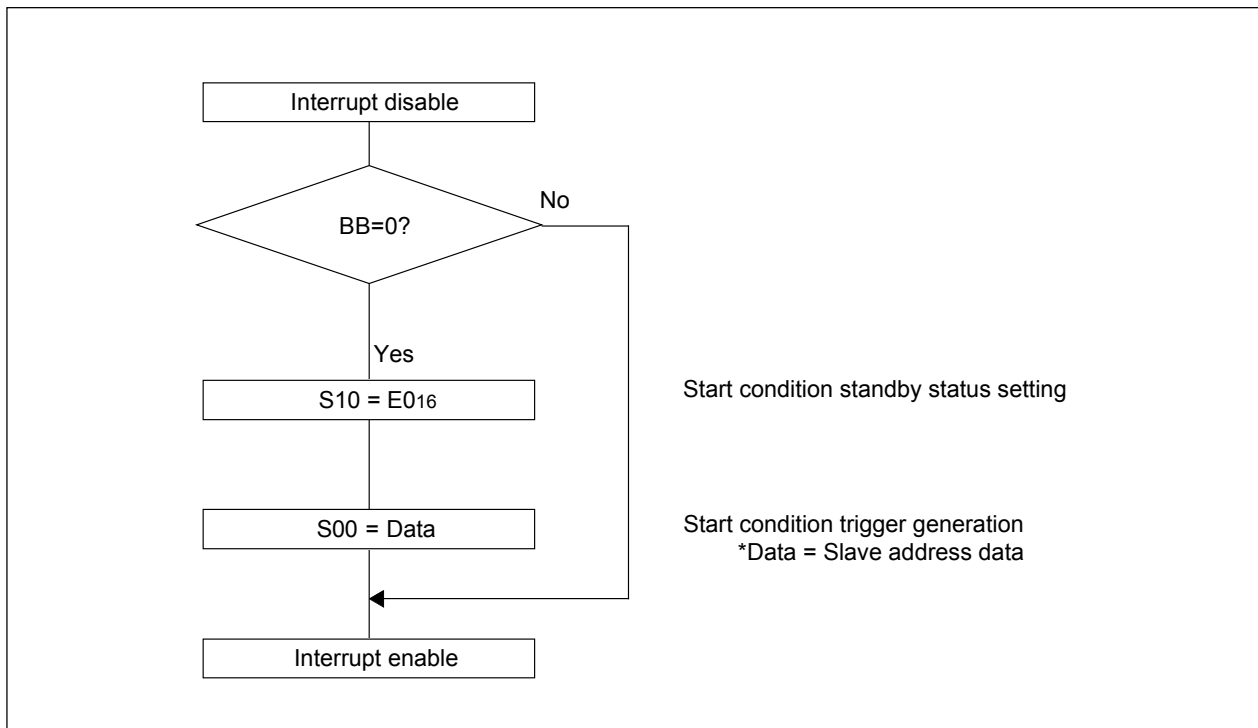


Figure 16.14 Start condition generation flow chart

16.10 START Condition Duplicate Protect Function

A START condition is generated when verifying that the BB flag in the S10 register does not use buses. However, if the BB flag is set to 1 (bus busy) by the START condition which other master device generates immediately after the BB flag is verified, the START condition is suspended by the START condition duplicate protect function. When the START condition duplicate protect function starts, it operates as follows:

- Disable the start condition standby setting

If the function has already been set, first exit START condition standby mode and then set bits MST and TRX in the S10 register to 0.

- Writing to the S00 register is disabled. (The START condition trigger generation is disabled)
- If the START condition generation is interrupted, the AL flag in the S10 register becomes 1.(arbitration lost detection)

The START condition duplicate protect function is valid between the SDA falling edge of the START condition and the receive completion of the slave address. **Figure 16.15** shows the duration of the START condition duplicate protect function.

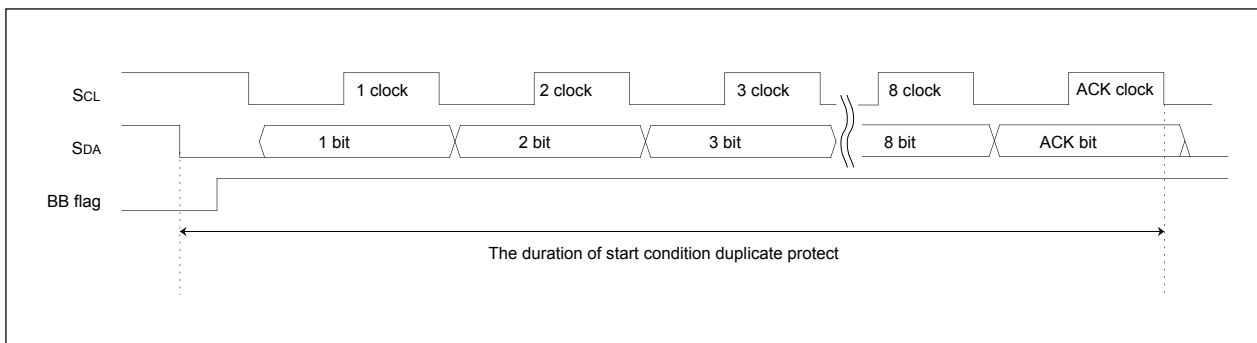


Figure 16.15 The duration of the start condition duplicate protect function

16.11 STOP Condition Generation Method

When the ES0 bit in the S1D0 register is set to 1 (I²C bus interface enabled) and bits MST and TRX in the S10 register are set to 1 at the same time, set the BB flag, PIN bit and 4 low-order bits in the S10 register to 0 simultaneously, to enter STOP condition standby mode. When dummy data is written to the S00 register next, the STOP condition is generated. The STOP condition generation timing varies between standard clock mode and high-speed clock mode. See **Figure 16.17** and **Table 16.8**.

Until the BB flag in the S10 register becomes 0 (bus free) after an instruction to generate the STOP condition is executed, do not write data to registers S10 and S00. Otherwise, the STOP condition waveform may not be generated correctly.

If an input signal level of the SCL pin is set to low ("L") after the instruction to generate the STOP condition is executed, a signal level of the SCL pin becomes high ("H"), and the BB flag is set to 0 (bus free), the MCU outputs an "L" signal to SCL pin.

In that case, the MCU can stop an "L" signal output to the SCL pin by generating the STOP condition, writing 0 to the ES0 bit in the S1D0 register (disabled), or writing 1 to the IHR bit in the S1D0 register (reset release).

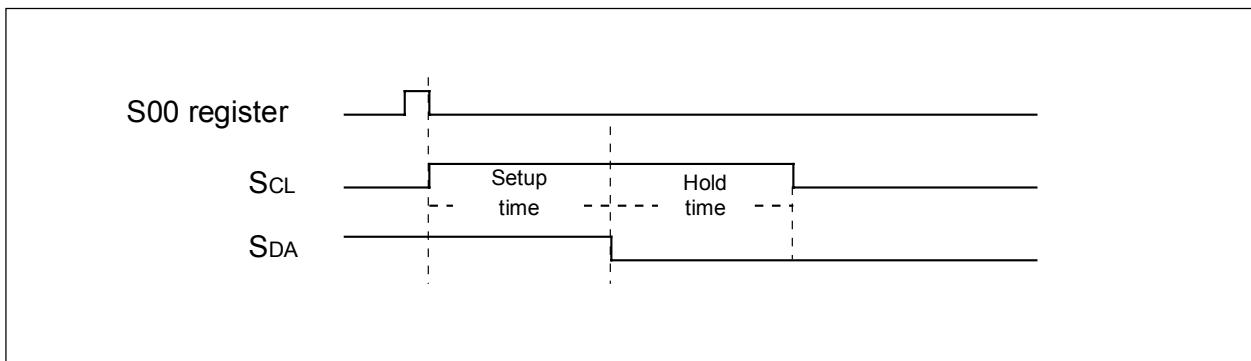


Figure 16.16 Start condition generation timing diagram

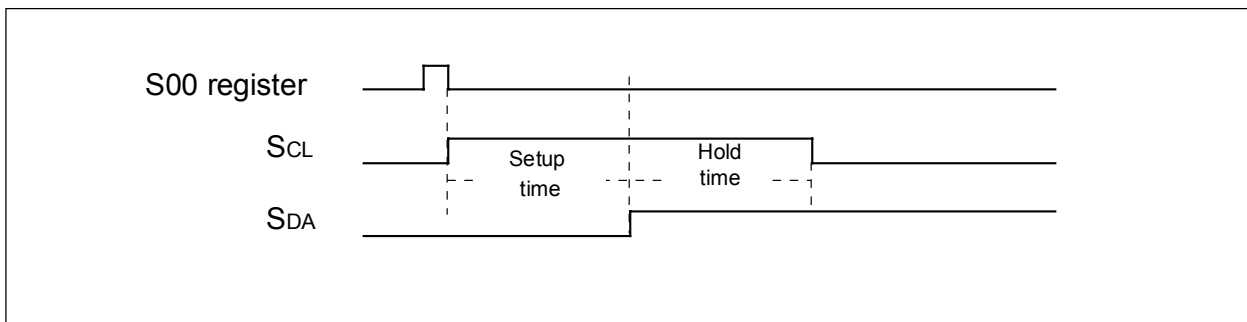


Figure 16.17 Stop condition generation timing diagram

Table 16.8 Start/Stop generation timing table

| | Start/Stop Condition Generation Select Bit | Standard Clock Mode | High-speed Clock Mode |
|------------|--|---------------------|-----------------------|
| Setup time | 0 | 5.0 μs (20 cycles) | 2.5 μs (10 cycles) |
| | 1 | 13.0 μs (52 cycles) | 6.5 μs (26 cycles) |
| Hold time | 0 | 5.0 μs (20 cycles) | 2.5 μs (10 cycles) |
| | 1 | 13.0 μs (52 cycles) | 6.5 μs (26 cycles) |

NOTE:

- Actual time at the time of V_{IIC} = 4MHz, The contents in () denote cycle numbers.

As mentioned above, when bits MST and TRX are set to 1, START condition or STOP condition mode is entered by writing 1 or 0 to the BB flag in the S10 register and writing 0 to the PIN bit and 4 low-order bits in the S10 register at the same time. Then SDAMM is left open in the START condition standby mode and SDAMM is set to low-level ("L") in the STOP condition standby mode. When the S00 register is set, the START/STOP conditions are generated. In order to set bits MST and TRX to 1 without generating the START/STOP conditions, write 1 to the 4 low-order bits simultaneously. **Table 16.9** lists functions along with the S10 register settings.

Table 16.9 S10 Register Settings and Functions

| S10 Register Settings | | | | | | | | Function |
|-----------------------|-----|----|-----|----|-----|-----|-----|---|
| MST | TRX | BB | PIN | AL | AAS | AS0 | LRB | |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | Setting up the START condition stand by in master transmit mode |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | Setting up the STOP condition stand by in master transmit mode |
| 0/1 | 0/1 | - | 0 | 1 | 1 | 1 | 1 | Setting up each communication mode (refer to 16.5 I²C status register) |

16.12 START/STOP Condition Detect Operation

Figure 16.18, Figure 16.19 and Table 16.10 show START/STOP condition detect operations. Bits SSC4 to SSC0 in the S2D0 register set the START/STOP conditions. The START/STOP condition can be detected only when the input signal of the SCLMM and SDAMM met the following conditions: the SCL release time, the set-up time, and the hold time (see Table 16.10). The BB flag in the S10 register is set to 1 when the START condition is detected and it is set to 0 when the STOP condition is detected. The BB flag set and reset timing varies between standard clock mode and high-speed clock mode. See Table 16.10.

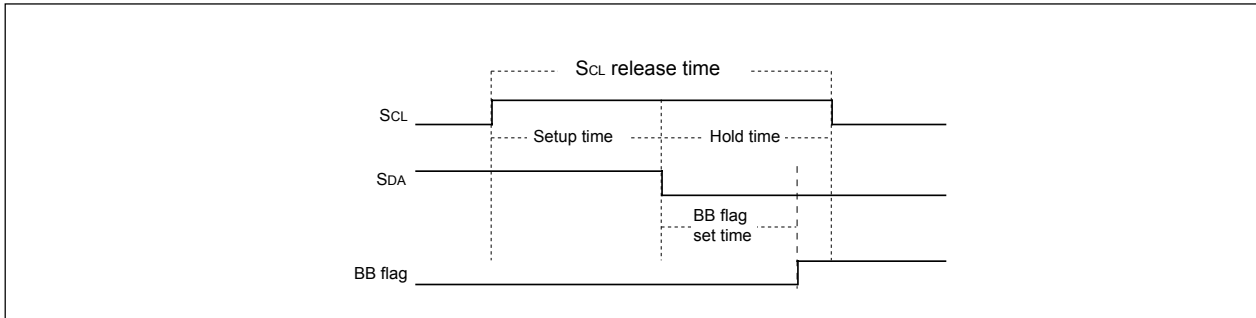


Figure 16.18 Start condition detection timing diagram

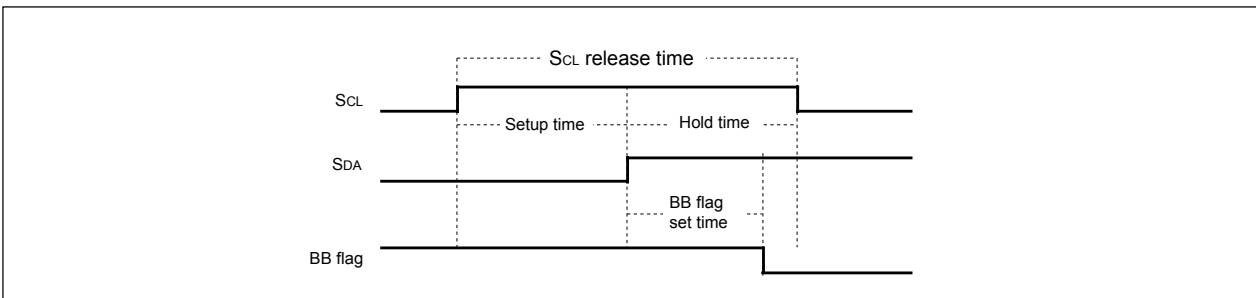


Figure 16.19 Stop condition detection timing diagram

Table 16.10 Start/Stop detection timing table

| | Standard clock mode | High-speed clock mode |
|------------------------|--|-----------------------|
| SCL release time | SSC value + 1 cycle (6.25μs) | 4 cycles (1.0μs) |
| Setup time | $\frac{\text{SSC value}}{2} + 1 \text{ cycle} < 4.0\mu\text{s}$ (3.25μs) | 2 cycles (0.5μs) |
| Hold time | $\frac{\text{SSC value}}{2} \text{ cycle} < 4.0\mu\text{s}$ (3.0μs) | 2 cycles (0.5μs) |
| BB flag set/reset time | $\frac{\text{SSC value} - 1}{2} + 2 \text{ cycles}$ (3.375μs) | 3.5 cycles (0.875μs) |

NOTE:

1. Unit : number of cycle for I²C system clock V_{IC}

The SSC value is the decimal notation value of bits SSC4 to SSC0. Do not set 0 or odd numbers to the SSC setting. The values in () are examples when the S2D0 register is set to 1816 at V_{IC} = 4 MHz.

16.13 Address Data Communication

This section describes data transmit control when a master transfers data or a slave receives data in 7-bit address format. **Figure 16.20 (1)** shows a master transmit format.

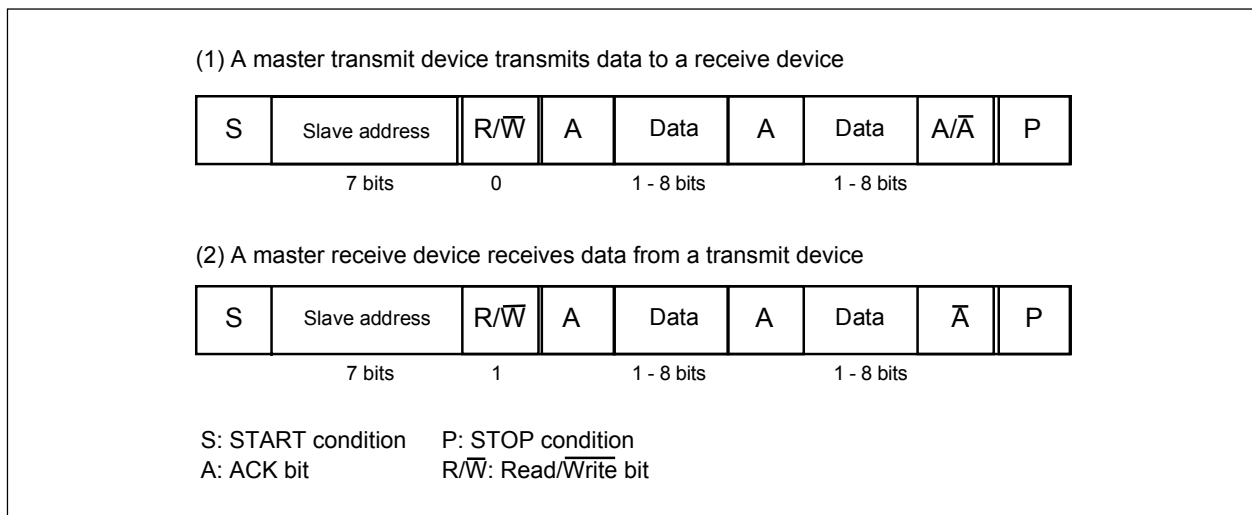


Figure 16.20 Address data communication format

16.13.1 Example of Master Transmit

For example, a master transmits data as shown below when following conditions are met: standard clock mode, SCL clock frequency of 100kHz and ACK clock added.

- 1) Set s slave address to the 7 high-order bits in the S0D0 register
- 2) Set 85₁₆ to the S20 register, 000₂ to bits ICK4 to ICK2 in the S4D0 register and 00₁₆ to the S3D0 register to generate an ACK clock and set SCL clock frequency t 100 kHz (f₁=8MHz, f_{IC}=f₁)
- 3) Set 00₁₆ to the S10 register to reset transmit/receive
- 4) Set 08₁₆ to the S1D0 register to enable data communication
- 5) Confirm whether the bus is free by BB flag setting in the S10 register
- 6) Set E0₁₆ to the S10 register to enter START condition standby mode
- 7) Set the destination address in 7 high-order bits and 0 to a least significant bit in the S00 register to generate START condition. At this time, the first byte consisting of SCL and ACK clock are automatically generated
- 8) Set a transmit data to the S00 register. At this time, SCL and an ACK clock are automatically generated
- 9) When transmitting more than 1-byte control data, repeat the above step 8).
- 10) Set C0₁₆ in the S10 register to enter STOP condition standby mode if ACK is not returned from the slave receiver or if the transmit is completed
- 11) Write dummy data to the S00 register to generate STOP condition

16.13.2 Example of Slave Receive

For example, a slave receives data as shown below when following conditions are met: high-speed clock mode, SCL frequency of 400 kHz, ACK clock added and addressing format.

- 1) Set a slave address in the 7 high-order bits in the S0D0 register
- 2) Set A5₁₆ to the S20 register, 000₂ to bits ICK4 to ICK2 in the S4D0 register, and 00₁₆ to the S3D0 register to generate an ACK clock and set SCL clock frequency at 400kHz ($f_1 = 8 \text{ MHz}$, $f_{iic} = f_1$)
- 3) Set 00₁₆ to the S10 register to reset transmit/receive mode
- 4) Set 08₁₆ to the S1D0 register to enable data communication
- 5) When a START condition is received, addresses are compared
- 6) •When the transmitted addresses are all 0 (general call), the ADR0 bit in the S10 register is set to 1 and an I²C bus interface interrupt request signal is generated.
•When the transmitted addresses match with the address set in 1), the ASS bit in the S10 register is set to 1 and an I²C bus interface interrupt request signal is generated.
•In other cases, bits ADR0 and AAS are set to 0 and I²C bus interface interrupt request signal is not generated.
- 7) Write dummy data to the S00 register.
- 8) After receiving 1-byte data, an ACK-CLK bit is automatically returned and an I²C bus interface interrupt request signal is generated.
- 9) To determine whether the ACK should be returned depending on contents in the received data, set dummy data to the S00 register to receive data after setting the WIT bit in the S3D0 register to 1 (enable the I²C bus interface interrupt of data receive completion). Because the I²C bus interface interrupt is generated when the 1-byte data is received, set the ACKBIT bit to 1 or 0 to output a signal from the ACKBIT bit.
- 10) When receiving more than 1-byte control data, repeat steps 7) and 8) or 7) and 9).
- 11) When a STOP condition is detected, the communication is ended.

16.14 Precautions

(1) Access to the registers of I²C bus interface circuit

The following is precautions when read or write the control registers of I²C bus interface circuit

- S00 register

Do not rewrite the S00 register during data transfer. If the bits in the S00 register are rewritten, the bit counter for transfer is reset and data may not be transferred successfully.

- S1D0 register

Bits BC2 o BC0 are set to 0002 when START condition is detected or when 1-byte data transfer is completed. Do not read or write the S1D0 register at this timing. Otherwise, data may be read or written unsuccessfully. **Figure 16.22** and **Figure 16.23** show the bit counter reset timing.

- S20 register

Do not rewrite the S20 register except the ACKBIT bit during transfer. If the bits in the S20 register except ACKBIT bit are rewritten, the I²C bus clock circuit is reset and data may be transferred incompletely.

- S3D0 register

Rewrite bits ICK4 to ICK0 in the S3D0 register when the ES0 bit in the S1D0 register is set to 0 (I²C bus interface is disabled). When the WIT bit is read, the internal WAIT flag is read. Therefore, do not use the bit managing instruction(read-modify-write instruction) to access the S3D0 register.

- S10 register

Do not use the bit managing instruction (read-modify-write instruction) because all bits in the S10 register will be changed, depending on the communication conditions. Do not read/write when te communication mode select bits, bits MST and TRX, are changing their value. Otherwise, data may be read or written unsuccessfully. **Figure16.21** to **Figure 16.23** show the timing when bits MST and TRX change.

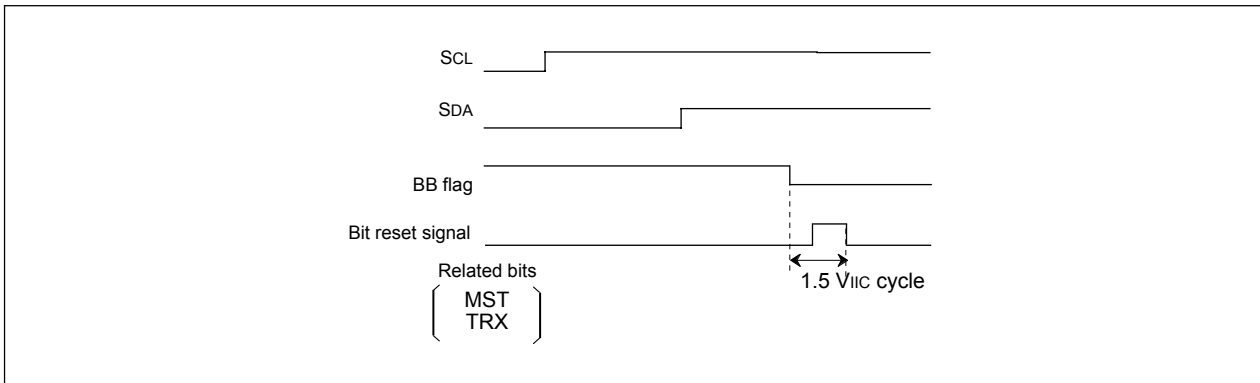


Figure 16.21 The bit reset timing (The STOP condition detection)

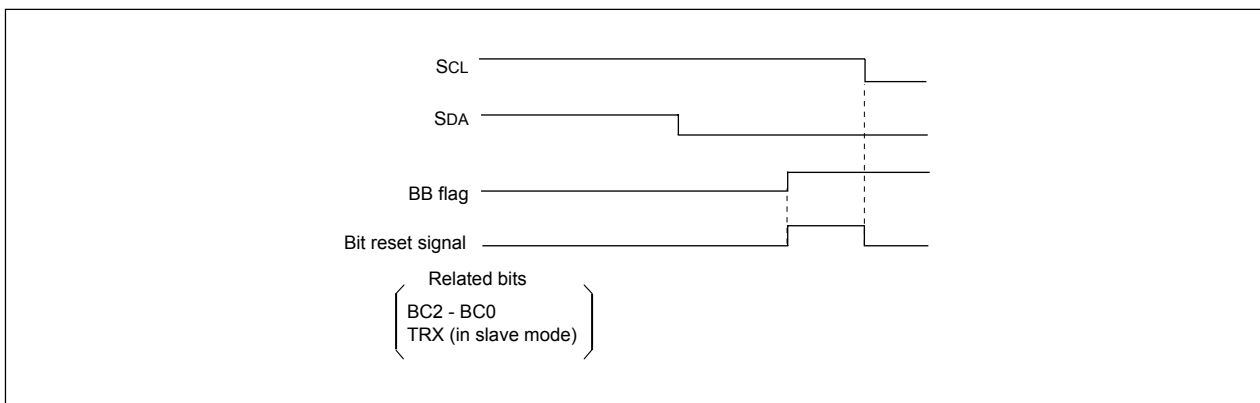


Figure 16.22 The bit reset timing (The START condition detection)

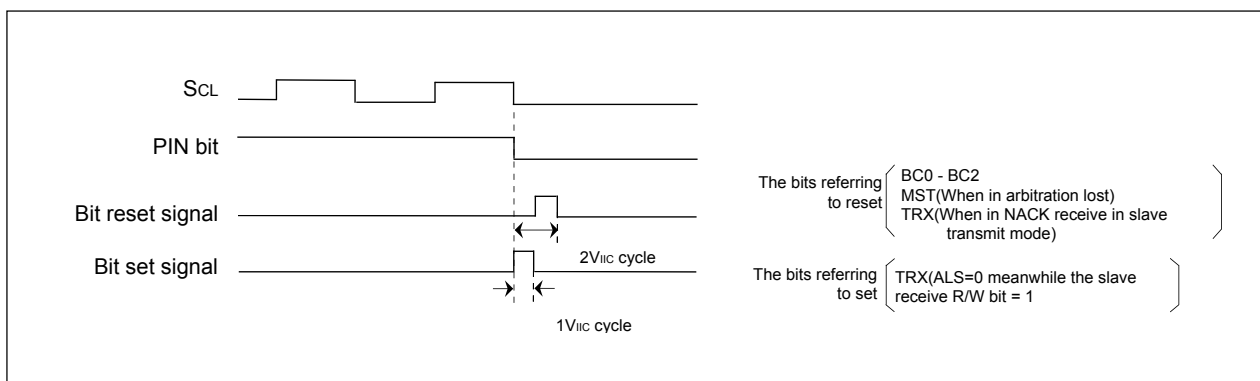


Figure 16.23 Bit set/reset timing (at the completion of data transfer)

(2) Generation of RESTART condition

In order to generate a RESTART condition after 1-byte data transfer, write E016 to the S10 register, enter START condition standby mode and leave the SDAMM open. Generate a START condition trigger by setting the S00 register after inserting a sufficient software wait until the SDAMM outputs a high-level ("H") signal. **Figure 16.24** shows the RESTART condition generation timing.

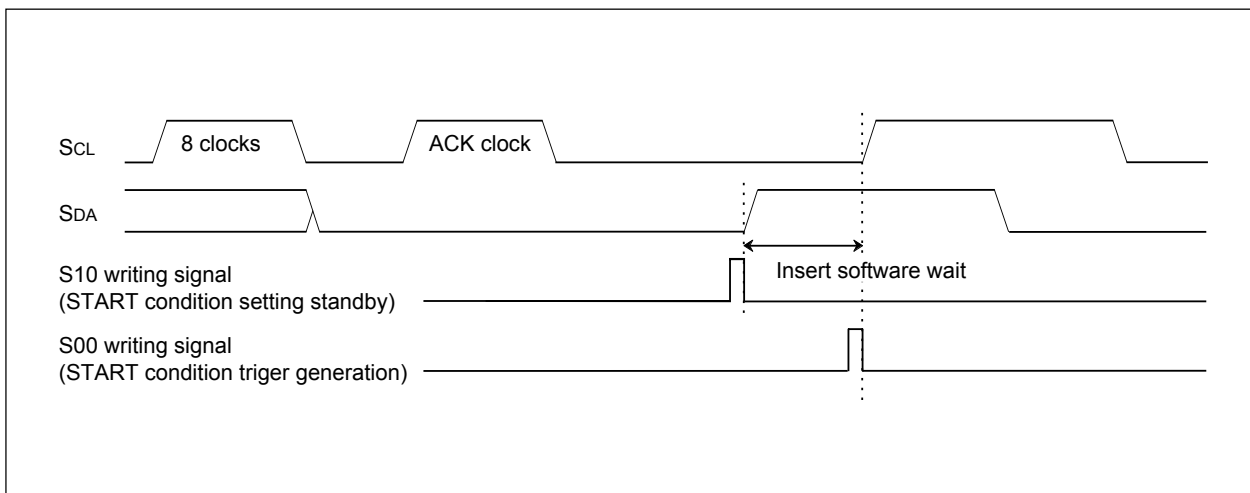


Figure 16.24 The time of generation of RESTART condition

(3) limitation of CPU clock

When the CM07 bit in the CM0 register is set to 1 (subclock), each register of the I²C bus interface circuit cannot be read or written. Read or write data when the CM07 bit is set to 0 (main clock, PLL clock, or on-chip oscillator clock).

17. CAN Module

The CAN (Controller Area Network) module for the M16C/29 Group of MCUs is a communication controller implementing the CAN 2.0B protocol. The M16C/29 Group contains one CAN module which can transmit and receive messages in both standard (11-bit) ID and extended (29-bit) ID formats.

Figure 17.1 shows a block diagram of the CAN module.

External CAN bus driver and receiver are required.

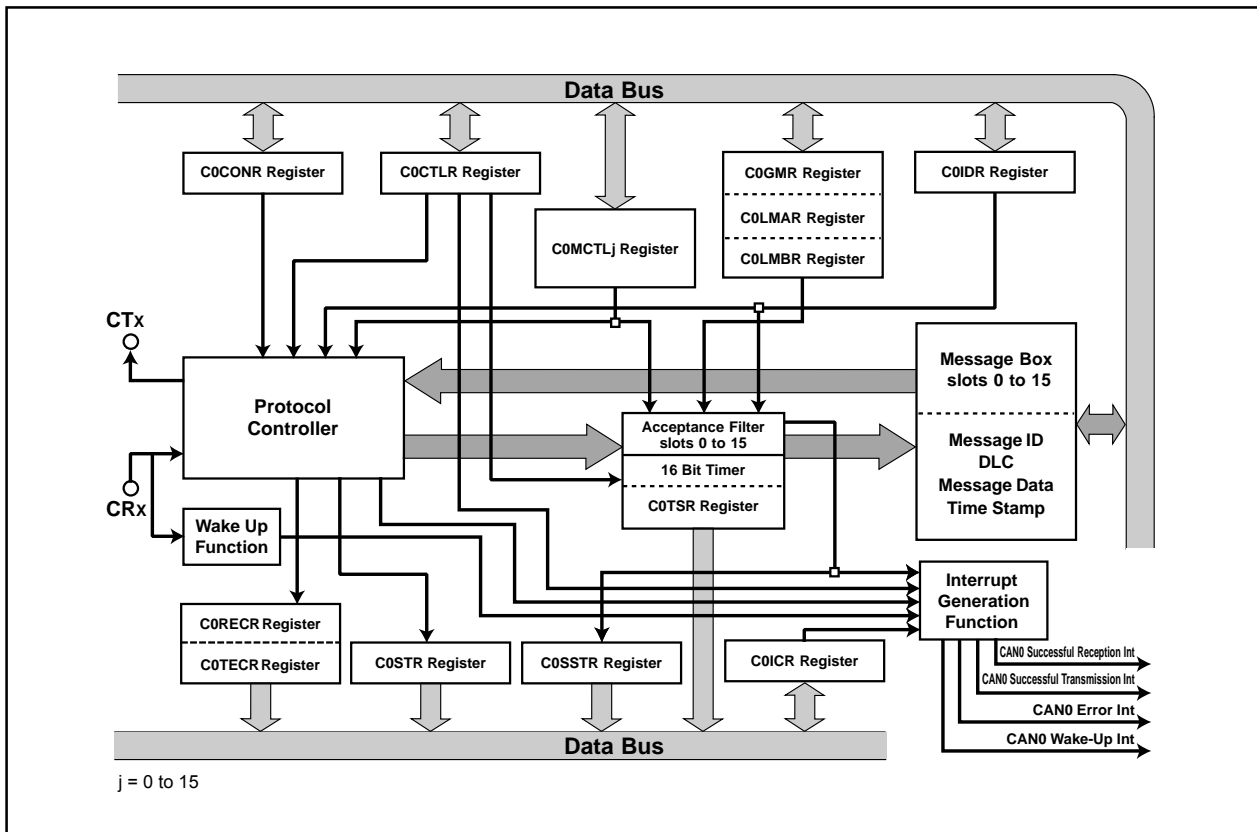


Figure 17.1 Block Diagram of CAN Module

| | |
|--------------------------------|--|
| CTx/CRx: | CAN I/O pins. |
| Protocol controller: | This controller handles the bus arbitration and the CAN protocol services, i.e. bit timing, stuffing, error status etc. |
| Message box: | This memory block consists of 16 slots that can be configured either as transmitter or receiver. Each slot contains an individual ID, data length code, a data field (8 bytes) and a time stamp. |
| Acceptance filter: | This block performs filtering operation for received messages. For the filtering operation, the COGMR register, the COLMAR register, or the COLMBR register is used. |
| 16 bit timer: | Used for the time stamp function. When the received message is stored in the message memory, the timer value is stored as a time stamp. |
| Wake-up function: | CAN0 wake-up interrupt request is generated by a message from the CAN bus. |
| Interrupt generation function: | The interrupt requests are generated by the CAN module. CAN0 successful reception interrupt, CAN0 successful transmission interrupt, CAN0 error interrupt and CAN0 wake-up interrupt. |

17.1 CAN Module-Related Registers

The CAN0 module has the following registers.

(1) CAN Message Box

A CAN module is equipped with 16 slots (16 bytes or 8 words each). Slots 14 and 15 can be used as Basic CAN.

- Priority of the slots: The smaller the number of the slot, the higher the priority, in both transmission and reception.
- A program can define whether a slot is defined as transmitter or receiver.

(2) Acceptance Mask Registers

A CAN module is equipped with 3 masks for the acceptance filter.

- CAN0 global mask register (COGMR register: 6 bytes)
Configuration of the masking condition for acceptance filtering processing to slots 0 to 13
- CAN0 local mask A register (COLMAR register: 6 bytes)
Configuration of the masking condition for acceptance filtering processing to slot 14
- CAN0 local mask B register (COLMBR register: 6 bytes)
Configuration of the masking condition for acceptance filtering processing to slot 15

(3) CAN SFR Registers

- CAN0 message control register j (COMCTLj register: 8 bits × 16) (j = 0 to 15)
Control of transmission and reception of a corresponding slot
- CANi control register (CiCLR register: 16 bits) (i = 0, 1)
Control of the CAN protocol
- CAN0 status register (COSTR register: 16 bits)
Indication of the protocol status
- CAN0 slot status register (COSSTR register: 16 bits)
Indication of the status of contents of each slot
- CAN0 interrupt control register (COICR register: 16 bits)
Selection of "interrupt enabled or disabled" for each slot
- CAN0 extended ID register (COIDR register: 16 bits)
Selection of ID format (standard or extended) for each slot
- CAN0 configuration register (COCONR register: 16 bits)
Configuration of the bus timing
- CAN0 receive error count register (CORECR register: 8 bits)
Indication of the error status of the CAN module in reception: the counter value is incremented or decremented according to the error occurrence.
- CAN0 transmit error count register (COTEER register: 8 bits)
Indication of the error status of the CAN module in transmission: the counter value is incremented or decremented according to the error occurrence.
- CAN0 time stamp register (COTSR register: 16 bits)
Indication of the value of the time stamp counter
- CAN0 acceptance filter support register (COAFS register: 16 bits)
Decoding the received ID for use by the acceptance filter support unit

Explanation of each register is given as follows.

17.1.1 CAN0 Message Box

Table 17.1 shows the memory mapping of the CAN0 message box.

It is possible to access to the message box in byte or word.

Mapping of the message contents differs from byte access to word access. Byte access or word access can be selected by the MsgOrder bit of the C0CTRL register.

Table 17.1 Memory Mapping of CAN0 Message Box

| Address | Message content (Memory mapping) | |
|-------------------------------|--|--|
| | Byte access (8 bits) | Word access (16 bits) |
| $0060_{16} + n \cdot 16 + 0$ | SID ₁₀ to SID ₆ | SID ₅ to SID ₀ |
| $0060_{16} + n \cdot 16 + 1$ | SID ₅ to SID ₀ | SID ₁₀ to SID ₆ |
| $0060_{16} + n \cdot 16 + 2$ | EID ₁₇ to EID ₁₄ | EID ₁₃ to EID ₆ |
| $0060_{16} + n \cdot 16 + 3$ | EID ₁₃ to EID ₆ | EID ₁₇ to EID ₁₄ |
| $0060_{16} + n \cdot 16 + 4$ | EID ₅ to EID ₀ | Data Length Code (DLC) |
| $0060_{16} + n \cdot 16 + 5$ | Data Length Code (DLC) | EID ₅ to EID ₀ |
| $0060_{16} + n \cdot 16 + 6$ | Data byte 0 | Data byte 1 |
| $0060_{16} + n \cdot 16 + 7$ | Data byte 1 | Data byte 0 |
| ⋮ | ⋮ | ⋮ |
| $0060_{16} + n \cdot 16 + 13$ | Data byte 7 | Data byte 6 |
| $0060_{16} + n \cdot 16 + 14$ | Time stamp high-order byte | Time stamp low-order byte |
| $0060_{16} + n \cdot 16 + 15$ | Time stamp low-order byte | Time stamp high-order byte |

n = 0 to 15: the number of the slot

Figures 17.2 and 17.3 show the bit mapping in each slot in byte access and word access. The content of each slot remains unchanged unless transmission or reception of a new message is performed.

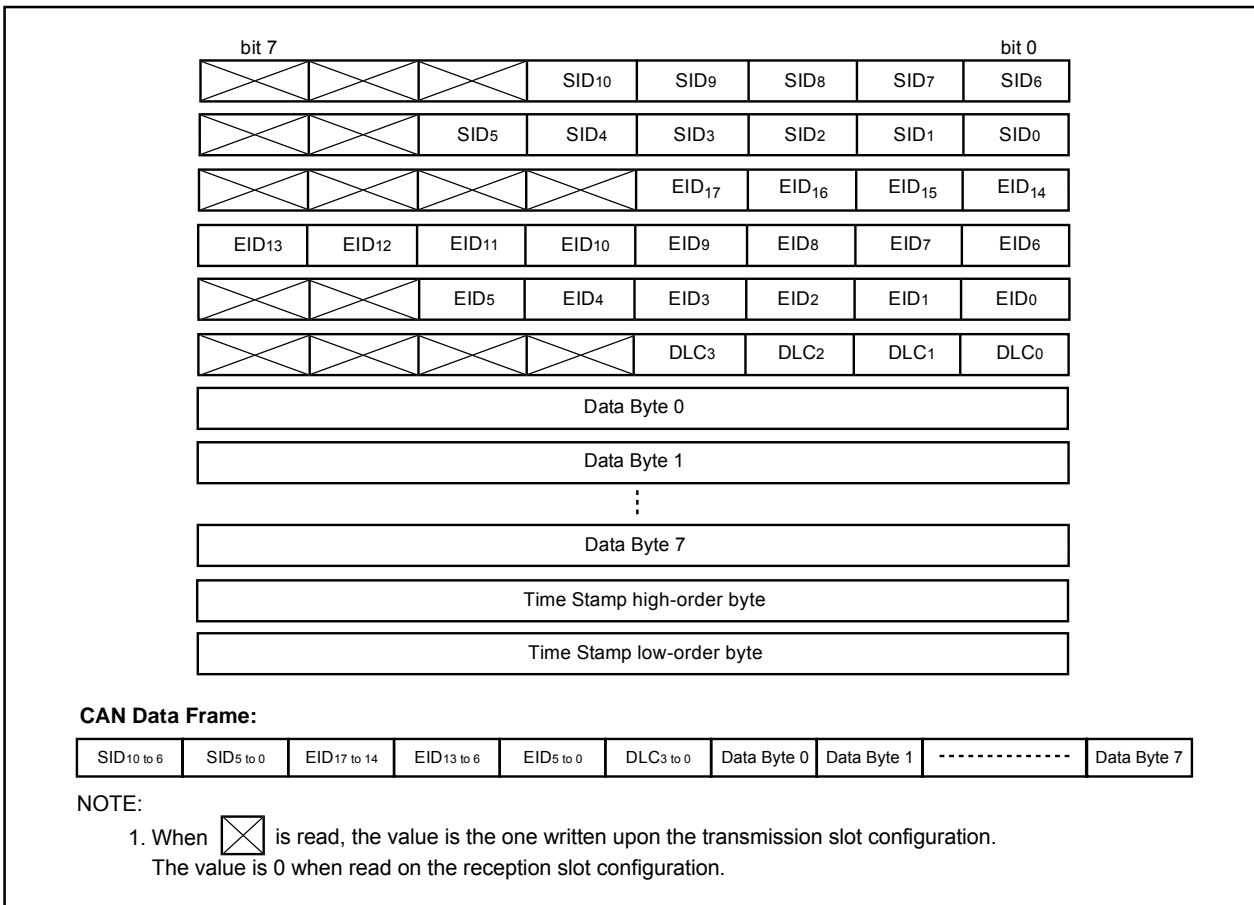


Figure 17.2 Bit Mapping in Byte Access

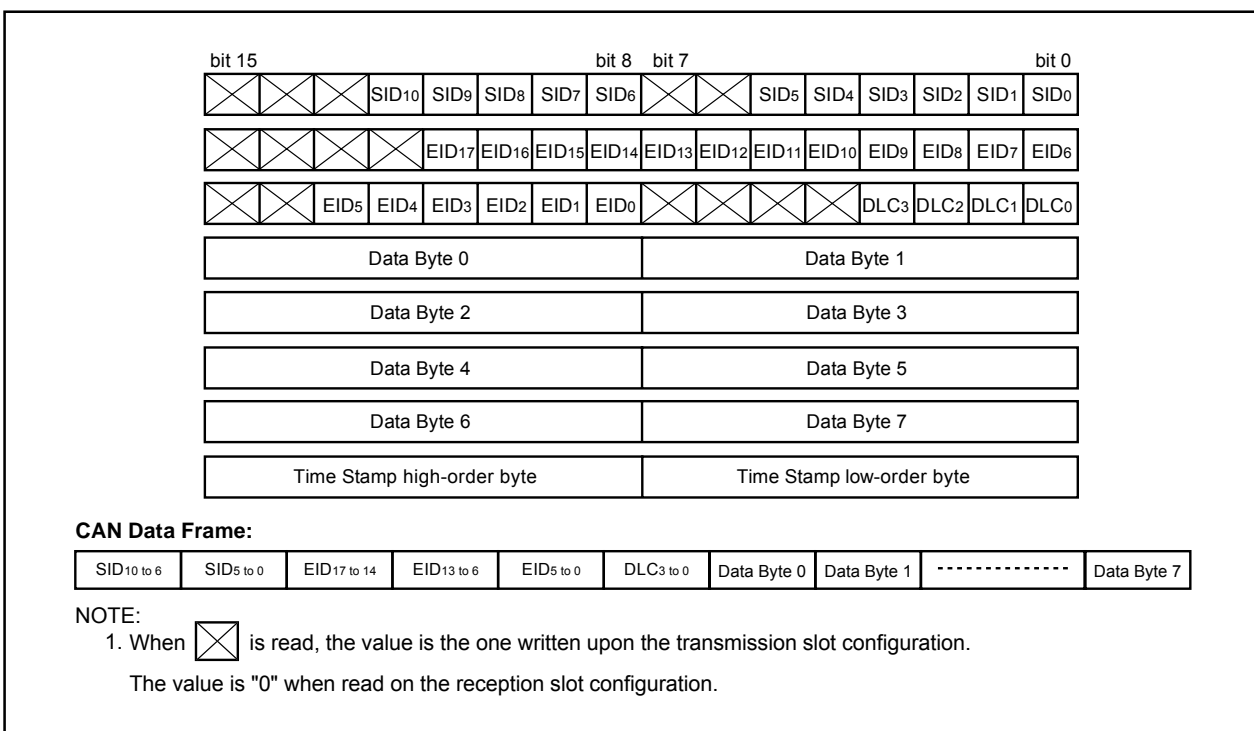


Figure 17.3 Bit Mapping in Word Access

17.1.2 Acceptance Mask Registers

Figures 17.4 and 17.5 show the COGMR register, the COLMAR register, and the COLMBR register, in which bit mapping in byte access and word access are shown.

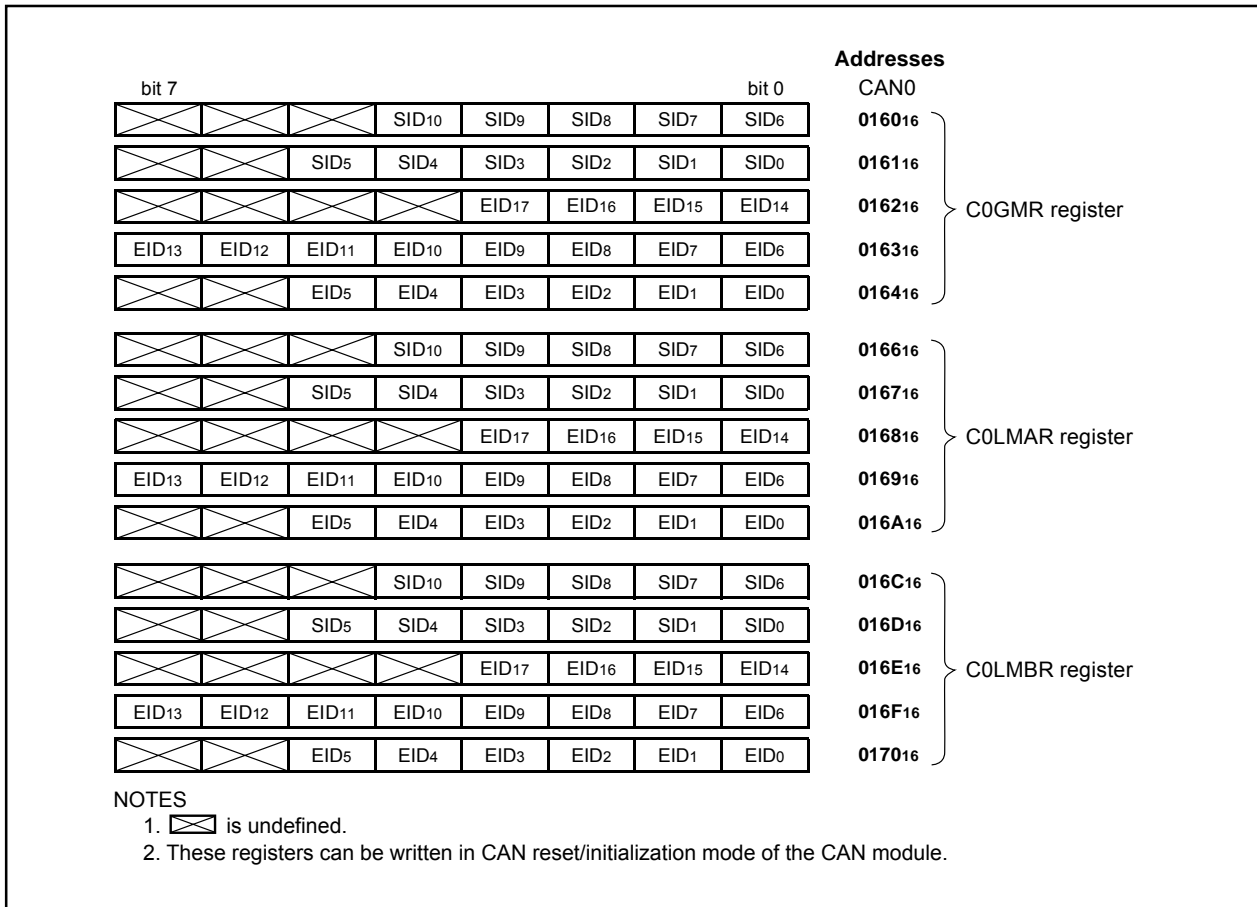


Figure 17.4 Bit Mapping of Mask Registers in Byte Access

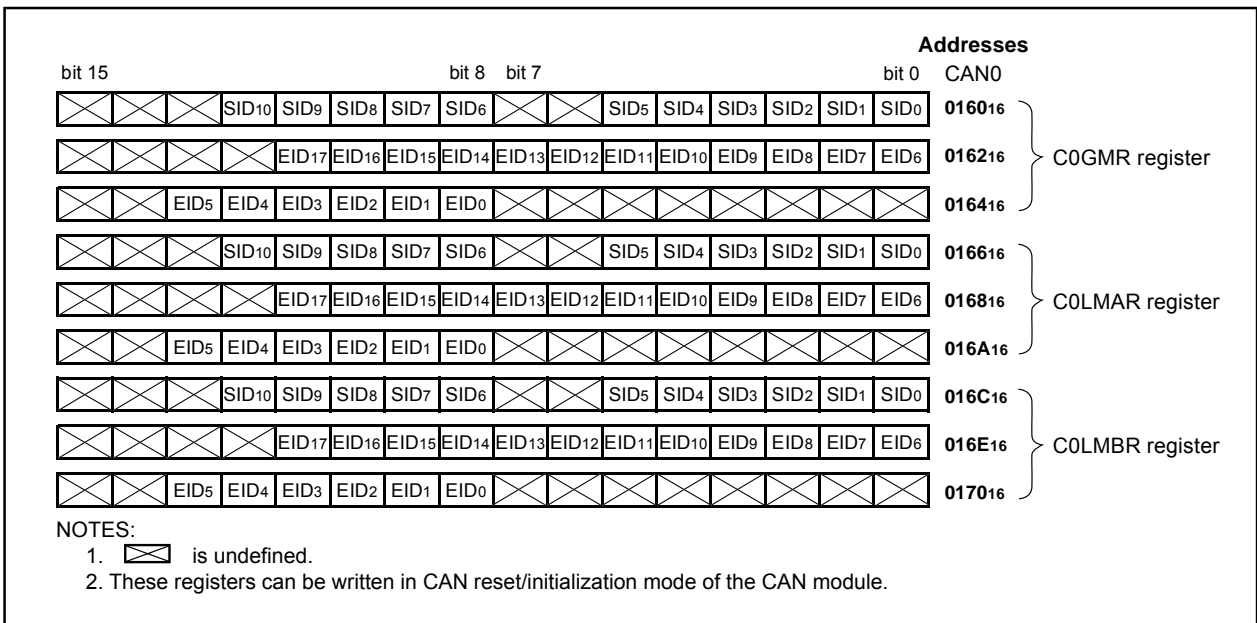


Figure 17.5 Bit Mapping of Mask Registers in Word Access

17.1.3 CAN SFR Registers

17.1.3.1 COMCTLj Register (j = 0 to 15)

Figure 17.6 shows the COMCTLj register.

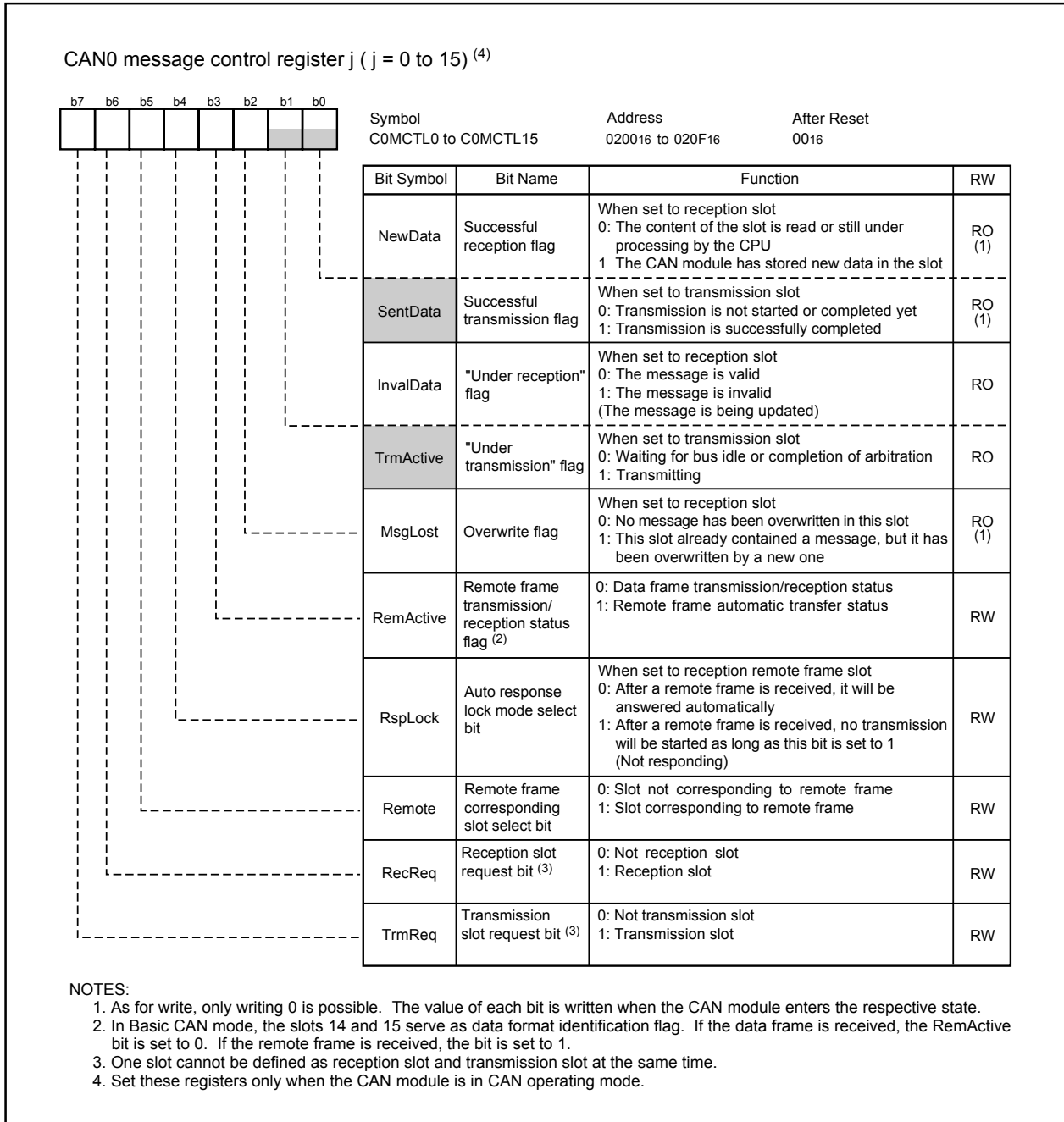


Figure 17.6 COMCTLj Register

17.1.3.2 C0CTRL Register

Figure 17.7 shows the C0CTRL register.

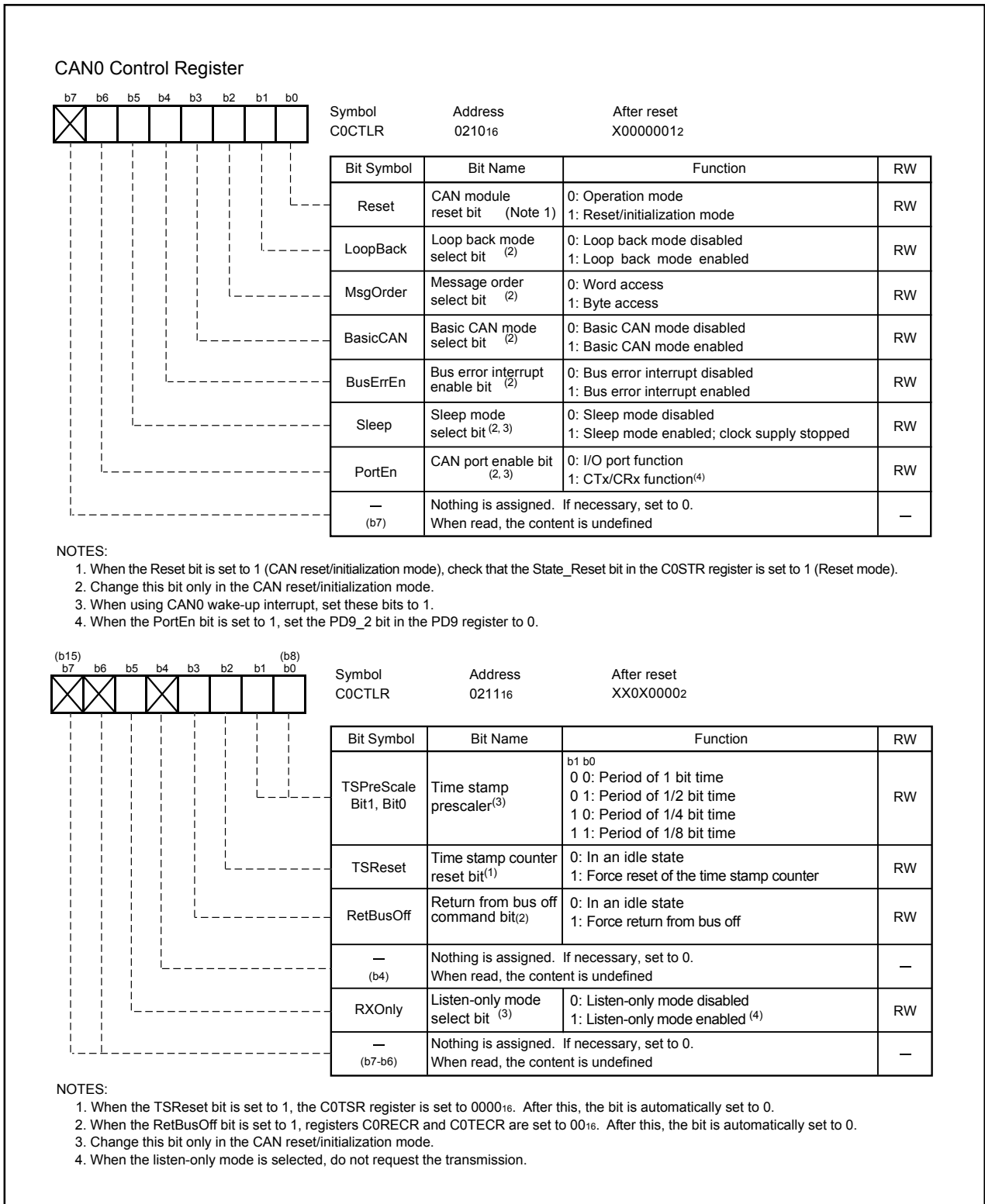


Figure 17.7 C0CTRL Register

17.1.3.3 C0STR Register

Figure 17.8 shows the C0STR register.

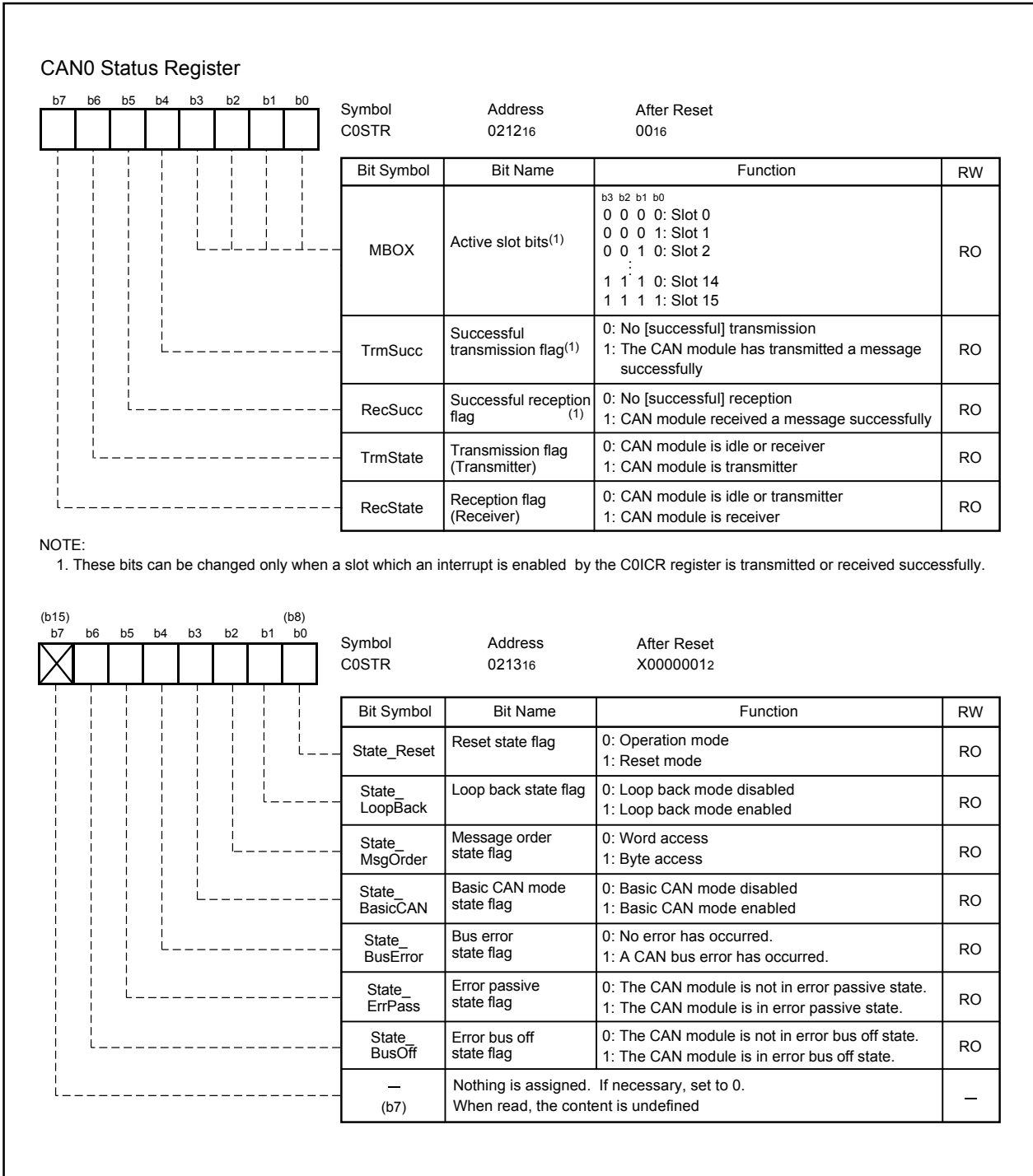


Figure 17.8 C0STR Register

17.1.3.4 C0SSTR Register

Figure 17.9 shows the C0SSTR register.

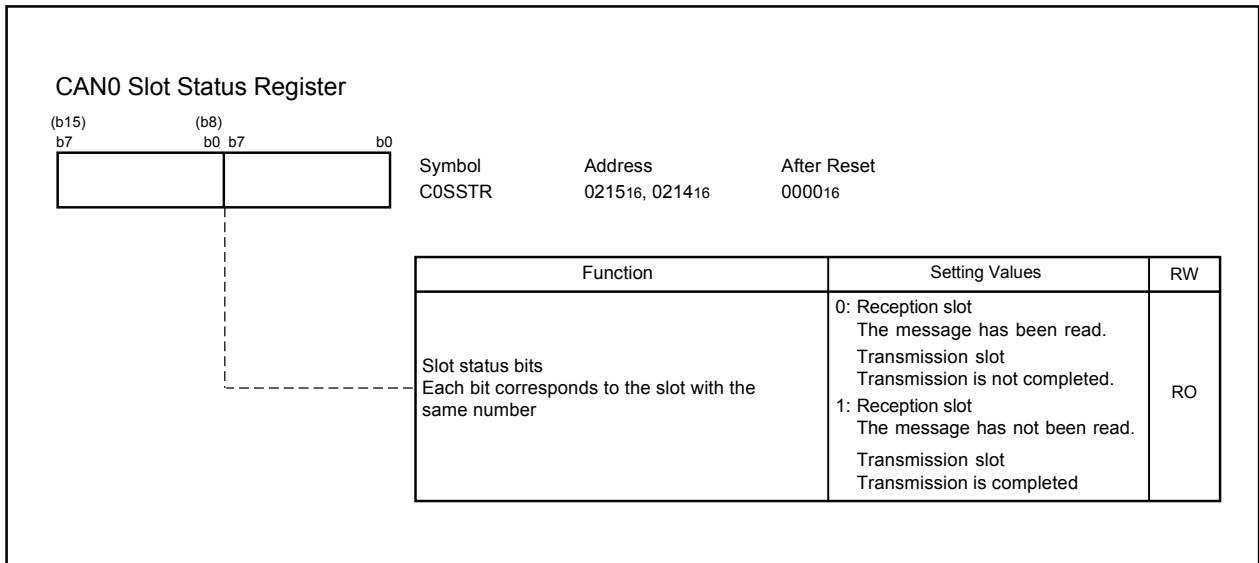


Figure 17.9 C0SSTR Register

17.1.3.5 C0ICR Register

Figure 17.10 shows the C0ICR register.

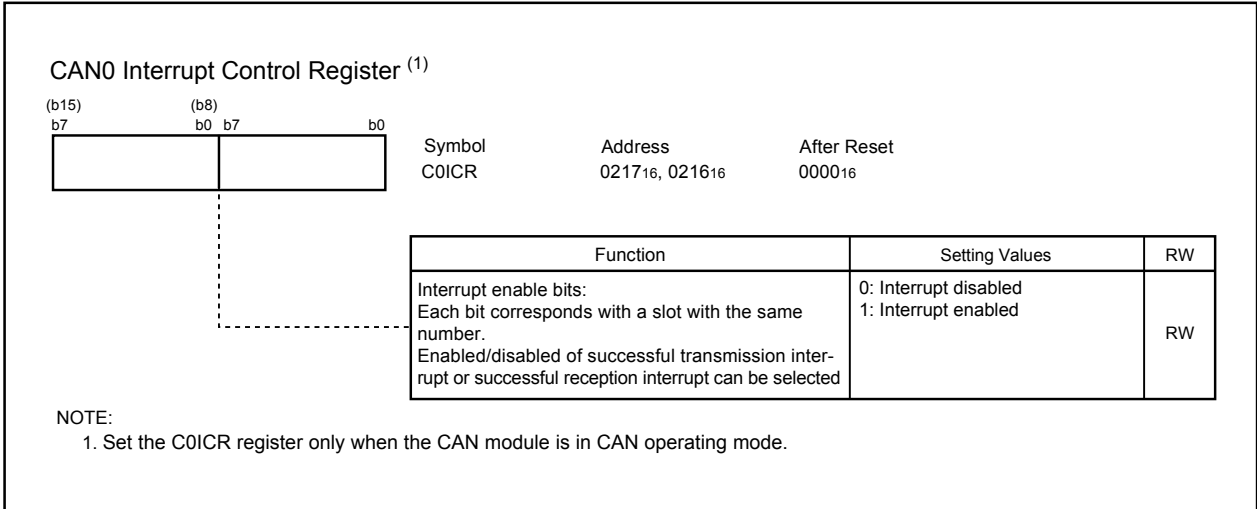


Figure 17.10 C0ICR Register

17.1.3.6 C0IDR Register

Figure 17.11 shows the C0IDR register.

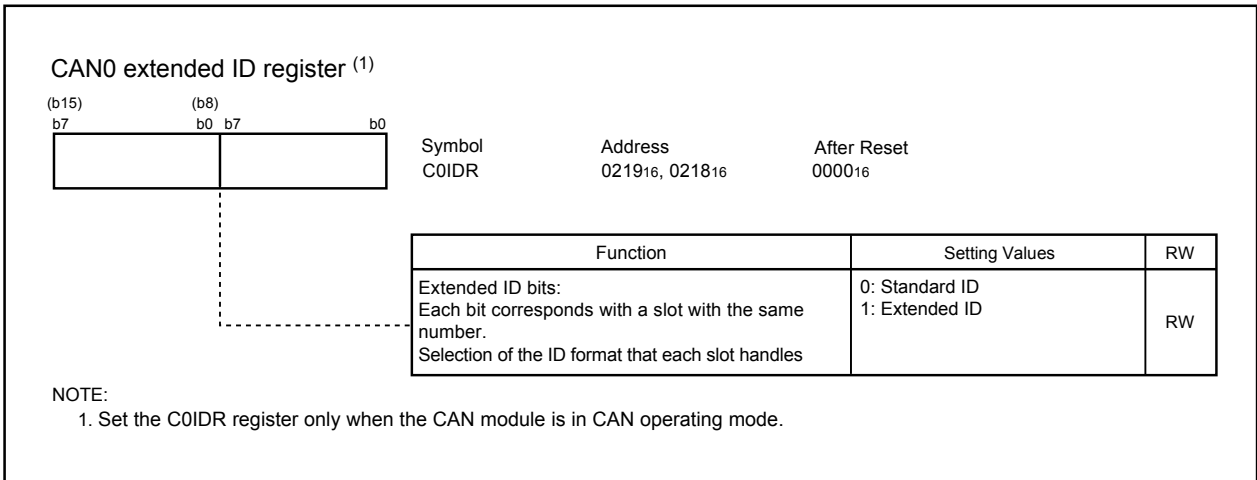


Figure 17.11 C0IDR Register

17.1.3.7 C0CONR Register

Figure 17.12 shows the C0CONR register.

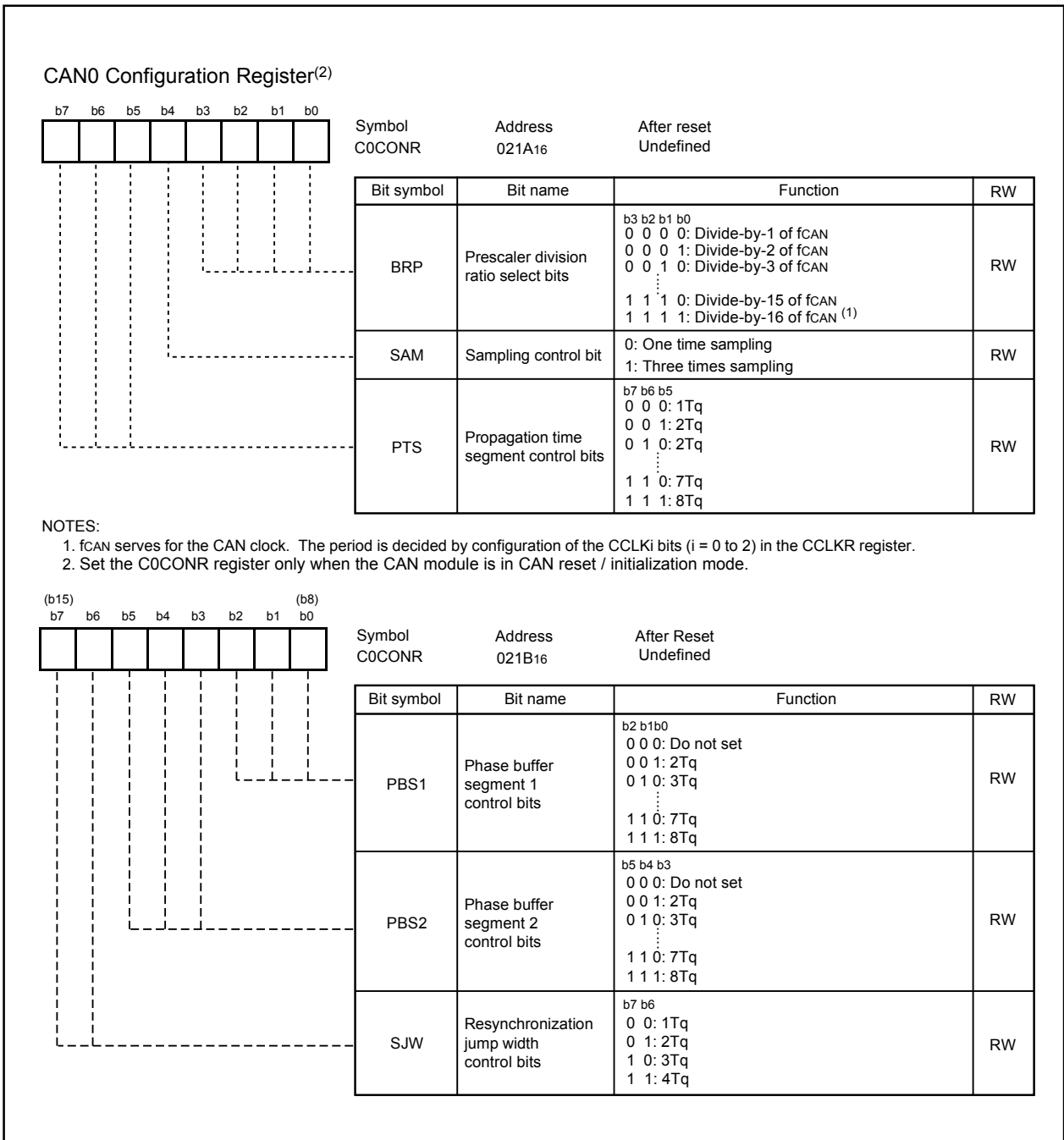


Figure 17.12 C0CONR Register

17.1.3.8 C0RECR Register

Figure 17.13 shows the C0RECR register.

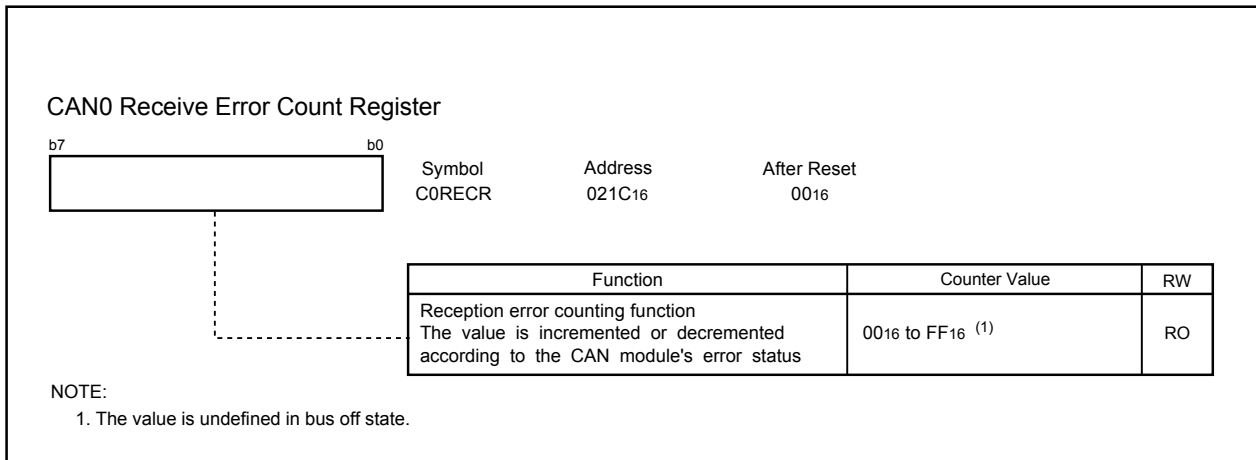


Figure 17.13 C0RECR Register

17.1.3.9 C0TECR Register

Figure 17.14 shows the C0TECR register.

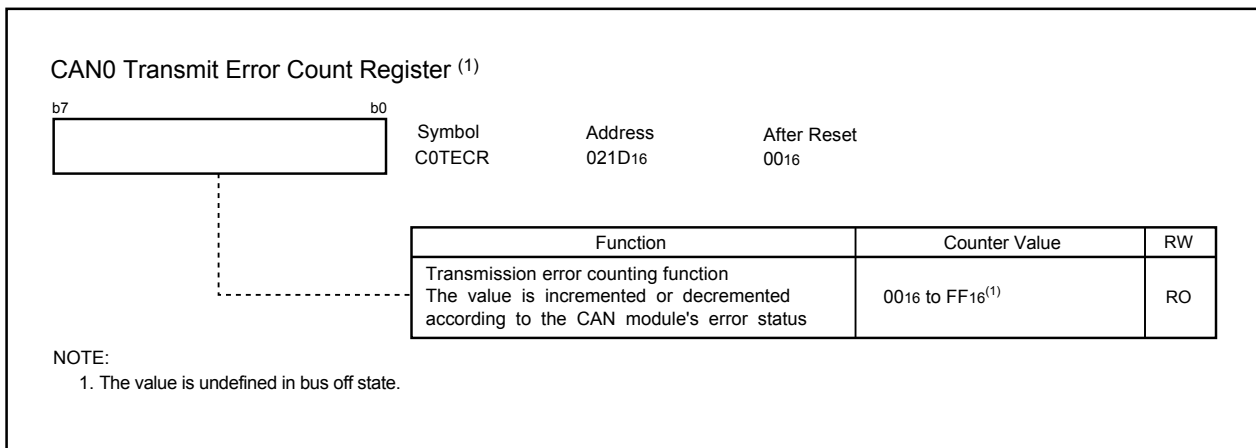


Figure 17.14 C0TECR Register

17.1.3.10 C0TSR Register

Figure 17.15 shows the C0TSR register.

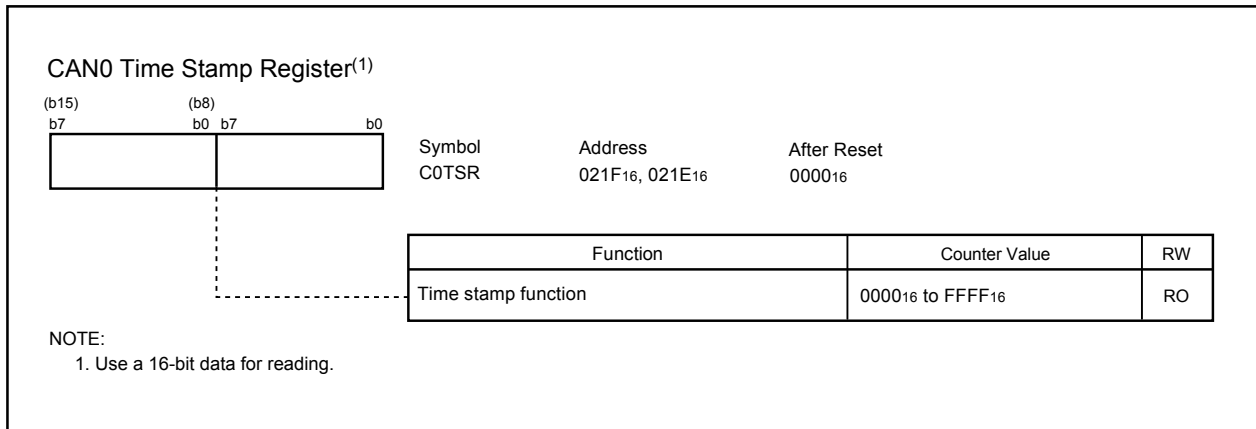


Figure 17.15 C0TSR Register

17.1.3.11 C0AFS Register

Figure 17.16 shows the C0AFS register.

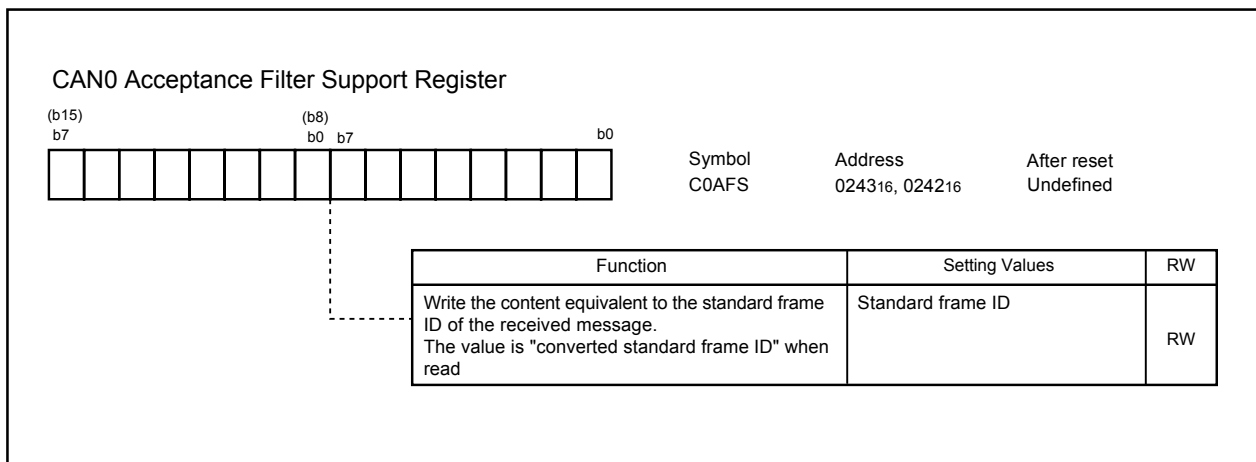


Figure 17.16 C0AFS Register

17.2 Operating Modes

The CAN module has the following four operating modes.

- CAN Reset/Initialization Mode
- CAN Operating Mode
- CAN Sleep Mode
- CAN Interface Sleep Mode

Figure 17.17 shows transition between operating modes.

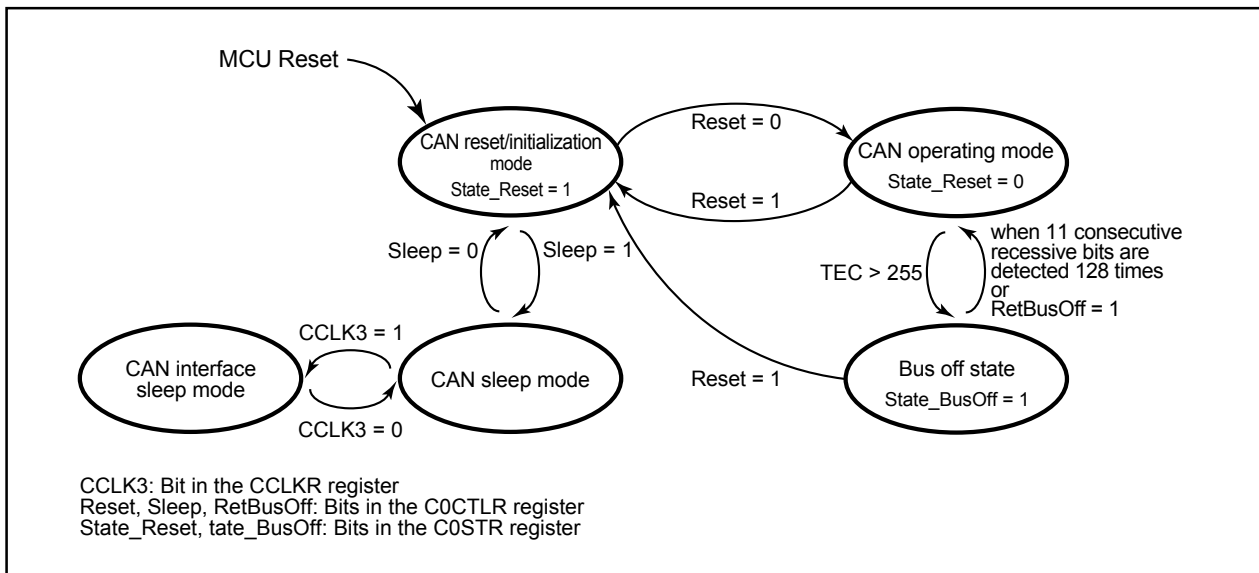


Figure 17.17 Transition Between Operating Modes

17.2.1 CAN Reset/Initialization Mode

The CAN reset/initialization mode is activated upon MCU reset or by setting the Reset bit in the C0CTLR register to 1. If the Reset bit is set to 1, check that the State_Reset bit in the C0STR register is set to 1.

Entering the CAN reset/initialization mode initiates the following functions by the module:

- CAN communication is impossible.
- When the CAN reset/initialization mode is activated during an ongoing transmission in operation mode, the module suspends the mode transition until completion of the transmission (successful, arbitration loss, or error detection). Then, the State_Reset bit is set to 1, and the CAN reset/initialization mode is activated.
- Registers C0MCTLj (j = 0 to 15), C0STR, C0ICR, C0IDR, C0RECR, C0TECR, and C0TSR are initialized. All these registers are locked to prevent CPU modification.
- Registers C0CTLR, C0CONR, C0GMR, C0LMAR, and C0LMBR and the CAN0 message box retain their contents and are available for CPU access.

17.2.2 CAN Operating Mode

The CAN operating mode is activated by setting the Reset bit in the C0CTLR register to 0. If the Reset bit is set to 0, check that the State_Reset bit in the C0STR register is set to 0.

If 11 consecutive recessive bits are detected after entering the CAN operating mode, the module initiates the following functions:

- The module's communication functions are released and it becomes an active node on the network and may transmit and receive CAN messages.
- Release the internal fault confinement logic including receive and transmit error counters. The module may leave the CAN operating mode depending on the error counts.

Within the CAN operating mode, the module may be in three different sub modes, depending on which type of communication functions are performed:

- Module idle : The modules receive and transmit sections are inactive.
- Module receives : The module receives a CAN message sent by another node.
- Module transmits : The module transmits a CAN message. The module may receive its own message simultaneously when the LoopBack bit in the C0CTLR register = 1 (Loop back mode enabled).

Figure 17.18 shows sub modes of the CAN operating mode.

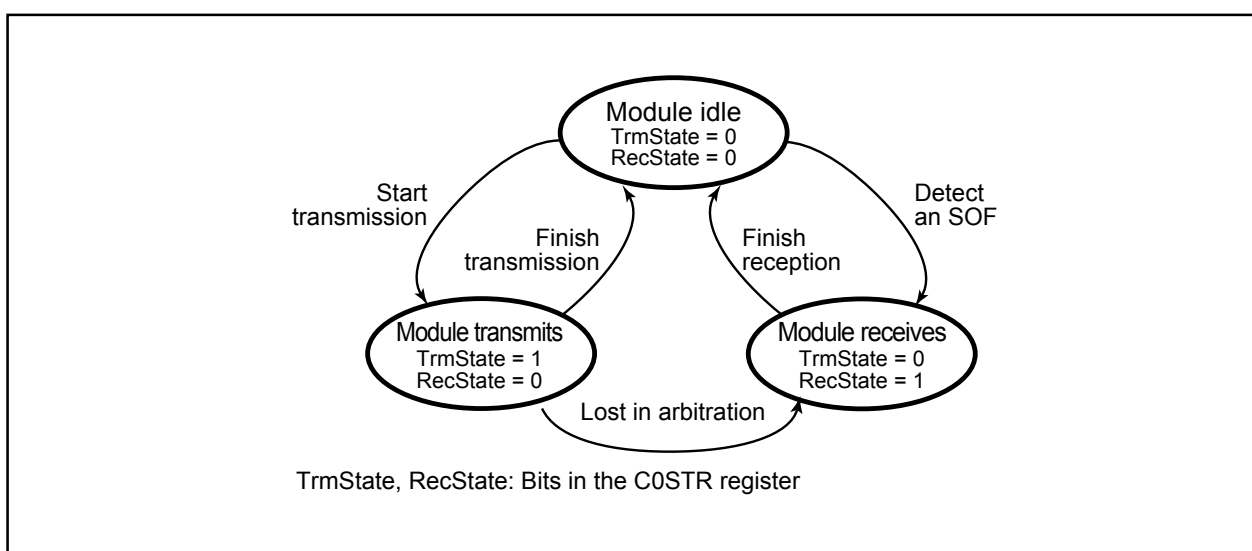


Figure 17.18 Sub Modes of CAN Operating Mode

17.2.3 CAN Sleep Mode

The CAN sleep mode is activated by setting the Sleep bit in the C0CTLR register to 1. It should never be activated from the CAN operating mode but only via the CAN reset/initialization mode.

Entering the CAN sleep mode instantly stops the clock supply to the module and thereby reduces power dissipation.

17.2.4 CAN Interface Sleep Mode

The CAN interface sleep mode is activated by setting the CCLK3 bit in the CCLKR register to 1. It should never be activated but only via the CAN sleep mode.

Entering the CAN interface sleep mode instantly stops the clock supply to the CPU Interface in the module and thereby reduces power dissipation.

17.2.5 Bus Off State

The bus off state is entered according to the fault confinement rules of the CAN specification. When returning to the CAN operating mode from the bus off state, the module has the following two cases.

In this time, the value of any CAN registers, except registers C0STR, C0RECR, and C0TECR, does not change.

- (1) When 11 consecutive recessive bits are detected 128 times

The module enters instantly into error active state and the CAN communication becomes possible immediately.

- (2) When the RetBusOff bit in the C0CTLR register = 1 (Force return from buss off)

The module enters instantly into error active state, and the CAN communication becomes possible again after 11 consecutive recessive bits are detected.

17.3 Configuration of the CAN Module System Clock

The M16C/29 Group has a CAN module system clock select circuit.

Configuration of the CAN module system clock can be done through manipulating the CCLKR register and the BRP bit in the C0CONR register.

For the CCLKR register, refer to **7. Clock Generation Circuit**.

Figure 17.19 shows a block diagram of the clock generation circuit of the CAN module system.

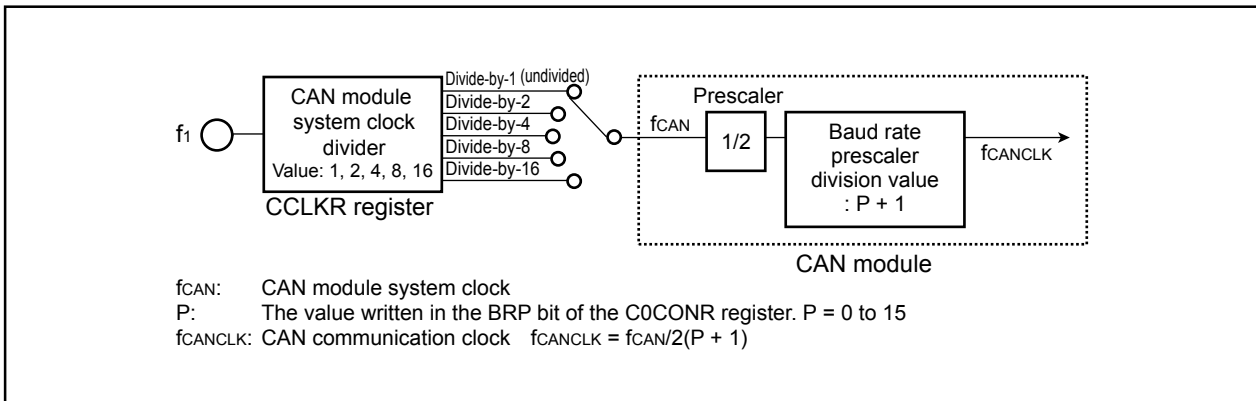


Figure 17.19 Block Diagram of CAN Module System Clock Generation Circuit

17.3.1 Bit Timing Configuration

The bit time consists of the following four segments:

- Synchronization segment (SS)
This serves for monitoring a falling edge for synchronization.
- Propagation time segment (PTS)
This segment absorbs physical delay on the CAN network which amounts to double the total sum of delay on the CAN bus, the input comparator delay, and the output driver delay.
- Phase buffer segment 1 (PBS1)
This serves for compensating the phase error. When the falling edge of the bit falls later than expected, the segment can become longer by the maximum of the value defined in SJW.
- Phase buffer segment 2 (PBS2)
This segment has the same function as the phase buffer segment 1. When the falling edge of the bit falls earlier than expected, the segment can become shorter by the maximum of the value defined in SJW.

Figure 17.20 shows the bit timing.

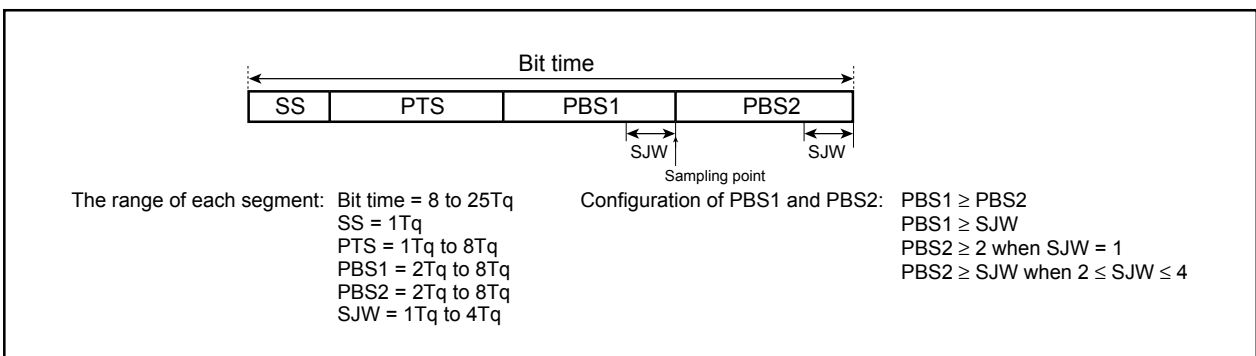


Figure 17.20 Bit Timing

17.3.2 Bit-rate

Bit-rate depends on f_1 , the division value of the CAN module system clock, the division value of the baud rate prescaler, and the number of Tq of one bit.

Table 17.2 shows the examples of bit-rate.

Table 17.2 Examples of Bit-rate

| Bit-rate | 20MHz | 16MHz | 10MHz | 8MHz |
|----------|-----------|-----------|-----------|----------|
| 1Mbps | 10Tq (1) | 8Tq (1) | – | – |
| 500kbps | 10Tq (2) | 8Tq (2) | 10Tq (1) | 8Tq (1) |
| | 20Tq (1) | 16Tq (1) | – | – |
| 125kbps | 10Tq (8) | 8Tq (8) | 10Tq (4) | 8Tq (4) |
| | 20Tq (4) | 16Tq (4) | 20Tq (2) | 16Tq (2) |
| 83.3kbps | 10Tq (12) | 8Tq (12) | 10Tq (6) | 8Tq (6) |
| | 20Tq (6) | 16Tq (6) | 20Tq (3) | 16Tq (3) |
| 33.3kbps | 10Tq (30) | 8Tq (30) | 10Tq (15) | 8Tq (15) |
| | 20Tq (15) | 16Tq (15) | – | – |

NOTE:

1. The number in () indicates a value of “ f_{CAN} division value” multiplied by “baud rate prescaler division value”.

■ Calculation of Bit-rate

$$\frac{f_1}{2 \times \text{“}f_{CAN} \text{ division value (Note 1)“} \times \text{“baud rate prescaler division value (Note 2)“} \times \text{“number of Tq of one bit“}}$$

Note 1: f_{CAN} division value = 1, 2, 4, 8, 16

f_{CAN} division value: a value selected in the CCLKR register

Note 2: Baud rate prescaler division value = P + 1 (P: 0 to 15)

P: a value selected in the BRP bit in the C0CONR register

17.4 Acceptance Filtering Function and Masking Function

These functions serve the users to select and receive a facultative message. The C0GMR register, the COLMAR register, and the COLMBR register can perform masking to the standard ID and the extended ID of 29 bits. The C0GMR register corresponds to slots 0 to 13, the COLMAR register corresponds to slot 14, and the COLMBR register corresponds to slot 15. The masking function becomes valid to 11 bits or 29 bits of a received ID according to the value in the corresponding slot of the C0IDR register upon acceptance filtering operation. When the masking function is employed, it is possible to receive a certain range of IDs. **Figure 17.21** shows correspondence of the mask registers and slots, **Figure 17.22** shows the acceptance function.

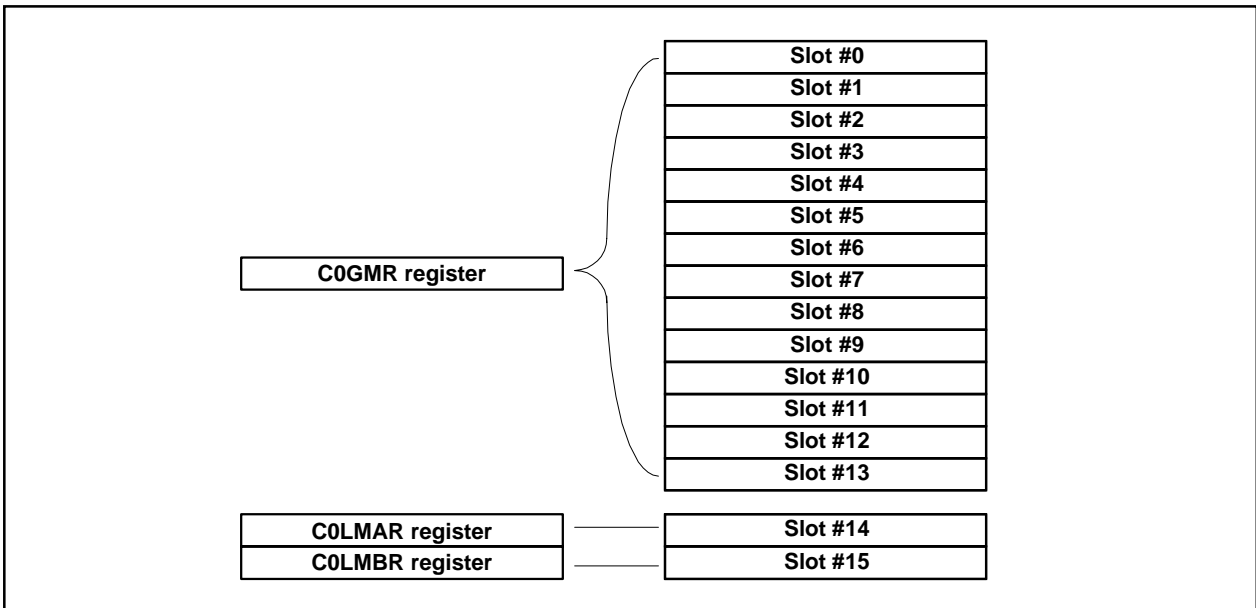


Figure 17.21 Correspondence of Mask Registers to Slots

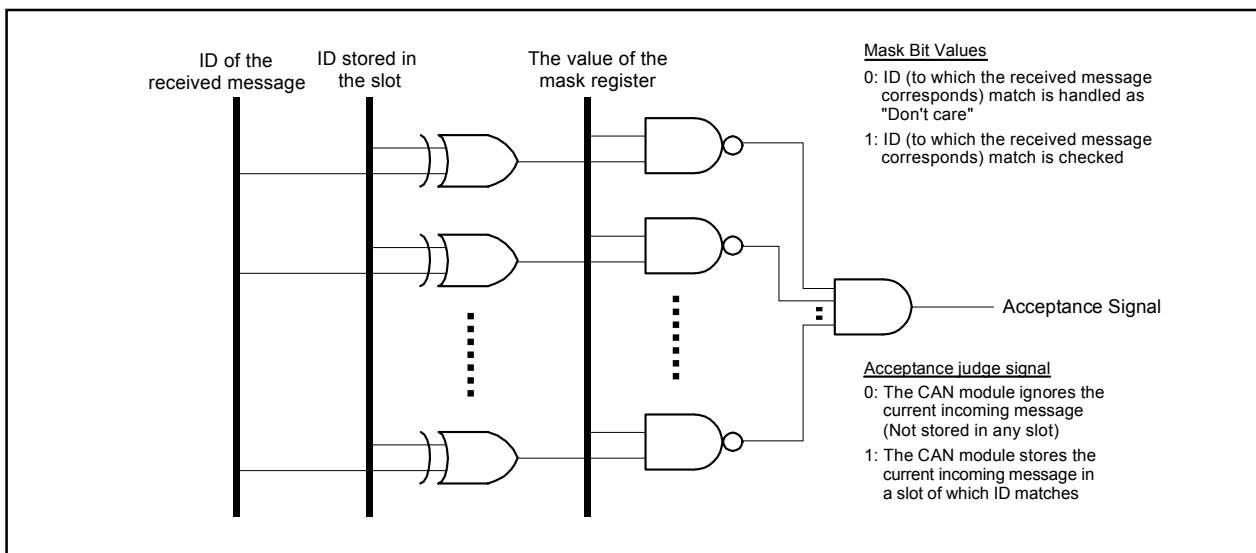


Figure 17.22 Acceptance Function

When using the acceptance function, note the following points.

- (1) When one ID is defined in two slots, the one with a smaller number alone is valid.
- (2) When it is configured that slots 14 and 15 receive all IDs with Basic CAN mode, slots 14 and 15 receive all IDs which are not stored into slots 0 to 13.

17.5 Acceptance Filter Support Unit (ASU)

The acceptance filter support unit has a function to judge valid/invalid of a received ID through table search. The IDs to receive are registered in the data table; a received ID is stored in the C0AFS register, and table search is performed with a decoded received ID. The acceptance filter support unit can be used for the IDs of the standard frame only.

The acceptance filter support unit is valid in the following cases.

- When the ID to receive cannot be masked by the acceptance filter.
(Example) IDs to receive: 078¹⁶, 087¹⁶, 111¹⁶
- When there are too many IDs to receive; it would take too much time to filter them by software.

Figure 17.23 shows the write and read of the C0AFS register in word access.

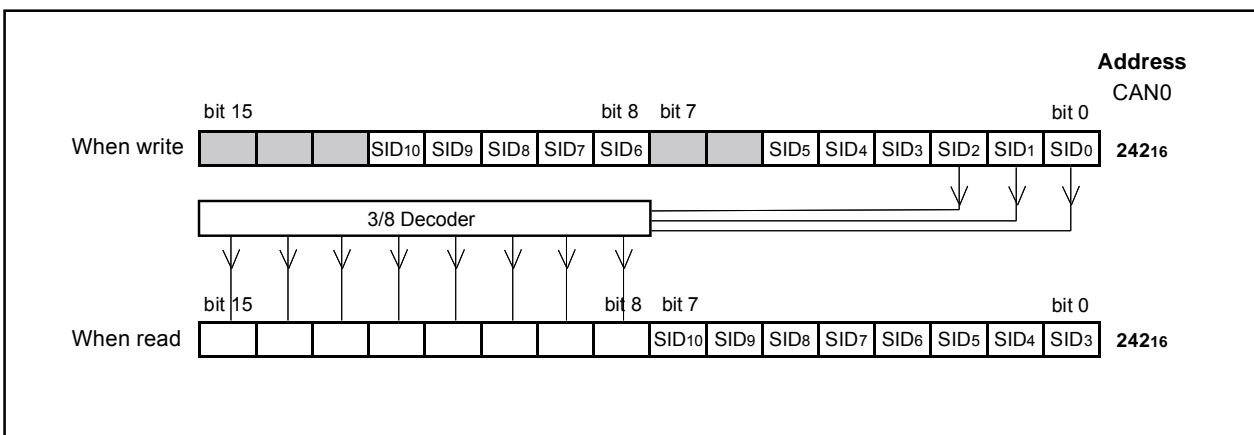


Figure 17.23 Write/read of C0AFS Register in Word Access

17.6 BasicCAN Mode

When the BasicCAN bit in the C0CTRL register is set to 1 (Basic CAN mode enabled), slots 14 and 15 correspond to Basic CAN mode. During normal operations, individual slots can select either data frame or remote frame by CPU setting. However, in Basic CAN mode, both frames can be selected.

When slots 14 and 15 are defined as reception slots in Basic CAN mode, received messages are stored in slots 14 and 15 alternately.

The received message data format can be determined by the RemActive bit in the C0MCTLj register ($j = 0$ to 15).

Figure 17.24 shows the operation of slots 14 and 15 in Basic CAN mode.

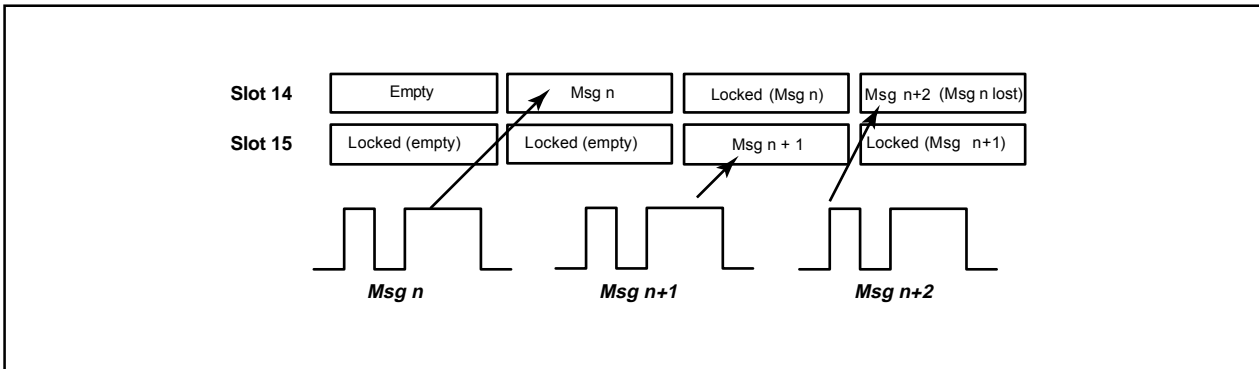


Figure 17.24 Operation of Slots 14 and 15 in Basic CAN Mode

When using Basic CAN mode, note the following points.

- (1) Setting of Basic CAN mode has to be done in CAN reset/initialization mode.
- (2) Select the same ID for slots 14 and 15. Also, setting of the C0LMAR and C0LMBR register has to be the same.
- (3) Define slots 14 and 15 as reception slot only.
- (4) There is no protection available against message overwrite. A message can be overwritten by a new message.
- (5) Slots 0 to 13 can be used in the same way as in normal CAN operating mode.

17.7 Return from Bus off Function

When the protocol controller enters bus off state, it is possible to make it forced return from bus off state by setting the RetBusOff bit in the C0CTLR register to 1 (Force return from bus off). At this time, the error state changes from bus off state to error active state. If the RetBusOff bit is set to 1, registers C0RECR and C0TECR are initialized and the State_Reset bit in the C0STR register is set to 0 (The CAN module is not in error bus off state). However, registers of the CAN module such as C0CONR register and the content of each slot are not initialized.

17.8 Time Stamp Counter and Time Stamp Function

When the C0TSR register is read, the value of the time stamp counter at the moment is read. The period of the time stamp counter reference clock is the same as that of 1 bit time that is configured by the C0CONR register. The time stamp counter functions as a free run counter.

The 1 bit time period can be divided by 1 (undivided), 2, 4 or 8 to produce the time stamp counter reference clock. Use the TSPreScale bit in the C0CTLR register to select the divide-by-n value.

The time stamp counter is equipped with a register that captures the counter value when the protocol controller regards it as a successful reception. The captured value is stored when a time stamp value is stored in a reception slot.

17.9 Listen-Only Mode

When the RXOnly bit in the C0CTLR register is set to 1, the module enters listen-only mode.

In listen-only mode, no transmission -- data frames, error frames, and ACK response -- is performed to bus.

When listen-only mode is selected, do not request the transmission.

17.10 Reception and Transmission

Configuration of CAN Reception and Transmission Mode

Table 17.3 shows configuration of CAN reception and transmission mode.

Table 17.3 Configuration of CAN Reception and Transmission Mode

| TrmReq | RecReq | Remote | RspLock | Communication mode of the slot |
|--------|--------|--------|---------|--|
| 0 | 0 | – | – | Communication environment configuration mode: configure the communication mode of the slot. |
| 0 | 1 | 0 | 0 | Configured as a reception slot for a data frame. |
| 1 | 0 | 1 | 0 | Configured as a transmission slot for a remote frame. (At this time the RemActive = 1.) After completion of transmission, this functions as a reception slot for a data frame. (At this time the RemActive = 0.) However, when an ID that matches on the CAN bus is detected before remote frame transmission, this immediately functions as a reception slot for a data frame. |
| 1 | 0 | 0 | 0 | Configured as a transmission slot for a data frame. |
| 0 | 1 | 1 | 1/0 | Configured as a reception slot for a remote frame. (At this time the RemActive = 1.) After completion of reception, this functions as a transmission slot for a data frame. (At this time the RemActive = 0.) However, transmission does not start as long as RspLock bit remains 1; thus no automatic response. Response (transmission) starts when the RspLock bit is set to 0. |

TrmReq, RecReq, Remote, RspLock, RemActive, RspLock: Bits in the C0MCTLj register (j = 0 to 15)

When configuring a slot as a reception slot, note the following points.

- (1) Before configuring a slot as a reception slot, be sure to set the C0MCTLj register (j = 0 to 15) to 00₁₆.
- (2) A received message is stored in a slot that matches the condition first according to the result of reception mode configuration and acceptance filtering operation. Upon deciding in which slot to store, the smaller the number of the slot is, the higher priority it has.
- (3) In normal CAN operating mode, when a CAN module transmits a message of which ID matches, the CAN module never receives the transmitted data. In loop back mode, however, the CAN module receives back the transmitted data. In this case, the module does not return ACK.

When configuring a slot as a transmission slot, note the following points.

- (1) Before configuring a slot as a transmission slot, be sure to set the C0MCTLj registers to 00₁₆.
- (2) Set the TrmReq bit in the C0MCTLj register to 0 (not transmission slot) before rewriting a transmission slot.
- (3) A transmission slot should not be rewritten when the TrmActive bit in the C0MCTLj register is 1 (transmitting).
If it is rewritten, an undefined data will be transmitted.

17.10.1 Reception

Figure 17.25 shows the behavior of the module when receiving two consecutive CAN messages, that fit into the slot of the shown C0MCTLj register (j = 0 to 15) and leads to losing/overwriting of the first message.

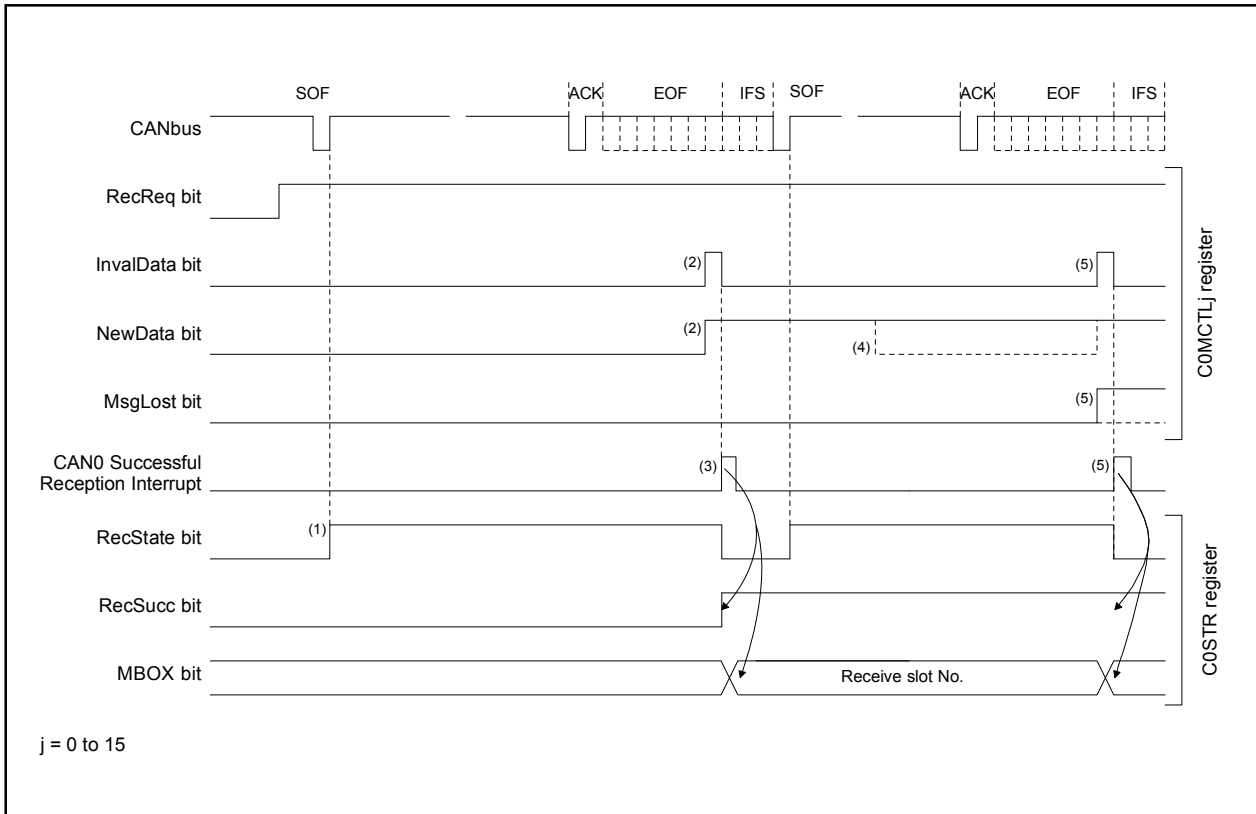


Figure 17.25 Timing of Receive Data Frame Sequence

- (1) On monitoring a SOF on the CAN bus the RecState bit in the C0STR register becomes 1 (CAN module is receiver) immediately, given the module has no transmission pending.
- (2) After successful reception of the message, the NewData bit in the C0MCTLj register (j = 0 to 15) of the receiving slot becomes 1 (stored new data in slot). The InvalData bit in the C0MCTLj register becomes 1 (message is being updated) at the same time and the InvalData bit becomes 0 (message is valid) again after the complete message was transferred to the slot.
- (3) When the interrupt enable bit in the C0ICR register of the receiving slot = 1 (interrupt enabled), the CAN0 successful reception interrupt request is generated and the MBOX bit in the C0STR register is changed. It shows the slot number where the message was stored and the RecSucc bit in the C0STR register is active.
- (4) Read the message out of the slot after setting the New Data bit to 0 (the content of the slot is read or still under processing by the CPU) by program.
- (5) If the NewData bit is set to 0 by program or the next CAN message is received successfully before the receive request for the slot is canceled, the MsgLost bit in the C0MCTLj register is set to 1 (message has been overwritten). The new received message is transferred to the slot. Generating of an interrupt request and change of the C0STR register are same as in 3).

17.10.2 Transmission

Figure 17.26 shows the timing of the transmit sequence.

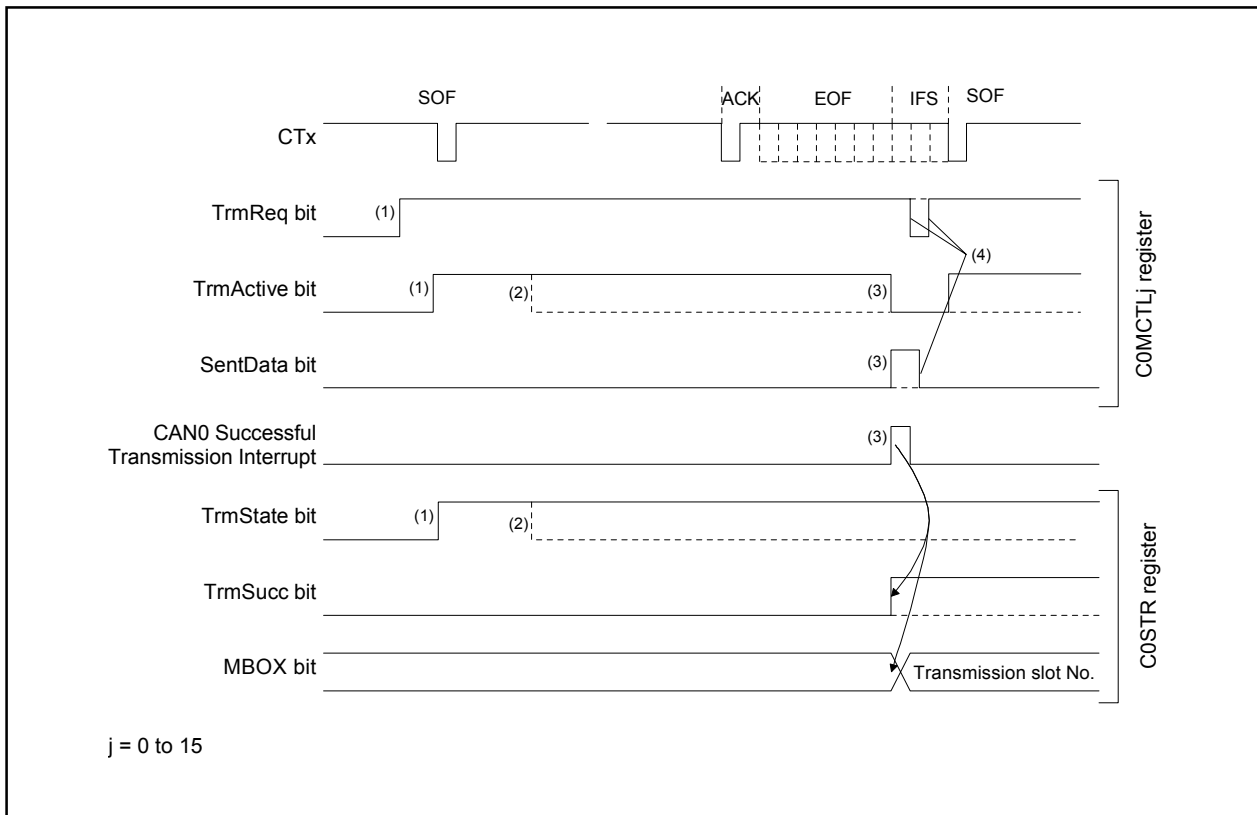


Figure 17.26 Timing of Transmit Sequence

- (1) If the TrmReq bit in the C0MCTLj register ($j = 0$ to 15) is set to 1 (Transmission slot) in the bus idle state, the TrmActive bit in the C0MCTLj register and the TrmState bit in the C0STR register are set to 1 (Transmitting/Transmitter), and CAN module starts the transmission.
- (2) If the arbitration is lost after the CAN module starts the transmission, the TrmActive and TrmState bits are set to 0.
- (3) If the transmission has been successful without lost in arbitration, the SentData bit in the C0MCTLj register is set to 1 (Transmission is successfully completed) and TrmActive bit in the C0MCTLj register is set to 0 (Waiting for bus idle or completion of arbitration). And when the interrupt enable bits in the C0ICR register = 1 (Interrupt enabled), CAN0 successful transmission interrupt request is generated and the MBOX (the slot number which transmitted the message) and TrmSucc bit in the C0STR register are changed.
- (4) When starting the next transmission, set bits SentData and TrmReq to 0. And set the TrmReq bit to 1 after checking that bits SentData and TrmReq are set to 0.

17.11 CAN Interrupts

The CAN module provides the following CAN interrupts.

- CAN0 Successful Reception Interrupt
- CAN0 Successful Transmission Interrupt
- CAN0 Error Interrupt
 - Error Passive State
 - Error BusOff State
 - Bus Error (this feature can be disabled separately)
- CAN0 Wake-up Interrupt

When the CPU detects the CAN0 successful reception/transmission interrupt request, the MBOX bit in the C0STR register must be read to determine which slot has generated the interrupt request.

18. CRC Calculation Circuit

The Cyclic Redundancy Check (CRC) calculation detects errors in blocks of data. The MCU uses a generator polynomial of CRC_CCITT ($X^{16} + X^{12} + X^5 + 1$) or CRC-16 ($X^{16} + X^{15} + X^2 + 1$) to generate CRC code.

The CRC code is a 16-bit code generated for a block of a given data length in multiples of bytes. The code is updated in the CRC data register everytime one byte of data is transferred to a CRC input register. The data register must be initialized before use. Generation of CRC code for one byte of data is completed in two machine cycles.

Figure 18.1 shows the block diagram of the CRC circuit. **Figure 18.2** shows the CRC-related registers. **Figure 18.3** shows the calculation example using the CRC_CCITT operation.

18.1 CRC Snoop

The CRC circuit includes the ability to snoop reads and writes to certain SFR addresses. This can be used to accumulate the CRC value on a stream of data without using extra bandwidth to explicitly write data into the CRCIN register. All SFR addresses after 0020_{16} are subject to the CRC snoop. The CRC snoop is useful to snoop the writes to a UART TX buffer, or the reads from a UART RX buffer.

To snoop an SFR address, the target address is written to the CRC snoop Address Register (CRCSAR). The two most significant bits of this register enable snooping on reads or writes to the target address. If the target SFR is written to by the CPU or DMA, and the CRC snoop write bit is set (CRCSW=1), the CRC will latch the data into the CRCIN register. The new CRC code will be set in the CRCD register.

Similarly, if the target SFR is read by the CRC or DMA, and the CRC snoop read bit is set (CRCSR=1), the CRC will latch the data from the target into the CRCIN register and calculate the CRC.

The CRC circuit can only calculate CRC codes on data byte at a time. Therefore, if a target SFR is accessed in word (16 bit), only one low-order byte data is stored into the CRCIN register.

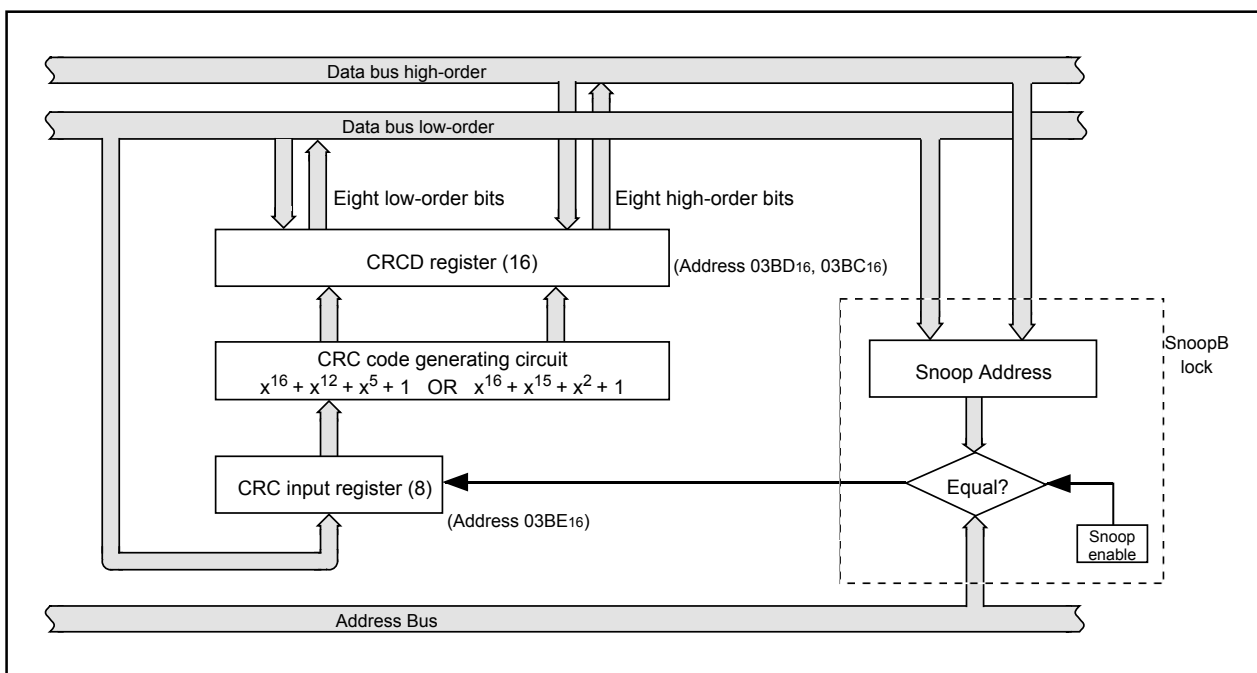


Figure 18.1 CRC circuit block diagram

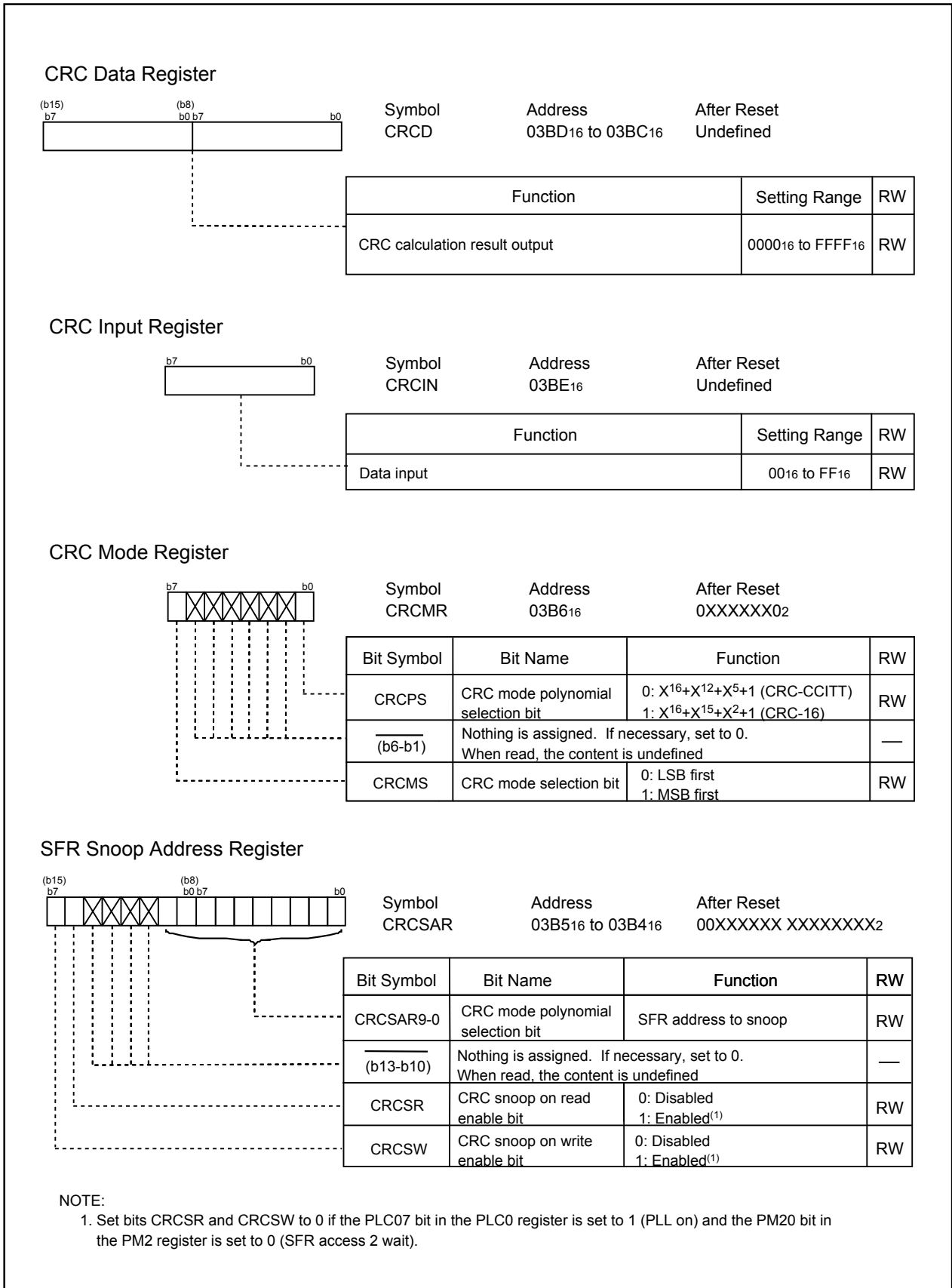


Figure 18.2. CRCD, CRCIN, CRCMR, CRCSAR Register

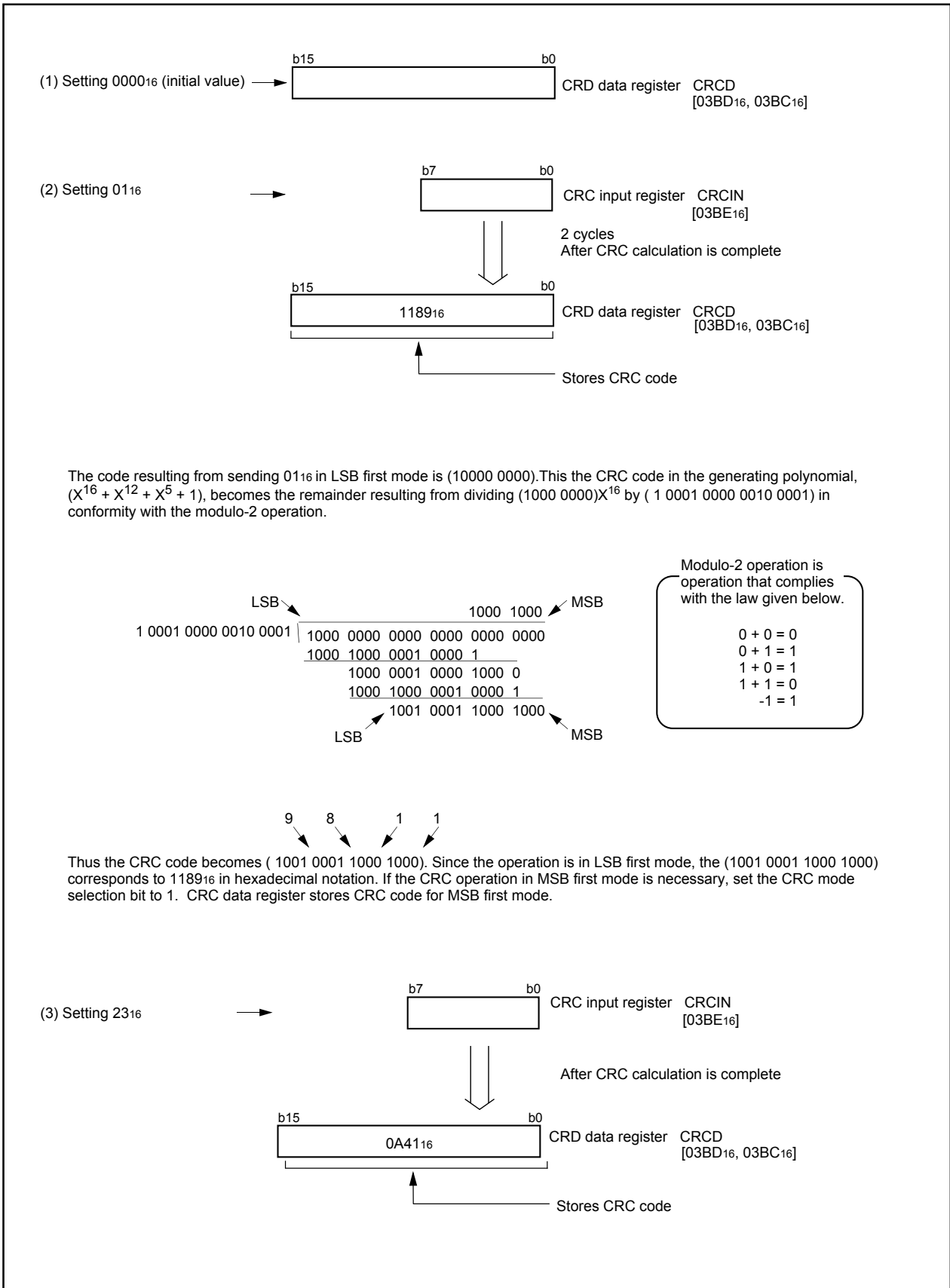


Figure 18.3. CRC Calculation

19. Programmable I/O Ports

Note

Ports P04 to P07, P10 to P14, P34 to P37 and P95 to P97 are not available in 64-pin package.

The programmable input/output ports (hereafter referred to simply as “I/O ports”) consist of 71 lines P0, P1, P2, P3, P6, P7, P8, P9, P10 (except P94) for the 80-pin package, or 55 lines P00 to P03, P15 to P17, P2, P30 to P33, P6, P7, P8, P90 to P93, P10 for the 64-pin package. Each port can be set for input or output every line by using a direction register, and can also be chosen to be or not be pulled high in sets of 4 lines.

Figures 19.1 to 19.4 show the I/O ports. **Figure 19.5** shows the I/O pins.

Each pin functions as an I/O port, a peripheral function input/output.

For details on how to set peripheral functions, refer to each functional description in this manual. If any pin is used as a peripheral function input, set the direction bit for that pin to 0 (input mode). Any pin used as an output pin for peripheral functions is directed for output no matter how the corresponding direction bit is set.

19.1 Port Pi Direction Register (PDi Register, i = 0 to 3, 6 to 10)

Figure 19.6 shows the direction registers.

This register selects whether the I/O port is to be used for input or output. The bits in this register correspond one for one to each port.

19.2 Port Pi Register (Pi Register, i = 0 to 3, 6 to 10)

Figure 19.7 shows the Pi registers.

Data input/output to and from external devices are accomplished by reading and writing to the Pi register. The Pi register consists of a port latch to hold the output data and a circuit to read the pin status. For ports set for input mode, the input level of the pin can be read by reading the corresponding Pi register, and data can be written to the port latch by writing to the Pi register.

For ports set for output mode, the port latch can be read by reading the corresponding Pi register, and data can be written to the port latch by writing to the Pi register. The data written to the port latch is output from the pin. The bits in the Pi register correspond one for one to each port.

19.3 Pull-up Control Register 0 to 2 (PUR0 to PUR2 Registers)

Figure 19.8 shows registers PUR0 to PUR2.

Registers PUR0 to PUR2 select whether the pins, divided into groups of four pins, are pulled up or not. The pins, selected by setting the bits in registers PUR0 to PUR2 to 1 (pull-up), are pulled up when the direction registers are set to 0 (input mode). The pins are pulled up regardless of the pins' function.

19.4 Port Control Register (PCR Register)

Figure 19.9 shows the port control register.

When the P1 register is read after setting the PCR0 bit in the PCR register to 1, the corresponding port latch can be read no matter how the PD1 register is set.

19.5 Pin Assignment Control Register (PACR)

Figure 19.10 shows the PACR register. After reset, set bits PACR2 to PACR0 in the PACR register before a signal is input or output to each pin. When bits PACR2 to PACR0 are not set, some pins do not function as I/O ports.

Bits PACR2 to PACR0: control pins to be used

Value after reset: 0002.

To select the 80-pin package, set the bits to 0112.

To select the 64-pin package, set the bits to 0102.

U1MAP bit: controls pin assignments for the UART1 function.

To assign the UART1 function to P64/ $\overline{\text{CTS1}}/\overline{\text{RTS1}}$, P65/CLK1, P66/RxD1, and P67/TxD1, set the U1MAP bit to 0 (P67 to P64).

To assign the function to P70/ $\overline{\text{CTS1}}/\overline{\text{RTS1}}$, P71/CLK1, P72/RxD1, and P73/TxD1, set the U1MAP bit to 1 (P73 to P70)

The PRC2 bit in the PRCR protects the PACR register. Set the PACR register after setting the PRC2 bit in the PRCR register.

19.6 Digital Debounce Function

Two digital debounce function circuits are provided. Level is determined when level is held, after applying either a falling edge or rising edge to the pin, longer than the programmed filter width time. This enables noise reduction.

This function is assigned to $\overline{\text{INT5}}/\overline{\text{INPC17}}$ and $\overline{\text{NMI}}/\overline{\text{SD}}$. Digital filter width is set in the NDDR register and the P17DDR register respectively. **Figure 19.11** shows the NDDR register and the P17DDR register. Additionally, a digital debounce function is disabled to the port P17 input and the port P85 input.

Filter width : $(n+1) \times 1/f8$ n: count value set in the NDDR register and P17DDR register

The NDDR register and the P17DDR register decrement count value with f8 as the count source. The NDDR register and the P17DDR register indicate count time. Count value is reloaded if a falling edge or a rising edge is applied to the pin.

The NDDR register and the P17DDR register can be set 00₁₆ to FF₁₆ when using the digital debounce function. Setting to FF₁₆ disables the digital filter. See **Figure 19.12** for details.

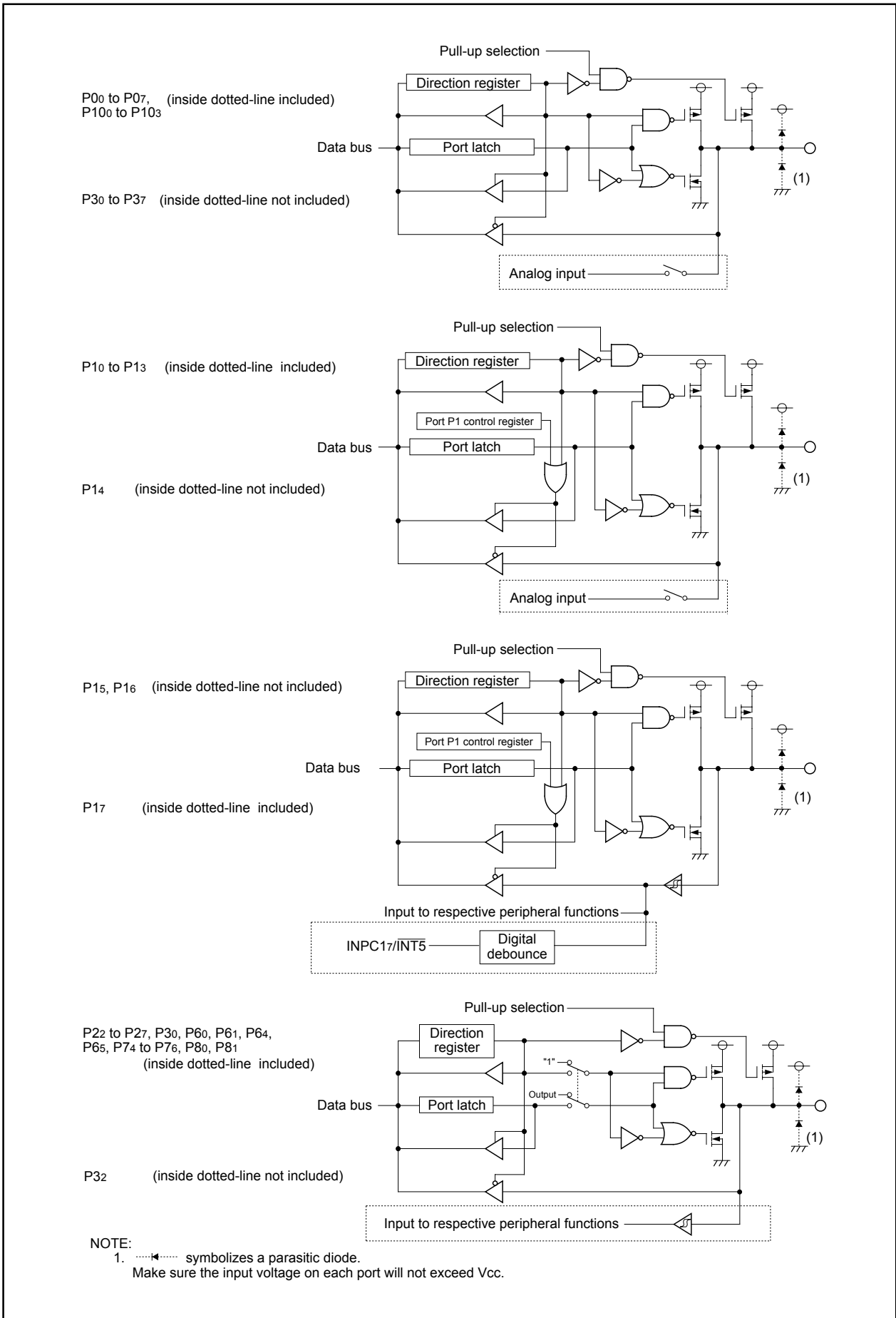


Figure 19.1 I/O Ports (1)

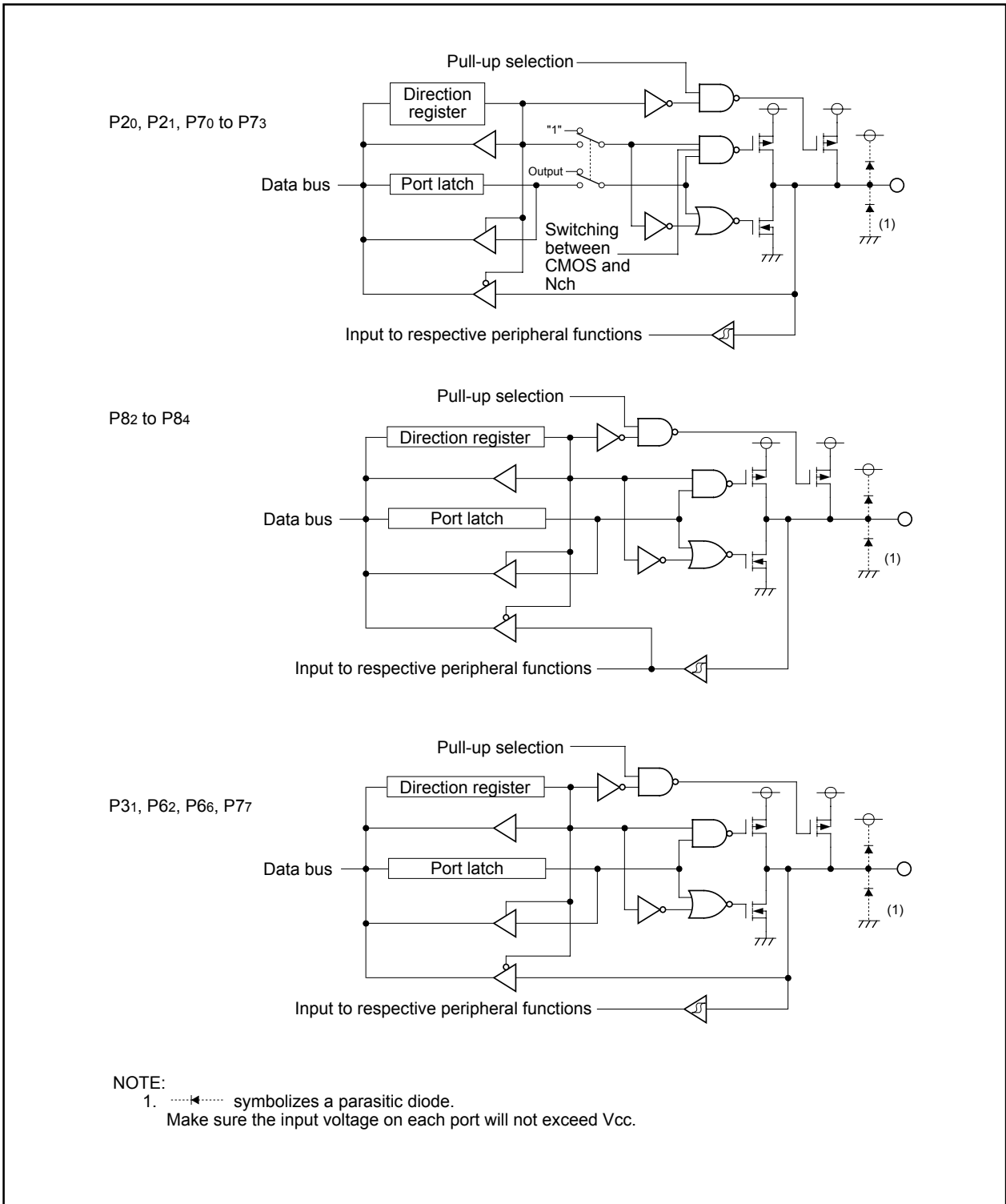


Figure 19.2 I/O Ports (2)

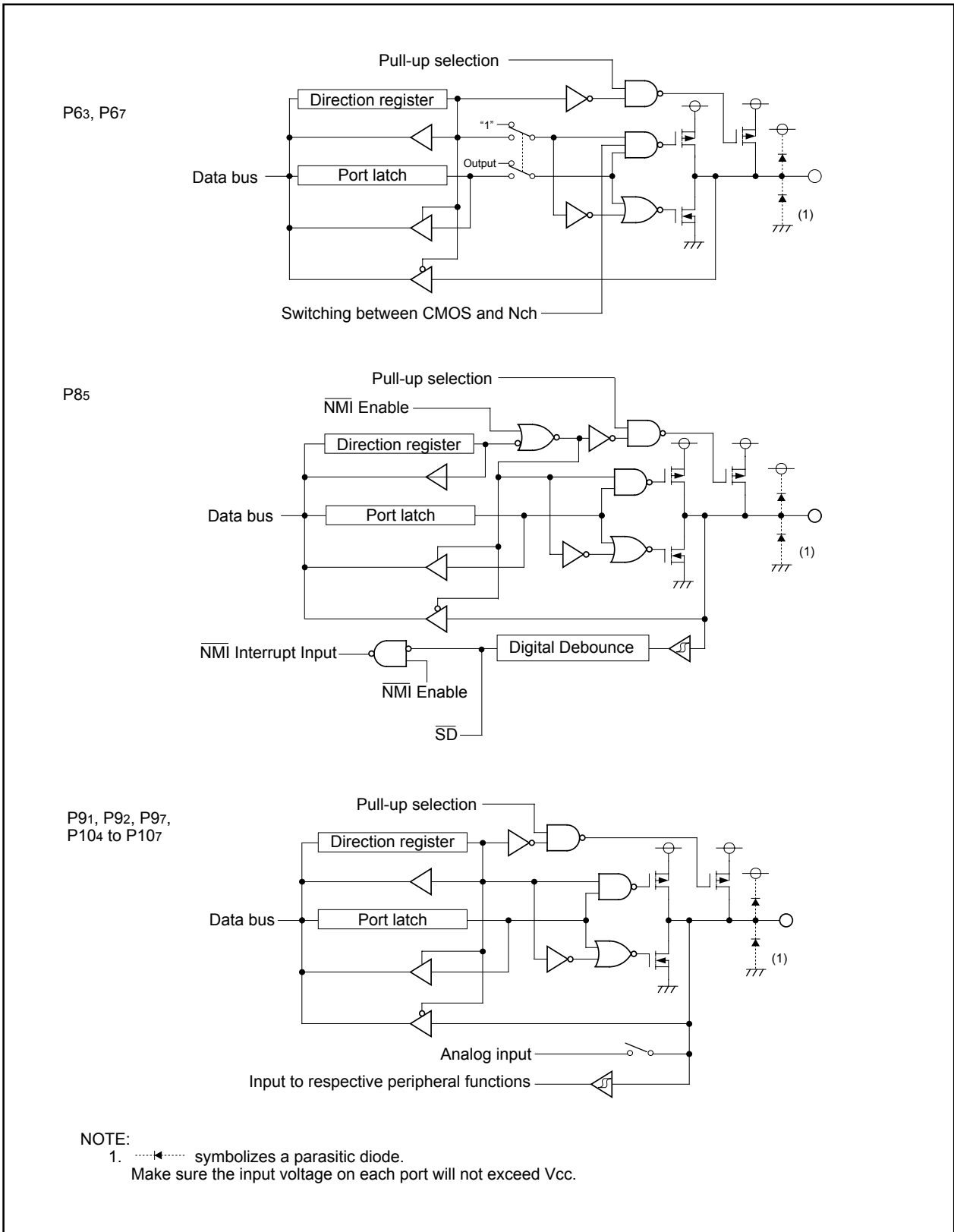


Figure 19.3 I/O Ports (3)

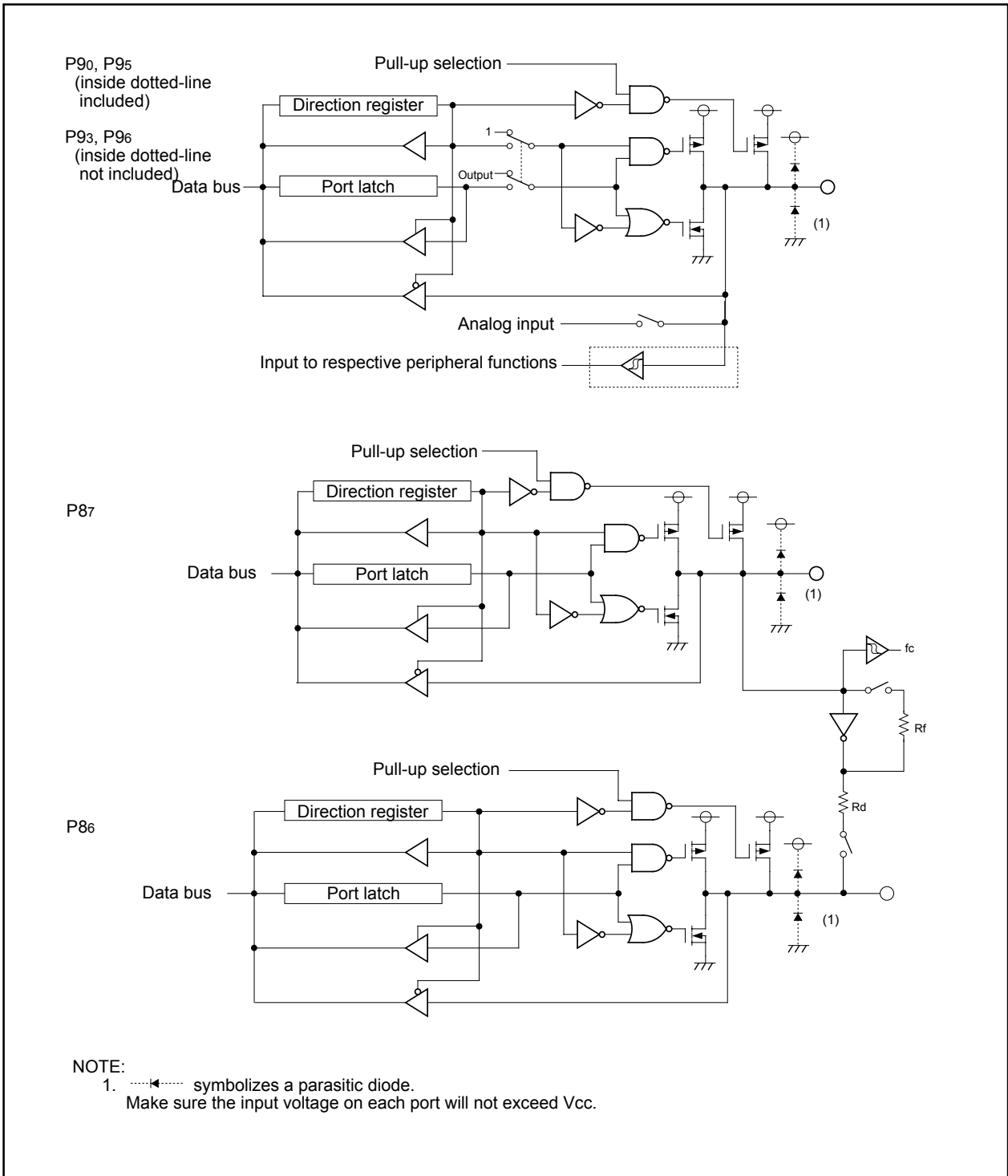


Figure 19.4 I/O Ports (4)

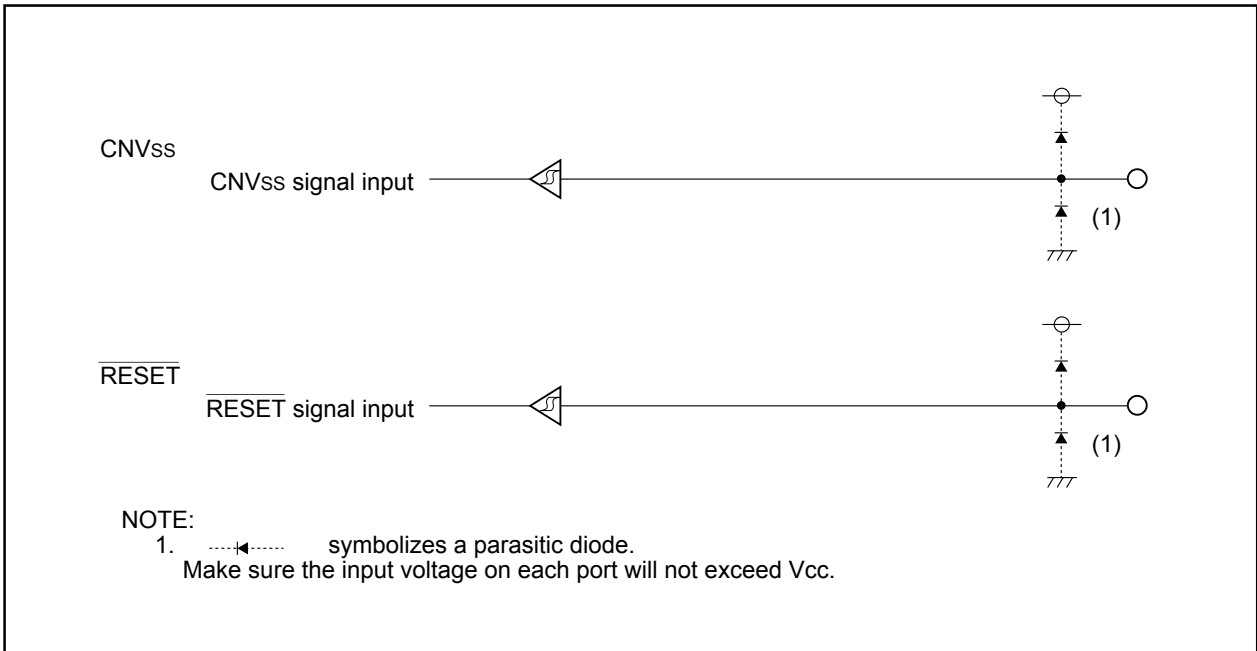


Figure 19.5 I/O Pins

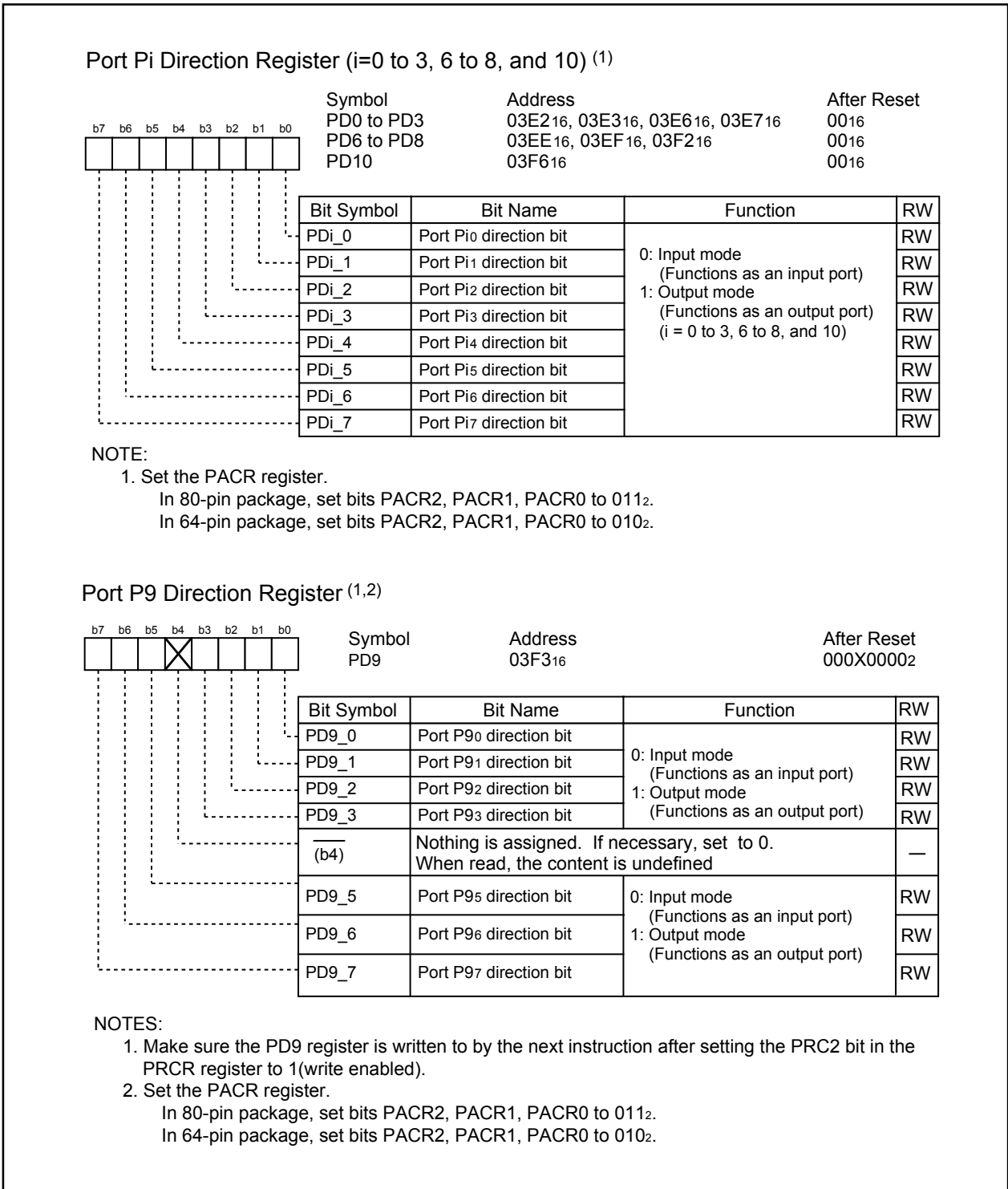


Figure 19.6 PD0 to PD3 and PD6 to PD10 Registers

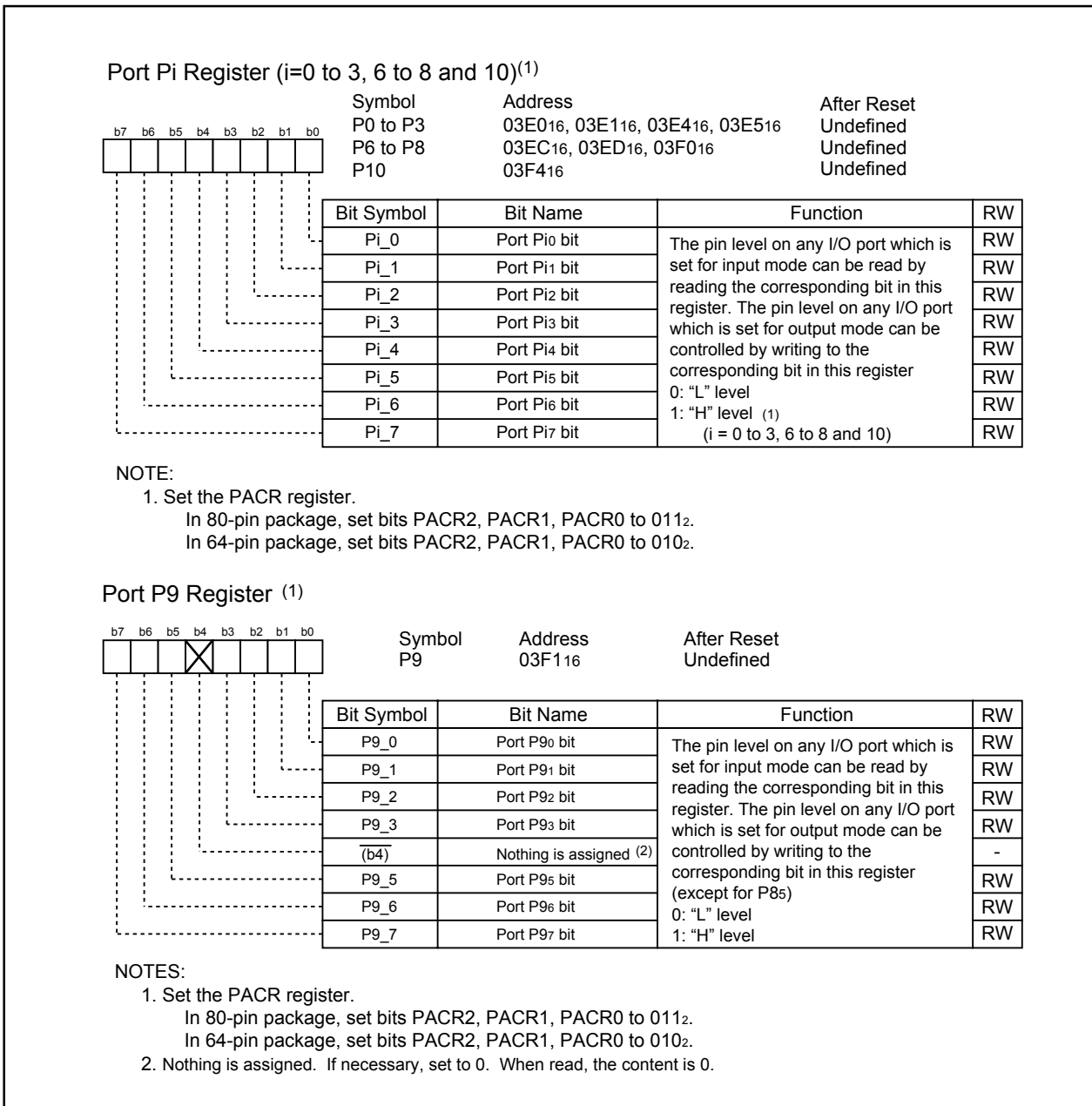


Figure 19.7 P0 to P3 and P6 to P10 Registers

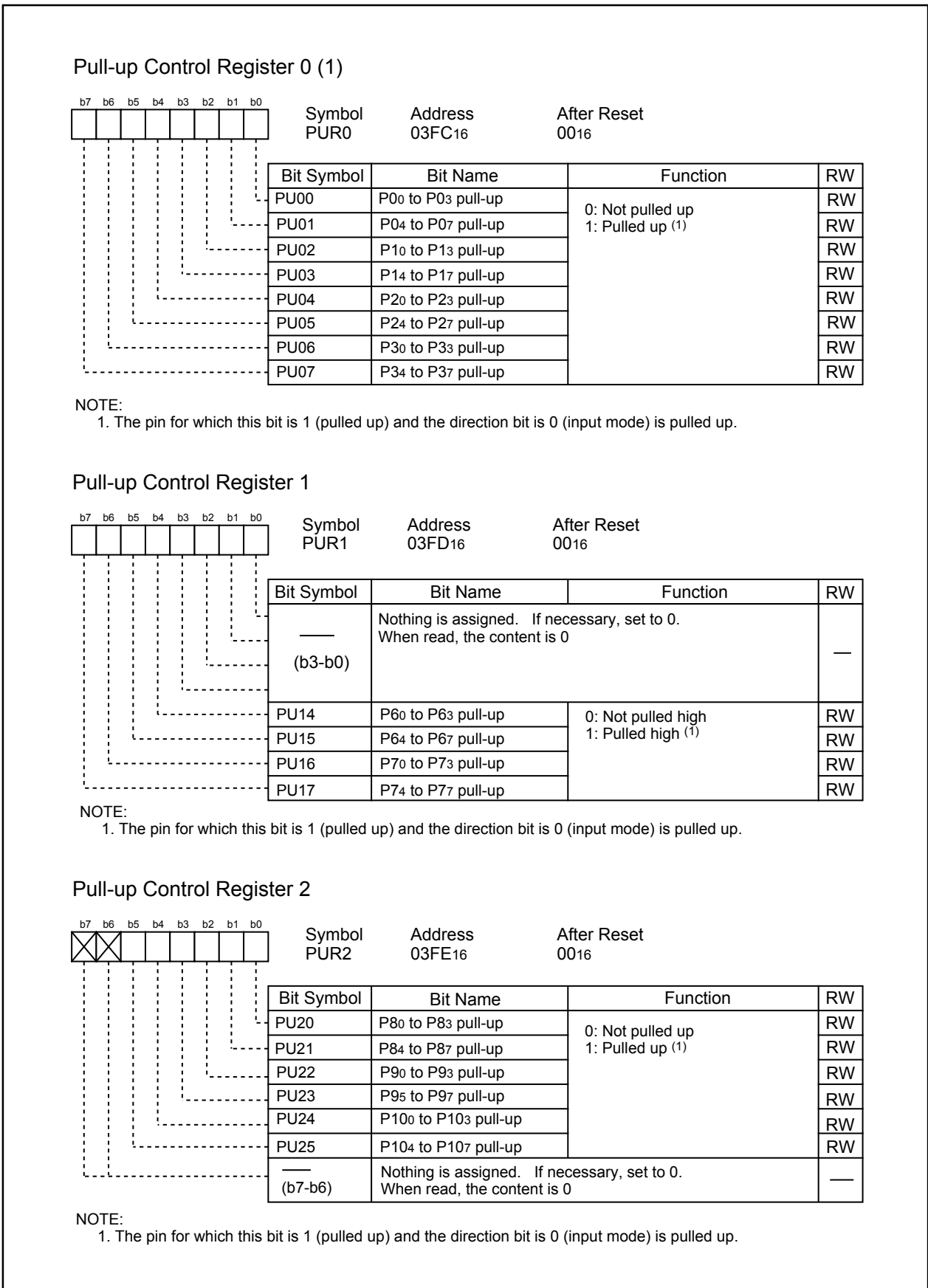


Figure 19.8 PUR0 to PUR2 Registers

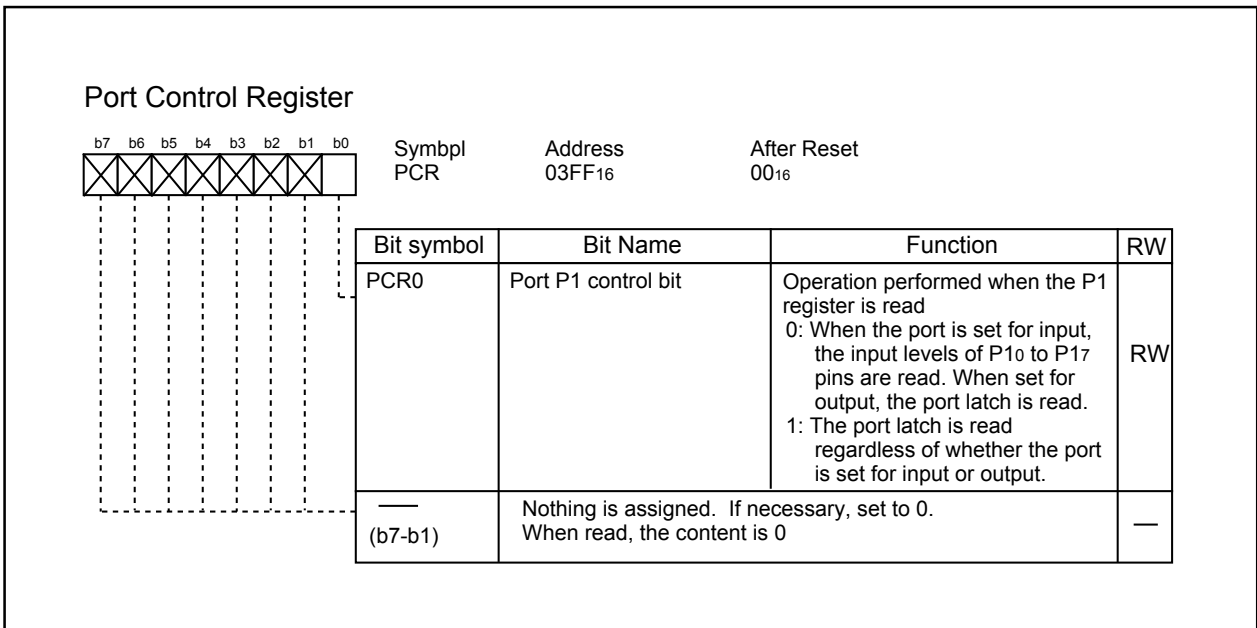


Figure 19.9 PCR Register

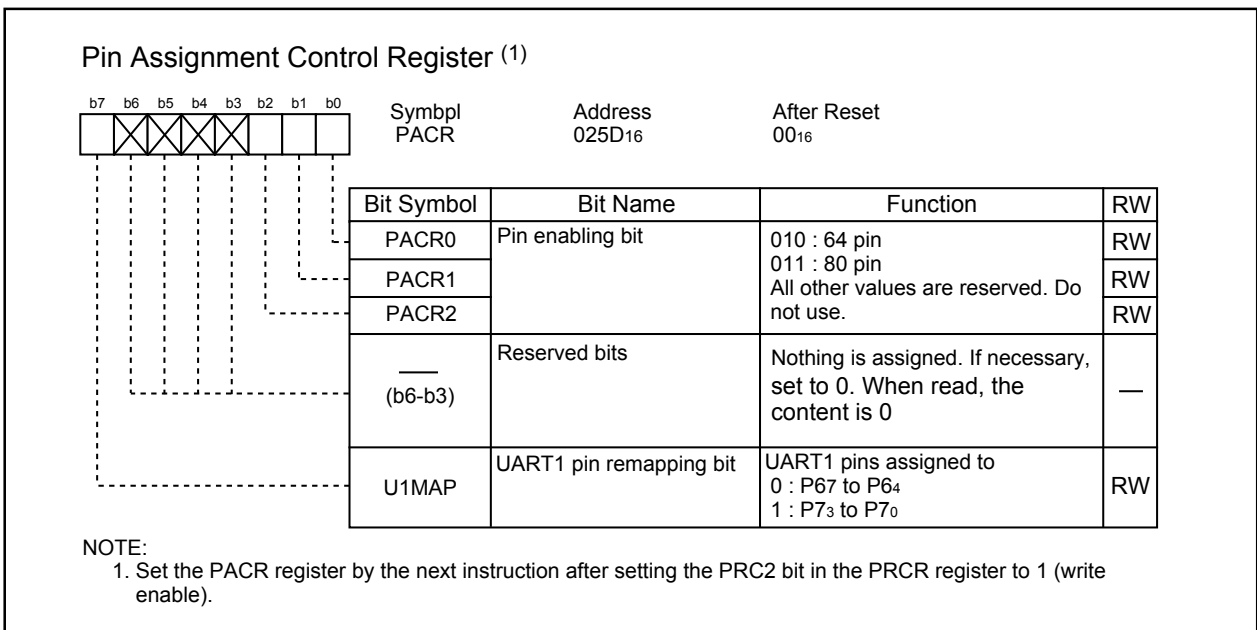


Figure 19.10 PACR Register

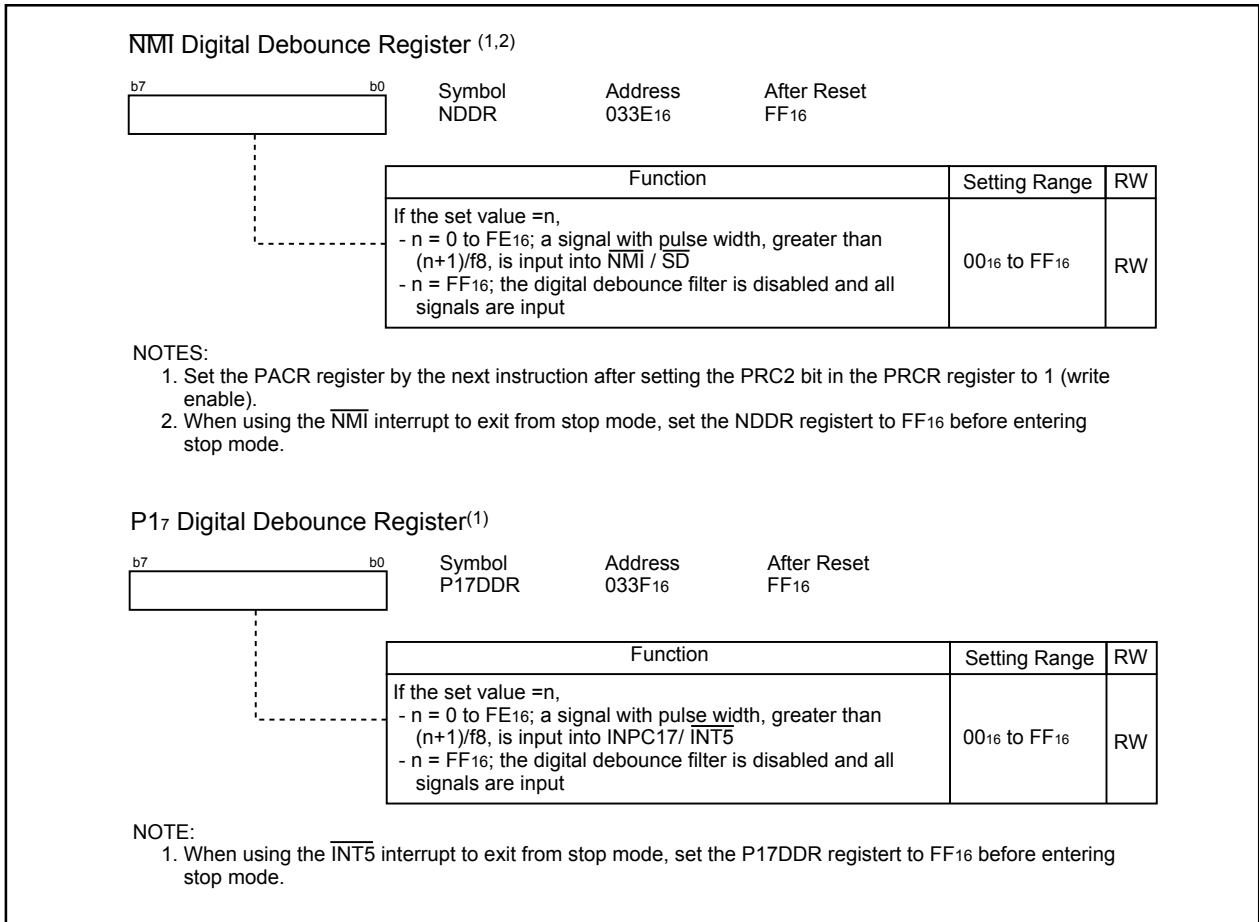


Figure 19.11 NDDR and P17DDR Registers

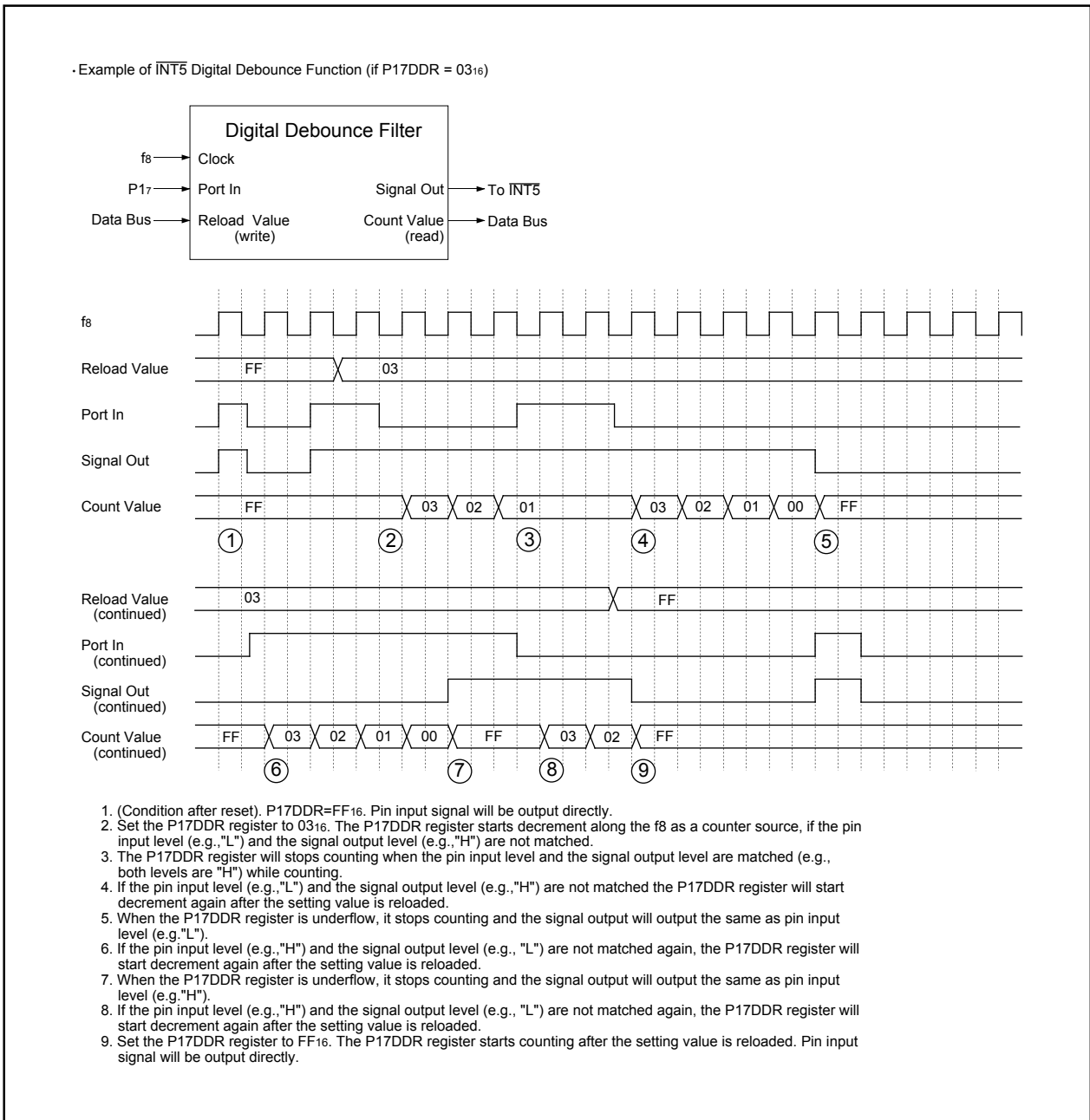


Figure 19.12 Functioning of Digital Debounce Filter

Table 19.1 Unassigned Pin Handling in Single-chip Mode

| Pin Name | Setting |
|---------------------------|--|
| Ports P0 to P3, P6 to P10 | Enter input mode and connect each pin to Vss via a resistor (pull-down); or enter output mode and leave the pins open ^(1,2,4) |
| XOUT | Leave pin open ⁽³⁾ |
| XIN | Connect pin to Vcc via a resistor (pull-up) ⁽⁵⁾ |
| AVCC | Connect pin to Vcc |
| AVSS, VREF | Connect pin to Vss |

NOTES:

1. If the port enters output mode and is left open, it is in input mode before output mode is entered by program after reset. While the port is in input mode, voltage level on the pins is indeterminate and power consumption may increase. Direction register setting may be changed by noise or failure caused by noise. Configure direction register settings regularly to increase the reliability of the program.
2. Use the shortest possible wiring to connect the MCU pins to unassigned pins (within 2 cm).
3. When the external clock is applied to the XIN pin, set the pin as written above.
4. In the 64-pin package, set bits PACR2, PACR1, and PACR0 in the PACR register to 010₂. In the 80-pin package, set bits PACR2, PACR1, and PACR0 to 011₂.
5. When the main clock oscillation is not used, set the CM05 bit in the CM0 register to 1 (main clock stops) to reduce power consumption.

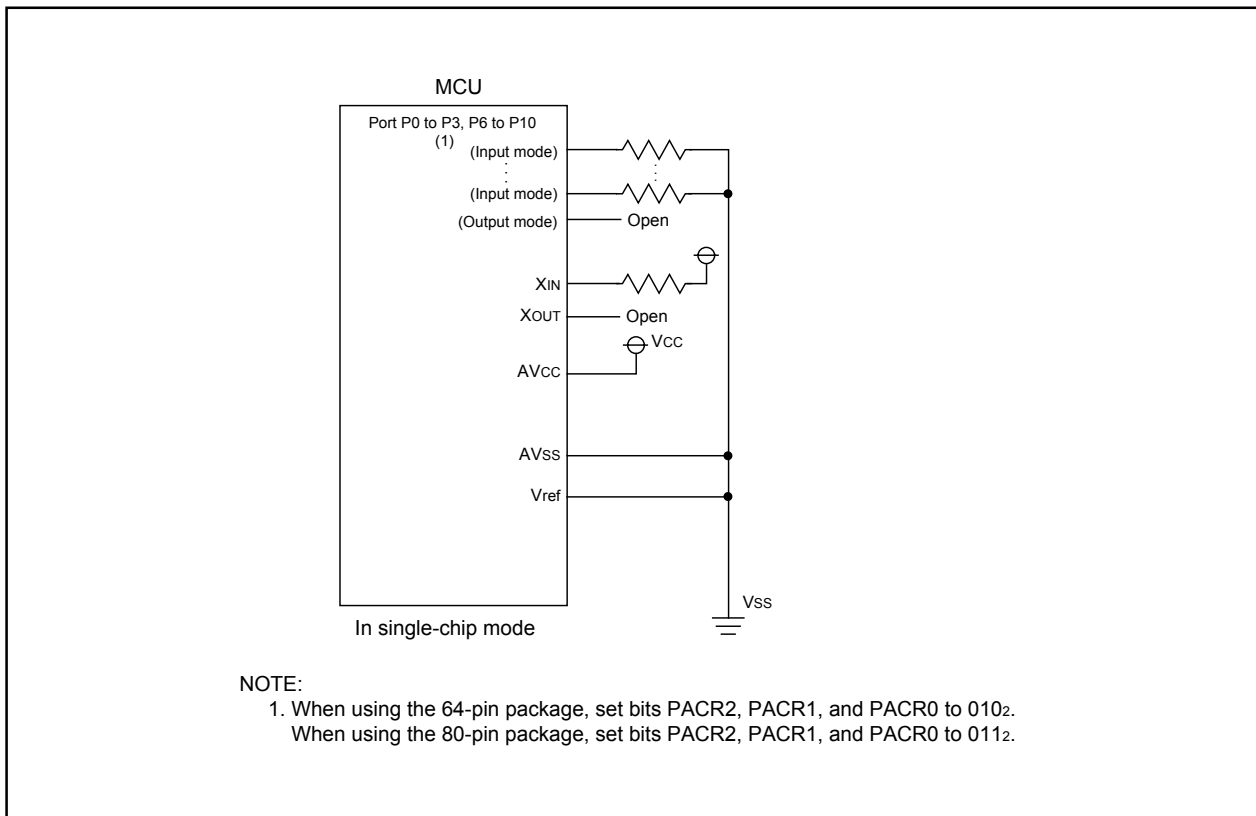


Figure 19.13 Unassigned Pin Handling

20. Flash Memory Version

20.1 Flash Memory Performance

In the flash memory version, rewrite operation to the flash memory can be performed in four modes: CPU rewrite mode, standard serial I/O mode, parallel I/O mode, and CAN I/O mode.

Table 20.1 lists specifications of the flash memory version. (Refer to **Table 1.1** or **Table 1.2** for the items not listed in **Table 20.1**.)

Table 20.1 Flash Memory Version Specifications

| Item | | Specification |
|--|---|--|
| Flash memory operating mode | | 4 modes (CPU rewrite, standard serial I/O, parallel I/O, CAN I/O) ⁽³⁾ |
| Erase block | | See Figures 20.1 to 20.3 Flash Memory Block Diagram |
| Program method | | In units of word |
| Erase method | | Block erase |
| Program, erase control method | | Program and erase controlled by software command |
| Protect method | | Blocks 0 to 5 are write protected by FMR16 bit. In addition, the block 0 and block 1 are write protected by FMR02 bit |
| Number of commands | | 5 commands |
| Program/Erase Endurance ⁽¹⁾ | Block 0 to 5 (program area) | 100 times 1,000 times (See Tables 1.6 to 1.8) |
| | Block A and B (data are) ⁽²⁾ | 100 times 10,000 times (See Tables 1.6 to 1.8) |
| Data Retention | | 20 years (Topr = 55°C) |
| ROM code protection | | Parallel I/O, standard serial I/O, and CAN I/O modes are supported. |

NOTES:

1. Program and erase endurance definition

Program and erase endurance are the erase endurance of each block. If the program and erase endurance are n times (n=100,1000,10000), each block can be erased n times. For example, if a 2-Kbyte block A is erased after writing 1 word data 1024 times, each to different addresses, this is counted as one program and erasure.

However, data cannot be written to the same address more than once without erasing the block. (Rewrite disabled)

2. To use the limited number of erasure efficiently, write to unused address within the block instead of rewrite. Erase block only after all possible address are used. For example, an 8-word program can be written 128 times before erase is necessary. Maintaining an equal number of erasure between Block A and B will also improve efficiency. We recommend keeping track of the number of times erasure is used.

3. The M16C/29 Group, T-ver./V-ver. does not support the CAN I/O mode.

Table 20.2. Flash Memory Rewrite Modes Overview

| Flash memory rewrite mode | CPU rewrite mode | Standard serial I/O mode | Parallel I/O mode | CAN I/O mode |
|------------------------------|---|---|---|---|
| Function | The user ROM area is rewritten when the CPU executes software command from the CPU. EW0 mode: Rewrite in area other than flash memory EW1 mode: Rewrite in flash memory | The user ROM area is rewritten using a dedicated serial programmer. Standard serial I/O mode 1: Clock synchronous serial I/O Standard serial I/O mode 2: UART | The user ROM areas are rewritten using a dedicated parallel programmer. | The user ROM areas is rewritten using a dedicated CAN programmer. |
| Areas which can be rewritten | User ROM area | User ROM area | User ROM area | User ROM area |
| Operation mode | Single chip mode | Boot mode | Parallel I/O mode | Boot mode |
| ROM programmer | None | Serial programmer | Parallel programmer | CAN programmer |

20.1.1 Boot Mode

The MCU enters boot mode when a hardware reset is performed while a high-level ("H") signal is applied to pins CNVss and P86 or while an "H" signal is applied to pins CNVss and P16 and a low-level ("L") signal is applied to the P85. A program in the boot ROM area is executed.

The boot ROM area is reserved. The boot ROM area stores the rewrite control program for a standard serial I/O mode before shipping. Do not rewrite the boot ROM area.

20.2 Memory Map

The flash memory contains the user ROM area and the boot ROM area (reserved area). **Figures 20.1 to 20.3** show a block diagram of the flash memory. The user ROM area has space to store the MCU operation program in single-chip mode and two 2-Kbyte spaces: the block A and B.

The user ROM area is divided into several blocks. The user ROM area can be rewritten in CPU rewrite, standard serial I/O, parallel I/O, or CAN I/O mode.

However, to rewrite program in block 0 and 1 in CPU rewrite mode, set the FMR02 bit in the FMR0 register to 1 (block 0, 1 rewrite enabled) and the FMR16 bit in the FMR1 register to 1 (blocks 0 to 5 rewrite enabled). Also, to rewrite program in blocks 2 to 5 in CPU rewrite mode, set the FMR16 bit in the FMR1 register to 1 (blocks 0 to 5 rewrite enabled). When the PM10 bit in the PM1 register is set to 1 (data space access enabled), blocks A and B can be available for use.

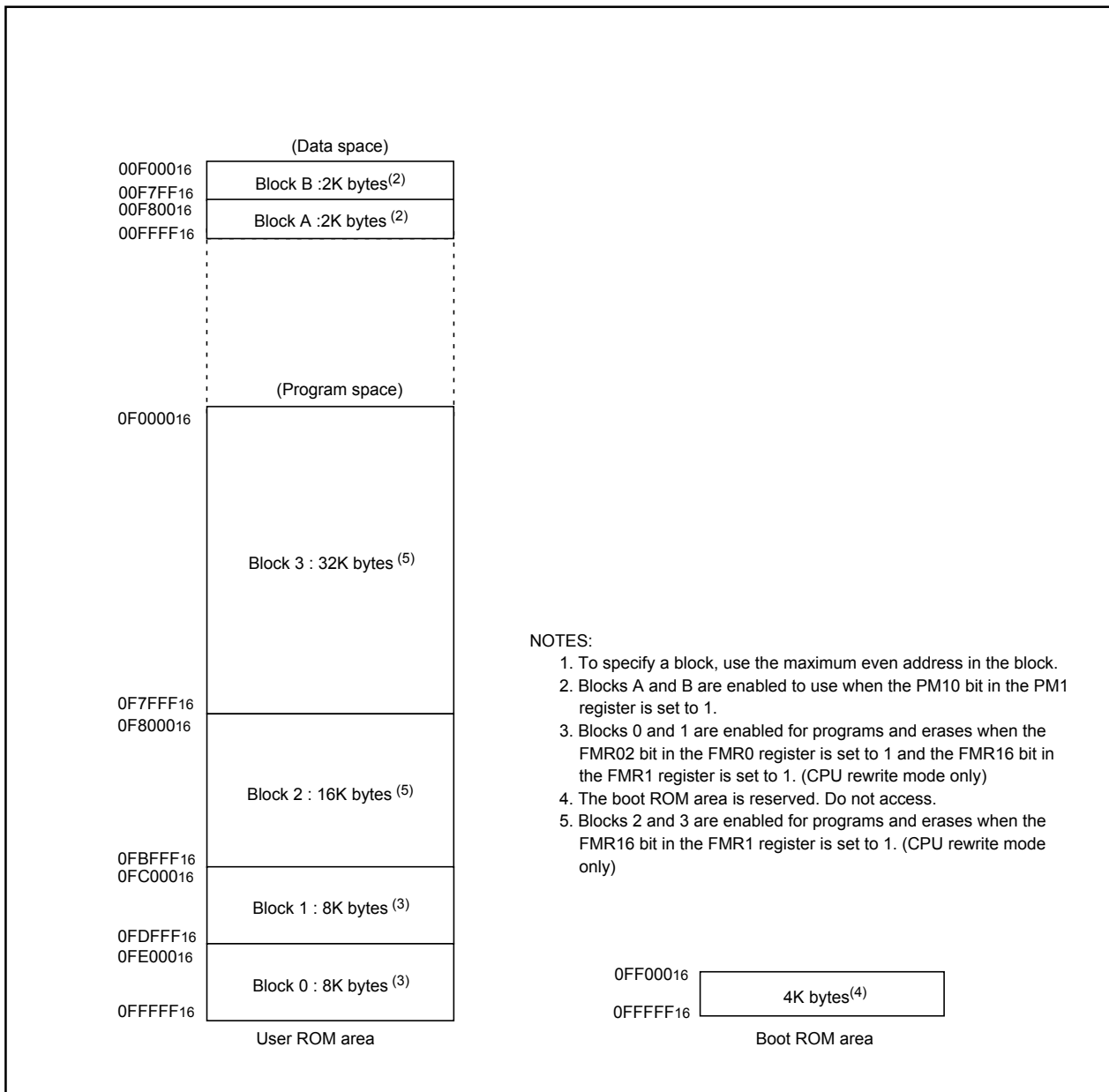


Figure 20.1 Flash Memory Block Diagram (ROM capacity 64 Kbytes)

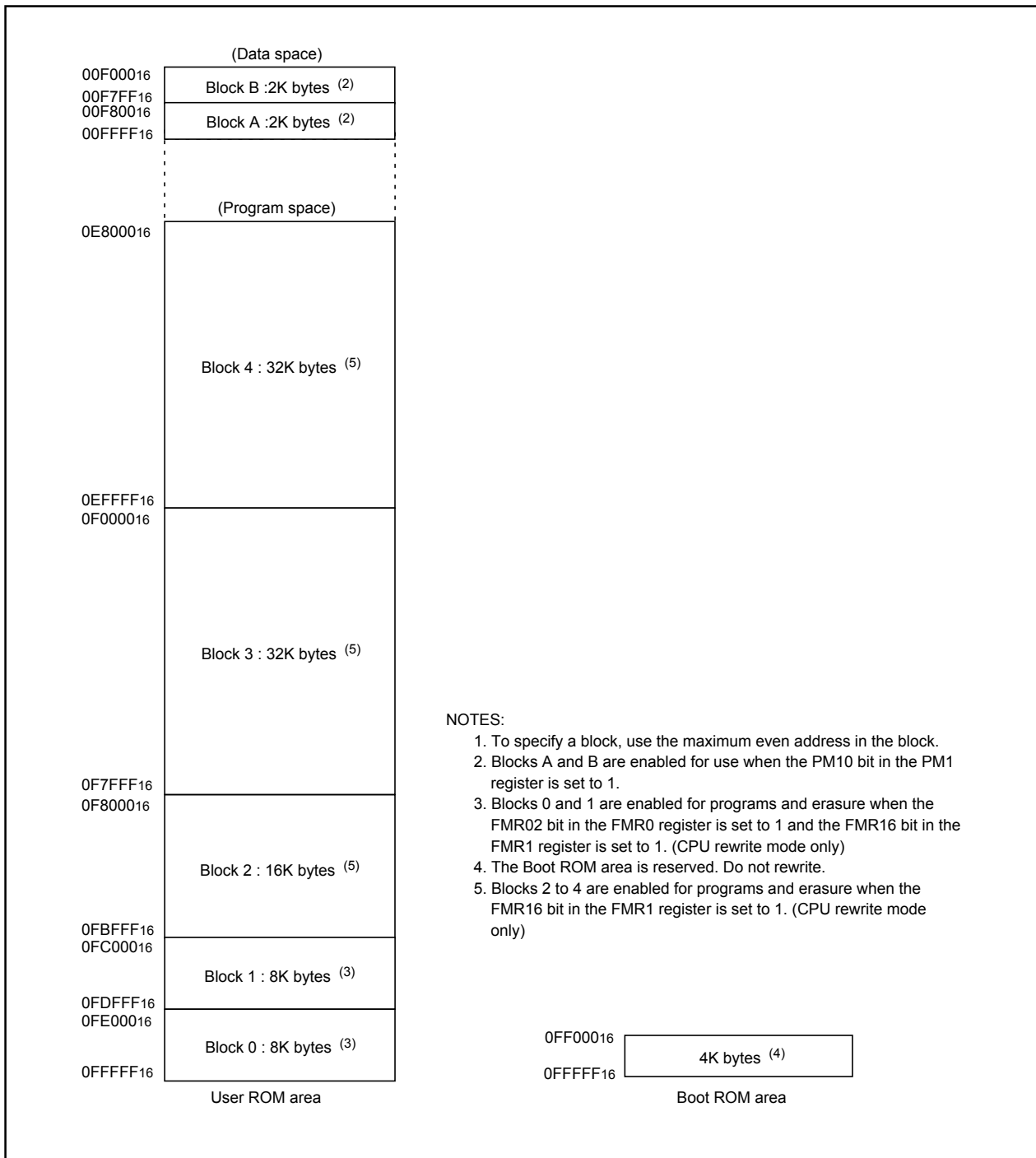


Figure 20.2 Flash Memory Block Diagram (ROM capacity 96 Kbytes)

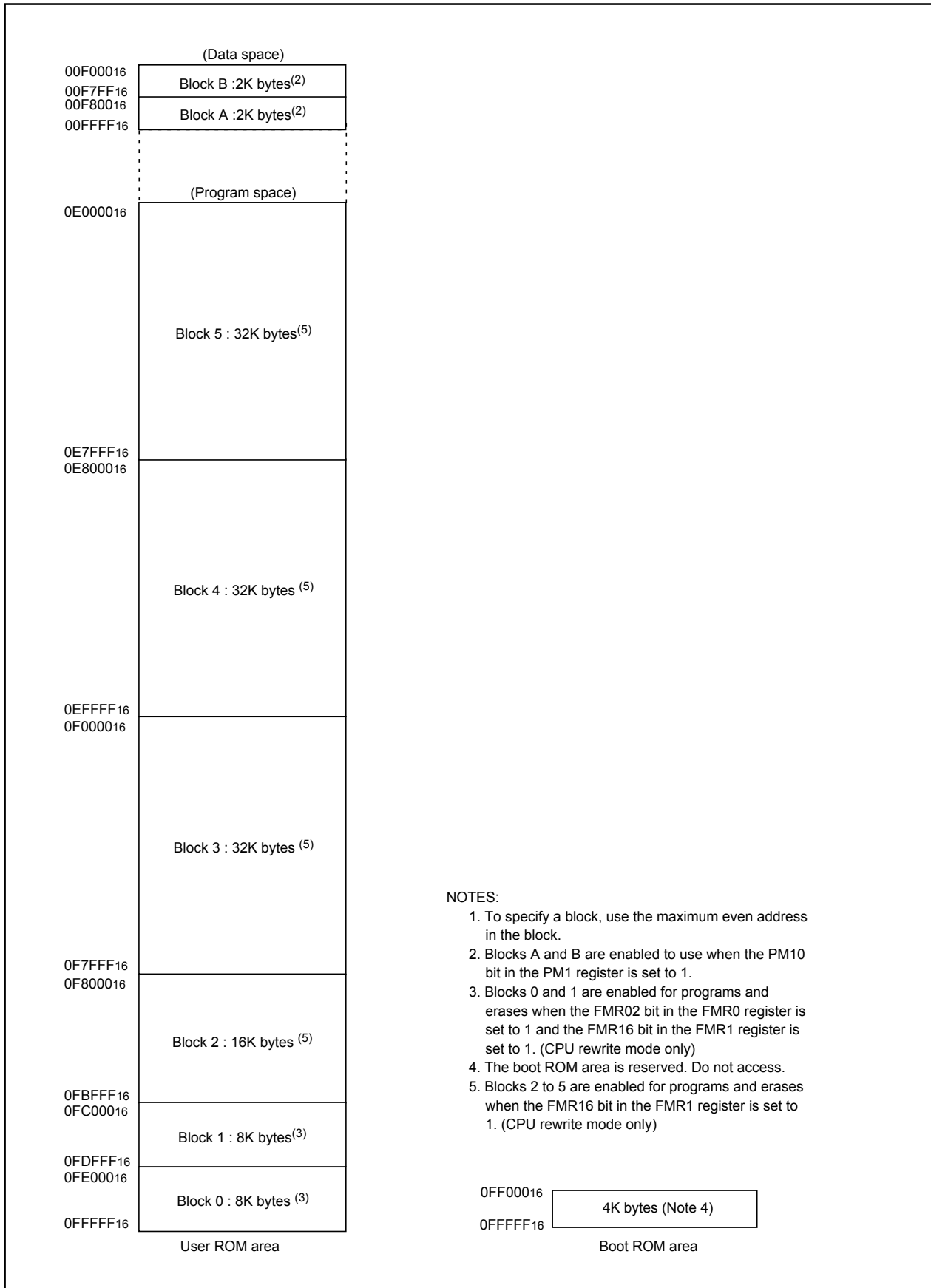


Figure 20.3 Flash Memory Block Diagram (ROM capacity 128 Kbytes)

20.3 Functions To Prevent Flash Memory from Rewriting

The flash memory has a built-in ROM code protect function for parallel I/O mode and a built-in ID code check function for standard input/output mode to prevent the flash memory from reading or rewriting.

20.3.1 ROM Code Protect Function

The ROM code protect function disables reading or changing the contents of the on-chip flash memory in parallel I/O mode. **Figure 20.4** shows the ROMCP address. The ROMCP address is located in a user ROM area. To enable ROM code protect, set the ROMCP1 bit to “002”, “012”, or “102” and set the bit 5 to bit 0 to “1111112”.

To cancel ROM code protect, erase the block including the the ROMCP register in CPU rewrite mode or standard serial I/O mode.

20.3.2 ID Code Check Function

Use the ID code check function in standard serial input/output mode. Unless the flash memory is blank, the ID code sent from the programmer and the 7-byte ID code written in the flash memory are compared for 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, starting with the first byte, into addresses, 0FFFD_{F16}, 0FFFE₃₁₆, 0FFFE_{B16}, 0FFFE_{F16}, 0FFFF₃₁₆, 0FFFF₇₁₆, and 0FFFF_{B16}. The flash memory must have a program with the ID code set in these addresses.

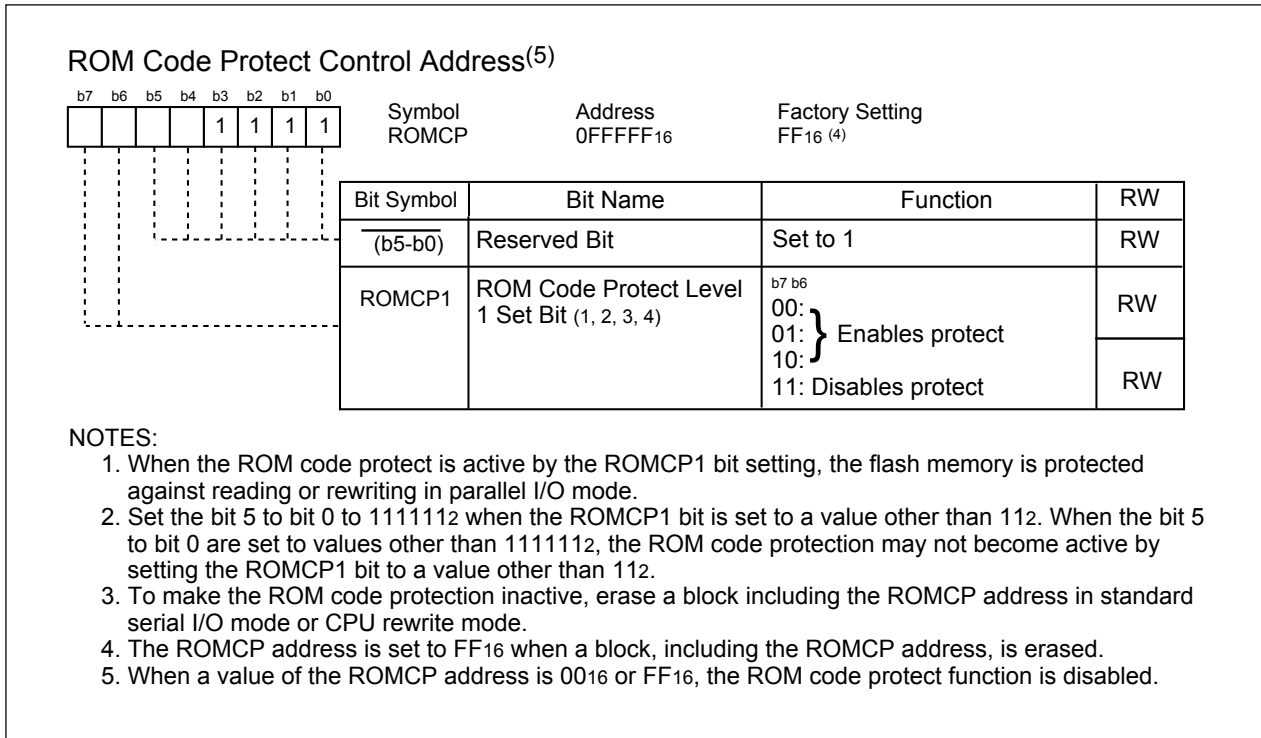


Figure 20.4 ROMCP Address

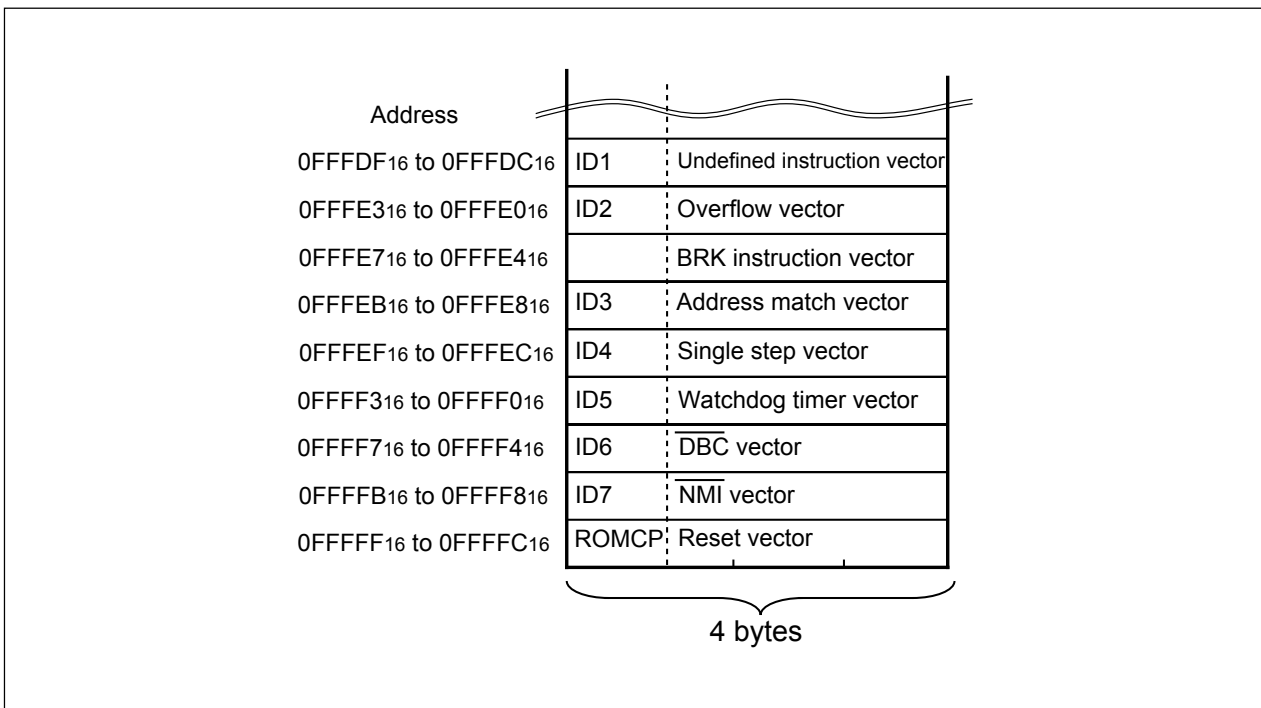


Figure 20.5 Address for ID Code Stored

20.4 CPU Rewrite Mode

In CPU rewrite mode, the user ROM area can be rewritten when the CPU executes software commands. The user ROM area can be rewritten with MCU mounted on a board without using the ROM writer. The program and block erase commands are executed only in the user ROM area.

When the interrupt requests are generated during the erase operation in CPU rewrite mode, the flash memory offers an erase suspend function to suspend the erase operation and process the interrupt operation. During the erase suspend function is operated, the user ROM area can be read by program.

Erase-write(EW) 0 mode and erase-write 1 mode are provided as CPU rewrite mode. **Table 20.3** lists differences between EW mode 0 and EW mode 1. One wait is required for the CPU erase-write control.

Table 20.3 EW Mode 0 and EW Mode 1

| Item | EW mode 0 | EW mode 1 |
|--|---|--|
| Operation mode | Single chip mode | Single chip mode |
| Areas in which a rewrite control program can be located | User ROM area | User ROM area |
| Areas where rewrite control program can be executed ⁽²⁾ | The rewrite control program must be transferred to any other than the flash memory (e.g., RAM) before being executed | The rewrite control program can be executed in the user ROM area |
| Areas which can be rewritten | User ROM area | User ROM area However, this excludes blocks with the rewrite control program |
| Software command Restrictions | None | <ul style="list-style-type: none"> • Program, block erase command Cannot be executed in a block having the rewrite control program • Read Status Register command Cannot be executed |
| Mode after programming or erasing | Read Status Register Mode | Read Array mode |
| CPU state during auto-write and auto-erase | Operating | In a hold state (I/O ports retain the state before the command is executed ⁽¹⁾) |
| Flash memory status detection | <ul style="list-style-type: none"> • Read the FMR00, FMR06, and FMR07 bits in the FMR0 register by program • Execute the read status register command to read bits SR7, SR5, and SR4. | Read the FMR00, FMR06, and FMR07 bits in the FMR0 register by program |
| Condition for transferring to erase-suspend ⁽³⁾ | Set bits FMR40 and FMR41 in the FMR4 register to 1 by program. | The FMR40 bit in the FMR4 register is set to 1 and the interrupt request of an acknowledged interrupt is generated |

NOTES:

1. Do not generate a DMA transfer.
2. Block 1 and Block 0 are enabled for rewrite by setting FMR02 bit in the FMR0 register to 1 and setting FMR16 bit in the FMR1 register to 1. Block 2 to Block 5 are enabled for rewrite by setting FMR16 bit in the FMR1 register to 1.
3. The time, until entering erase suspend and reading flash is enabled, is maximum $t_d(SR-ES)$ after satisfying the conditions.

20.4.1 EW Mode 0

The MCU enters CPU rewrite mode by setting the FMR01 bit in the FMR0 register to 1 (CPU rewrite mode enabled) and is ready to accept software commands. EW mode 0 is selected by setting the FMR11 bit in the FMR1 register to 0.

To set the FMR01 bit to 1, set to 1 after first writing 0. The software commands control programming and erasing. The FMR0 register or the status register indicates whether a programming or erasing operations is completed.

When entering the erase-suspend during the auto-erasing, set the FMR40 bit to 1 (erase-suspend enabled) and the FMR41 bit to 1 (suspend request). After waiting for td(SR-ES) and verifying the FMR46 bit is set to 1 (auto-erase stop), access to the user ROM area. When setting the FMR41 bit to 0 (erase restart), auto-erasing is restarted.

20.4.2 EW Mode 1

EW mode 1 is selected by setting the FMR11 bit to 1 after the FMR01 bit is set to 1 (set to 1 after first writing 0).

The FMR0 register indicates whether or not a programming or an erasing operation is completed. Read status register cannot be read in EW mode 1.

When an erase/program command is initiated, the CPU halts all program execution until the command operation is completed or erase-suspend request is generated.

When enabling an erase-suspend function, set the FMR40 bit to 1 (erase suspend enabled) and execute block erase commands. Also, the interrupt to transfer to erase-suspend must be set enabled preliminarily. When entering erase-suspend after td(SR-ES) from an interrupt is requested, interrupts can be accepted.

When an interrupt request is generated, the FMR41 bit is automatically set to 1 (suspend request) and an auto-erasing is suspended. If an auto-erasing has not completed (when the FMR00 bit is 0) after an interrupt process is completed, set the FMR41 bit to 0 (erase restart) and execute block erase commands again.

20.5 Register Description

Figure 20.6 shows the flash memory control register 0 and flash memory control register 1. **Figure 20.7** shows the flash memory control register 4.

20.5.1 Flash Memory Control Register 0 (FMR0)

•FMR00 Bit

The FMR00 bit indicates the operating state of the flash memory. Its value is 0 while the program, erase, or erase-suspend command is being executed, otherwise, it is 1.

•FMR01 Bit

The MCU can accept commands when the FMR01 bit is set to 1 (CPU rewrite mode). To set the FMR01 bit to 1, first set it to 0 and then 1. The FMR01 bit is set to 0 only by writing 0.

•FMR02 Bit

The combined settings of bits FMR02 and FMR16 enable program and erase in the user ROM area. See **Table 20.4** for setting details. To set the FMR02 bit to 1, first set it to 0 and then 1. The FMR02 bit is valid only when the FMR01 bit is set to 1 (CPU rewrite mode enable).

•FMSTP Bit

The FMSTP bit initializes the flash memory control circuits and minimizes power consumption in the flash memory. Access to the on-chip flash memory is disabled when the FMSTP bit is set to 1. Set the FMSTP bit by program in a space other than the flash memory.

Set the FMSTP bit to 1 if one of the following occurs:

- A flash memory access error occurs during erasing or programming in EW mode 0 (FMR00 bit does not switch back to 1 (ready)).
- Low-power consumption mode or on-chip oscillator low-power consumption mode is entered.

Figure 20.10 shows a flow chart illustrating how to start and stop the flash memory before and after entering low power mode. Follow the procedure in this flow chart.

When entering stop or wait mode while the CPU rewrite mode is disabled, do not set the FMR0 register because the on-chip flash memory is automatically turned off and turned back on when exiting.

•FMR06 Bit

The FMR06 bit is a read-only bit indicating an auto-program operation state. The FMR06 bit is set to 1 when a program error occurs; otherwise, it is set to 0. For details, refer to **20.8.4 Full Status Check**.

•FMR07 Bit

The FMR07 bit is a read-only bit indicating an auto-erase operation status. The FMR07 bit is set to 1 when an erase error occurs; otherwise, it is set to 0. For details, refer to **20.8.4 Full Status Check**.

Figure 20.8 shows a EW mode 0 set/reset flowchart, **Figure 20.9** shows a EW mode 1 set/reset flowchart.

20.5.2 Flash Memory Control Register 1 (FMR1)

•FMR11 Bit

EW mode 1 is entered by setting the FMR11 bit to 1 (EW mode 1). The FMR11 bit is valid only when the FMR01 bit is set to 1.

•FMR16 Bit

The combined setting of bits FMR02 and FMR16 enables program and erase in the user ROM area. To set the FMR16 bit to 1, first set it to 0 and then 1. The FMR16 bit is valid only when the FMR01 bit is set to 1 (CPU rewrite mode enable).

•FMR17 Bit

If the FMR17 bit is set to 1 (with wait state), 1 wait state is inserted when blocks A and B are accessed, regardless of the content of the PM17 bit in the PM1 register. The PM17 bit setting is reflected to access other blocks and internal RAM, regardless of the FMR17 bit setting.

Set the FMR17 bit to 1 (with wait state) to rewrite more than 100 times (U7, U9).

Table 20.4 Protection using FMR16 and FMR02

| FMR16 | FMR02 | Block A, Block B | Block 0, Block 1 | other user block |
|-------|-------|------------------|------------------|------------------|
| 0 | 0 | write enabled | write disabled | write disabled |
| 0 | 1 | write enabled | write disabled | write disabled |
| 1 | 0 | write enabled | write disabled | write enabled |
| 1 | 1 | write enabled | write enabled | write enabled |

20.5.3 Flash Memory Control Register 4 (FMR4)

•FMR40 Bit

The erase-suspend function is enabled when the FMR40 bit is set to 1 (enabled).

•FMR41 Bit

When the FMR41 bit is set to 1 by program during auto-erasing in EW mode 0, erase-suspend mode is entered. In EW mode 1, the FMR41 bit is automatically set to 1 (suspend request) to enter erase-suspend mode when an enabled interrupt request is generated. Set the FMR41 bit to 0 (erase restart) to restart an auto-erasing operation.

•FMR46 Bit

The FMR46 bit is set to 0 during auto-erasing. It is set to 1 in erase-suspend mode.

Do not access to flash memory when the FMR46 bit is set to 0.

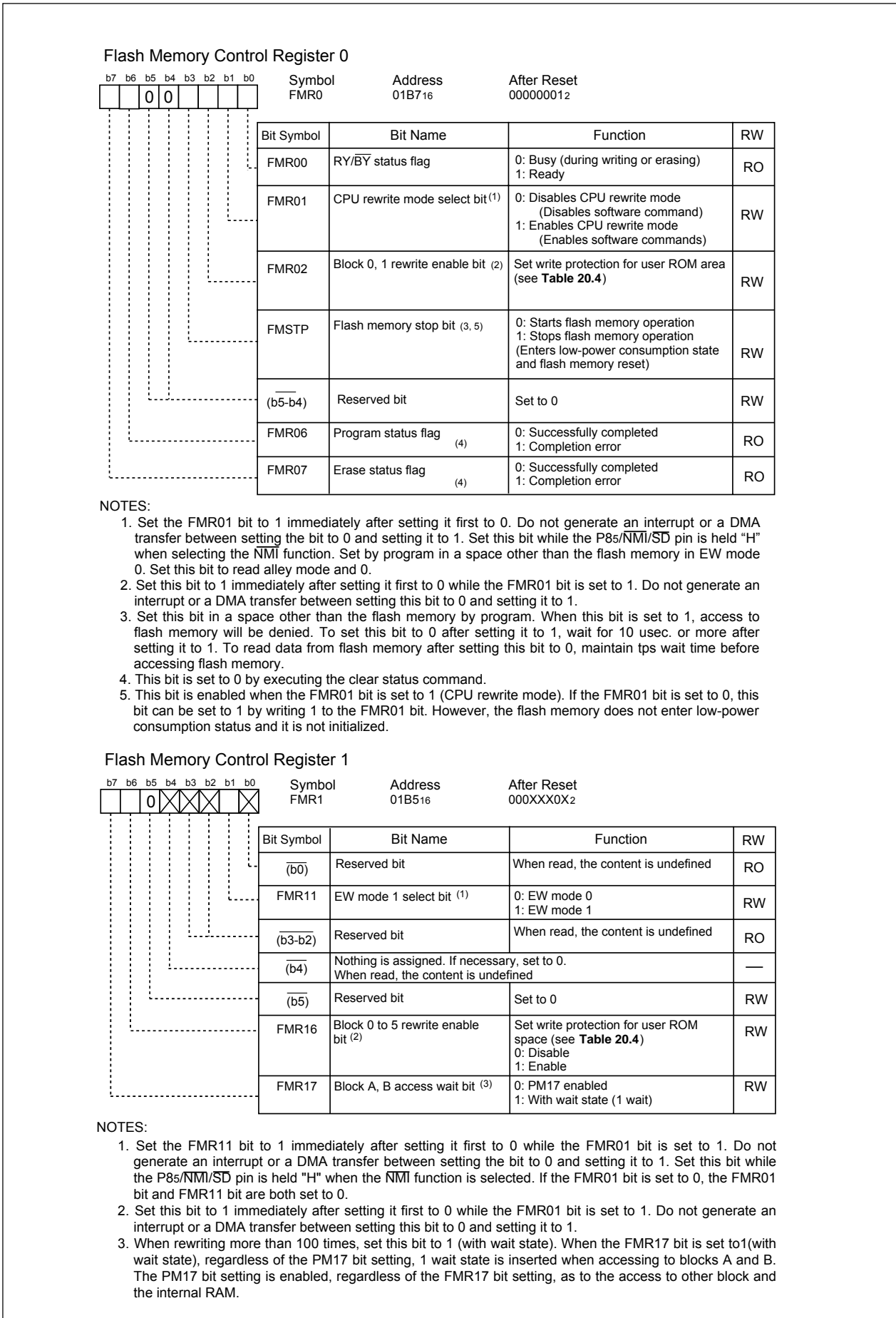


Figure 20.6 FMR0 and FMR1 Registers

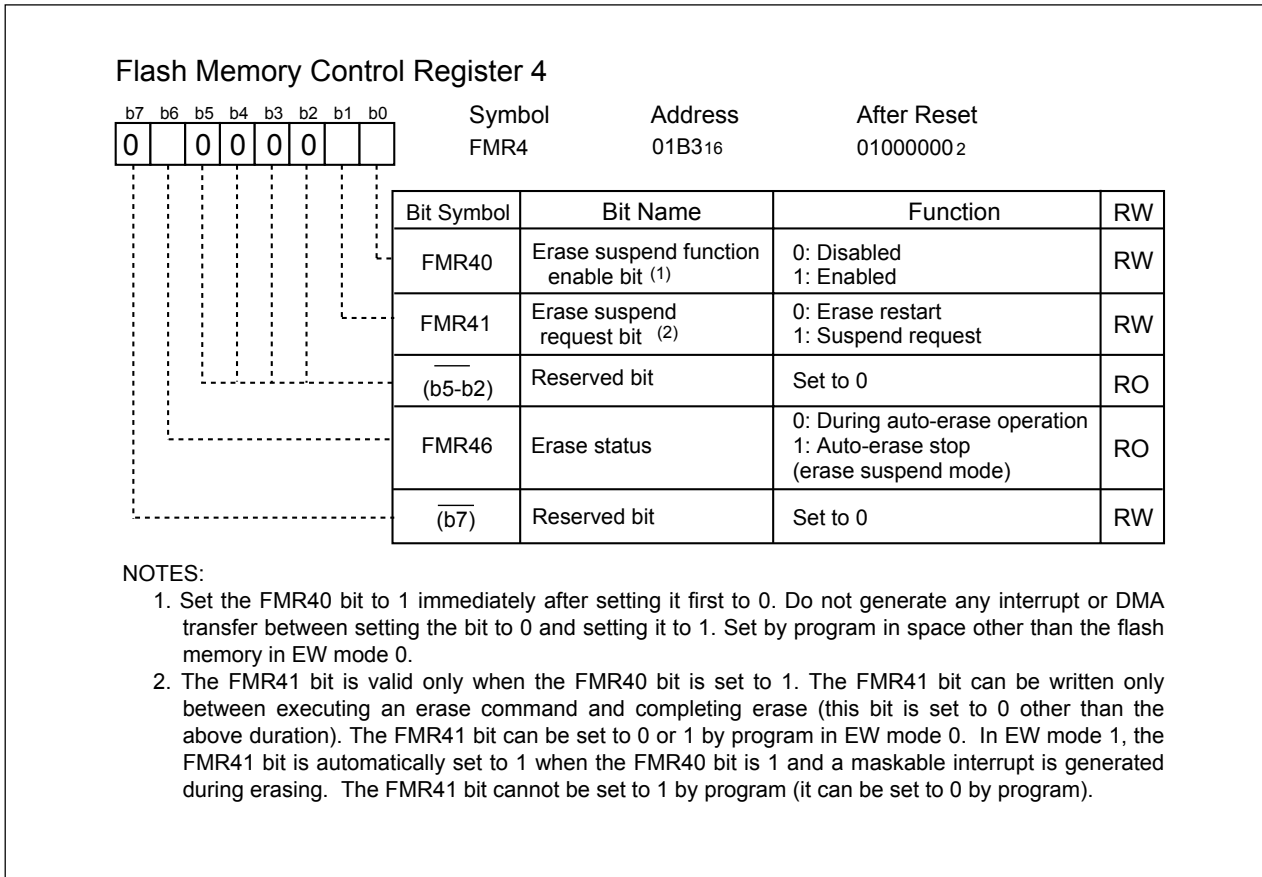


Figure 20.7 FMR4 Register

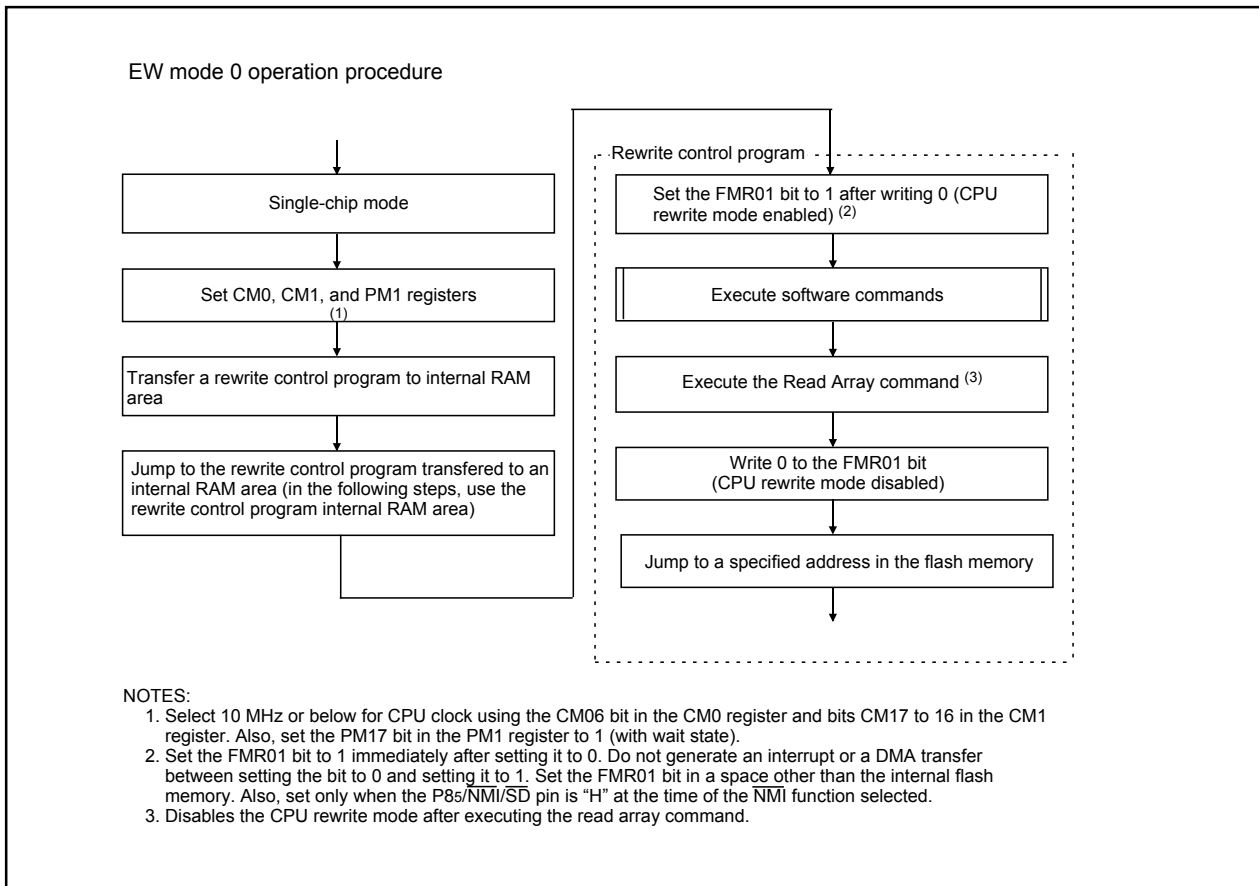


Figure 20.8 Setting and Resetting of EW Mode 0

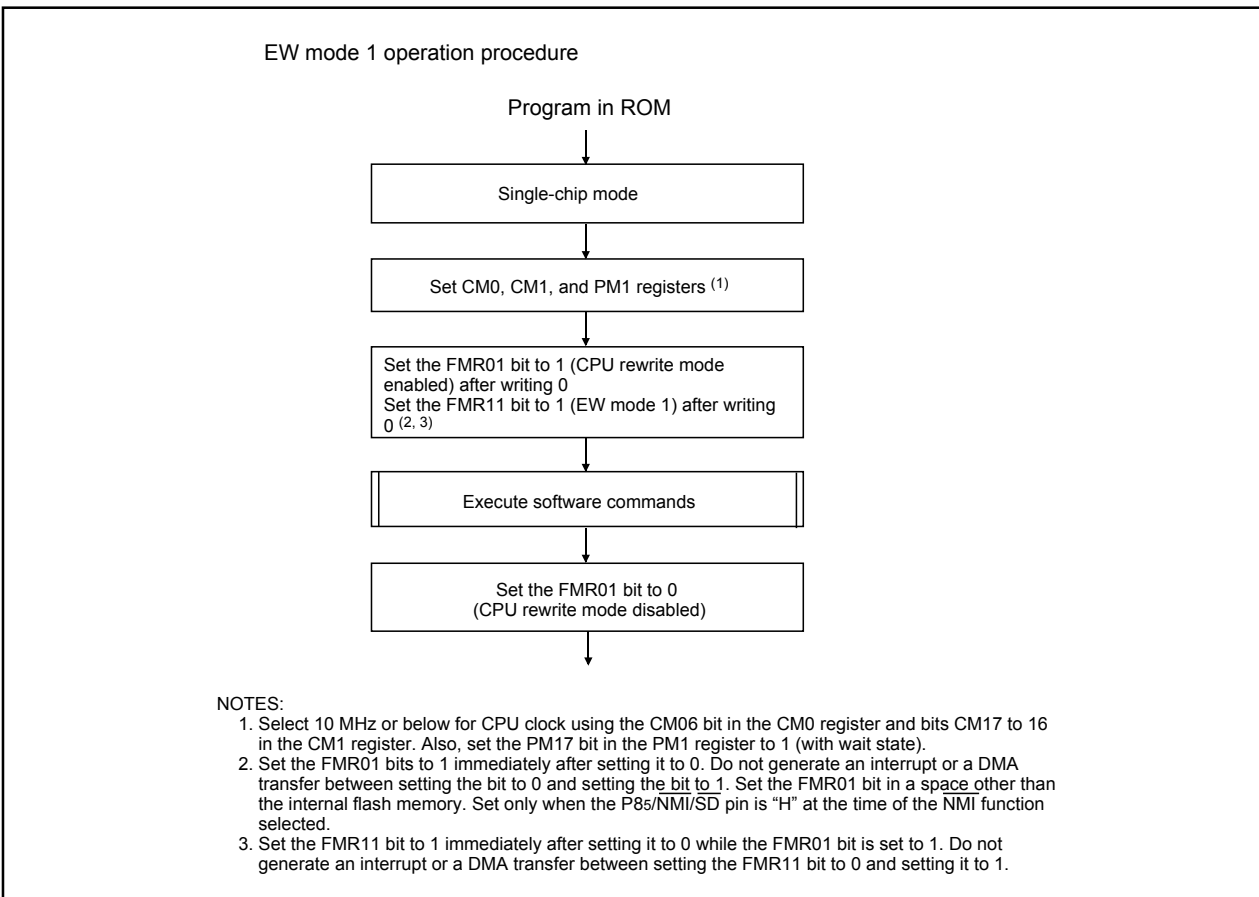


Figure 20.9 Setting and Resetting of EW Mode 1

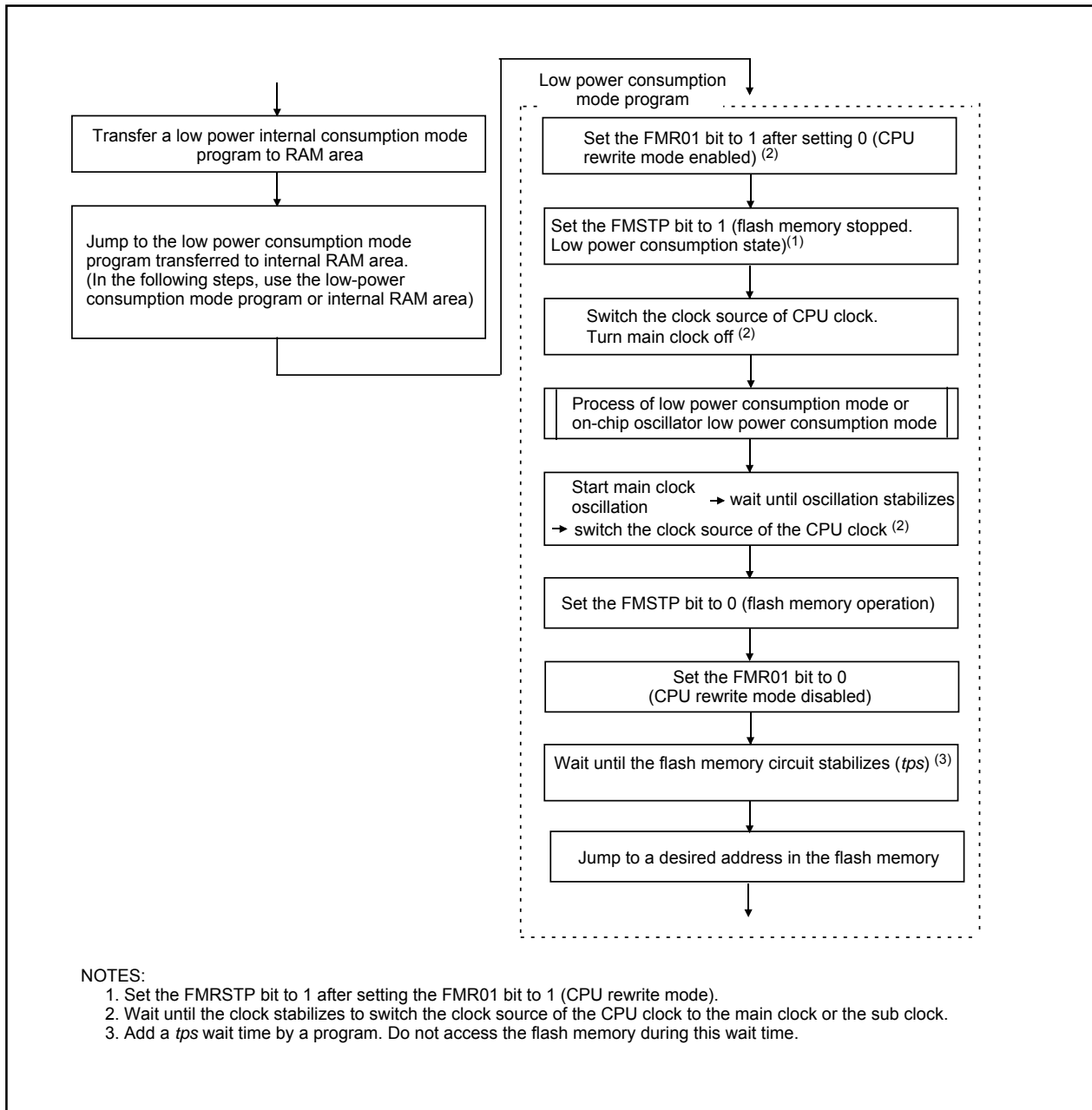


Figure 20.10 Processing Before and After Low Power Dissipation Mode

20.6 Precautions in CPU Rewrite Mode

Described below are the precautions to be observed when rewriting the flash memory in CPU rewrite mode.

20.6.1 Operation Speed

When the CPU clock source is the main clock, set the CPU clock frequency at 10 MHz or less with the CM06 bit in the CM0 register and bits CM17 and CM16 in the CM1 register, before entering CPU rewrite mode (EW mode 0 or EW mode 1). Also, when selecting f₃(ROC) of a on-chip oscillator as a CPU clock source, set bits ROCR3 and ROCR2 in the ROCR register to the CPU clock division rate at “divide-by-4” or “divide-by-8”, before entering CPU rewrite mode (EW mode 0 or EW mode 1).

In both cases, set the PM17 bit in the PM1 register to 1 (with wait state).

20.6.2 Prohibited Instructions

The following instructions cannot be used in EW mode 0 because the CPU tries to read data in the flash memory: UND instruction, INTO instruction, JMPS instruction, JSRS instruction, and BRK instruction

20.6.3 Interrupts

EW Mode 0

- To use interrupts having vectors in a relocatable vector table, the vectors must be relocated to the RAM area.
- The $\overline{\text{NMI}}$ and watchdog timer interrupts are available since registers FMR0 and FMR1 are forcibly reset when either interrupt occurs. However, the interrupt program, which allocates the jump addresses for each interrupt routine to the fixed vector table, is needed. Flash memory rewrite operation is aborted when the $\overline{\text{NMI}}$ or watchdog timer interrupt occurs. Set the FMR01 bit to 1 and execute the rewrite and erase program again after exiting the interrupt routine.
- The address match interrupt can not be used since the CPU tries to read data in the flash memory.

EW Mode 1

- Do not acknowledge any interrupts with vectors in the relocatable vector table or the address match interrupt during the auto program period or auto erase period with erase-suspend function disabled.

20.6.4 How to Access

To set bit FMR01, FMR02, FMR11 or FMR16 to 1, write 1 immediately after setting to 0. Do not generate an interrupt or a DMA transfer between the instruction to set the bit to 0 and the instruction to set it to 1. When the $\overline{\text{NMI}}$ function is selected, set the bit while an “H” signal is applied to the P85/ $\overline{\text{NMI}}/\overline{\text{SD}}$ pin.

20.6.5 Writing in the User ROM Area

20.6.5.1 EW Mode 0

- If the supply voltage drops while rewriting the block where the rewrite control program is stored, the flash memory can not be rewritten, because the rewrite control program is not correctly rewritten. If this error occurs, rewrite the user ROM area in standard serial I/O mode or parallel I/O mode.

20.6.5.2 EW Mode 1

- Do not rewrite the block where the rewrite control program is stored.

20.6.6 DMA Transfer

In EW mode 1, do not generate a DMA transfer while the FMR00 bit in the FMR0 register is set to 0. (during the auto-programming or auto-erasing).

20.6.7 Writing Command and Data

Write the command codes and data to even addresses in the user ROM area.

20.6.8 Wait Mode

When entering wait mode, set the FMR01 bit to 0 (CPU rewrite mode disabled) before executing the WAIT instruction.

20.6.9 Stop Mode

When entering stop mode, the following settings are required:

- Set the FMR01 bit to 0 (CPU rewrite mode disabled) and disable the DMA transfer before setting the CM10 bit to 1 (stop mode).

20.6.10 Low Power Consumption Mode and On-Chip Oscillator-Low Power Consumption Mode

If the CM05 bit is set to 1 (main clock stopped), do not execute the following commands.

- Program
- Block erase

20.7 Software Commands

Read or write 16-bit commands and data from or to even addresses in the user ROM area. When writing a command code, 8 high-order bits (D15–D8) are ignored.

Table 20.5 Software Commands

| Command | First bus cycle | | | Second bus cycle | | |
|-----------------------|-----------------|---------|--------------------|------------------|---------|--------------------|
| | Mode | Address | Data (D15 to D0) | Mode | Address | Data (D15 to D0) |
| Read array | Write | X | xxFF ₁₆ | | | |
| Read status register | Write | X | xx70 ₁₆ | Read | X | SRD |
| Clear status register | Write | X | xx50 ₁₆ | | | |
| Program | Write | WA | xx40 ₁₆ | Write | WA | WD |
| Block erase | Write | X | xx20 ₁₆ | Write | BA | xxD0 ₁₆ |

SRD: Status register data (D7 to D0)

WA : Write address (However, even address)

WD : Write data (16 bits)

BA : Highest-order block address (However, even address)

X : Any even address in the user ROM area

xx : 8 high-order bits of command code (ignored)

20.7.1 Read Array Command (FF₁₆)

The read array command reads the flash memory.

Read array mode is entered by writing command code xxFF₁₆ in the first bus cycle. Content of a specified address can be read in 16-bit unit after the next bus cycle. The MCU remains in read array mode until another command is written. Therefore, contents of multiple addresses can be read consecutively.

20.7.2 Read Status Register Command (70₁₆)

The read status register command reads the status register.

By writing command code xx70₁₆ in the first bus cycle, the status register can be read in the second bus cycle (Refer to **20.8 Status Register**). Read an even address in the user ROM area. Do not execute this command in EW mode 1.

20.7.3 Clear Status Register Command (50₁₆)

The clear status register command clears the status register to 0.

By writing xx50₁₆ in the first bus cycle, and bits FMR06 to FMR07 in the FMR0 register and bits SR4 to SR5 in the status register are set to 0.

20.7.4 Program Command (4016)

The program command writes 2-byte data to the flash memory.

Auto program operation (data program and verify) start by writing xx4016 in the first bus cycle and data to the write address specified in the second bus cycle. The address value specified in the first bus cycle must be the same even address as the write address specified in the second bus cycle.

The FMR00 bit in the FMR0 register indicates whether an auto-programming operation has been completed. The FMR00 bit is set to 0 during the auto-program and 1 when the auto-program operation is completed.

After the completion of auto-program operation, the FMR06 bit in the FMR0 register indicates whether or not the auto-program operation has been successfully completed. (Refer to **20.8.4 Full Status Check**).

Also, each block can disable programming command (Refer to **Table 20.4**).

An address that is already written cannot be altered or rewritten.

When commands other than the program command are executed immediately after executing the program command, set the same address as the write address specified in the second bus cycle of the program command, to the specified address value in the first bus cycle of the following command.

In EW mode 1, do not execute this command on the blocks where the rewrite control program is allocated.

In EW mode 0, the MCU enters read status register mode as soon as the auto-program operation starts and the status register can be read. The SR7 bit in the status register is set to 0 as soon as the auto-program operation starts. This bit is set to 1 when the auto-program operation is completed. The MCU remains in read status register mode until the read array command is written. After completion of the auto-program operation, the status register indicates whether or not the auto-program operation has been successfully completed.

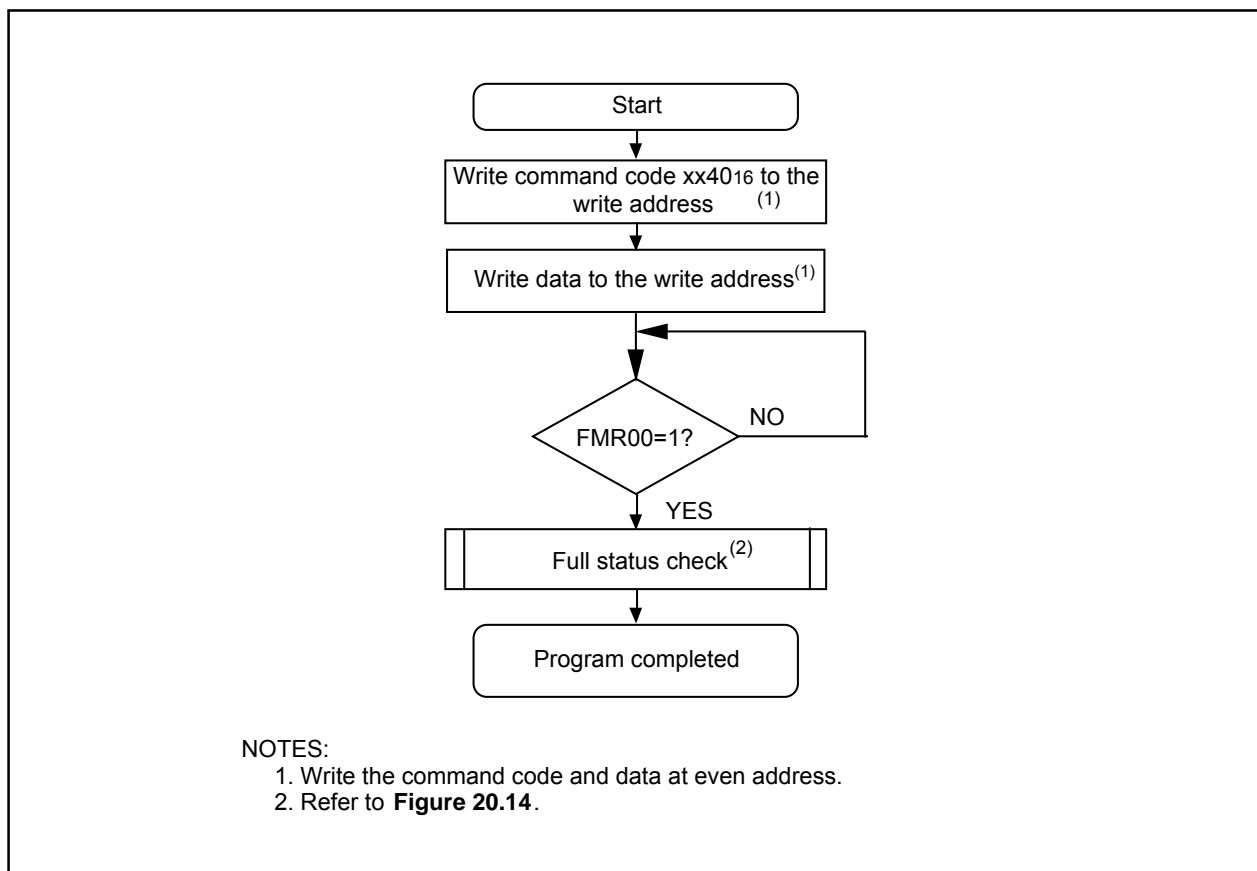


Figure 20.11 Flow Chart of Program Command

20.7.5 Block Erase

Auto erase operation (erase and verify) start in the specified block by writing $xx20_{16}$ in the first bus cycle and $xxD0_{16}$ to the highest-order even address of a block in the second bus cycle.

The FMR00 bit in the FMR0 register indicates whether the auto-erase operation has been completed.

The FMR00 bit is set to 0 (busy) during the auto-erase and 1 (ready) when the auto-erase operation is completed.

When using the erase-suspend function in EW mode 0, verify whether a flash memory has entered erase suspend mode, by the FMR46 bit in the FMR4 register. The FMR46 bit is set to 0 during auto-erase operation and 1 when the auto-erase operation is completed (entering erase-suspend).

After the completion of an auto-erase operation, the FMR07 bit in the FMR0 register indicates whether or not the auto-erase operation has been successfully completed. (Refer to **20.8.4 Full Status Check**).

Also, each block can disable erasing. (Refer to **Table 20.4**).

Figure 20.12 shows a flow chart of the block erase command programming when not using the erase-suspend function. **Figure 20.12** shows a flow chart of the block erase command programming when using an erase-suspend function.

In EW mode 1, do not execute this command on the block where the rewrite control program is allocated. In EW mode 0, the MCU enters read status register mode as soon as the auto-erase operation starts and the status register can be read. The SR7 bit in the status register is set to 0 at the same time the auto-erase operation starts. This bit is set to 1 when the auto-erase operation is completed. The MCU remains in read status register mode until the read array command is written.

When the erase error occurs, execute the clear status register command and block erase command at least three times until an erase error does not occur.

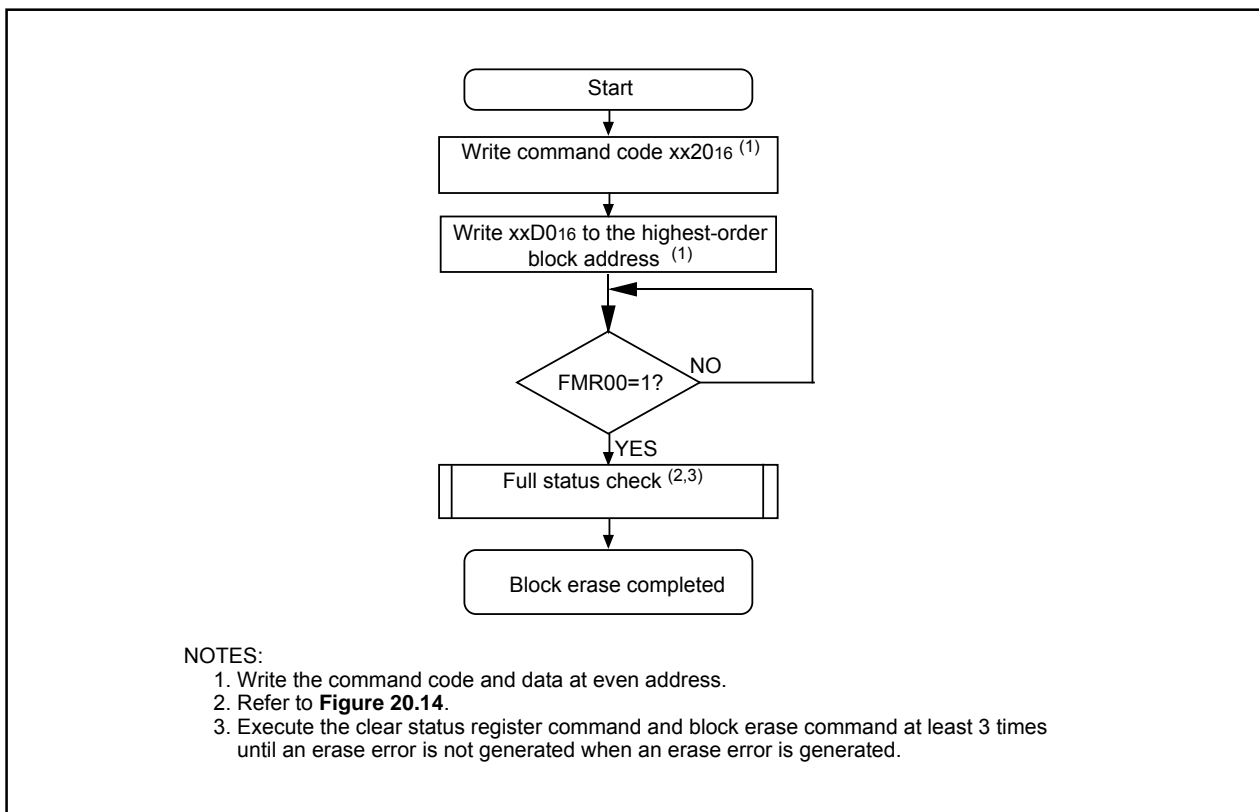


Figure 20.12 Flow Chart of Block Erase Command (when not using erase suspend function)

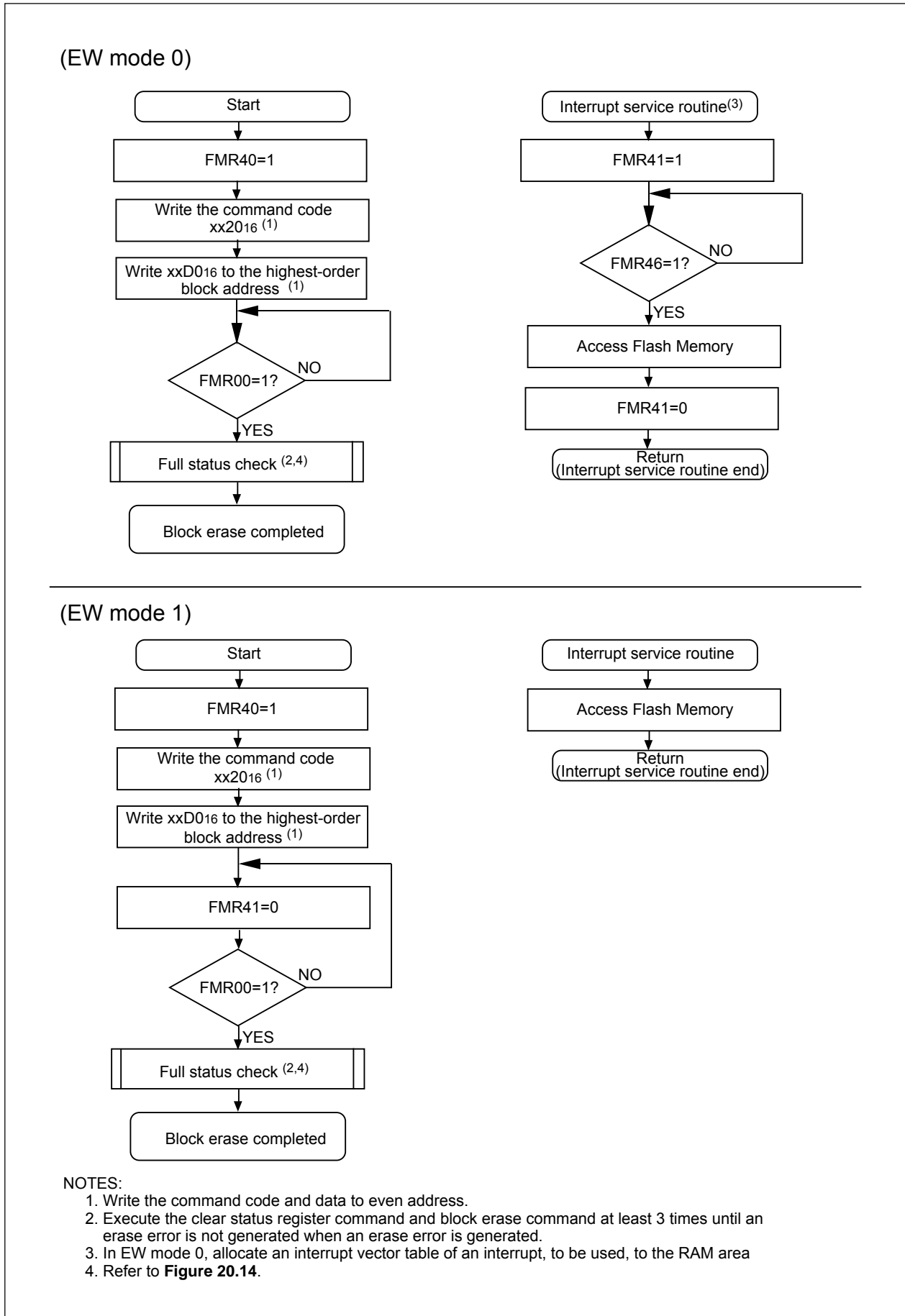


Figure 20.13 Block Erase Command (at use erase suspend)

20.8 Status Register

The status register indicates the operating status of the flash memory and whether or not erase or program operation is successfully completed. Bits FMR00, FMR06, and FMR07 in the FMR0 register indicate the status of the status register.

Table 20.6 lists the status register.

In EW mode 0, the status register can be read in the following cases:

- (1) Any even address in the user ROM area is read after writing the read status register command
- (2) Any even address in the user ROM area is read from when the program or block erase command is executed until when the read array command is executed.

20.8.1 Sequence Status (SR7 and FMR00 Bits)

The sequence status indicates the flash memory operating status. It is set to 0 (busy) while the auto-program and auto-erase operation is being executed and 1 (ready) as soon as these operations are completed. This bit indicates 0 (busy) in erase-suspend mode.

20.8.2 Erase Status (SR5 and FMR07 Bits)

Refer to **20.8.4 Full Status Check**.

20.8.3 Program Status (SR4 and FMR06 Bits)

Refer to **20.8.4 Full Status Check**.

Table 20.6 Status Register

| Bits in the SRD Register | Bits in the FMR0 Register | Status Name | Contents | | Value After Reset |
|--------------------------|---------------------------|-----------------|--------------------|---------------------|-------------------|
| | | | 0 | 1 | |
| SR7 (D7) | FMR00 | Sequence status | Busy | Ready | 1 |
| SR6 (D6) | — | Reserved | - | - | — |
| SR5 (D5) | FMR07 | Erase status | Completed normally | Terminated by error | 0 |
| SR4 (D4) | FMR06 | Program status | Completed normally | Terminated by error | 0 |
| SR3 (D3) | — | Reserved | - | - | — |
| SR2 (D2) | — | Reserved | - | - | — |
| SR1 (D1) | — | Reserved | - | - | — |
| SR0 (D0) | — | Reserved | - | - | — |

- D7 to D0: Indicates the data bus which is read out when executing the read status register command.
- The FMR07 bit (SR5) and FMR06 bit (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 are not accepted.

20.8.4 Full Status Check

If an error occurs, bits FMR06 to FMR07 in the FMR0 register are set to 1, indicating a specific error. Therefore, execution results can be confirmed by verifying these status bits (full status check).

Table 20.7 lists errors and FMR0 register state. **Figure 20.14** shows a flow chart of the full status check and handling procedure for each error.

Table 20.7 Errors and FMR0 Register Status

| FMR0 register (SRD register) status | | Error | Error occurrence condition |
|---|----------------|------------------------|--|
| FMR07 (SR5) | FMR06 (SR4) | | |
| 1 | 1 | Command sequence error | <ul style="list-style-type: none"> • An incorrect commands is written • A value other than <code>xxD016</code> or <code>xxFF16</code> is written in the second bus cycle of the block erase command ⁽¹⁾ • When the block erase command is executed on an protected block • When the program command is executed on protected blocks |
| 1 | 0 | Erase error | <ul style="list-style-type: none"> • The block erase command is executed on an unprotected block but the program operation is not successfully completed |
| 0 | 1 | Program error | <ul style="list-style-type: none"> • The program command is executed on an unprotected block but the program operation is not successfully completed |

Note 1: The flash memory enters read array mode by writing command code `xxFF16` in the second bus cycle of these commands. The command code written in the first bus cycle becomes invalid.

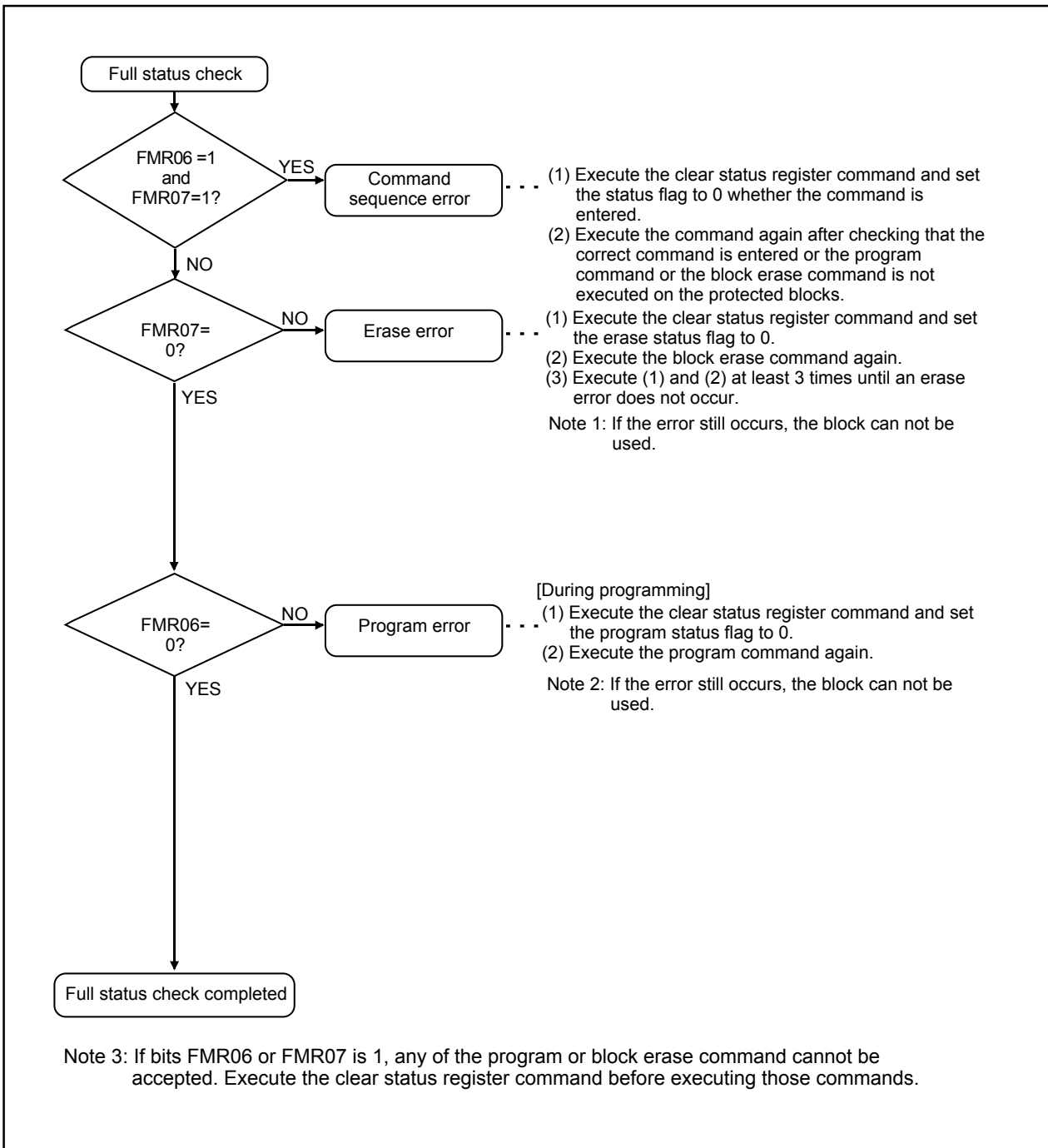


Figure 20.14 Full Status Check and Handling Procedure for Each Error

20.9 Standard Serial I/O Mode

In standard serial I/O mode, the serial programmer supporting the M16C/29 group can be used to rewrite the flash memory user ROM area, while the MCU is mounted on a board. For more information about the serial programmer, contact your serial programmer manufacturer. Refer to the user's manual included with your serial programmer for instruction.

Table 20.8 lists pin description (flash memory standard serial input/output mode). **Figures 20.15** and **20.16** show pin connections for standard serial input/output mode.

20.9.1 ID Code Check Function

The ID code check function determines whether or not the ID codes sent from the serial programmer matches those written in the flash memory. (Refer to **20.3 Functions To Prevent Flash Memory from Rewriting.**)

Table 20.8 Pin Descriptions (Flash Memory Standard Serial I/O Mode)

| Pin | Name | I/O | Description | |
|------------------------|------------------------------|---------------|---|---|
| Vcc, Vss | Power input | | Apply the voltage guaranteed for Program and Erase to Vcc pin and 0 V to Vss pin. | |
| CNVss | CNVs S | I | Connect to Vcc pin. | |
| RESET | Reset input | I | Reset input pin. While $\overline{\text{RESET}}$ pin is "L", wait for td(ROC). | |
| XIN | Clock input | I | Connect a ceramic resonator or crystal oscillator between XIN and XOUT pins. To input an externally generated clock, input it to XIN pin and open XOUT pin. | |
| XOUT | Clock output | O | | |
| AVcc, AVss | Analog power supply input | | Connect AVss to Vss and AVcc to Vcc, respectively. | |
| VREF | Reference voltage input | I | Enter the reference voltage for AD conversion. | |
| P00 to P07 | Input port P0 | I | Input "H" or "L" signal or leave open. | |
| P10 to P15, P17 | Input port P1 | I | Input "H" or "L" signal or leave open. | |
| P16 | Input port P1 | I | Connect this pin to Vcc while RESET pin is "L". (2) | |
| P20 to P27 | Input port P2 | I | Input "H" or "L" level signal or leave open. | |
| P30 to P37 | Input port P3 | I | Input "H" or "L" level signal or leave open. | |
| P60 to P63 | Input port P6 | I | Input "H" or "L" level signal or leave open. | |
| P64 | BUSY output | O | Standard serial I/O mode 1: BUSY signal output pin Standard serial I/O mode 2: Monitor signal output pin for boot program operation check | |
| P65 | SCLK input | I | Standard serial I/O mode 1: Serial clock input pin Standard serial I/O mode 2: Input "L". | |
| P66 | RxD input | I | Serial data input pin | |
| P67 | TxD output | O | Serial data output pin (1) | |
| P70 to P77 | Input port P7 | I | Input "H" or "L" signal or leave open. | |
| P80 to P84, P87 | Input port P8 | I | Input "H" or "L" signal or leave open. | |
| P85 | $\overline{\text{RP}}$ input | I | Connect this pin to Vss while $\overline{\text{RESET}}$ pin is "L". (2) | |
| P86 | $\overline{\text{CE}}$ input | I | Connect this pin to Vcc while $\overline{\text{RESET}}$ pin is "L". (2) | |
| P90 to P92, P95 to P97 | Input port P9 | I | Input "H" or "L" signal or leave open. | |
| P93 | Input port P93 | Normal-ver. | I/O | "H" signal is output for specific time. Input "H" signal or leave open. |
| | | T-ver./V-ver. | I | Input "H" or "L" signal or leave open. |
| P100 to P107 | Input port P10 | I | Input "H" or "L" signal or leave open. | |

NOTES:

- When using standard serial I/O mode 1, to input "H" to the TxD pin is necessary while the $\overline{\text{RESET}}$ pin is held "L". Therefore, connect this pin to Vcc via a resistor. Adjust the pull-up resistor value on a system not to affect a data transfer after reset, because this pin changes to a data-output pin
- Set the following, either or both.
 - Connect the $\overline{\text{CE}}$ pin to Vcc.
 - Connect the $\overline{\text{RP}}$ pin to VSS and P16 pin to Vcc.

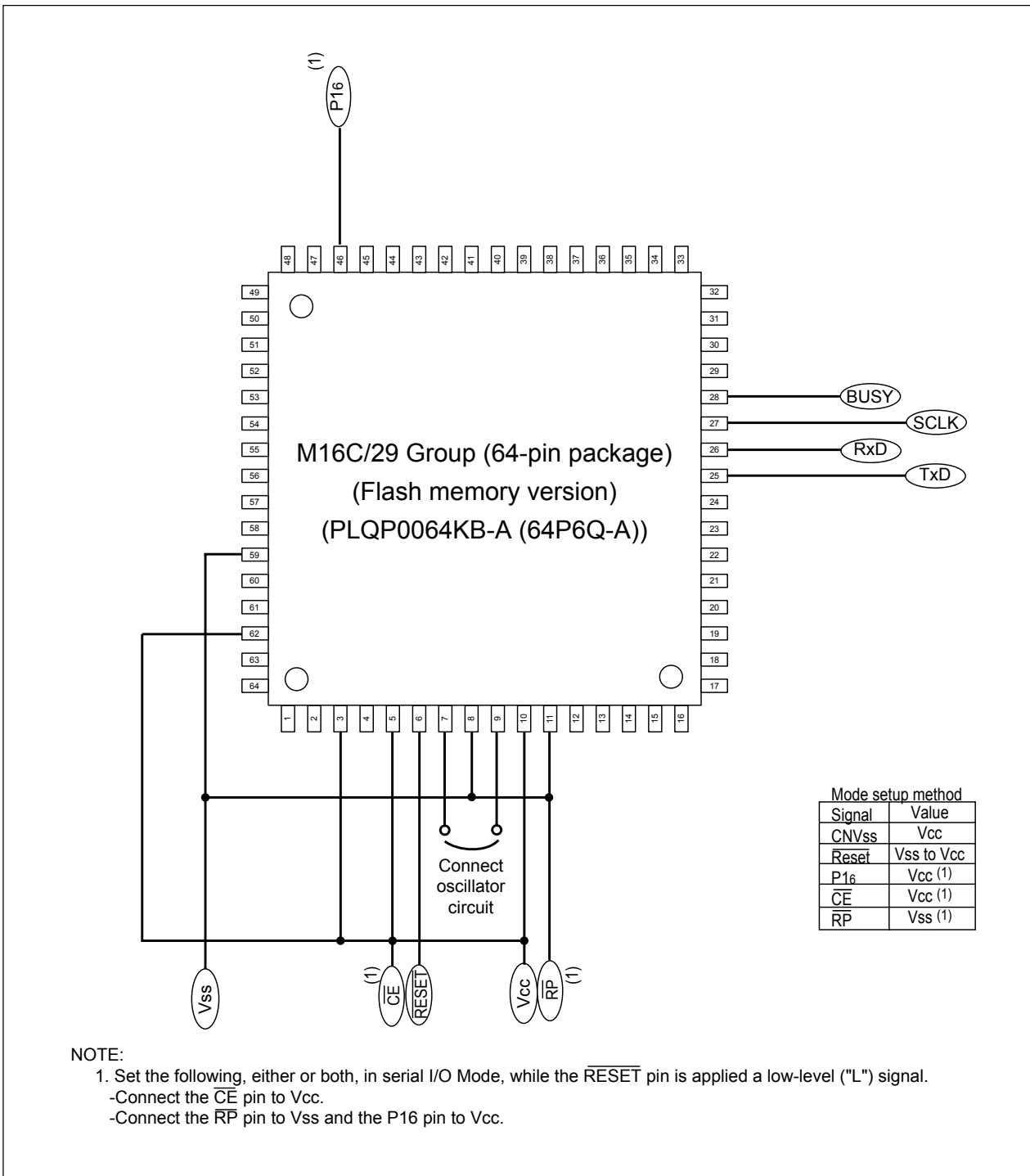


Figure 20.15 Pin Connections for Serial I/O Mode (1)

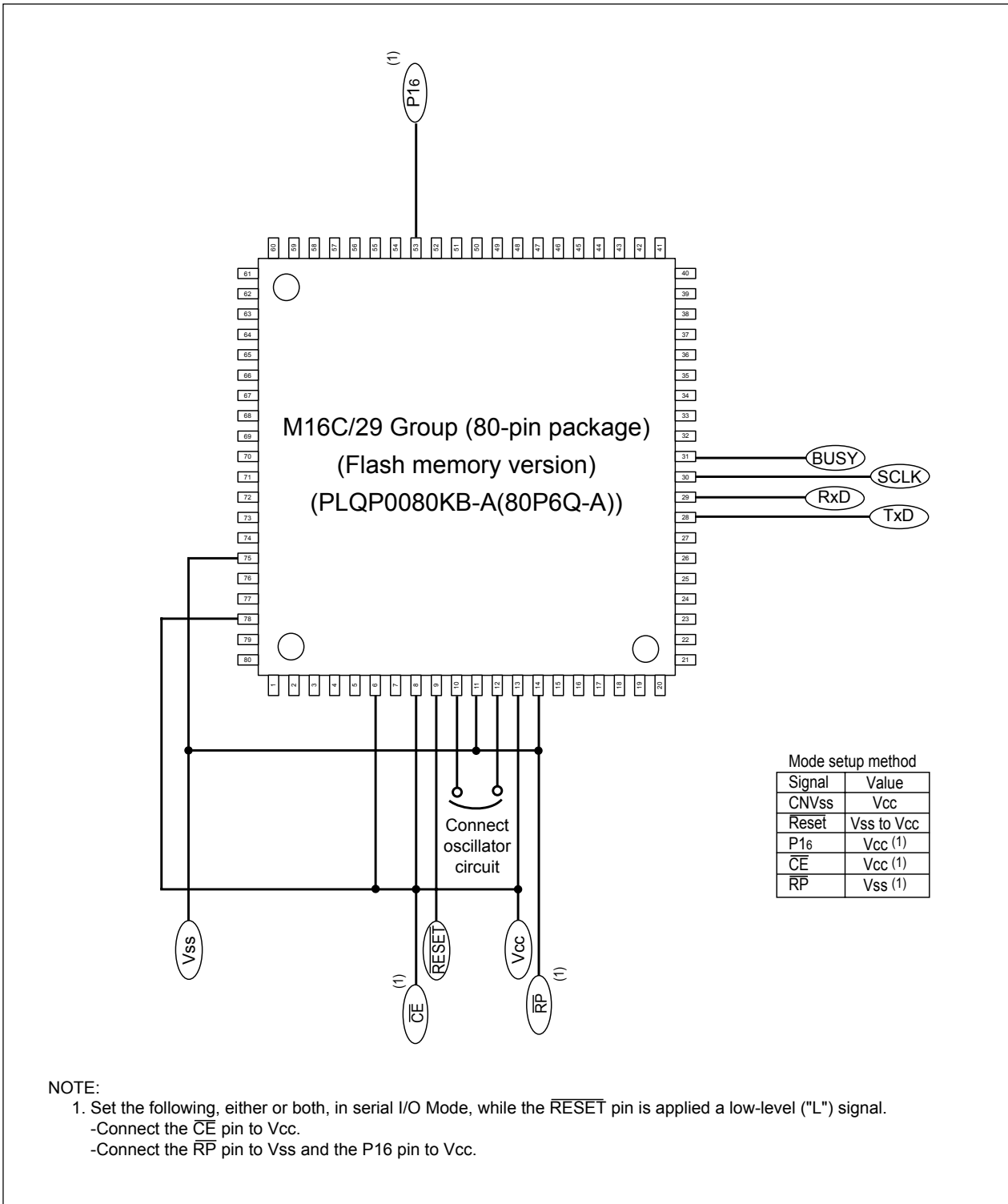


Figure 20.16 Pin Connections for Serial I/O Mode (2)

20.9.2 Example of Circuit Application in Standard Serial I/O Mode

Figure 20.17 shows an example of a circuit application in standard serial I/O mode 1 and Figure 20.18 shows an example of a circuit application in standard serial I/O mode 2. Refer to the user's manual of your serial programmer to handle pins controlled by the serial programmer.

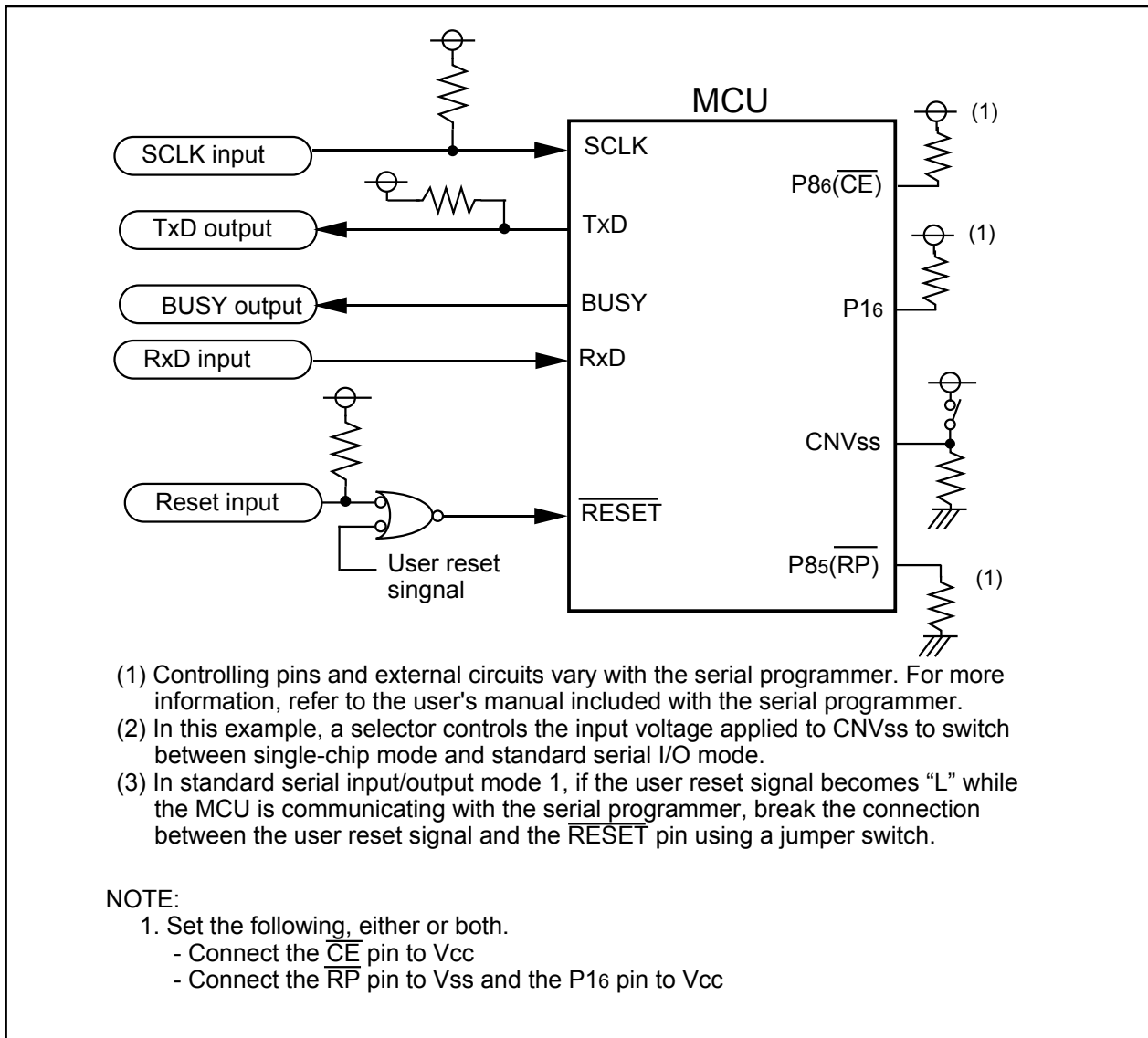


Figure 20.17 Circuit Application in Standard Serial I/O Mode 1

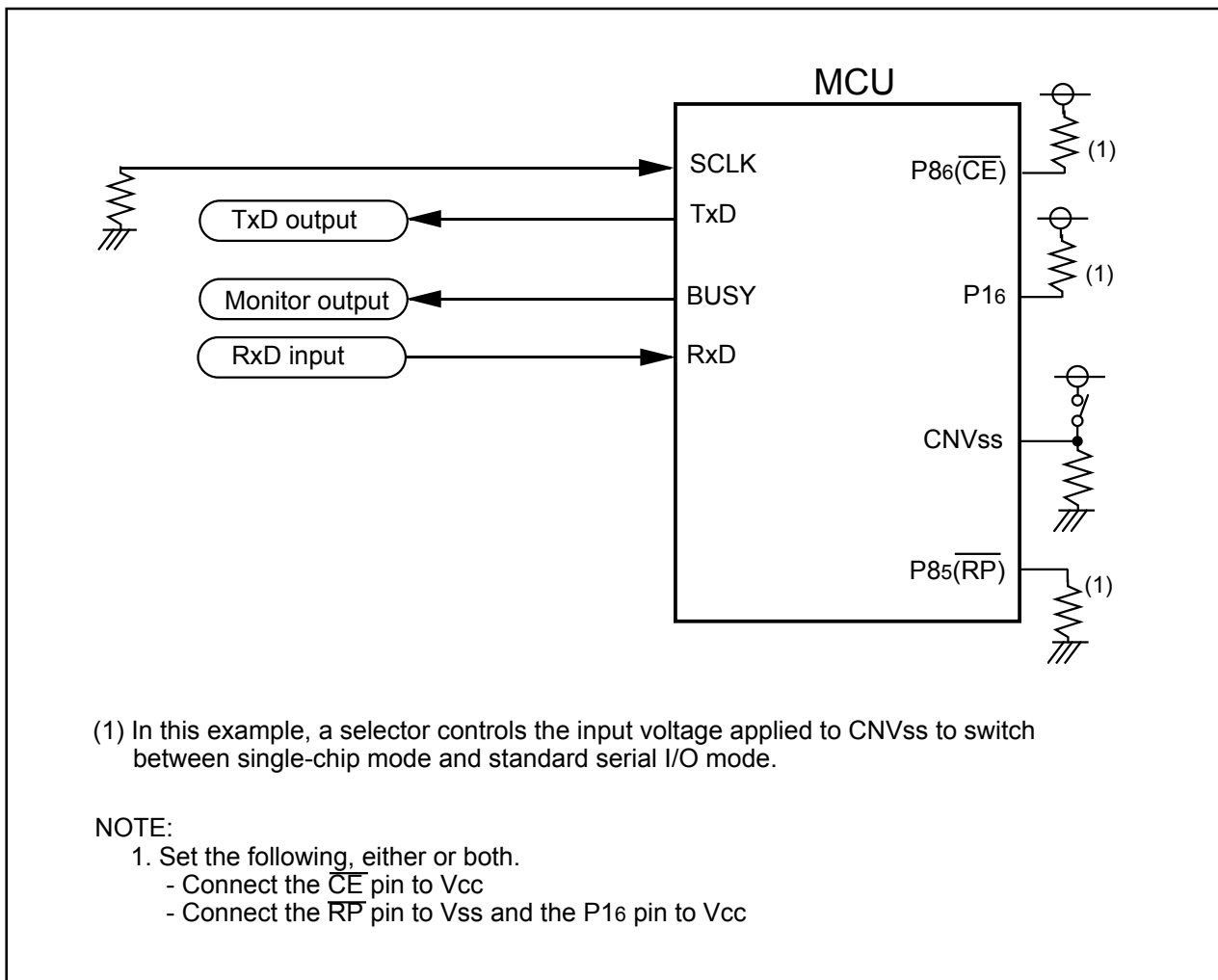


Figure 20.18 Circuit Application in Standard Serial I/O Mode 2

20.10 Parallel I/O Mode

In parallel input/output mode, the user ROM can be rewritten by a parallel programmer supporting the M16C/29 group. Contact your parallel programmer manufacturer for more information on the parallel programmer. Refer to the user's manual included with your parallel programmer for instructions.

20.10.1 ROM Code Protect Function

The ROM code protect function prevents the flash memory from being read or rewritten. (Refer to **20.3 Functions To Prevent Flash Memory from Rewriting**).

20.11 CAN I/O Mode

Note

The CAN I/O mode is not available in M16C/29 T-ver./V-ver.

In CAN I/O mode, the user ROM area can be rewritten while the MCU is mounted on-board by using a CAN programmer which is applicable for the M16C/29 group. For more information about CAN programmers, contact the manufacturer of your CAN programmer. For details on how to use, refer to the user's manual included with your CAN programmer.

Table 20.9 lists pin functions for CAN I/O mode. Figures 20.19 and 20.20 show pin connections for CAN I/O mode.

20.11.1 ID code check function

This function determines whether the ID codes sent from the CAN programmer and those written in the flash memory match. (Refer to **20.3 Functions To Prevent Flash Memory from Rewriting.**)

Table 20.9 Pin Functions for CAN I/O Mode

| Pin | Name | I/O | Description |
|------------------------|---------------------------|-----|---|
| Vcc,Vss | Power input | | Apply the voltage guaranteed for Program and Erase to Vcc pin and 0 V to Vss pin. |
| CNVss | CNVss | I | Connect to Vcc pin. |
| RESET | Reset input | I | Reset input pin. While RESET pin is "L" level, wait for td(ROC). |
| XIN | Clock input | I | Connect a ceramic resonator or crystal oscillator between XIN and XOUT pins. To input an externally generated clock, input it to XIN pin and open XOUT pin. |
| XOUT | Clock output | O | |
| AVcc, AVss | Analog power supply input | | Connect AVss to Vss and AVcc to Vcc, respectively. |
| VREF | Reference voltage input | I | Enter the reference voltage for AD from this pin. |
| P00 to P07 | Input port P0 | I | Input "H" or "L" level signal or leave open. |
| P10 to P15, P17 | Input port P1 | I | Input "H" or "L" level signal or leave open. |
| P16 | Input port P1 | I | Connect this pin to Vcc while RESET is low. (Note 1) |
| P20 to P27 | Input port P2 | I | Input "H" or "L" level signal or leave open. |
| P30 to P37 | Input port P3 | I | Input "H" or "L" level signal or leave open. |
| P60 to P64, P66 | Input port P6 | I | Input "H" or "L" level signal or leave open. |
| P65 | SCLK input | I | Input "L" level signal. |
| P67 | TxD output | O | Input "H" level signal. |
| P70 to P77 | Input port P7 | I | Input "H" or "L" level signal or leave open. |
| P80 to P84, P87 | Input port P8 | I | Input "H" or "L" level signal or leave open. |
| P85 | RP input | I | Connect this pin to Vss while RESET is low. (Note 1) |
| P86 | CE input | I | Connect this pin to Vcc while RESET is low. (Note 1) |
| P90 to P91, P95 to P97 | Input port P9 | I | Input "H" or "L" level signal or leave open. |
| P92 | CRX input | I | Connect this pin to a CAN transceiver. |
| P93 | CTX output | O | Connect this pin to a CAN transceiver. |
| P100 to P107 | Input port P10 | I | Input "H" or "L" level signal or leave open. |

NOTE:

- Set following either or both.
 - Connect the CE pin to Vcc.
 - Connect the RP pin to Vss and the P16 pin to Vcc.

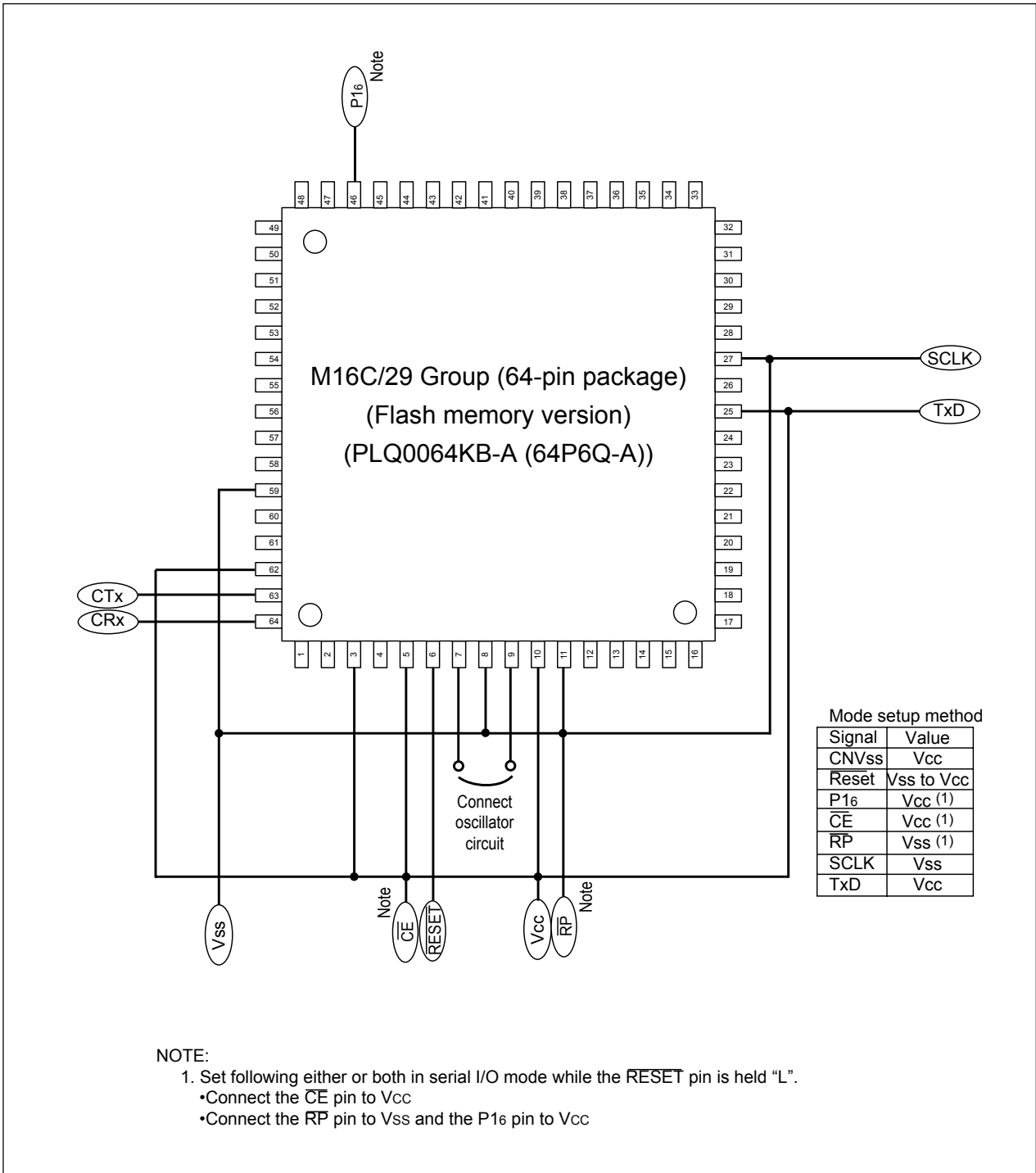


Figure 20.19 Pin Connections for CAN I/O Mode (1)

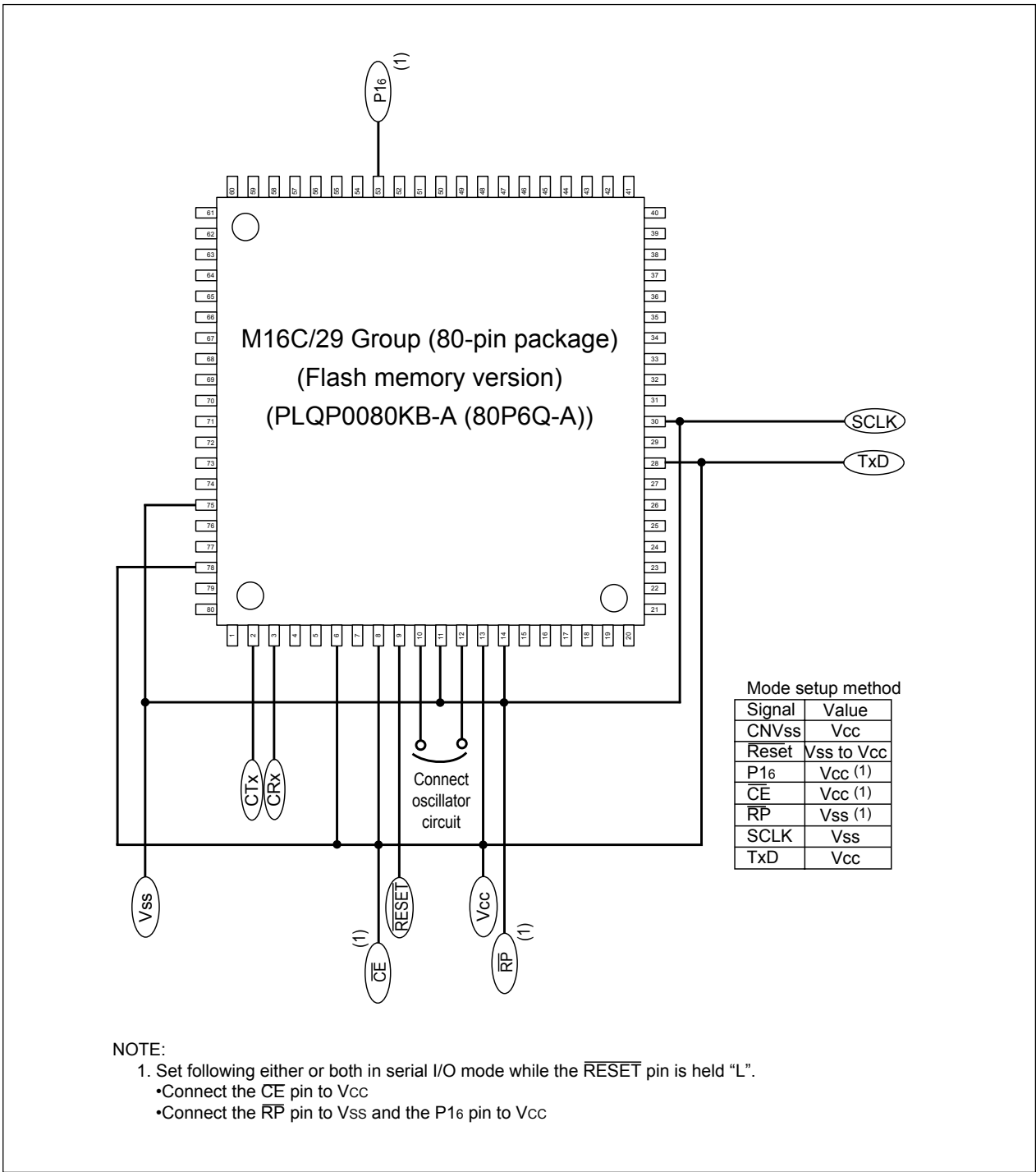


Figure 20.20 Pin Connections for CAN I/O Mode (2)

20.11.2 Example of Circuit Application in CAN I/O Mode

Figure 20.21 shows example of circuit application in CAN I/O mode. Refer to the user's manual for CAN programmer to handle pins controlled by a CAN programmer.

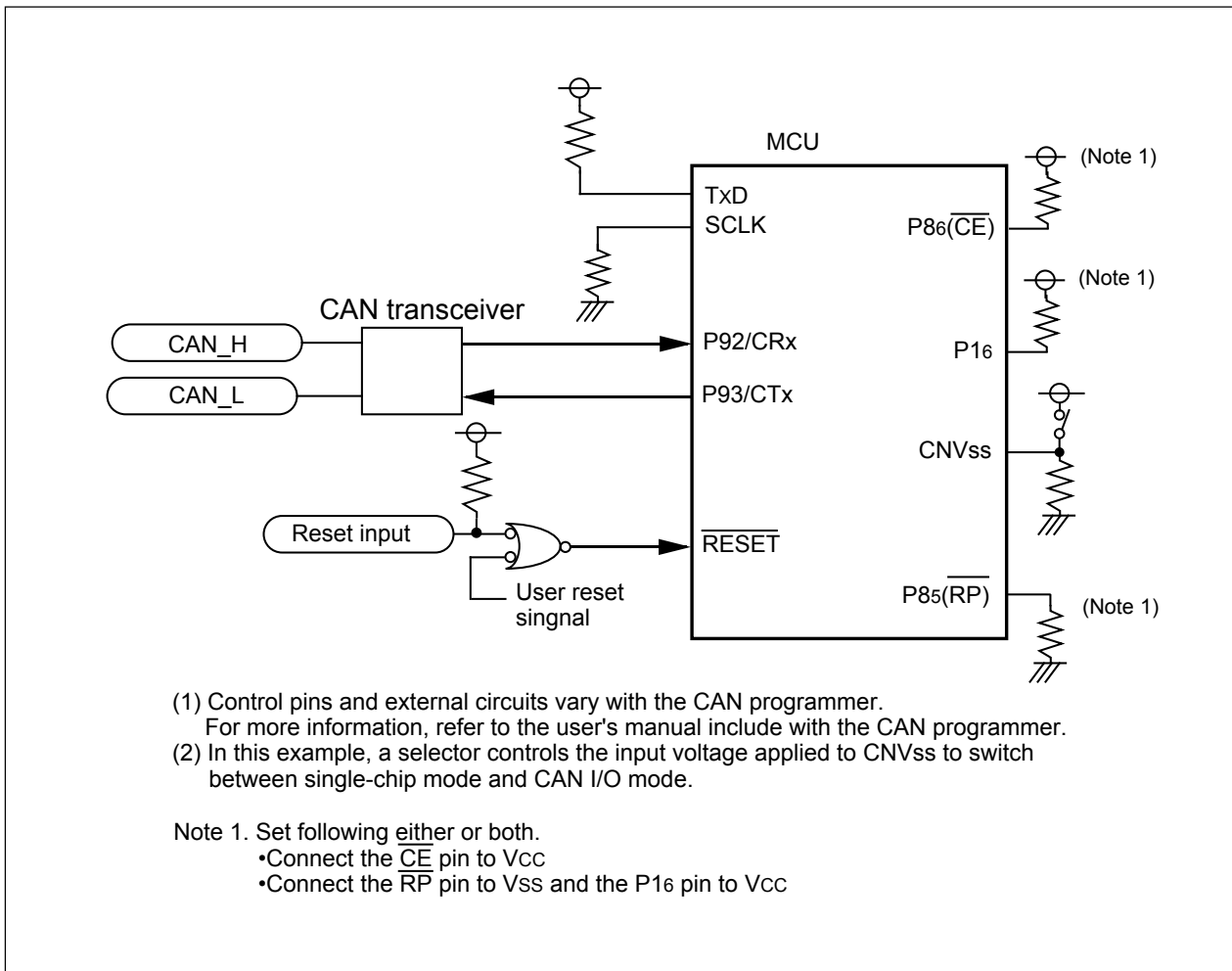


Figure 20.21 Circuit Application in CAN I/O Mode

21. Electrical Characteristics

21.1 Normal version

Table 21.1 Absolute Maximum Ratings

| Symbol | Parameter | | Condition | 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 | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ , X _{IN} , V _{REF} , RESET, CNV _{SS} | | -0.3 to V _{CC} +0.3 | V |
| V _O | Output Voltage | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ , X _{OUT} | | -0.3 to V _{CC} +0.3 | V |
| P _d | Power Dissipation | | -40≤Topr≤85° C | 300 | mW |
| Topr | Operating Ambient Temperature | during CPU operation | | -20 to 85 / -40 to 85 ⁽¹⁾ | ° C |
| | | during flash memory program and erase operation | Program Space (Block 0 to Block 5) | 0 to 60 | ° C |
| | | | Data Space (Block A, Block B) | -20 to 85 / -40 to 85 ⁽¹⁾ | ° C |
| Tstg | Storage Temperature | | | -65 to 150 | ° C |

NOTE:

1. Refer to **Table 1.6**.

Table 21.2 Recommended Operating Conditions (Note 1)

| Symbol | Parameter | | Standard | | | Unit | |
|-----------------------|---|---|---|------------------------------------|--------------------------|--------------------|---|
| | | | Min. | Typ. | Max. | | |
| V _{CC} | Supply Voltage | | 2.7 | | 5.5 | V | |
| AV _{CC} | Analog Supply Voltage | | | V _{CC} | | V | |
| V _{SS} | Supply Voltage | | | 0 | | V | |
| AV _{SS} | Analog Supply Voltage | | | 0 | | V | |
| V _{IH} | Input High ("H") Voltage | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | 0.7V _{CC} | | V _{CC} | V | |
| | | XIN, RESET, CNVSS | 0.8V _{CC} | | V _{CC} | V | |
| | | SDA _{MM} , SCL _{MM} | When I ² C bus input level is selected | 0.7V _{CC} | | V _{CC} | V |
| | | | | When SMBUS input level is selected | 1.4 | V _{CC} | V |
| V _{IL} | Input Low ("L") Voltage | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | 0 | | 0.3V _{CC} | V | |
| | | XIN, RESET, CNVSS | 0 | | 0.2V _{CC} | V | |
| | | SDA _{MM} , SCL _{MM} | When I ² C bus input level is selected | 0 | | 0.3V _{CC} | V |
| | | | | When SMBUS input level is selected | 0 | 0.6 | V |
| I _{OH(peak)} | Peak Output High ("H") Current | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | | | -10.0 | mA | |
| I _{OH(avg)} | Average Output High ("H") Current | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | | | -5.0 | mA | |
| I _{OL(peak)} | Peak Output Low ("L") Current | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | | | 10.0 | mA | |
| I _{OL(avg)} | Average Output Low ("L") Current | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | | | 5.0 | mA | |
| f(XIN) | Main Clock Input Oscillation Frequency ⁽⁴⁾ | | V _{CC} =3.0 to 5.5V | 0 | 20 | MHz | |
| | | | V _{CC} =2.7 to 3.0V | 0 | 33 X V _{CC} -80 | MHz | |
| f(XCIN) | Sub Clock Oscillation Frequency | | | 32.768 | 50 | kHz | |
| f ₁ (ROC) | On-chip Oscillator Frequency 1 | | 0.5 | 1 | 2 | MHz | |
| f ₂ (ROC) | On-chip Oscillator Frequency 2 | | 1 | 2 | 4 | MHz | |
| f ₃ (ROC) | On-chip Oscillator Frequency 3 | | 8 | 16 | 26 | MHz | |
| f(PLL) | PLL Clock Oscillation Frequency ⁽⁴⁾ | | V _{CC} =3.0 to 5.5V | 10 | 20 | MHz | |
| | | | V _{CC} =2.7 to 3.0V | 10 | 33 X V _{CC} -80 | MHz | |
| f(BCLK) | CPU Operation Clock Frequency | | 0 | | 20 | MHz | |
| t _{SU} (PLL) | Wait Time to Stabilize PLL Frequency Synthesizer | | V _{CC} =5.0V | | 20 | ms | |
| | | | V _{CC} =3.0V | | 50 | ms | |

NOTES:

1. Referenced to V_{CC} = 2.7 to 5.5V at Topr = -20 to 85 ° C / -40 to 85 ° C unless otherwise specified.
2. The mean output current is the mean value within 100ms.
3. The total I_{OL(peak)} for all ports must be 80mA or less. The total I_{OH(peak)} for all ports must be -80mA or less.
4. Relationship among main clock oscillation frequency, PLL clock oscillation frequency and supply voltage.

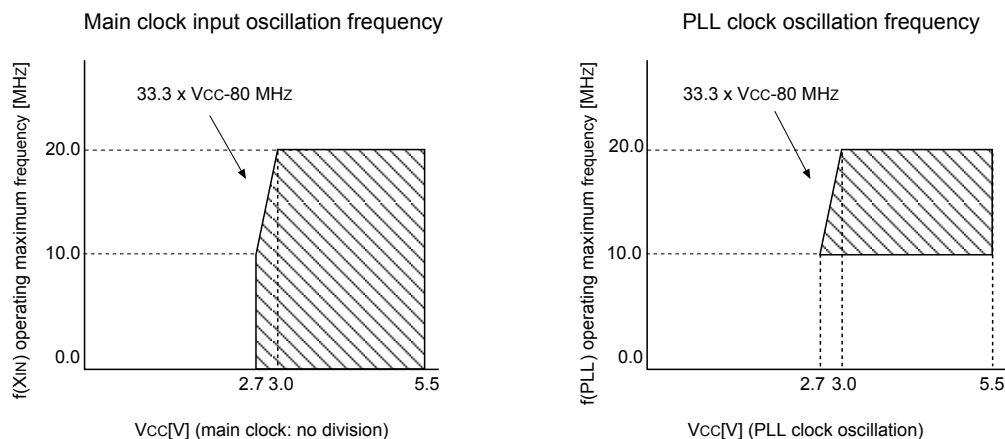


Table 21.3 A/D Conversion Characteristics (Note 1)

| Symbol | Parameter | | Measurement Condition | Standard | | | Unit |
|---------------------|--|--------|---|----------|------|------------------|------------|
| | | | | Min. | Typ. | Max. | |
| - | Resolution | | $V_{REF}=V_{CC}$ | | | 10 | Bits |
| INL | Integral Nonlinearity Error | 10 bit | $V_{REF}=V_{CC}=5V$ | | | ± 3 | LSB |
| | | | $V_{REF}=V_{CC}=3.3V$ | | | ± 5 | LSB |
| | | 8 bit | $V_{REF}=V_{CC}=3.3V$ | | | ± 2 | LSB |
| - | Absolute Accuracy | 10 bit | $V_{REF}=V_{CC}=5V$ | | | ± 3 | LSB |
| | | | $V_{REF}=V_{CC}=3.3V$ | | | ± 5 | LSB |
| | | 8 bit | $V_{REF}=V_{CC}=3.3V$ | | | ± 2 | LSB |
| DNL | Differential Nonlinearity Error | | | | | ± 1 | LSB |
| - | Offset Error | | | | | ± 3 | LSB |
| - | Gain Error | | | | | ± 3 | LSB |
| R _{LADDER} | Resistor Ladder | | $V_{REF}=V_{CC}$ | 10 | | 40 | k Ω |
| t _{CONV} | 10-bit Conversion Time Sample & Hold Function Available | | $V_{REF}=V_{CC}=5V$, $\phi_{AD}=10MHz$ | 3.3 | | | μs |
| t _{CONV} | 8-bit Conversion Time Sample & Hold Function Available | | $V_{REF}=V_{CC}=5V$, $\phi_{AD}=10MHz$ | 2.8 | | | μs |
| V _{REF} | Reference Voltage | | | 2.0 | | V _{CC} | V |
| V _{IA} | Analog Input Voltage | | | 0 | | V _{REF} | V |

NOTES:

1. Referenced to $V_{CC}=AV_{CC}=V_{REF}= 3.3$ to $5.5V$, $V_{SS}=AV_{SS}=0V$ at $T_{opr} = -20$ to $85^\circ C$ / -40 to $85^\circ C$ unless otherwise specified.
2. Keep ϕ_{AD} frequency at 10 MHz or less. Additionally, divide the f_{AD} if V_{CC} is less than 4.2V, and make ϕ_{AD} frequency equal to or lower than $f_{AD}/2$.
3. When sample & hold function is disabled, keep ϕ_{AD} frequency at 250 kHz or more in addition to the limitation in Note 2. When sample & hold function is enabled, keep ϕ_{AD} frequency at 1 MHz or more in addition to the limitation in Note 2.
4. When sample & hold function is enabled, sampling time is $3/\phi_{AD}$ frequency. When sample & hold function is disabled, sampling time is $2/\phi_{AD}$ frequency.

Table 21.4 Flash Memory Version Electrical Characteristics ⁽¹⁾ for 100/1000 E/W cycle products**[Program Space and Data Space in U3 and U5: Program Space in U7 and U9]**

| Symbol | Parameter | Standard | | | Unit |
|-----------------|--|-----------------------------|---------------------|------|--------|
| | | Min. | Typ. ⁽²⁾ | Max. | |
| - | Program and Erase Endurance ⁽³⁾ | 100/1000 ^(4, 11) | | | cycles |
| - | Word Program Time (V _{CC} =5.0V, T _{opr} =25° C) | | 75 | 600 | μs |
| - | Block Erase Time (V _{CC} =5.0V, T _{opr} =25° C) | 2-Kbyte Block | 0.2 | 9 | s |
| | | 8-Kbyte Block | 0.4 | 9 | s |
| | | 16-Kbyte Block | 0.7 | 9 | s |
| | | 32-Kbyte Block | 1.2 | 9 | s |
| td(SR-ES) | Duration between Suspend Request and Erase Suspend | | | 8 | ms |
| t _{rs} | Wait Time to Stabilize Flash Memory Circuit | | | 15 | μs |
| - | Data Hold Time ⁽⁵⁾ | 20 | | | years |

Table 21.5 Flash Memory Version Electrical Characteristics ⁽⁶⁾ 10000 E/W cycle products (Option)**[Data Space in U7 and U9⁽⁷⁾]**

| Symbol | Parameter | Standard | | | Unit |
|-----------------|---|--------------------------|---------------------|------|--------|
| | | Min. | Typ. ⁽²⁾ | Max. | |
| - | Program and Erase Endurance ^(3, 8, 9) | 10000 ^(4, 10) | | | cycles |
| - | Word Program Time (V _{CC} = 5.0 V, T _{opr} = 25° C) | | 100 | | μs |
| - | Block Erase Time (V _{CC} = 5.0 V, T _{opr} = 25° C) (2-Kbyte block) | | 0.3 | | s |
| td(SR-ES) | Duration between Suspend Request and Erase Suspend | | | 8 | ms |
| t _{rs} | Wait Time to Stabilize Flash Memory Circuit | | | 15 | μs |
| - | Data Hold Time ⁽⁵⁾ | 20 | | | years |

NOTES:

1. Referenced to V_{CC} = 2.7 to 5.5 V at T_{opr} = 0 to 60° C (program space), unless otherwise specified.
2. V_{CC} = 5.0 V; T_{opr} = 25° C
3. Program and erase endurance is defined as number of program-erase cycles per block.
If program and erase endurance is *n* cycle (*n* = 100, 1000, 10000), each block can be erased and programmed *n* cycles.
For example, if a 2-Kbyte block A is erased after programming one-word data to each address 1,024 times, this counts as one program and erase endurance. Data cannot be programmed to the same address more than once without erasing the block. (rewrite prohibited).
4. Number of E/W cycles for which operation is guaranteed (1 to minimum value are guaranteed).
5. T_{opr} = 55° C
6. Referenced to V_{CC} = 2.7 to 5.5 V at T_{opr} = -40 to 85° C (U7) / -20 to 85° C (U9) unless otherwise specific.
7. **Table 21.5** applies for data space in U7 and U9 when program and erase endurance is more than 1,000 cycles. Otherwise, use **Table 21.4**.
8. To reduce the number of program and erase endurance when working with systems requiring numerous rewrites, write to unused word addresses within the block instead of rewrite. Erase block only after all possible addresses are used. For example, an 8-word program can be written 128 times maximum before erase becomes necessary. Maintaining an equal number of times erasure between block A and block B will also improve efficiency. It is recommended to track the total number of erasure performed per block and to limit the number of erasure.
9. If an erase error is generated during block erase, execute the clear status register command and block erase command at least 3 times until an erase error is not generated.
10. When executing more than 100 times rewrites, set one wait state per block access by setting the FMR17 bit in the FMR1 register 1 to 1 (wait state). When accessing to all other blocks and internal RAM, wait state can be set by the PM17 bit, regardless of the FMR17 bit setting value.
11. The program and erase endurance is 100 cycles for program space and data space in U3 and U5; 1,000 cycles for program space in U7 and U9.
12. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for further details on the E/W failure rate.

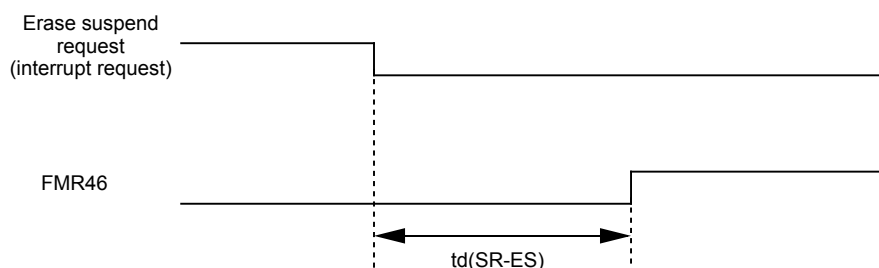


Table 21.6 Low Voltage Detection Circuit Electrical Characteristics (Note 1, Note 3)

| Symbol | Parameter | Measurement Condition | Standard | | | Unit |
|--------|---|------------------------------|----------|------|------|------|
| | | | Min. | Typ. | Max. | |
| Vdet4 | Low Voltage Detection Voltage ⁽¹⁾ | V _{CC} =0.8 to 5.5V | 3.2 | 3.8 | 4.45 | V |
| Vdet3 | Reset Space Detection Voltage ⁽¹⁾ | | 2.3 | 2.8 | 3.4 | V |
| Vdet3s | Low Voltage Reset Hold Voltage ⁽²⁾ | | | | 1.7 | V |
| Vdet3r | Low Voltage Reset Release Voltage | | 2.35 | 2.9 | 3.5 | V |

NOTES:

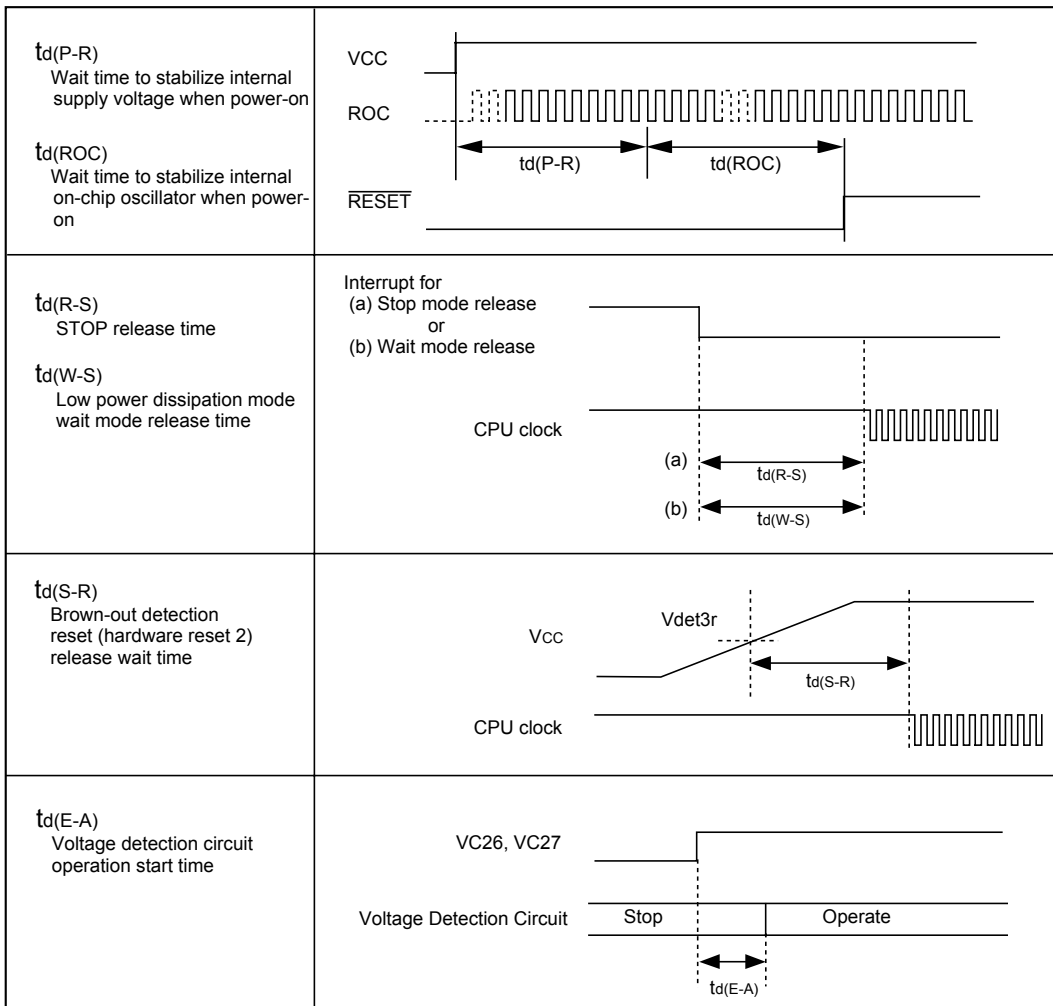
1. Vdet4 > Vdet3
2. Vdet3s is the minmum voltage to maintain brown-out detection reset (hardware reset 2).
3. The low Voltage detection circuit is designed to use when V_{CC} is set to 5V.
4. If the supply power voltage is greater than the reset level detection voltage when the reset level detection voltage is less than 2.7V, the operation at f(BCLK) < 10MHz is guaranteed. However, A/D conversion, serial I/O, flash memory program and erase are excluded.

Table 21.7 Power Supply Circuit Timing Characteristics

| Symbol | Parameter | Measurement Condition | Standard | | | Unit |
|---------|--|-----------------------------------|----------|------------------|------|------|
| | | | Min. | Typ. | Max. | |
| td(P-R) | Wait Time to Stabilize Internal Supply Voltage when Power-on | V _{CC} = 2.7 to 5.5 V | | | 2 | ms |
| td(ROC) | Wait Time to Stabilize Internal On-chip Oscillator when Power-on | | | | 40 | μs |
| td(R-S) | STOP Release Time | | | | 150 | μs |
| td(W-S) | Low Power Dissipation Mode Wait Mode Release Time | | | | 150 | μs |
| td(S-R) | Hardware Reset 2 Release Wait Time | V _{CC} = Vdet3r to 5.5 V | | 6 ⁽¹⁾ | 20 | ms |
| td(E-A) | Low Voltage Detection Circuit Operation Start Time | V _{CC} = 2.7 to 5.5 V | | | 20 | μs |

NOTE:

1. When V_{CC}=5



VCC = 5V**Table 21.8 Electrical Characteristics (Note 1)**

| Symbol | Parameter | | Condition | Standard | | | Unit |
|----------------------------------|---------------------------|---|------------------------|------------------------|----------------------|-----------------|------|
| | | | | Min. | Typ. | Max. | |
| V _{OH} | Output High ("H") Voltage | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | I _α =-5mA | V _{CC} -2.0 | | V _{CC} | V |
| V _{OH} | Output High ("H") Voltage | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | I _α =-200μA | V _{CC} -0.3 | | V _{CC} | V |
| V _{OH} | Output High ("H") Voltage | X _{OUT} | High Power | I _α =-1mA | V _{CC} -2.0 | V _{CC} | V |
| | | | Low Power | I _α =-0.5mA | V _{CC} -2.0 | V _{CC} | |
| | Output High ("H") Voltage | X _{COU} T | High Power | No load applied | | 2.5 | V |
| | | | Low Power | No load applied | | 1.6 | |
| V _{OL} | Output Low ("L") Voltage | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | I _α =5mA | | | 2.0 | V |
| V _{OL} | Output Low ("L") Voltage | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | I _α =200μA | | | 0.45 | V |
| V _{OL} | Output Low ("L") Voltage | X _{OUT} | High Power | I _α =1mA | | 2.0 | V |
| | | | Low Power | I _{OL} =0.5mA | | 2.0 | |
| | Output Low ("L") Voltage | X _{COU} T | High Power | No load applied | | 0 | V |
| | | | Low Power | No load applied | | 0 | |
| V _{T+} -V _{T-} | Hysteresis | TA0 _{IN} -TA4 _{IN} , TB0 _{IN} -TB2 _{IN} , INT ₀ -INT ₅ , NMI, AD _{TRG} , CTS ₀ -CTS ₂ , SCL, SDA, CLK ₀ -CLK ₂ , TA2 _{OUT} -TA4 _{OUT} , K1 ₀ -K1 ₃ , RxD ₀ -RxD ₂ , S _{IN} ₃ , S _{IN} ₄ | | 0.2 | | 1.0 | V |
| V _{T+} -V _{T-} | Hysteresis | RESET | | 0.2 | | 2.5 | V |
| V _{T+} -V _{T-} | Hysteresis | X _{IN} | | 0.2 | | 0.8 | V |
| I _{IH} | Input High ("H") Current | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ X _{IN} , RESET, CNV _{SS} | V _I =5V | | | 5.0 | μA |
| I _{IL} | Input Low ("L") Current | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ X _{IN} , RESET, CNV _{SS} | V _I =0V | | | -5.0 | μA |
| R _{FULLUP} | Pull-up Resistance | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | V _I =0V | 30 | 50 | 170 | kΩ |
| R _{FXIN} | Feedback Resistance | X _{IN} | | | 1.5 | | MΩ |
| R _{FXGIN} | Feedback Resistance | X _{GIN} | | | 15 | | MΩ |
| V _{RAM} | RAM Standby Voltage | | In stop mode | 2.0 | | | V |

NOTES:

1. Referenced to V_{CC}=4.2 to 5.5V, V_{SS}=0V at T_{opr}=-20 to 85 ° C / -40 to 85 ° C, f(BCLK)=20MHz unless otherwise specified.

V_{CC} = 5V**Table 21.9 Electrical Characteristics (2) (Note 1)**

| Symbol | Parameter | Measurement Condition | | | Standard | | | Unit |
|-------------------|--|---|------------------------|---|----------|------|------|------|
| | | | | | Min. | Typ. | Max. | |
| I _{CC} | Power Supply Current (V _{CC} =4.2 to 5.5V) | Output pins are left open and other pins are connected to V _{SS} | Mask ROM | f(BCLK) = 20 MHz, main clock, no division | | 18 | 25 | mA |
| | | | | On-chip oscillation, f _{2(ROC)} selected, f(BCLK) = 1 MHz | | 2 | | mA |
| | | | Flash memory | f(BCLK) = 20 MHz, main clock, no division | | 18 | 25 | mA |
| | | | | On-chip oscillation, f _{2(ROC)} selected, f(BCLK) = 1 MHz | | 2 | | mA |
| | | | Flash memory program | f(BCLK) = 10 MHz, V _{CC} = 5.0 V | | 11 | | mA |
| | | | Flash memory erase | f(BCLK) = 10 MHz, V _{CC} = 5.0 V | | 11 | | mA |
| | | | Mask ROM | f(BCLK) = 32 kHz, In low-power consumption mode, Program running on ROM ⁽³⁾ | | 25 | | μA |
| | | | | On-chip oscillation, f _{2(ROC)} selected, f(BCLK) = 1 MHz, In wait mode | | 50 | | μA |
| | | | Flash memory | f(BCLK) = 32 kHz, In low-power consumption mode, Program running on RAM ⁽³⁾ | | 25 | | μA |
| | | | | f(BCLK) = 32 kHz, In low-power consumption mode, Program running on flash memory ⁽³⁾ | | 450 | | μA |
| | | | | On-chip oscillation, f _{2(ROC)} selected, f(BCLK) = 1 MHz, In wait mode ⁽⁴⁾ | | 50 | | μA |
| | | | Mask ROM, Flash memory | f(BCLK) = 32 kHz, In wait mode ⁽²⁾ , Oscillation capacity high | | 8.5 | | μA |
| | | | | f(BCLK) = 32 kHz, In wait mode ⁽²⁾ , Oscillation capacity low | | 3 | | μA |
| | | | | While clock stops, Topr = 25° C | | 0.8 | 3 | μA |
| I _{det4} | Low voltage detection dissipation current ⁽⁴⁾ | | | 0.7 | 4 | μA | | |
| I _{det3} | Reset area detection dissipation current ⁽⁴⁾ | | | 1.2 | 8 | μA | | |

NOTES:

1. Referenced to V_{CC} = 4.2 to 5.5 V, V_{SS} = 0 V at Topr = -20 to 85° C / -40 to 85° C, f(BCLK) = 20 MHz unless otherwise specified.
2. With one timer operates, using fc32.
3. This indicates the memory in which the program to be executed exists.
4. I_{det} is dissipation current when the following bit is set to 1 (detection circuit enabled).
I_{det4}: VC27 bit in the VCR2 register
I_{det3}: VC26 bit in the VCR2 register

$V_{CC} = 5V$

Timing Requirements**($V_{CC} = 5V$, $V_{SS} = 0V$, at $T_{opr} = -20$ to $85^{\circ}C$ / -40 to $85^{\circ}C$ unless otherwise specified)****Table 21.10 External Clock Input (XIN input)**

| Symbol | Parameter | Standard | | Unit |
|------------|---------------------------------------|----------|------|------|
| | | Min. | Max. | |
| t_c | External Clock Input Cycle Time | 50 | | ns |
| $t_{w(H)}$ | External Clock Input High ("H") Width | 20 | | ns |
| $t_{w(L)}$ | External Clock Input Low ("L") Width | 20 | | ns |
| t_r | External Clock Rise Time | | 9 | ns |
| t_f | External Clock Fall Time | | 9 | ns |

$V_{CC} = 5V$

Timing Requirements(V_{CC} = 5V, V_{SS} = 0V, at Topr = – 20 to 85°C / – 40 to 85°C unless otherwise specified)**Table 21.11 Timer A Input (Counter Input in Event Counter Mode)**

| Symbol | Parameter | Standard | | Unit |
|----------------------|------------------------------|----------|------|------|
| | | Min. | Max. | |
| t _c (TA) | TAiIN input cycle time | 100 | | ns |
| t _w (TAH) | TAiIN input HIGH pulse width | 40 | | ns |
| t _w (TAL) | TAiIN input LOW pulse width | 40 | | ns |

Table 21.12 Timer A Input (Gating Input in Timer Mode)

| Symbol | Parameter | Standard | | Unit |
|----------------------|------------------------------|----------|------|------|
| | | Min. | Max. | |
| t _c (TA) | TAiIN input cycle time | 400 | | ns |
| t _w (TAH) | TAiIN input HIGH pulse width | 200 | | ns |
| t _w (TAL) | TAiIN input LOW pulse width | 200 | | ns |

Table 21.13 Timer A Input (External Trigger Input in One-shot Timer Mode)

| Symbol | Parameter | Standard | | Unit |
|----------------------|------------------------------|----------|------|------|
| | | Min. | Max. | |
| t _c (TA) | TAiIN input cycle time | 200 | | ns |
| t _w (TAH) | TAiIN input HIGH pulse width | 100 | | ns |
| t _w (TAL) | TAiIN input LOW pulse width | 100 | | ns |

Table 21.14 Timer A Input (External Trigger Input in Pulse Width Modulation Mode)

| Symbol | Parameter | Standard | | Unit |
|----------------------|------------------------------|----------|------|------|
| | | Min. | Max. | |
| t _w (TAH) | TAiIN input HIGH pulse width | 100 | | ns |
| t _w (TAL) | TAiIN input LOW pulse width | 100 | | ns |

Table 21.15 Timer A Input (Counter Increment/decrement Input in Event Counter Mode)

| Symbol | Parameter | Standard | | Unit |
|--------------------------|-------------------------------|----------|------|------|
| | | Min. | Max. | |
| t _c (UP) | TAiOUT input cycle time | 2000 | | ns |
| t _w (UPH) | TAiOUT input HIGH pulse width | 1000 | | ns |
| t _w (UPL) | TAiOUT input LOW pulse width | 1000 | | ns |
| t _{su} (UP-TiN) | TAiOUT input setup time | 400 | | ns |
| t _h (TiN-UP) | TAiOUT input hold time | 400 | | ns |

Table 21.16 Timer A Input (Two-phase Pulse Input in Event Counter Mode)

| Symbol | Parameter | Standard | | Unit |
|------------------------------|-------------------------|----------|------|------|
| | | Min. | Max. | |
| t _c (TA) | TAiIN input cycle time | 800 | | ns |
| t _{su} (TAiN-TAOUT) | TAiOUT input setup time | 200 | | ns |
| t _{su} (TAOUT-TAiN) | TAiIN input setup time | 200 | | ns |

$V_{CC} = 5V$

Timing Requirements**($V_{CC} = 5V$, $V_{SS} = 0V$, at $T_{op} = -20$ to $85^{\circ}C$ / -40 to $85^{\circ}C$ unless otherwise specified)****Table 21.17 Timer B Input (Counter Input in Event Counter Mode)**

| Symbol | Parameter | Standard | | Unit |
|--------------|--|----------|------|------|
| | | Min. | Max. | |
| $t_{c(TB)}$ | TBiIN input cycle time (counted on one edge) | 100 | | ns |
| $t_{w(TBH)}$ | TBiIN input HIGH pulse width (counted on one edge) | 40 | | ns |
| $t_{w(TBL)}$ | TBiIN input LOW pulse width (counted on one edge) | 40 | | ns |
| $t_{c(TB)}$ | TBiIN input cycle time (counted on both edges) | 200 | | ns |
| $t_{w(TBH)}$ | TBiIN input HIGH pulse width (counted on both edges) | 80 | | ns |
| $t_{w(TBL)}$ | TBiIN input LOW pulse width (counted on both edges) | 80 | | ns |

Table 21.18 Timer B Input (Pulse Period Measurement Mode)

| Symbol | Parameter | Standard | | Unit |
|--------------|------------------------------|----------|------|------|
| | | Min. | Max. | |
| $t_{c(TB)}$ | TBiIN input cycle time | 400 | | ns |
| $t_{w(TBH)}$ | TBiIN input HIGH pulse width | 200 | | ns |
| $t_{w(TBL)}$ | TBiIN input LOW pulse width | 200 | | ns |

Table 21.19 Timer B Input (Pulse Width Measurement Mode)

| Symbol | Parameter | Standard | | Unit |
|--------------|------------------------------|----------|------|------|
| | | Min. | Max. | |
| $t_{c(TB)}$ | TBiIN input cycle time | 400 | | ns |
| $t_{w(TBH)}$ | TBiIN input HIGH pulse width | 200 | | ns |
| $t_{w(TBL)}$ | TBiIN input LOW pulse width | 200 | | ns |

Table 21.20 A/D Trigger Input

| Symbol | Parameter | Standard | | Unit |
|--------------|--|----------|------|------|
| | | Min. | Max. | |
| $t_{c(AD)}$ | \overline{ADTRG} input cycle time (trigger able minimum) | 1000 | | ns |
| $t_{w(ADL)}$ | \overline{ADTRG} input LOW pulse width | 125 | | ns |

Table 21.21 Serial I/O

| Symbol | Parameter | Standard | | Unit |
|---------------|-----------------------------|----------|------|------|
| | | Min. | Max. | |
| $t_{c(CK)}$ | CLKi input cycle time | 200 | | ns |
| $t_{w(CKH)}$ | CLKi input HIGH pulse width | 100 | | ns |
| $t_{w(CKL)}$ | CLKi input LOW pulse width | 100 | | ns |
| $t_{d(C-Q)}$ | TxDi output delay time | | 80 | ns |
| $t_{h(C-Q)}$ | TxDi hold time | 0 | | ns |
| $t_{su(D-C)}$ | RxDi input setup time | 70 | | ns |
| $t_{h(C-D)}$ | RxDi input hold time | 90 | | ns |

Table 21.22 External Interrupt \overline{INTi} Input

| Symbol | Parameter | Standard | | Unit |
|--------------|--|----------|------|------|
| | | Min. | Max. | |
| $t_{w(INH)}$ | \overline{INTi} input HIGH pulse width | 250 | | ns |
| $t_{w(INL)}$ | \overline{INTi} input LOW pulse width | 250 | | ns |

$$V_{CC} = 5V$$

Timing Requirements

($V_{CC} = 5V$, $V_{SS} = 0V$, at $T_{op} = -20$ to $85^{\circ}C$ / -40 to $85^{\circ}C$ unless otherwise specified)

Table 21.23 Multi-master I²C bus Line

| Symbol | Parameter | Standard clock mode | | High-speed clock mode | | Unit |
|---------|-------------------------------------|---------------------|------|-----------------------|------|------|
| | | Min. | Max. | Min. | Max. | |
| tBUF | Bus free time | 4.7 | | 1.3 | | μs |
| tHD;STA | The hold time in start condition | 4.0 | | 0.6 | | μs |
| tLOW | The hold time in SCL clock 0 status | 4.7 | | 1.3 | | μs |
| tR | SCL, SDA signals' rising time | | 1000 | 20+0.1Cb | 300 | ns |
| tHD;DAT | Data hold time | 0 | | 0 | 0.9 | μs |
| tHIGH | The hold time in SCL clock 1 status | 4.0 | | 0.6 | | μs |
| tF | SCL, SDA signals' falling time | | 300 | 20+0.1Cb | 300 | ns |
| tsu;DAT | Data setup time | 250 | | 100 | | ns |
| tsu;STA | The setup time in restart condition | 4.7 | | 0.6 | | μs |
| tsu;STO | Stop condition setup time | 4.0 | | 0.6 | | μs |

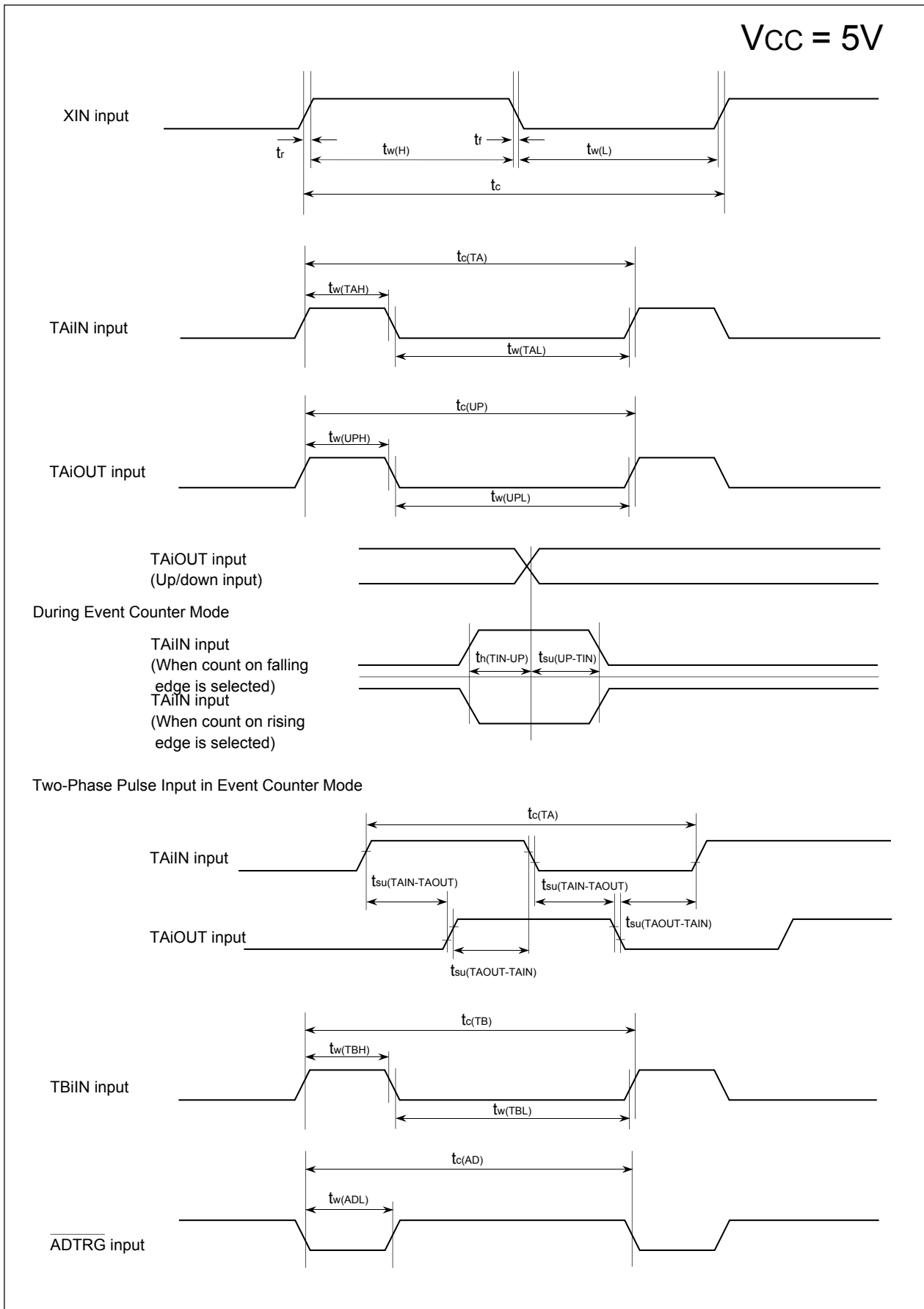


Figure 21.1 Timing Diagram (1)

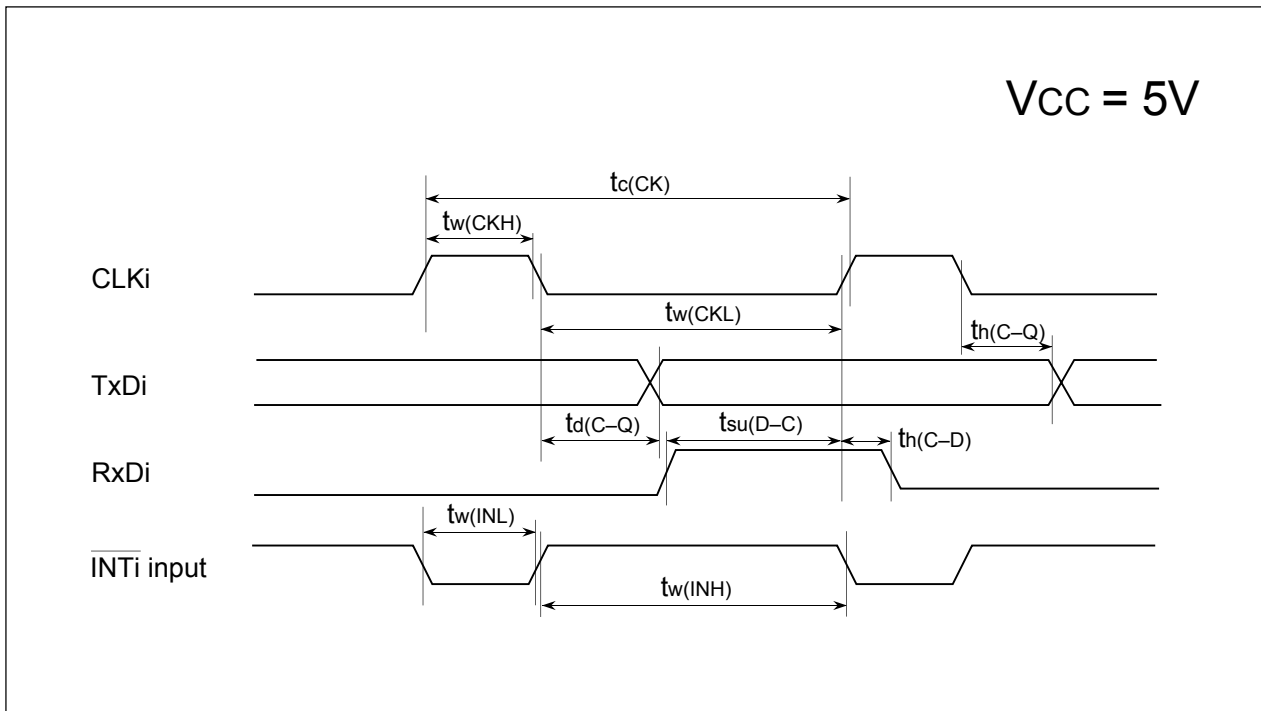


Figure 21.2 Timing Diagram (2)

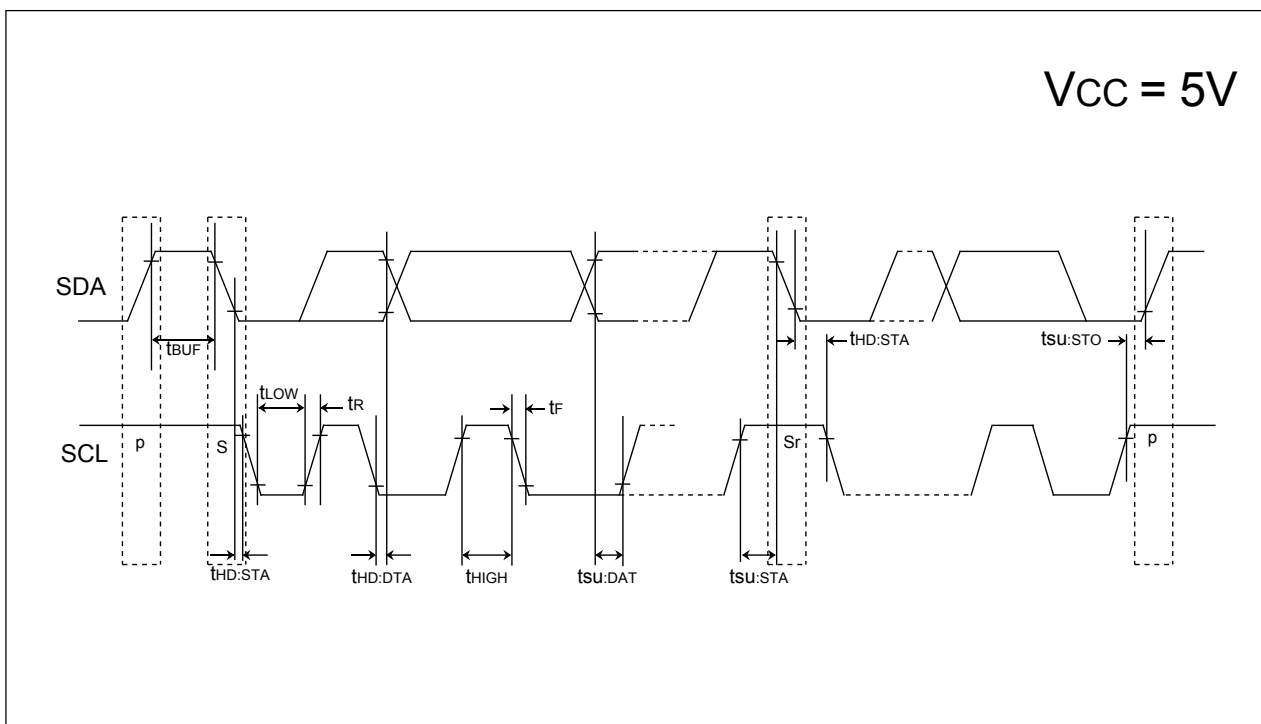


Figure 21.3 Timing Diagram (3)

V_{CC} = 3V**Table 21.24 Electrical Characteristics (Note 1)**

| Symbol | Parameter | | Condition | Standard | | | Unit |
|----------------------------------|---------------------------|---|-------------------------|---------------------------|----------------------|-----------------|------|
| | | | | Min. | Typ. | Max. | |
| V _{OH} | Output High ("H") Voltage | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | I _{OH} = -1 mA | V _{CC} -0.5 | | V _{CC} | V |
| V _{OH} | Output High ("H") Voltage | X _{OUT} | High Power | I _{OH} = -0.1 mA | V _{CC} -0.5 | V _{CC} | V |
| | | | Low Power | I _{OH} = -50 μA | V _{CC} -0.5 | V _{CC} | |
| | Output High ("H") Voltage | X _{COUT} | High Power | No load applied | | 2.5 | V |
| | | | Low Power | No load applied | | 1.6 | |
| V _{OL} | Output Low ("L") Voltage | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | I _{OL} = 1 mA | | | 0.5 | V |
| V _{OL} | Output Low ("L") Voltage | X _{OUT} | High Power | I _{OL} = 0.1 mA | | 0.5 | V |
| | | | Low Power | I _{OL} = 50 μA | | 0.5 | |
| | Output Low ("L") Voltage | X _{COUT} | High Power | No load applied | | 0 | V |
| | | | Low Power | No load applied | | 0 | |
| V _{T+} -V _{T-} | Hysteresis | TA0 _{IN} -TA4 _{IN} , TB0 _{IN} -TB2 _{IN} , INT ₀ -INT ₅ , NMI, AD _{TRG} , CTS ₀ -CTS ₂ , SCL, SDA, CLK ₀ -CLK ₂ , TA2 _{OUT} -TA4 _{OUT} , KI ₀ -KI ₃ , R _{XDD} -R _{XD2} , S _{IN3} , S _{IN4} | | | | 0.8 | V |
| V _{T+} -V _{T-} | Hysteresis | RESET | | | | 1.8 | V |
| V _{T+} -V _{T-} | Hysteresis | X _{IN} | | | | 0.8 | V |
| I _{IH} | Input High ("H") Current | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ X _{IN} , RESET, CNV _{SS} | V _I = 3 V | | | 4.0 | μA |
| I _{IL} | Input Low ("L") Current | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ X _{IN} , RESET, CNV _{SS} | V _I = 0 V | | | -4.0 | μA |
| R _{PULLUP} | Pull-up Resistance | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | V _I = 0 V | 50 | 100 | 500 | kΩ |
| R _{fXIN} | Feedback Resistance | X _{IN} | | | 3.0 | | MΩ |
| R _{fXCIN} | Feedback Resistance | X _{CIN} | | | 25 | | MΩ |
| V _{RAM} | RAM Standby Voltage | | In stop mode | 2.0 | | | V |

NOTE:

1. Referenced to V_{CC} = 2.7 to 3.6 V, V_{SS} = 0 V at Topr = -20 to 85 °C / -40 to 85 °C, f(BCLK) = 10MHz unless otherwise specified.

$V_{CC} = 3V$ **Table 21.25 Electrical Characteristics (2) (Note 1)**

| Symbol | Parameter | Measurement Condition | | | Standard | | | Unit |
|-----------------|--|---|------------------------|---|----------|------|------|------|
| | | | | | Min. | Typ. | Max. | |
| I _{CC} | Power Supply Current (V _{CC} = 2.7 to 3.6V) | Output pins are left open and other pins are connected to V _{SS} | Mask ROM | f(BCLK) = 10 MHz, main clock, no division | | 8 | 13 | mA |
| | | | | On-chip oscillation, f _{2(ROC)} selected, f(BCLK) = 1 MHz | | 1 | | mA |
| | | | Flash memory | f(BCLK) = 10 MHz, main clock, no division | | 8 | 13 | mA |
| | | | Flash memory program | f(BCLK) = 10 MHz, V _{CC} = 3.0 V | | 11 | | mA |
| | | | Flash memory erase | f(BCLK) = 10 MHz, V _{CC} = 3.0 V | | 11 | | mA |
| | | | Mask ROM | f(BCLK) = 32 kHz, In low-power consumption mode, Program running on ROM ⁽³⁾ | | 20 | | μA |
| | | | | On-chip oscillation, f _{2(ROC)} selected, f(BCLK) = 1 MHz, In wait mode | | 25 | | μA |
| | | | Flash memory | f(BCLK) = 32 kHz, In low-power consumption mode, Program running on RAM ⁽³⁾ | | 20 | | μA |
| | | | | f(BCLK) = 32 kHz, In low-power consumption mode, Program running on flash memory ⁽³⁾ | | 450 | | μA |
| | | | | On-chip oscillation, f _{2(ROC)} selected, f(BCLK) = 1 MHz, In wait mode ⁽⁴⁾ | | 45 | | μA |
| | | | Mask ROM, Flash memory | f(BCLK) = 32 kHz, In wait mode ⁽²⁾ , Oscillation capacity high | | 6.6 | | μA |
| | | | | f(BCLK) = 32 kHz, In wait mode ⁽²⁾ , Oscillation capacity low | | 2.2 | | μA |
| | | | | While clock stops, Topr = 25° C | | 0.7 | 3 | μA |
| Idet4 | Low voltage detection dissipation current ⁽⁴⁾ | | 0.6 | 4 | μA | | | |
| Idet3 | Reset level detection dissipation current ⁽⁴⁾ | | 1.0 | 5 | μA | | | |

NOTES:

1. Referenced to V_{CC} = 2.7 to 3.6 V, V_{SS} = 0 V at Topr = -20 to 85 ° C / -40 to 85 ° C, f(BCLK) = 10 MHz unless otherwise specified.
2. With one timer operates, using f_{C32}.
3. This indicates the memory in which the program to be executed exists.
4. I_{det} is dissipation current when the following bit is set to 1 (detection circuit enabled).
 I_{det4}: the VC27 bit of the VCR2 register
 I_{det3}: the VC26 bit in the VCR2 register

$V_{CC} = 3V$

Timing Requirements**($V_{CC} = 3V$, $V_{SS} = 0V$, at $T_{opr} = -20$ to $85^{\circ}C$ / -40 to $85^{\circ}C$ unless otherwise specified)****Table 21.26 External Clock Input (XIN input)**

| Symbol | Parameter | Standard | | Unit |
|------------|---------------------------------------|----------|------|------|
| | | Min. | Max. | |
| t_c | External clock input cycle time | 100 | | ns |
| $t_{w(H)}$ | External clock input HIGH pulse width | 40 | | ns |
| $t_{w(L)}$ | External clock input LOW pulse width | 40 | | ns |
| t_r | External clock rise time | | 18 | ns |
| t_f | External clock fall time | | 18 | ns |

$$V_{CC} = 3V$$

Timing Requirements

($V_{CC} = 3V$, $V_{SS} = 0V$, at $T_{opr} = -20$ to $85^{\circ}C$ / -40 to $85^{\circ}C$ unless otherwise specified)

Table 21.27 Timer A Input (Counter Input in Event Counter Mode)

| Symbol | Parameter | Standard | | Unit |
|--------------|--|----------|------|------|
| | | Min. | Max. | |
| $t_{c(TA)}$ | TA _{IIN} input cycle time | 150 | | ns |
| $t_{w(TAH)}$ | TA _{IIN} input HIGH pulse width | 60 | | ns |
| $t_{w(TAL)}$ | TA _{IIN} input LOW pulse width | 60 | | ns |

Table 21.28 Timer A Input (Gating Input in Timer Mode)

| Symbol | Parameter | Standard | | Unit |
|--------------|--|----------|------|------|
| | | Min. | Max. | |
| $t_{c(TA)}$ | TA _{IIN} input cycle time | 600 | | ns |
| $t_{w(TAH)}$ | TA _{IIN} input HIGH pulse width | 300 | | ns |
| $t_{w(TAL)}$ | TA _{IIN} input LOW pulse width | 300 | | ns |

Table 21.29 Timer A Input (External Trigger Input in One-shot Timer Mode)

| Symbol | Parameter | Standard | | Unit |
|--------------|--|----------|------|------|
| | | Min. | Max. | |
| $t_{c(TA)}$ | TA _{IIN} input cycle time | 300 | | ns |
| $t_{w(TAH)}$ | TA _{IIN} input HIGH pulse width | 150 | | ns |
| $t_{w(TAL)}$ | TA _{IIN} input LOW pulse width | 150 | | ns |

Table 21.30 Timer A Input (External Trigger Input in Pulse Width Modulation Mode)

| Symbol | Parameter | Standard | | Unit |
|--------------|--|----------|------|------|
| | | Min. | Max. | |
| $t_{w(TAH)}$ | TA _{IIN} input HIGH pulse width | 150 | | ns |
| $t_{w(TAL)}$ | TA _{IIN} input LOW pulse width | 150 | | ns |

Table 21.31 Timer A Input (Counter Increment/decrement Input in Event Counter Mode)

| Symbol | Parameter | Standard | | Unit |
|------------------|---|----------|------|------|
| | | Min. | Max. | |
| $t_{c(UP)}$ | TA _{IOUT} input cycle time | 3000 | | ns |
| $t_{w(UPH)}$ | TA _{IOUT} input HIGH pulse width | 1500 | | ns |
| $t_{w(UPL)}$ | TA _{IOUT} input LOW pulse width | 1500 | | ns |
| $t_{su(UP-TIN)}$ | TA _{IOUT} input setup time | 600 | | ns |
| $t_h(TIN-UP)$ | TA _{IOUT} input hold time | 600 | | ns |

Table 21.32 Timer A Input (Two-phase Pulse Input in Event Counter Mode)

| Symbol | Parameter | Standard | | Unit |
|----------------------|-------------------------------------|----------|------|------|
| | | Min. | Max. | |
| $t_{c(TA)}$ | TA _{IIN} input cycle time | 2 | | μs |
| $t_{su(TAIN-TAOUT)}$ | TA _{IOUT} input setup time | 500 | | ns |
| $t_{su(TAOUT-TAIN)}$ | TA _{IIN} input setup time | 500 | | ns |

$V_{CC} = 3V$

Timing Requirements**($V_{CC} = 3V$, $V_{SS} = 0V$, at $T_{op} = -20$ to $85^{\circ}C$ / -40 to $85^{\circ}C$ unless otherwise specified)****Table 21.33 Timer B Input (Counter Input in Event Counter Mode)**

| Symbol | Parameter | Standard | | Unit |
|--------------|--|----------|------|------|
| | | Min. | Max. | |
| $t_{c(TB)}$ | TBiIN input cycle time (counted on one edge) | 150 | | ns |
| $t_{w(TBH)}$ | TBiIN input HIGH pulse width (counted on one edge) | 60 | | ns |
| $t_{w(TBL)}$ | TBiIN input LOW pulse width (counted on one edge) | 60 | | ns |
| $t_{c(TB)}$ | TBiIN input cycle time (counted on both edges) | 300 | | ns |
| $t_{w(TBH)}$ | TBiIN input HIGH pulse width (counted on both edges) | 120 | | ns |
| $t_{w(TBL)}$ | TBiIN input LOW pulse width (counted on both edges) | 120 | | ns |

Table 21.34 Timer B Input (Pulse Period Measurement Mode)

| Symbol | Parameter | Standard | | Unit |
|--------------|------------------------------|----------|------|------|
| | | Min. | Max. | |
| $t_{c(TB)}$ | TBiIN input cycle time | 600 | | ns |
| $t_{w(TBH)}$ | TBiIN input HIGH pulse width | 300 | | ns |
| $t_{w(TBL)}$ | TBiIN input LOW pulse width | 300 | | ns |

Table 21.35 Timer B Input (Pulse Width Measurement Mode)

| Symbol | Parameter | Standard | | Unit |
|--------------|------------------------------|----------|------|------|
| | | Min. | Max. | |
| $t_{c(TB)}$ | TBiIN input cycle time | 600 | | ns |
| $t_{w(TBH)}$ | TBiIN input HIGH pulse width | 300 | | ns |
| $t_{w(TBL)}$ | TBiIN input LOW pulse width | 300 | | ns |

Table 21.36 A/D Trigger Input

| Symbol | Parameter | Standard | | Unit |
|--------------|--|----------|------|------|
| | | Min. | Max. | |
| $t_{c(AD)}$ | \overline{ADTRG} input cycle time (trigger able minimum) | 1500 | | ns |
| $t_{w(ADL)}$ | \overline{ADTRG} input LOW pulse width | 200 | | ns |

Table 21.37 Serial I/O

| Symbol | Parameter | Standard | | Unit |
|---------------|-----------------------------|----------|------|------|
| | | Min. | Max. | |
| $t_{c(CK)}$ | CLKi input cycle time | 300 | | ns |
| $t_{w(CKH)}$ | CLKi input HIGH pulse width | 150 | | ns |
| $t_{w(CKL)}$ | CLKi input LOW pulse width | 150 | | ns |
| $t_{d(C-Q)}$ | TxDi output delay time | | 160 | ns |
| $t_{h(C-Q)}$ | TxDi hold time | 0 | | ns |
| $t_{su(D-C)}$ | RxDi input setup time | 100 | | ns |
| $t_{h(C-D)}$ | RxDi input hold time | 90 | | ns |

Table 21.38 External Interrupt \overline{INTi} Input

| Symbol | Parameter | Standard | | Unit |
|--------------|--|----------|------|------|
| | | Min. | Max. | |
| $t_{w(INH)}$ | \overline{INTi} input HIGH pulse width | 380 | | ns |
| $t_{w(INL)}$ | \overline{INTi} input LOW pulse width | 380 | | ns |

$$V_{CC} = 3V$$

Timing Requirements

($V_{CC} = 3V$, $V_{SS} = 0V$, at $T_{opr} = -20$ to $85^{\circ}C$ / -40 to $85^{\circ}C$ unless otherwise specified)

Table 21.39 Multi-master I²C bus Line

| Symbol | Parameter | Standard clock mode | | High-speed clock mode | | Unit |
|---------|-------------------------------------|---------------------|------|-----------------------|------|------|
| | | Min. | Max. | Min. | Max. | |
| tBUF | Bus free time | 4.7 | | 1.3 | | μs |
| tHD;STA | The hold time in start condition | 4.0 | | 0.6 | | μs |
| tLOW | The hold time in SCL clock 0 status | 4.7 | | 1.3 | | μs |
| tR | SCL, SDA signals' rising time | | 1000 | 20+0.1Cb | 300 | ns |
| tHD;DAT | Data hold time | 0 | | 0 | 0.9 | μs |
| tHIGH | The hold time in SCL clock 1 status | 4.0 | | 0.6 | | μs |
| tF | SCL, SDA signals' falling time | | 300 | 20+0.1Cb | 300 | ns |
| tsu;DAT | Data setup time | 250 | | 100 | | ns |
| tsu;STA | The setup time in restart condition | 4.7 | | 0.6 | | μs |
| tsu;STO | Stop condition setup time | 4.0 | | 0.6 | | μs |

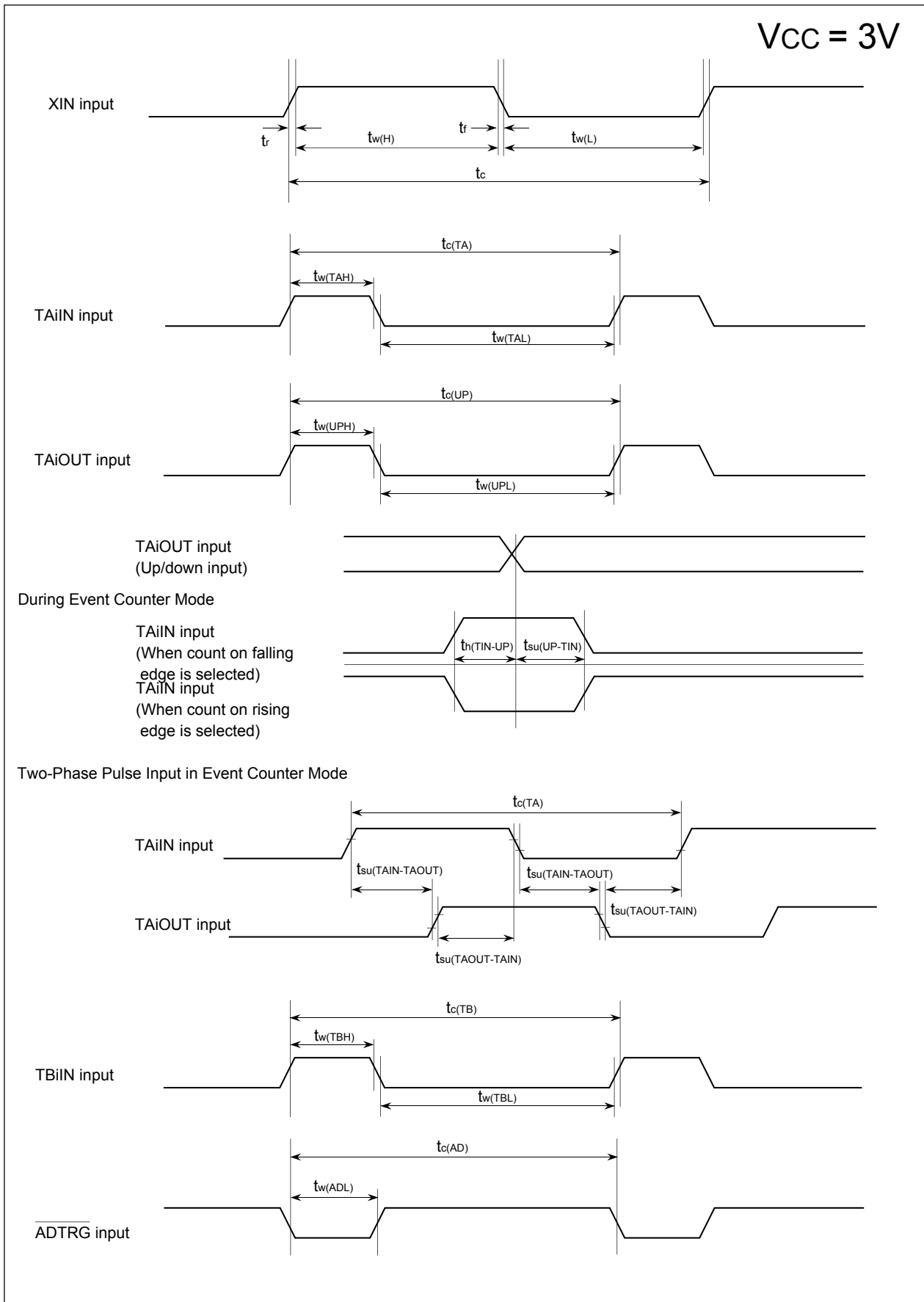


Figure 21.4 Timing Diagram (1)

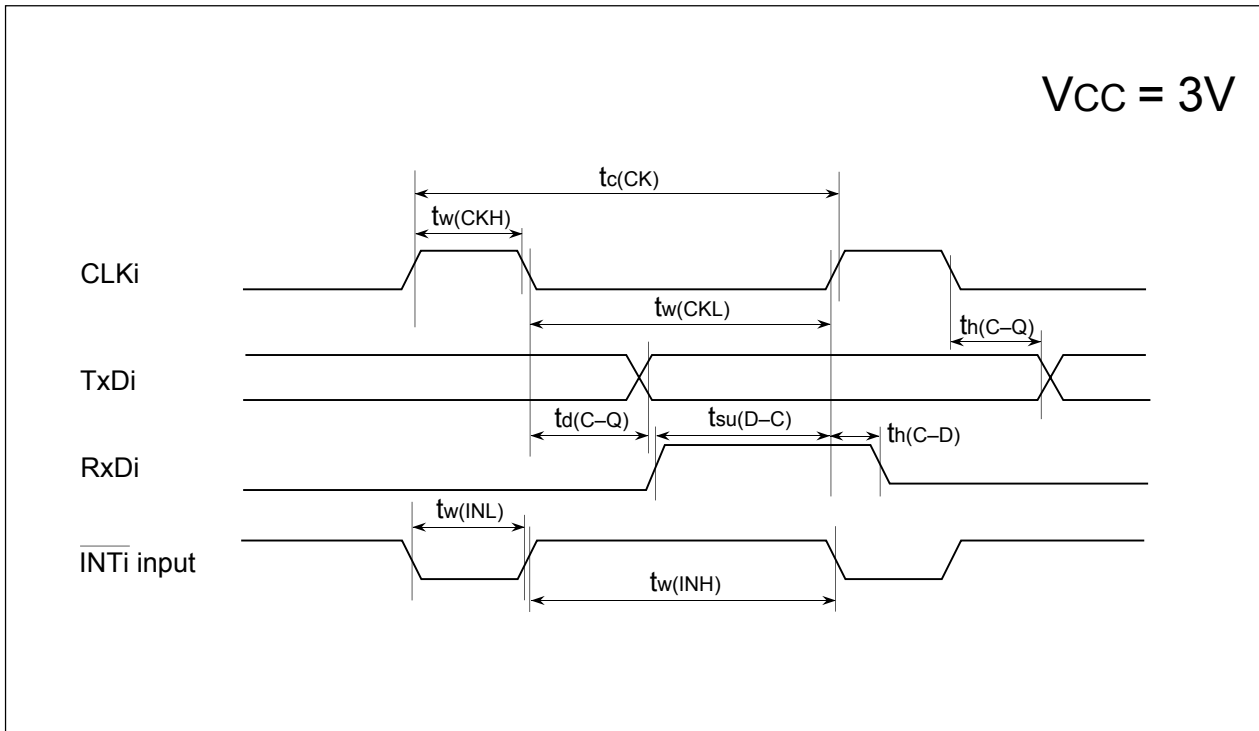


Figure 21.5 Timing Diagram (2)

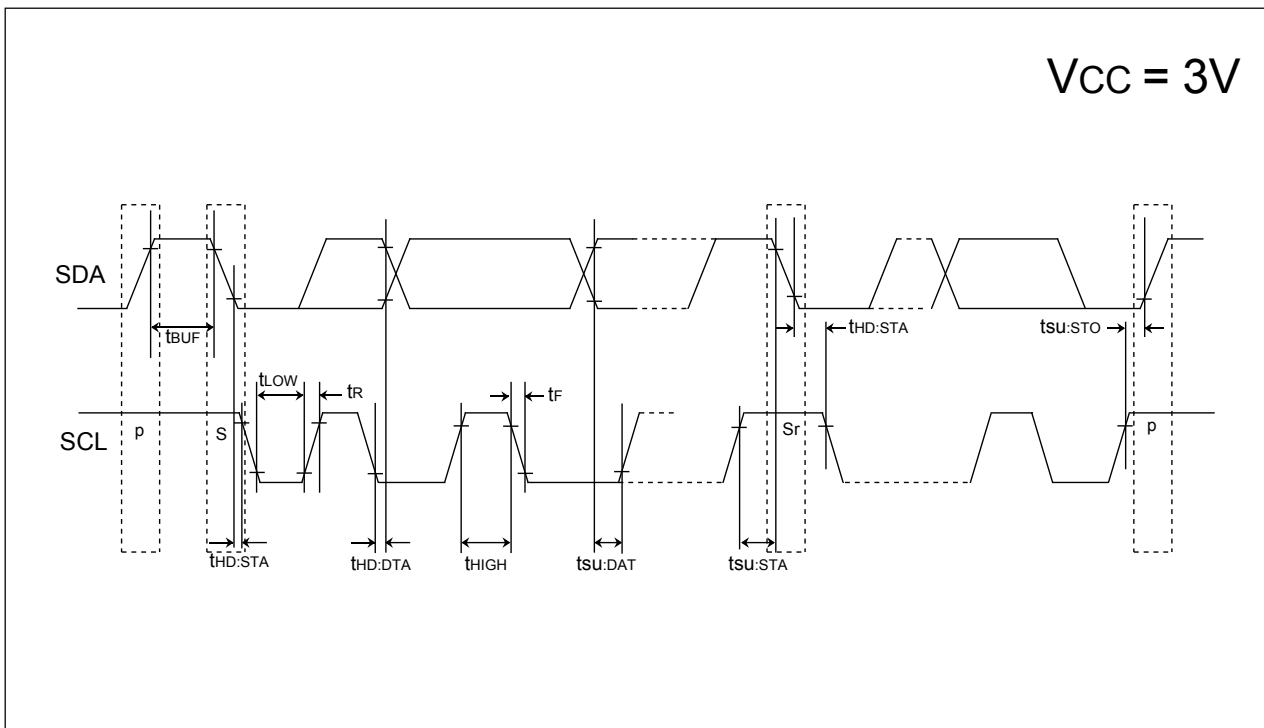


Figure 21.6 Timing Diagram (3)

21.2 T version

Table 21.40 Absolute Maximum Ratings

| Symbol | Parameter | | Condition | 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 | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ , X _{IN} , V _{REF} , RESET, CNV _{SS} | | -0.3 to V _{CC} +0.3 | V |
| V _O | Output Voltage | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ , X _{OUT} | | -0.3 to V _{CC} +0.3 | V |
| P _d | Power Dissipation | | -40≤T _{opr} ≤85° C | 300 | mW |
| T _{opr} | Operating Ambient Temperature | during CPU operation | | -40 to 85 | ° C |
| | | during flash memory program and erase operation | Program Space (Block 0 to Block 5) | 0 to 60 | ° C |
| | | | Data Space (Block A, Block B) | -40 to 85 | ° C |
| T _{stg} | Storage Temperature | | | -65 to 150 | ° C |

Table 21.41 Recommended Operating Conditions (Note 1)

| Symbol | Parameter | | Standard | | | Unit | | | |
|-----------------------|---|---|---|--------------------|------------------------------------|--------------------|---|-----------------|---|
| | | | Min. | Typ. | Max. | | | | |
| V _{CC} | Supply Voltage | | 3.0 | | 5.5 | V | | | |
| AV _{CC} | Analog Supply Voltage | | | V _{CC} | | V | | | |
| V _{SS} | Supply Voltage | | | 0 | | V | | | |
| AV _{SS} | Analog Supply Voltage | | | 0 | | V | | | |
| V _{IH} | Input High ("H") Voltage | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | 0.7V _{CC} | | V _{CC} | V | | | |
| | | XIN, RESET, CNVSS | 0.8V _{CC} | | V _{CC} | V | | | |
| | | SDA _{MM} , SCL _{MM} | When I ² C bus input level is selected | 0.7V _{CC} | | V _{CC} | V | | |
| | | | | | When SMBUS input level is selected | 1.4 | | V _{CC} | V |
| V _{IL} | Input Low ("L") Voltage | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | 0 | | 0.3V _{CC} | V | | | |
| | | XIN, $\overline{\text{RESET}}$, CNVSS | 0 | | 0.2V _{CC} | V | | | |
| | | SDA _{MM} , SCL _{MM} | When I ² C bus input level is selected | 0 | | 0.3V _{CC} | V | | |
| | | | | | When SMBUS input level is selected | 0 | | 0.6 | V |
| I _{OH(peak)} | Peak Output High ("H") Current | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | | | -10.0 | mA | | | |
| I _{OH(avg)} | Average Output High ("H") Current | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | | | -5.0 | mA | | | |
| I _{OL(peak)} | Peak Output Low ("L") Current | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | | | 10.0 | mA | | | |
| I _{OL(avg)} | Average Output Low ("L") Current | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | | | 5.0 | mA | | | |
| f(X _{IN}) | Main Clock Input Oscillation Frequency ⁽⁴⁾ | | 0 | | 20 | MHz | | | |
| f(X _{CIN}) | Sub Clock Oscillation Frequency | | | 32.768 | 50 | kHz | | | |
| f ₁ (ROC) | On-chip Oscillator Frequency 1 | | 0.5 | 1 | 2 | MHz | | | |
| f ₂ (ROC) | On-chip Oscillator Frequency 2 | | 1 | 2 | 4 | MHz | | | |
| f ₃ (ROC) | On-chip Oscillator Frequency 3 | | 8 | 16 | 26 | MHz | | | |
| f(PLL) | PLL Clock Oscillation Frequency ⁽⁴⁾ | | 10 | | 20 | MHz | | | |
| f(BCLK) | CPU Operation Clock Frequency | | 0 | | 20 | MHz | | | |
| t _{su} (PLL) | Wait Time to Stabilize PLL Frequency Synthesizer | V _{CC} =5.0V | | | 20 | ms | | | |
| | | V _{CC} =3.0V | | | 50 | ms | | | |

NOTES:

1. Referenced to V_{CC} = 3.0 to 5.5V at Topr = -40 to 85 °C unless otherwise specified.
2. The mean output current is the mean value within 100ms.
3. The total I_{OL(peak)} for all ports must be 80mA or less. The total I_{OH(peak)} for all ports must be -80mA or less.
4. Relationship among main clock oscillation frequency, PLL clock oscillation frequency and supply voltage.

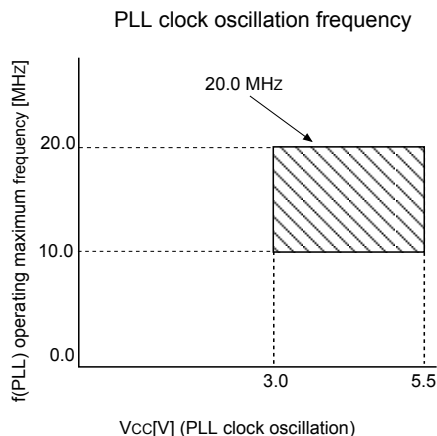
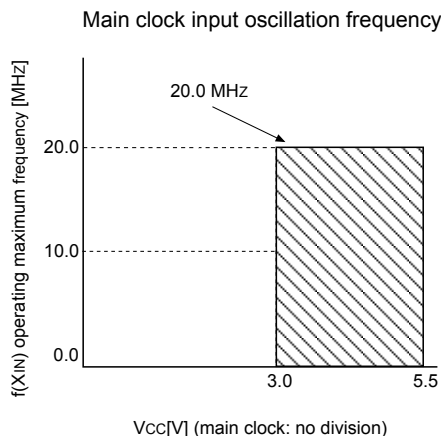


Table 21.42 A/D Conversion Characteristics (Note 1)

| Symbol | Parameter | | Measurement Condition | Standard | | | Unit |
|------------|--|--------|---|----------|------|-----------|---------------|
| | | | | Min. | Typ. | Max. | |
| - | Resolution | | $V_{REF} = V_{CC}$ | | | 10 | Bits |
| INL | Integral Nonlinearity Error | 10 bit | $V_{REF} = V_{CC} = 5\text{ V}$ | | | ± 3 | LSB |
| | | | $V_{REF} = V_{CC} = 3.3\text{ V}$ | | | ± 5 | LSB |
| | | 8 bit | $V_{REF} = V_{CC} = 3.3\text{ V}$ | | | ± 2 | LSB |
| - | Absolute Accuracy | 10 bit | $V_{REF} = V_{CC} = 5\text{ V}$ | | | ± 3 | LSB |
| | | | $V_{REF} = V_{CC} = 3.3\text{ V}$ | | | ± 5 | LSB |
| | | 8 bit | $V_{REF} = V_{CC} = 3.3\text{ V}$ | | | ± 2 | LSB |
| DNL | Differential Nonlinearity Error | | | | | ± 1 | LSB |
| - | Offset Error | | | | | ± 3 | LSB |
| - | Gain Error | | | | | ± 3 | LSB |
| RLADDER | Resistor Ladder | | $V_{REF} = V_{CC}$ | 10 | | 40 | k Ω |
| t_{CONV} | 10-bit Conversion Time Sample & Hold Function Available | | $V_{REF} = V_{CC} = 5\text{ V}$, $\phi_{AD} = 10\text{ MHz}$ | 3.3 | | | μs |
| t_{CONV} | 8-bit Conversion Time Sample & Hold Function Available | | $V_{REF} = V_{CC} = 5\text{ V}$, $\phi_{AD} = 10\text{ MHz}$ | 2.8 | | | μs |
| V_{REF} | Reference Voltage | | | 2.0 | | V_{CC} | V |
| V_{IA} | Analog Input Voltage | | | 0 | | V_{REF} | V |

NOTES:

1. Referenced to $V_{CC} = AV_{CC} = V_{REF} = 3.3$ to 5.5 V , $V_{SS} = AV_{SS} = 0\text{ V}$ at $T_{opr} = -40$ to 85° C unless otherwise specified.
2. Keep ϕ_{AD} frequency at 10 MHz or less. Additionally, divide the f_{AD} if V_{CC} is less than 4.2 V , and make ϕ_{AD} frequency equal to or lower than $f_{AD}/2$.
3. When sample & hold function is disabled, keep ϕ_{AD} frequency at 250 kHz or more in addition to the limitation in Note 2. When sample & hold function is enabled, keep ϕ_{AD} frequency at 1 MHz or more in addition to the limitation in Note 2.
4. When sample & hold function is enabled, sampling time is $3/\phi_{AD}$ frequency. When sample & hold function is disabled, sampling time is $2/\phi_{AD}$ frequency.

Table 21.43 Flash Memory Version Electrical Characteristics ⁽¹⁾ for 100/1000 E/W cycle products
[Program Space and Data Space in U3; Program Space in U7]

| Symbol | Parameter | Standard | | | Unit |
|-----------------|---|-----------------------------|---------------------|------|--------|
| | | Min. | Typ. ⁽²⁾ | Max. | |
| - | Program and Erase Endurance ⁽³⁾ | 100/1000 ^(4, 11) | | | cycles |
| - | Word Program Time (V _{CC} = 5.0 V, Topr = 25° C) | | 75 | 600 | μs |
| - | Block Erase Time (V _{CC} = 5.0 V, Topr = 25° C) | 2-Kbyte Block | 0.2 | 9 | s |
| | | 8-Kbyte Block | 0.4 | 9 | s |
| | | 16-Kbyte Block | 0.7 | 9 | s |
| | | 32-Kbyte Block | 1.2 | 9 | s |
| td(SR-ES) | Duration between Suspend Request and Erase Suspend | | | 8 | ms |
| t _{rs} | Wait Time to Stabilize Flash Memory Circuit | | | 15 | μs |
| - | Data Hold Time ⁽⁵⁾ | 20 | | | years |

Table 21.44 Flash Memory Version Electrical Characteristics ⁽⁶⁾ for 10000 E/W cycle products
[Data Space in U7⁽⁷⁾]

| Symbol | Parameter | Standard | | | Unit |
|-----------------|--|--------------------------|---------------------|------|--------|
| | | Min. | Typ. ⁽²⁾ | Max. | |
| - | Program and Erase Endurance ^(3, 8, 9) | 10000 ^(4, 10) | | | cycles |
| - | Word Program Time (V _{CC} = 5.0 V, Topr = 25° C) | | 100 | | μs |
| - | Block Erase Time (V _{CC} = 5.0V, Topr = 25° C) (2-Kbyte block) | | 0.3 | | s |
| td(SR-ES) | Duration between Suspend Request and Erase Suspend | | | 8 | ms |
| t _{rs} | Wait Time to Stabilize Flash Memory Circuit | | | 15 | μs |
| - | Data Hold Time ⁽⁵⁾ | 20 | | | years |

NOTES:

1. Referenced to VCC = 3.0 to 5.5 V at Topr = 0 to 60° C (program space)/ Topr = -40 to 85° C (data space), unless otherwise specified.
2. VCC = 5.0 V; TOPR = 25° C
3. Program and erase endurance is defined as number of program-erase cycles per block.
 If program and erase endurance is *n* cycle (*n* = 100, 1000, 10000), each block can be erased and programmed *n* cycles.
 For example, if a 2-Kbyte block A is erased after programming one-word data to each address 1,024 times, this counts as one program and erase endurance. Data cannot be programmed to the same address more than once without erasing the block. (rewrite prohibited).
4. Number of E/W cycles for which operation is guaranteed (1 to minimum value are guaranteed).
5. Topr = 55° C
6. Referenced to VCC = 3.0 to 5.5 V at Topr = -40 to 85° C unless otherwise specified.
7. **Table 21.44** applies for data space in U7 when program and erase endurance is more than 1,000 cycles. Otherwise, use **Table 21.43**.
8. To reduce the number of program and erase endurance when working with systems requiring numerous rewrites, write to unused word addresses within the block instead of rewrite. Erase block only after all possible addresses are used. For example, an 8-word program can be written 128 times maximum before erase becomes necessary. Maintaining an equal number of times erasure between block A and block B will also improve efficiency. It is recommended to track the total number of erasure performed per block and to limit the number of erasure.
9. If an erase error is generated during block erase, execute the clear status register command and block erase command at least 3 times until an erase error is not generated.
10. When executing more than 100 times rewrites, set one wait state per block access by setting the FMR17 bit in the FMR1 register to 1 (wait state). When accessing to all other blocks and internal RAM, wait state can be set by the PM17 bit, regardless of the FMR17 bit setting value.
11. The program and erase endurance is 100 cycles for program space and data space in U3; 1,000 cycles for program space in U7.
12. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for further details on the E/W failure rate.

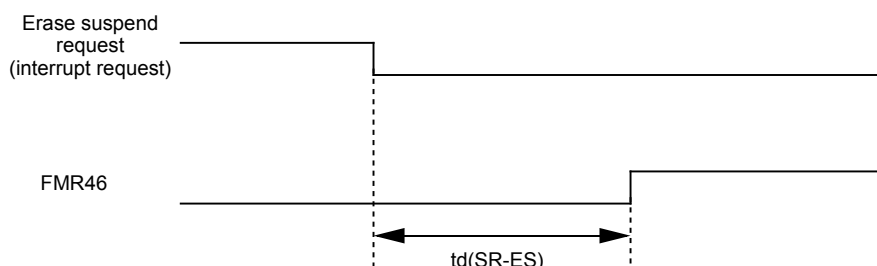
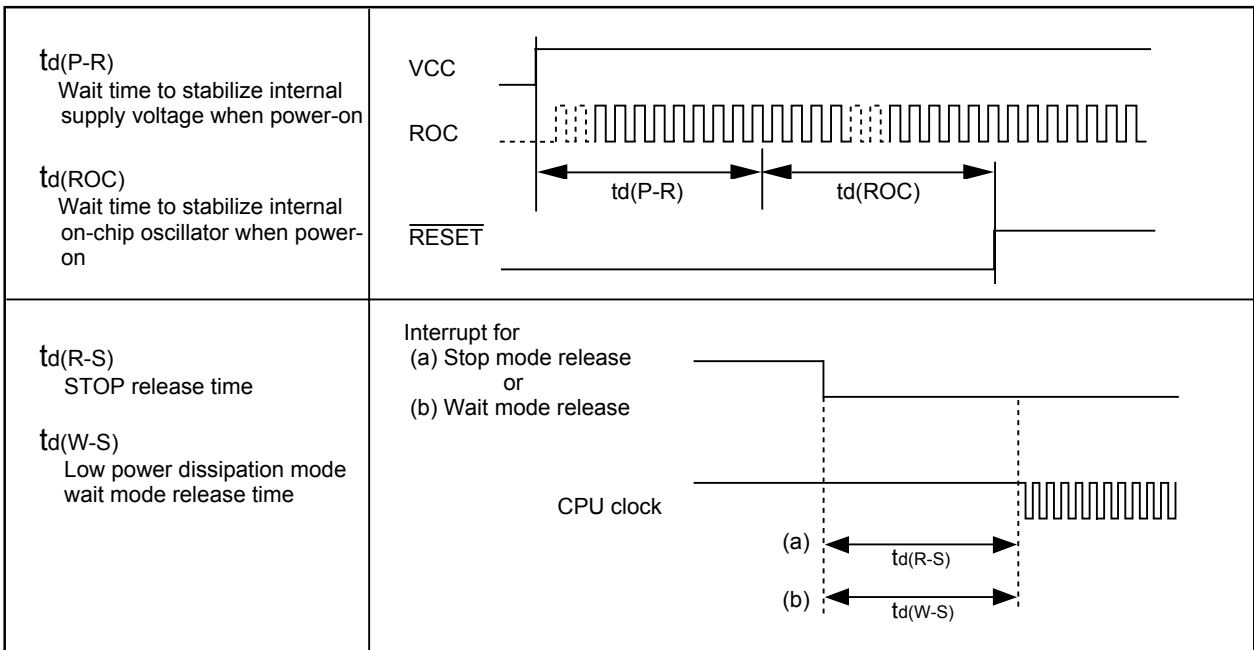


Table 21.45 Power Supply Circuit Timing Characteristics

| Symbol | Parameter | Measurement Condition | Standard | | | Unit |
|---------|--|-----------------------|----------|------|------|------|
| | | | Min. | Typ. | Max. | |
| td(P-R) | Wait Time to Stabilize Internal Supply Voltage when Power-on | Vcc = 3.0 to 5.5V | | | 2 | ms |
| td(ROC) | Wait Time to Stabilize Internal On-chip Oscillator when Power-on | | | | 40 | μs |
| td(R-S) | STOP Release Time ⁽¹⁾ | | | | 150 | μs |
| td(W-S) | Low Power Dissipation Mode Wait Mode Release Time | | | | 150 | μs |



V_{CC} = 5V**Table 21.46 Electrical Characteristics (Note 1)**

| Symbol | Parameter | | Condition | Standard | | | Unit |
|----------------------------------|---------------------------|---|---------------------------|---------------------------|----------------------|-----------------|------|
| | | | | Min. | Typ. | Max. | |
| V _{OH} | Output High ("H") Voltage | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | I _{OH} = -5 mA | V _{CC} -2.0 | | V _{CC} | V |
| V _{OH} | Output High ("H") Voltage | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | I _{OH} = -200 μA | V _{CC} -0.3 | | V _{CC} | V |
| V _{OH} | Output High ("H") Voltage | X _{OUT} | High Power | I _{OH} = -1 mA | V _{CC} -2.0 | V _{CC} | V |
| | | | Low Power | I _{OH} = -0.5 mA | V _{CC} -2.0 | V _{CC} | |
| | Output High ("H") Voltage | X _{COUT} | High Power | No load applied | | 2.5 | V |
| | | | Low Power | No load applied | | 1.6 | |
| V _{OL} | Output Low ("L") Voltage | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | I _{OL} = 5 mA | | | 2.0 | V |
| V _{OL} | Output Low ("L") Voltage | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | I _{OL} = 200 μA | | | 0.45 | V |
| V _{OL} | Output Low ("L") Voltage | X _{OUT} | High Power | I _{OL} = 1 mA | | 2.0 | V |
| | | | Low Power | I _{OL} = 0.5 mA | | 2.0 | |
| | Output Low ("L") Voltage | X _{COUT} | High Power | No load applied | | 0 | V |
| | | | Low Power | No load applied | | 0 | |
| V _{T+} -V _{T-} | Hysteresis | TA0 _{IN} -TA4 _{IN} , TB0 _{IN} -TB2 _{IN} , INT0-INT5, NMI, AD _{TRG} , CTS0-CTS2, SCL, SDA, CLK0-CLK2, TA2 _{OUT} -TA4 _{OUT} , K10-K13, RXD0-RXD2, SIN3, SIN4 | | 0.2 | | 1.0 | V |
| V _{T+} -V _{T-} | Hysteresis | RESET | | 0.2 | | 2.5 | V |
| V _{T+} -V _{T-} | Hysteresis | X _{IN} | | 0.2 | | 0.8 | V |
| I _{IH} | Input High ("H") Current | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ X _{IN} , RESET, CNV _{SS} | V _I = 5 V | | | 5.0 | μA |
| I _{IL} | Input Low ("L") Current | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ X _{IN} , RESET, CNV _{SS} | V _I = 0 V | | | -5.0 | μA |
| R _{PULLUP} | Pull-up Resistance | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | V _I = 0 V | 30 | 50 | 170 | kΩ |
| R _{fXIN} | Feedback Resistance | X _{IN} | | | 1.5 | | MΩ |
| R _{fXCIN} | Feedback Resistance | X _{CIN} | | | 15 | | MΩ |
| V _{RAM} | RAM Standby Voltage | | In stop mode | 2.0 | | | V |

NOTES:

1. Referenced to V_{CC}=4.2 to 5.5V, V_{SS}=0V at T_{opr}=-40 to 85 ° C, f(BCLK)=20MHz unless otherwise specified.

$V_{CC} = 5V$ **Table 21.47 Electrical Characteristics (2) (Note 1)**

| Symbol | Parameter | Measurement Condition | | | Standard | | | Unit |
|----------|--|--|--|---|---|------|------|---------|
| | | | | | Min. | Typ. | Max. | |
| I_{CC} | Power Supply Current ($V_{CC}=4.2$ to $5.5V$) | Output pins are left open and other pins are connected to V_{SS} | Mask ROM | $f(BCLK) = 20$ MHz, main clock, no division | | 18 | 25 | mA |
| | | | | On-chip oscillation, $f_{2(ROC)}$ selected, $f(BCLK) = 1$ MHz | | 2 | | mA |
| | | | Flash memory | $f(BCLK) = 20$ MHz, main clock, no division | | 18 | 25 | mA |
| | | | | On-chip oscillation, $f_{2(ROC)}$ selected, $f(BCLK) = 1$ MHz | | 2 | | mA |
| | | | Flash memory program | $f(BCLK) = 10$ MHz, $V_{CC} = 5.0$ V | | 11 | | mA |
| | | | Flash memory erase | $f(BCLK) = 10$ MHz, $V_{CC} = 5.0$ V | | 11 | | mA |
| | | | Mask ROM | $f(BCLK) = 32$ kHz, In low-power consumption mode, Program running on ROM ⁽³⁾ | | 25 | | μA |
| | | | | On-chip oscillation, $f_{2(ROC)}$ selected, $f(BCLK) = 1$ MHz, In wait mode | | 50 | | μA |
| | | | Flash memory | $f(BCLK) = 32$ kHz, In low-power consumption mode, Program running on RAM ⁽³⁾ | | 25 | | μA |
| | | | | $f(BCLK) = 32$ kHz, In low-power consumption mode, Program running on flash memory ⁽³⁾ | | 450 | | μA |
| | | | | On-chip oscillation, $f_{2(ROC)}$ selected, $f(BCLK) = 1$ MHz, In wait mode | | 50 | | μA |
| | | | | Mask ROM, Flash memory | $f(BCLK) = 32$ kHz, In wait mode ⁽²⁾ , Oscillation capacity high | | 8.5 | |
| | | | $f(BCLK) = 32$ kHz, In wait mode ⁽²⁾ , Oscillation capacity low | | | 3 | | μA |
| | | | While clock stops, $T_{opr} = 25^{\circ} C$ | | | 0.8 | 3 | μA |

NOTES:

1. Referenced to $V_{CC} = 4.2$ to 5.5 V, $V_{SS} = 0$ V at $T_{opr} = -40$ to $85^{\circ} C$, $f(BCLK) = 20$ MHz unless otherwise specified.
2. With one timer operates, using f_{C32} .
3. This indicates the memory in which the program to be executed exists.

$V_{CC} = 5V$

Timing Requirements**($V_{CC} = 5V$, $V_{SS} = 0V$, at $T_{opr} = -40$ to $85^{\circ}C$ unless otherwise specified)****Table 21.48 External Clock Input (XIN input)**

| Symbol | Parameter | Standard | | Unit |
|--------|---------------------------------------|----------|------|------|
| | | Min. | Max. | |
| tc | External Clock Input Cycle Time | 50 | | ns |
| tw(H) | External Clock Input High ("H") Width | 20 | | ns |
| tw(L) | External Clock Input Low ("L") Width | 20 | | ns |
| tr | External Clock Rise Time | | 9 | ns |
| tf | External Clock Fall Time | | 9 | ns |

$V_{CC} = 5V$

Timing Requirements(V_{CC} = 5V, V_{SS} = 0V, at Topr = – 40 to 85°C unless otherwise specified)**Table 21.49 Timer A Input (Counter Input in Event Counter Mode)**

| Symbol | Parameter | Standard | | Unit |
|---------------------|------------------------------|----------|------|------|
| | | Min. | Max. | |
| t _{c(TA)} | TAiIN input cycle time | 100 | | ns |
| t _{w(TAH)} | TAiIN input HIGH pulse width | 40 | | ns |
| t _{w(TAL)} | TAiIN input LOW pulse width | 40 | | ns |

Table 21.50 Timer A Input (Gating Input in Timer Mode)

| Symbol | Parameter | Standard | | Unit |
|---------------------|------------------------------|----------|------|------|
| | | Min. | Max. | |
| t _{c(TA)} | TAiIN input cycle time | 400 | | ns |
| t _{w(TAH)} | TAiIN input HIGH pulse width | 200 | | ns |
| t _{w(TAL)} | TAiIN input LOW pulse width | 200 | | ns |

Table 21.51 Timer A Input (External Trigger Input in One-shot Timer Mode)

| Symbol | Parameter | Standard | | Unit |
|---------------------|------------------------------|----------|------|------|
| | | Min. | Max. | |
| t _{c(TA)} | TAiIN input cycle time | 200 | | ns |
| t _{w(TAH)} | TAiIN input HIGH pulse width | 100 | | ns |
| t _{w(TAL)} | TAiIN input LOW pulse width | 100 | | ns |

Table 21.52 Timer A Input (External Trigger Input in Pulse Width Modulation Mode)

| Symbol | Parameter | Standard | | Unit |
|---------------------|------------------------------|----------|------|------|
| | | Min. | Max. | |
| t _{w(TAH)} | TAiIN input HIGH pulse width | 100 | | ns |
| t _{w(TAL)} | TAiIN input LOW pulse width | 100 | | ns |

Table 21.53 Timer A Input (Counter Increment/decrement Input in Event Counter Mode)

| Symbol | Parameter | Standard | | Unit |
|-------------------------|-------------------------------|----------|------|------|
| | | Min. | Max. | |
| t _{c(UP)} | TAiOUT input cycle time | 2000 | | ns |
| t _{w(UPH)} | TAiOUT input HIGH pulse width | 1000 | | ns |
| t _{w(UPL)} | TAiOUT input LOW pulse width | 1000 | | ns |
| t _{su(UP-TiN)} | TAiOUT input setup time | 400 | | ns |
| t _{h(TiN-UP)} | TAiOUT input hold time | 400 | | ns |

Table 21.54 Timer A Input (Two-phase Pulse Input in Event Counter Mode)

| Symbol | Parameter | Standard | | Unit |
|------------------------------|-------------------------|----------|------|------|
| | | Min. | Max. | |
| t _{c(TA)} | TAiIN input cycle time | 800 | | ns |
| t _{su(TAiN-TAiOUT)} | TAiOUT input setup time | 200 | | ns |
| t _{su(TAiOUT-TAiN)} | TAiIN input setup time | 200 | | ns |

$$V_{CC} = 5V$$

Timing Requirements

($V_{CC} = 5V$, $V_{SS} = 0V$, at $T_{op} = -40$ to $85^{\circ}C$ unless otherwise specified)

Table 21.55 Timer B Input (Counter Input in Event Counter Mode)

| Symbol | Parameter | Standard | | Unit |
|--------------|--|----------|------|------|
| | | Min. | Max. | |
| $t_{c(TB)}$ | TBiIN input cycle time (counted on one edge) | 100 | | ns |
| $t_{w(TBH)}$ | TBiIN input HIGH pulse width (counted on one edge) | 40 | | ns |
| $t_{w(TBL)}$ | TBiIN input LOW pulse width (counted on one edge) | 40 | | ns |
| $t_{c(TB)}$ | TBiIN input cycle time (counted on both edges) | 200 | | ns |
| $t_{w(TBH)}$ | TBiIN input HIGH pulse width (counted on both edges) | 80 | | ns |
| $t_{w(TBL)}$ | TBiIN input LOW pulse width (counted on both edges) | 80 | | ns |

Table 21.56 Timer B Input (Pulse Period Measurement Mode)

| Symbol | Parameter | Standard | | Unit |
|--------------|------------------------------|----------|------|------|
| | | Min. | Max. | |
| $t_{c(TB)}$ | TBiIN input cycle time | 400 | | ns |
| $t_{w(TBH)}$ | TBiIN input HIGH pulse width | 200 | | ns |
| $t_{w(TBL)}$ | TBiIN input LOW pulse width | 200 | | ns |

Table 21.57 Timer B Input (Pulse Width Measurement Mode)

| Symbol | Parameter | Standard | | Unit |
|--------------|------------------------------|----------|------|------|
| | | Min. | Max. | |
| $t_{c(TB)}$ | TBiIN input cycle time | 400 | | ns |
| $t_{w(TBH)}$ | TBiIN input HIGH pulse width | 200 | | ns |
| $t_{w(TBL)}$ | TBiIN input LOW pulse width | 200 | | ns |

Table 21.58 A/D Trigger Input

| Symbol | Parameter | Standard | | Unit |
|--------------|--|----------|------|------|
| | | Min. | Max. | |
| $t_{c(AD)}$ | \overline{ADTRG} input cycle time (trigger able minimum) | 1000 | | ns |
| $t_{w(ADL)}$ | \overline{ADTRG} input LOW pulse width | 125 | | ns |

Table 21.59 Serial I/O

| Symbol | Parameter | Standard | | Unit |
|---------------|-----------------------------|----------|------|------|
| | | Min. | Max. | |
| $t_{c(CK)}$ | CLKi input cycle time | 200 | | ns |
| $t_{w(CKH)}$ | CLKi input HIGH pulse width | 100 | | ns |
| $t_{w(CKL)}$ | CLKi input LOW pulse width | 100 | | ns |
| $t_{d(C-Q)}$ | TxDi output delay time | | 80 | ns |
| $t_{h(C-Q)}$ | TxDi hold time | 0 | | ns |
| $t_{su(D-C)}$ | RxDi input setup time | 70 | | ns |
| $t_{h(C-D)}$ | RxDi input hold time | 90 | | ns |

Table 21.60 External Interrupt \overline{INTi} Input

| Symbol | Parameter | Standard | | Unit |
|--------------|--|----------|------|------|
| | | Min. | Max. | |
| $t_{w(INH)}$ | \overline{INTi} input HIGH pulse width | 250 | | ns |
| $t_{w(INL)}$ | \overline{INTi} input LOW pulse width | 250 | | ns |

$V_{CC} = 5V$ **Timing Requirements****($V_{CC} = 5V$, $V_{SS} = 0V$, at $T_{opr} = -40$ to $85^{\circ}C$ unless otherwise specified)****Table 21.61 Multi-master I²C bus Line**

| Symbol | Parameter | Standard clock mode | | High-speed clock mode | | Unit |
|---------|-------------------------------------|---------------------|------|-----------------------|------|---------|
| | | Min. | Max. | Min. | Max. | |
| tBUF | Bus free time | 4.7 | | 1.3 | | μs |
| tHD;STA | The hold time in start condition | 4.0 | | 0.6 | | μs |
| tLOW | The hold time in SCL clock 0 status | 4.7 | | 1.3 | | μs |
| tR | SCL, SDA signals' rising time | | 1000 | $20+0.1C_b$ | 300 | ns |
| tHD;DAT | Data hold time | 0 | | 0 | 0.9 | μs |
| tHIGH | The hold time in SCL clock 1 status | 4.0 | | 0.6 | | μs |
| tF | SCL, SDA signals' falling time | | 300 | $20+0.1C_b$ | 300 | ns |
| tsu;DAT | Data setup time | 250 | | 100 | | ns |
| tsu;STA | The setup time in restart condition | 4.7 | | 0.6 | | μs |
| tsu;STO | Stop condition setup time | 4.0 | | 0.6 | | μs |

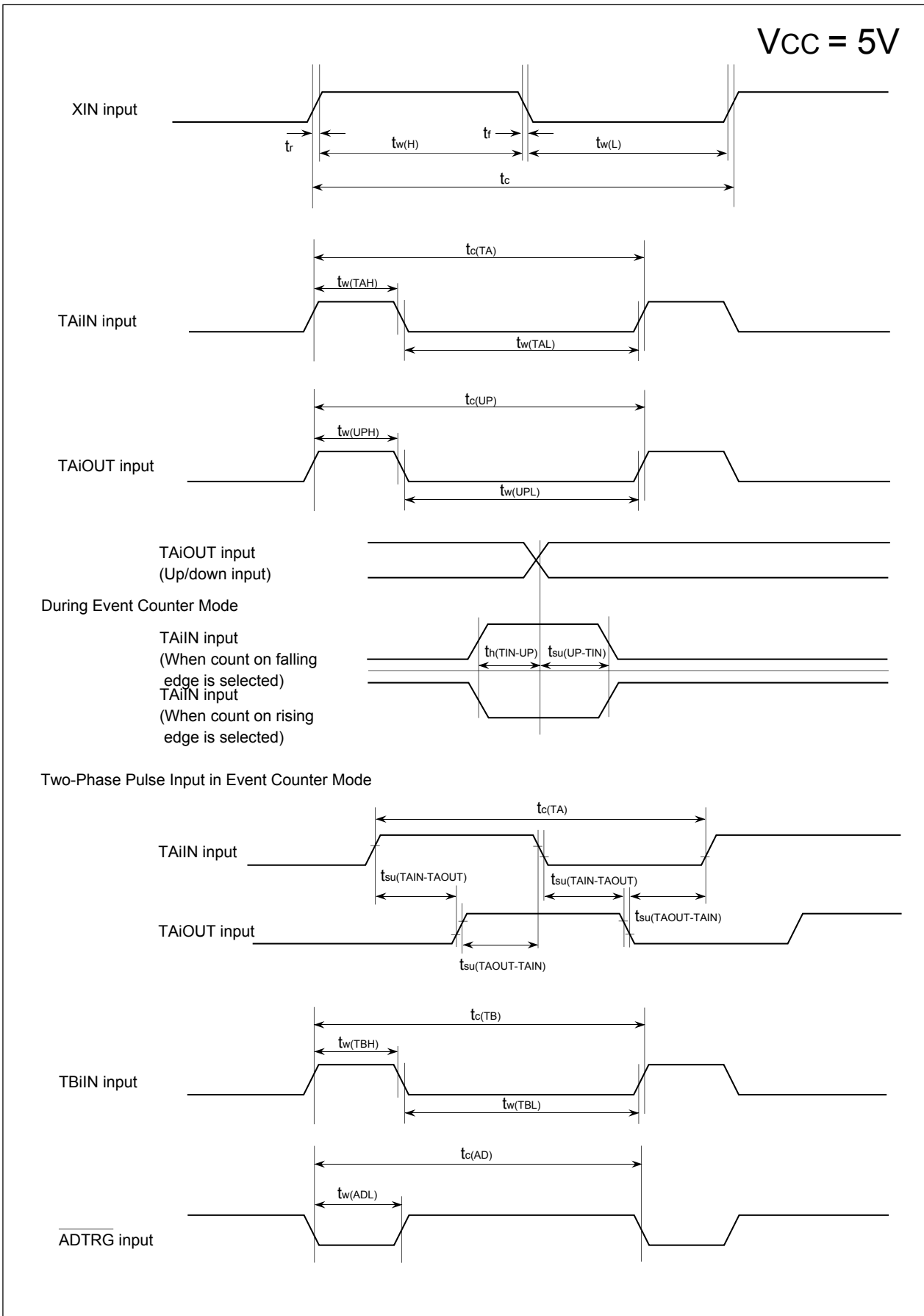


Figure 21.7 Timing Diagram (1)

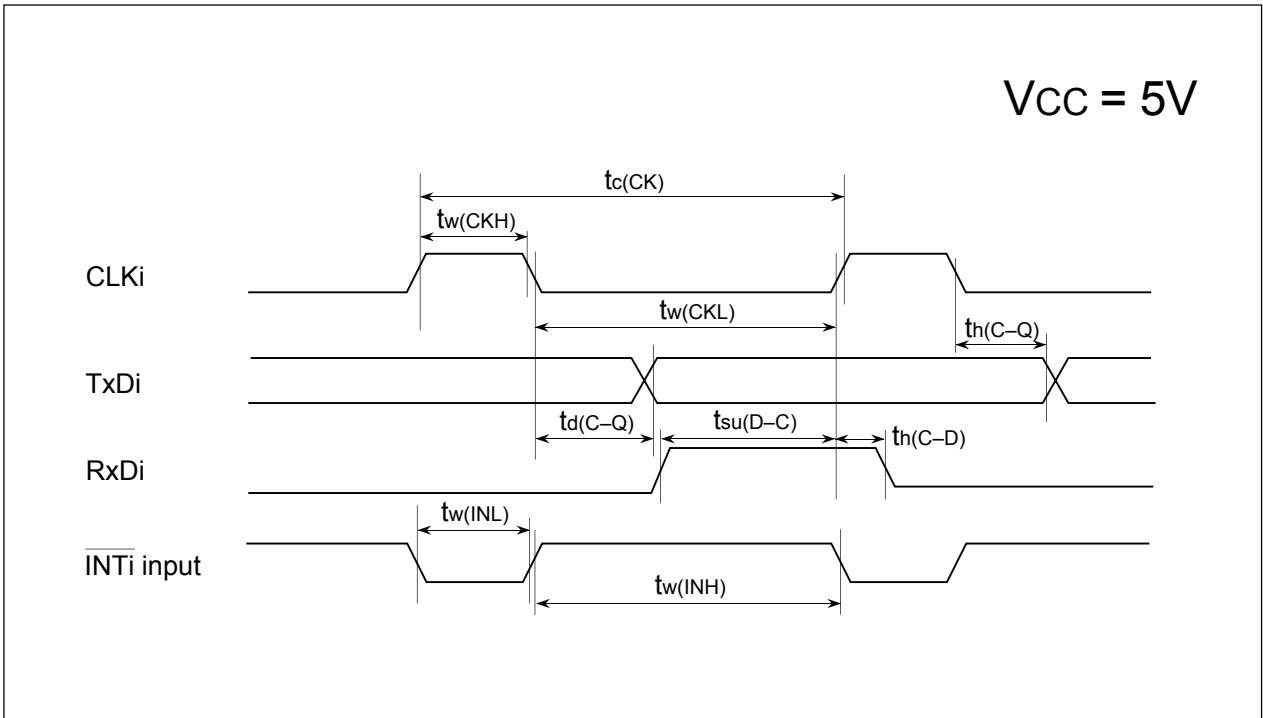


Figure 21.8 Timing Diagram (2)

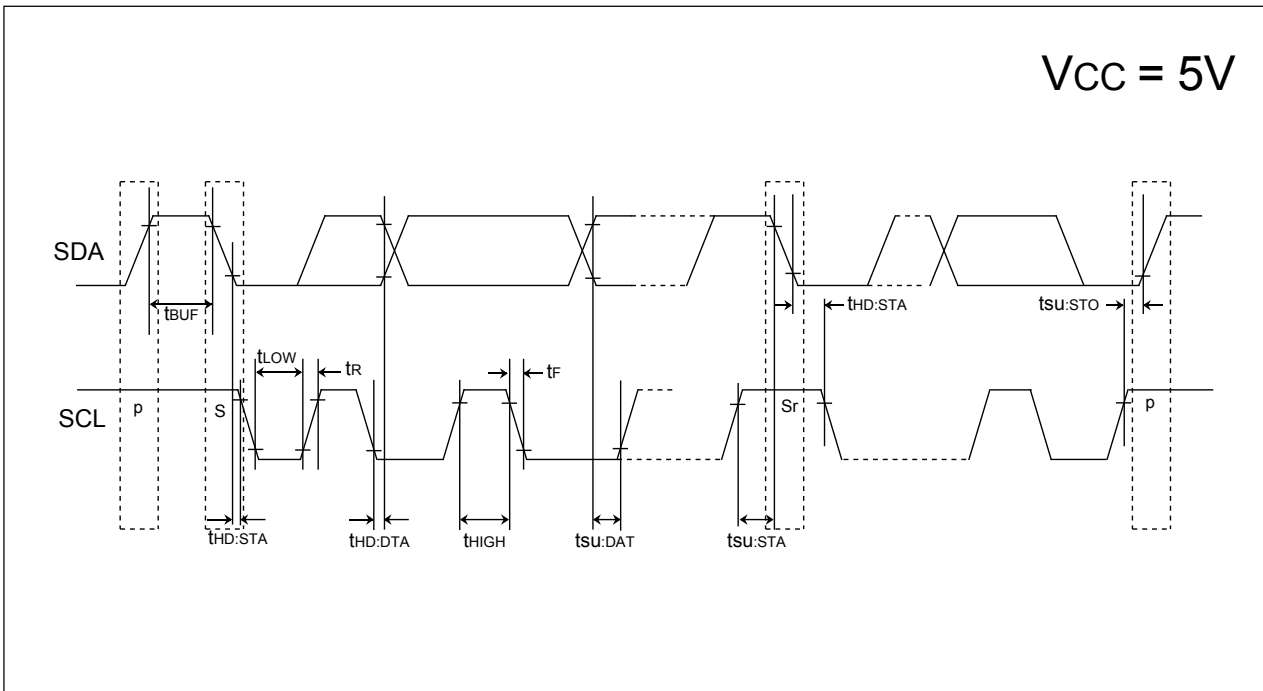


Figure 21.9 Timing Diagram (3)

V_{CC} = 3V**Table 21.62 Electrical Characteristics (Note)**

| Symbol | Parameter | | Condition | Standard | | | Unit |
|----------------------------------|---------------------------|---|-------------------------|---------------------------|----------------------|-----------------|------|
| | | | | Min. | Typ. | Max. | |
| V _{OH} | Output High ("H") Voltage | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | I _{OH} = -1 mA | V _{CC} -0.5 | | V _{CC} | V |
| V _{OH} | Output High ("H") Voltage | X _{OUT} | High Power | I _{OH} = -0.1 mA | V _{CC} -0.5 | V _{CC} | V |
| | | | Low Power | I _{OH} = -50 μA | V _{CC} -0.5 | V _{CC} | |
| | Output High ("H") Voltage | X _{COUT} | High Power | No load applied | | 2.5 | V |
| | | | Low Power | No load applied | | 1.6 | |
| V _{OL} | Output Low ("L") Voltage | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | I _{OL} = 1 mA | | | 0.5 | V |
| V _{OL} | Output Low ("L") Voltage | X _{OUT} | High Power | I _{OL} = 0.1 mA | | 0.5 | V |
| | | | Low Power | I _{OL} = 50 μA | | 0.5 | |
| | Output Low ("L") Voltage | X _{COUT} | High Power | No load applied | | 0 | V |
| | | | Low Power | No load applied | | 0 | |
| V _{T+} -V _{T-} | Hysteresis | TA0 _{IN} -TA4 _{IN} , TB0 _{IN} -TB2 _{IN} , INT0-INT5, NMI, AD _{TRG} , CTS0-C _{TS2} , SCL, SDA, CLK0-CLK2, TA2 _{OUT} -TA4 _{OUT} , KI0-KI3, RXD0-RXD2, SIN3, SIN4 | | | | 0.8 | V |
| V _{T+} -V _{T-} | Hysteresis | RESET | | | | 1.8 | V |
| V _{T+} -V _{T-} | Hysteresis | X _{IN} | | | | 0.8 | V |
| I _{IH} | Input High ("H") Current | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ X _{IN} , RESET, CNV _{SS} | V _I = 3 V | | | 4.0 | μA |
| I _{IL} | Input Low ("L") Current | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ X _{IN} , RESET, CNV _{SS} | V _I = 0 V | | | -4.0 | μA |
| R _{PULLUP} | Pull-up Resistance | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | V _I = 0 V | 50 | 100 | 500 | kΩ |
| R _{fXIN} | Feedback Resistance | X _{IN} | | | 3.0 | | MΩ |
| R _{fXCIN} | Feedback Resistance | X _{CIN} | | | 25 | | MΩ |
| V _{RAM} | RAM Standby Voltage | | In stop mode | 2.0 | | | V |

NOTE:

1. Referenced to V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V at Topr = -40 to 85 ° C, f(BCLK) = 20 MHz unless otherwise specified.

$V_{CC} = 3V$

Table 21.63 Electrical Characteristics (2) (Note 1)

| Symbol | Parameter | Measurement Condition | Standard | | | Unit | | |
|-----------------|--|---|------------------------|---|------|------|----|----|
| | | | Min. | Typ. | Max. | | | |
| I _{CC} | Power Supply Current (V _{CC} =3.0 to 3.6V) | Output pins are left open and other pins are connected to V _{SS} | Mask ROM | f(BCLK) = 10 MHz, main clock, no division | | 8 | 13 | mA |
| | | | | On-chip oscillation, f _{2(ROC)} selected, f(BCLK) = 1 MHz | | 1 | | mA |
| | | | Flash memory | f(BCLK) = 10 MHz, main clock, no division | | 8 | 13 | mA |
| | | | Flash memory program | f(BCLK) = 10 MHz, V _{CC} = 3.0 V | | 11 | | mA |
| | | | Flash memory erase | f(BCLK) = 10MHz, V _{CC} = 3.0 V | | 11 | | mA |
| | | | Mask ROM | f(BCLK) = 32 kHz, In low-power consumption mode, Program running on ROM ⁽³⁾ | | 20 | | μA |
| | | | | On-chip oscillation, f _{2(ROC)} selected, f(BCLK) = 1 MHz, In wait mode | | 25 | | μA |
| | | | Flash memory | f(BCLK) = 32 kHz, In low-power consumption mode, Program running on RAM ⁽³⁾ | | 20 | | μA |
| | | | | f(BCLK) = 32 kHz, In low-power consumption mode, Program running on flash memory ⁽³⁾ | | 450 | | μA |
| | | | | On-chip oscillation, f _{2(ROC)} selected, f(BCLK) = 1 MHz, In wait mode | | 45 | | μA |
| | | | Mask ROM, Flash memory | f(BCLK) = 32 kHz, In wait mode ⁽²⁾ , Oscillation capacity high | | 6.6 | | μA |
| | | | | f(BCLK) = 32 kHz, In wait mode ⁽²⁾ , Oscillation capacity low | | 2.2 | | μA |
| | | | | While clock stops, Topr = 25° C | | 0.7 | 3 | μA |

NOTES:

1. Referenced to V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V at Topr = -40 to 85 ° C, f(BCLK) = 20 MHz unless otherwise specified.
2. With one timer operates, using f_{CSZ}.
3. This indicates the memory in which the program to be executed exists.

$V_{CC} = 3V$

Timing Requirements**($V_{CC} = 3V$, $V_{SS} = 0V$, at $T_{opr} = -40$ to $85^{\circ}C$ unless otherwise specified)****Table 21.64 External Clock Input (X_{IN} input)**

| Symbol | Parameter | Standard | | Unit |
|------------|---------------------------------------|----------|------|------|
| | | Min. | Max. | |
| t_c | External clock input cycle time | 100 | | ns |
| $t_{w(H)}$ | External clock input HIGH pulse width | 40 | | ns |
| $t_{w(L)}$ | External clock input LOW pulse width | 40 | | ns |
| t_r | External clock rise time | | 18 | ns |
| t_f | External clock fall time | | 18 | ns |

$$V_{CC} = 3V$$

Timing Requirements

($V_{CC} = 3V$, $V_{SS} = 0V$, at $T_{opr} = -40$ to $85^{\circ}C$ unless otherwise specified)

Table 21.65 Timer A Input (Counter Input in Event Counter Mode)

| Symbol | Parameter | Standard | | Unit |
|--------------|------------------------------|----------|------|------|
| | | Min. | Max. | |
| $t_{c(TA)}$ | TAiIN input cycle time | 150 | | ns |
| $t_{w(TAH)}$ | TAiIN input HIGH pulse width | 60 | | ns |
| $t_{w(TAL)}$ | TAiIN input LOW pulse width | 60 | | ns |

Table 21.66 Timer A Input (Gating Input in Timer Mode)

| Symbol | Parameter | Standard | | Unit |
|--------------|------------------------------|----------|------|------|
| | | Min. | Max. | |
| $t_{c(TA)}$ | TAiIN input cycle time | 600 | | ns |
| $t_{w(TAH)}$ | TAiIN input HIGH pulse width | 300 | | ns |
| $t_{w(TAL)}$ | TAiIN input LOW pulse width | 300 | | ns |

Table 21.67 Timer A Input (External Trigger Input in One-shot Timer Mode)

| Symbol | Parameter | Standard | | Unit |
|--------------|------------------------------|----------|------|------|
| | | Min. | Max. | |
| $t_{c(TA)}$ | TAiIN input cycle time | 300 | | ns |
| $t_{w(TAH)}$ | TAiIN input HIGH pulse width | 150 | | ns |
| $t_{w(TAL)}$ | TAiIN input LOW pulse width | 150 | | ns |

Table 21.68 Timer A Input (External Trigger Input in Pulse Width Modulation Mode)

| Symbol | Parameter | Standard | | Unit |
|--------------|------------------------------|----------|------|------|
| | | Min. | Max. | |
| $t_{w(TAH)}$ | TAiIN input HIGH pulse width | 150 | | ns |
| $t_{w(TAL)}$ | TAiIN input LOW pulse width | 150 | | ns |

Table 21.69 Timer A Input (Counter Increment/decrement Input in Event Counter Mode)

| Symbol | Parameter | Standard | | Unit |
|------------------|-------------------------------|----------|------|------|
| | | Min. | Max. | |
| $t_{c(UP)}$ | TAiOUT input cycle time | 3000 | | ns |
| $t_{w(UPH)}$ | TAiOUT input HIGH pulse width | 1500 | | ns |
| $t_{w(UPL)}$ | TAiOUT input LOW pulse width | 1500 | | ns |
| $t_{su(UP-TIN)}$ | TAiOUT input setup time | 600 | | ns |
| $t_{h(TIN-UP)}$ | TAiOUT input hold time | 600 | | ns |

Table 21.70 Timer A Input (Two-phase Pulse Input in Event Counter Mode)

| Symbol | Parameter | Standard | | Unit |
|----------------------|-------------------------|----------|------|---------|
| | | Min. | Max. | |
| $t_{c(TA)}$ | TAiIN input cycle time | 2 | | μs |
| $t_{su(TAIN-TAOUT)}$ | TAiOUT input setup time | 500 | | ns |
| $t_{su(TAOUT-TAIN)}$ | TAiIN input setup time | 500 | | ns |

$$V_{CC} = 3V$$

Timing Requirements

($V_{CC} = 3V$, $V_{SS} = 0V$, at $T_{op} = -40$ to $85^{\circ}C$ unless otherwise specified)

Table 21.71 Timer B Input (Counter Input in Event Counter Mode)

| Symbol | Parameter | Standard | | Unit |
|--------------|--|----------|------|------|
| | | Min. | Max. | |
| $t_{c(TB)}$ | TBIiN input cycle time (counted on one edge) | 150 | | ns |
| $t_{w(TBH)}$ | TBIiN input HIGH pulse width (counted on one edge) | 60 | | ns |
| $t_{w(TBL)}$ | TBIiN input LOW pulse width (counted on one edge) | 60 | | ns |
| $t_{c(TB)}$ | TBIiN input cycle time (counted on both edges) | 300 | | ns |
| $t_{w(TBH)}$ | TBIiN input HIGH pulse width (counted on both edges) | 120 | | ns |
| $t_{w(TBL)}$ | TBIiN input LOW pulse width (counted on both edges) | 120 | | ns |

Table 21.72 Timer B Input (Pulse Period Measurement Mode)

| Symbol | Parameter | Standard | | Unit |
|--------------|------------------------------|----------|------|------|
| | | Min. | Max. | |
| $t_{c(TB)}$ | TBIiN input cycle time | 600 | | ns |
| $t_{w(TBH)}$ | TBIiN input HIGH pulse width | 300 | | ns |
| $t_{w(TBL)}$ | TBIiN input LOW pulse width | 300 | | ns |

Table 21.73 Timer B Input (Pulse Width Measurement Mode)

| Symbol | Parameter | Standard | | Unit |
|--------------|------------------------------|----------|------|------|
| | | Min. | Max. | |
| $t_{c(TB)}$ | TBIiN input cycle time | 600 | | ns |
| $t_{w(TBH)}$ | TBIiN input HIGH pulse width | 300 | | ns |
| $t_{w(TBL)}$ | TBIiN input LOW pulse width | 300 | | ns |

Table 21.74 A/D Trigger Input

| Symbol | Parameter | Standard | | Unit |
|--------------|--|----------|------|------|
| | | Min. | Max. | |
| $t_{c(AD)}$ | \overline{ADTRG} input cycle time (trigger able minimum) | 1500 | | ns |
| $t_{w(ADL)}$ | \overline{ADTRG} input LOW pulse width | 200 | | ns |

Table 21.75 Serial I/O

| Symbol | Parameter | Standard | | Unit |
|---------------|-----------------------------|----------|------|------|
| | | Min. | Max. | |
| $t_{c(CK)}$ | CLKi input cycle time | 300 | | ns |
| $t_{w(CKH)}$ | CLKi input HIGH pulse width | 150 | | ns |
| $t_{w(CKL)}$ | CLKi input LOW pulse width | 150 | | ns |
| $t_{d(C-Q)}$ | TxDi output delay time | | 160 | ns |
| $t_{h(C-Q)}$ | TxDi hold time | 0 | | ns |
| $t_{su(D-C)}$ | RxDi input setup time | 100 | | ns |
| $t_{h(C-D)}$ | RxDi input hold time | 90 | | ns |

Table 21.76 External Interrupt \overline{INTi} Input

| Symbol | Parameter | Standard | | Unit |
|--------------|--|----------|------|------|
| | | Min. | Max. | |
| $t_{w(INH)}$ | \overline{INTi} input HIGH pulse width | 380 | | ns |
| $t_{w(INL)}$ | \overline{INTi} input LOW pulse width | 380 | | ns |

$V_{CC} = 3V$

Timing Requirements**($V_{CC} = 3V$, $V_{SS} = 0V$, at $T_{opr} = -40$ to $85^{\circ}C$ unless otherwise specified)****Table 21.77 Multi-master I²C bus Line**

| Symbol | Parameter | Standard clock mode | | High-speed clock mode | | Unit |
|---------|-------------------------------------|---------------------|------|-----------------------|------|---------|
| | | Min. | Max. | Min. | Max. | |
| tBUF | Bus free time | 4.7 | | 1.3 | | μs |
| tHD;STA | The hold time in start condition | 4.0 | | 0.6 | | μs |
| tLOW | The hold time in SCL clock 0 status | 4.7 | | 1.3 | | μs |
| tR | SCL, SDA signals' rising time | | 1000 | $20+0.1C_b$ | 300 | ns |
| tHD;DAT | Data hold time | 0 | | 0 | 0.9 | μs |
| tHIGH | The hold time in SCL clock 1 status | 4.0 | | 0.6 | | μs |
| tF | SCL, SDA signals' falling time | | 300 | $20+0.1C_b$ | 300 | ns |
| tsu;DAT | Data setup time | 250 | | 100 | | ns |
| tsu;STA | The setup time in restart condition | 4.7 | | 0.6 | | μs |
| tsu;STO | Stop condition setup time | 4.0 | | 0.6 | | μs |

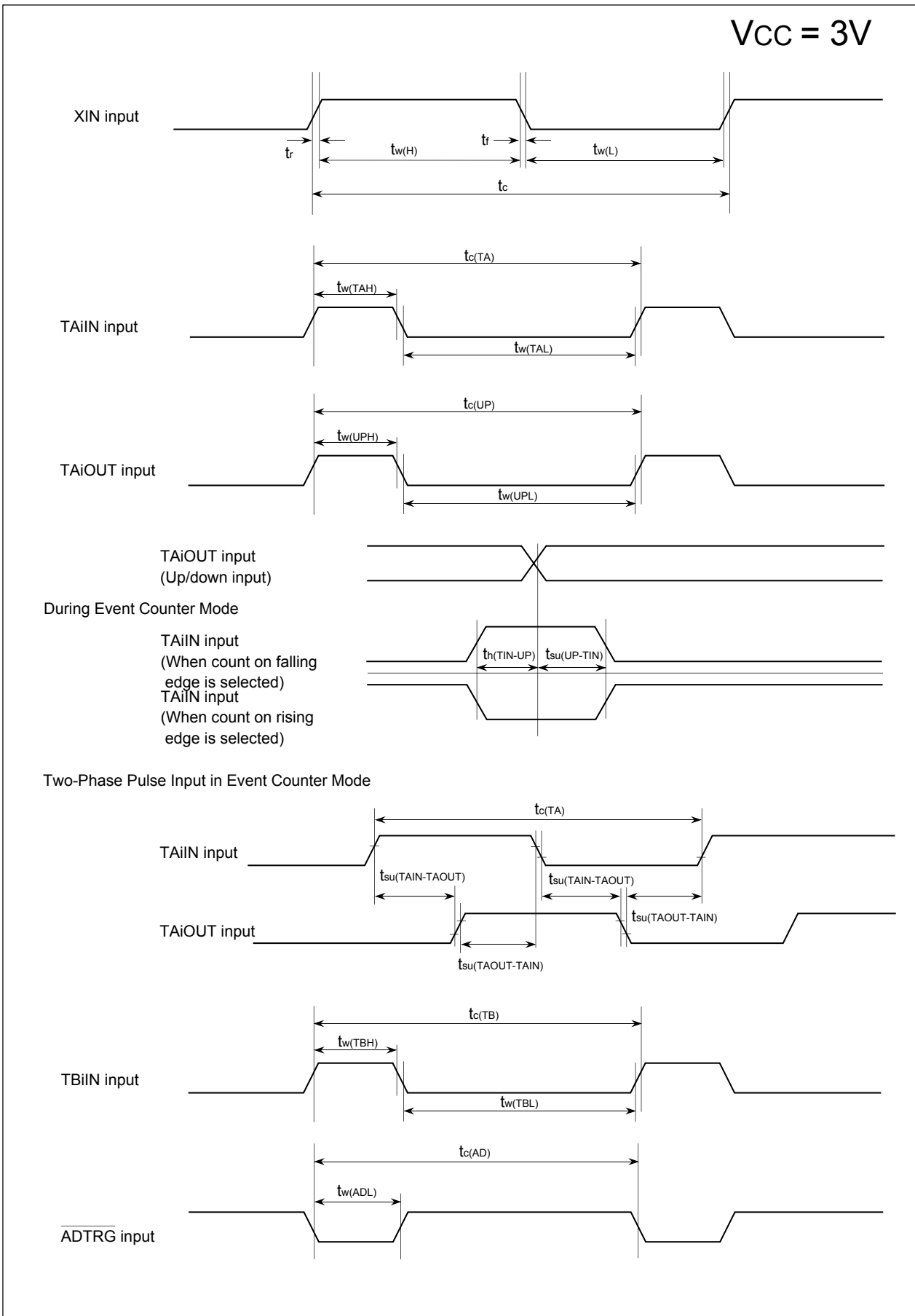


Figure 21.10 Timing Diagram (1)

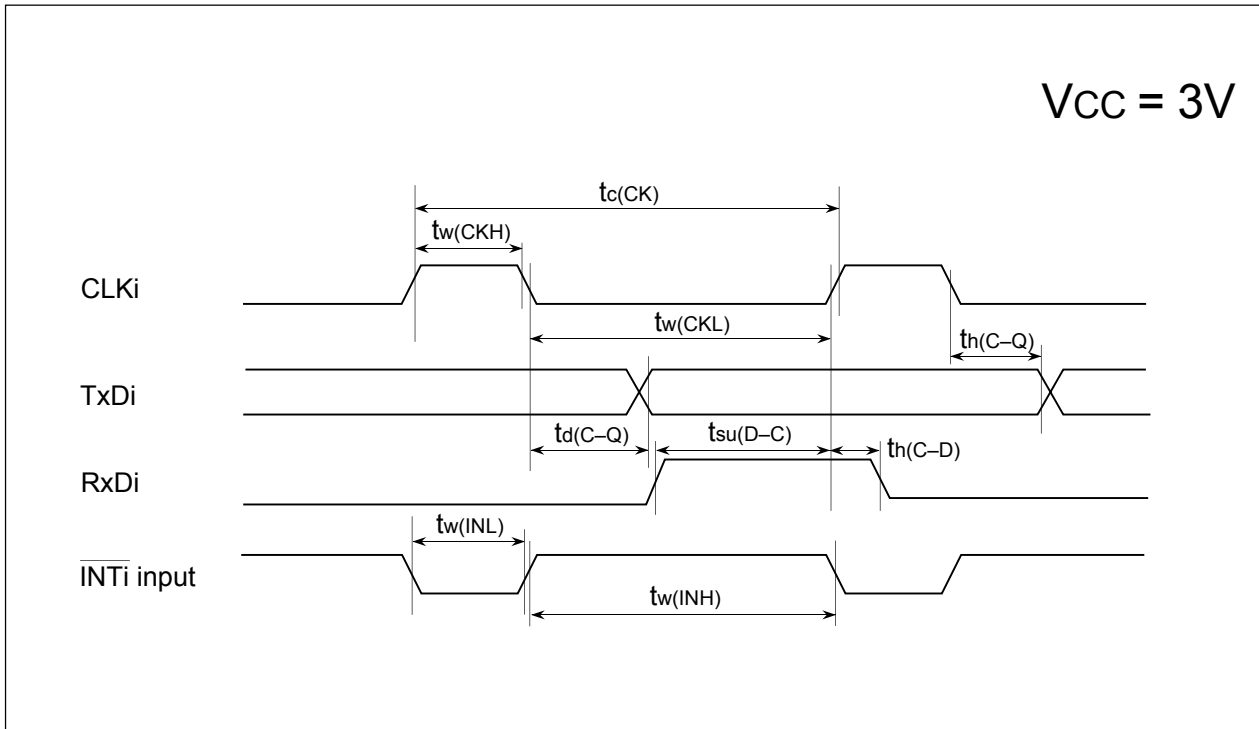


Figure 21.11 Timing Diagram (2)

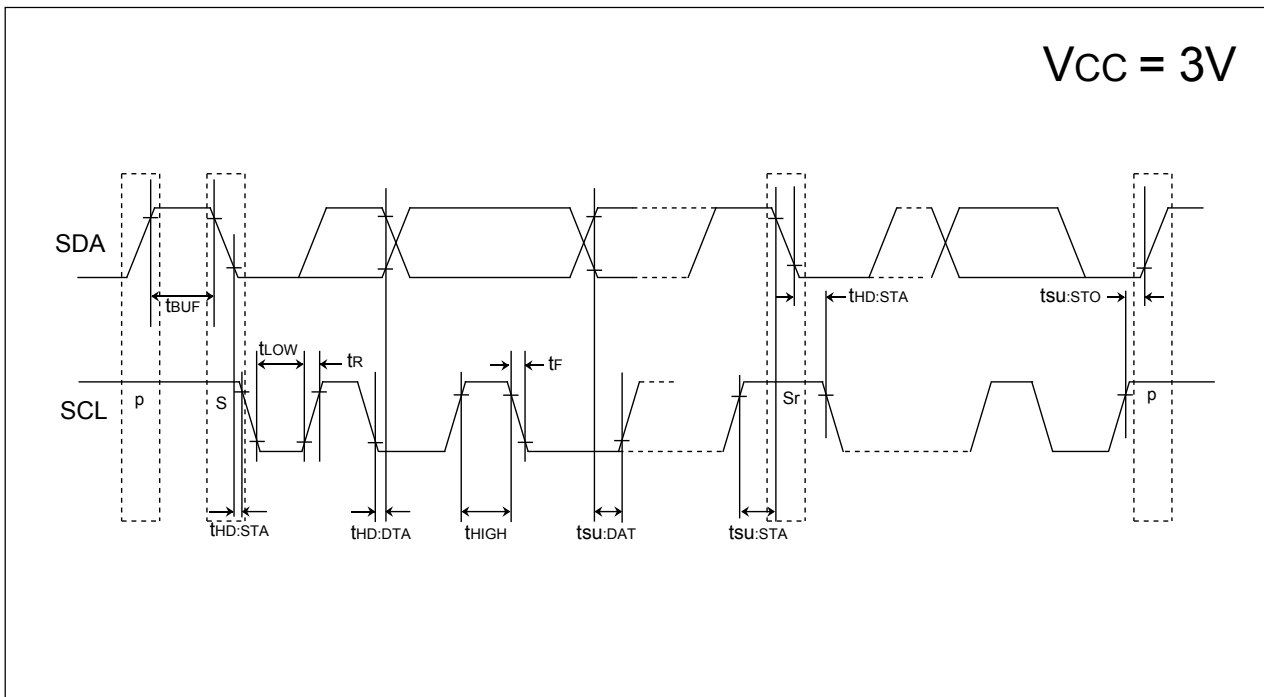


Figure 21.12 Timing Diagram (3)

21.3 V Version

Table 21.78 Absolute Maximum Ratings

| Symbol | Parameter | | Condition | 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 | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ , X _{IN} , V _{REF} , RESET, CNV _{SS} | | -0.3 to V _{CC} +0.3 | V |
| V _O | Output Voltage | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ , X _{OUT} | | -0.3 to V _{CC} +0.3 | V |
| P _d | Power Dissipation | | -40 ≤ T _{opr} ≤ 85 ° C | 300 | mW |
| | | | 85 ≤ T _{opr} ≤ 125 ° C | 200 | mW |
| T _{opr} | Operating Ambient Temperature | during CPU operation | | -40 to 125 | ° C |
| | | during flash memory program and erase operation | Program Space (Block 0 to Block 5) | 0 to 60 | ° C |
| | | | Data Space (Block A, Block B) | -40 to 125 | ° C |
| T _{stg} | Storage Temperature | | | -65 to 150 | ° C |

Table 21.79 Recommended Operating Conditions (1)

| Symbol | Parameter | | Standard | | | Unit | | | | |
|-----------------------|---|---|---|---------------------|--------------------|------------------------------------|-----|--|-----------------|---|
| | | | Min. | Typ. | Max. | | | | | |
| V _{CC} | Supply Voltage | | 4.2 | | 5.5 | V | | | | |
| AV _{CC} | Analog Supply Voltage | | | V _{CC} | | V | | | | |
| V _{SS} | Supply Voltage | | | 0 | | V | | | | |
| AV _{SS} | Analog Supply Voltage | | | 0 | | V | | | | |
| V _{IH} | Input High ("H") Voltage | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | 0.7 V _{CC} | | V _{CC} | V | | | | |
| | | XIN, $\overline{\text{RESET}}$, CNVSS | 0.8 V _{CC} | | V _{CC} | V | | | | |
| | | SDA _{MM} , SCL _{MM} | When I ² C bus input level is selected | 0.7 V _{CC} | | V _{CC} | V | | | |
| | | | | | | When SMBUS input level is selected | 1.4 | | V _{CC} | V |
| V _{IL} | Input Low ("L") Voltage | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | 0 | | 0.3V _{CC} | V | | | | |
| | | XIN, $\overline{\text{RESET}}$, CNVSS | 0 | | 0.2V _{CC} | V | | | | |
| | | SDA _{MM} , SCL _{MM} | When I ² C bus input level is selected | 0 | | 0.3V _{CC} | V | | | |
| | | | | | | When SMBUS input level is selected | 0 | | 0.6 | V |
| I _{OH(peak)} | Peak Output High ("H") Current | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | | | -10.0 | mA | | | | |
| I _{OH(avg)} | Average Output High ("H") Current | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | | | -5.0 | mA | | | | |
| I _{OL(peak)} | Peak Output Low ("L") Current | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | | | 10.0 | mA | | | | |
| I _{OL(avg)} | Average Output Low ("L") Current | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | | | 5.0 | mA | | | | |
| f(X _{IN}) | Main Clock Input Oscillation Frequency ⁽⁴⁾ | | Topr = -40 to 105 ° C | 0 | 20 | MHz | | | | |
| | | | Topr = -40 to 125 ° C | 0 | 16 | MHz | | | | |
| f(X _{CIN}) | Sub Clock Oscillation Frequency | | | 32.768 | 50 | kHz | | | | |
| f ₁ (ROC) | On-chip Oscillator Frequency 1 | | 0.5 | 1 | 2 | MHz | | | | |
| f ₂ (ROC) | On-chip Oscillator Frequency 2 | | 1 | 2 | 4 | MHz | | | | |
| f ₃ (ROC) | On-chip Oscillator Frequency 3 | | 8 | 16 | 26 | MHz | | | | |
| f(PLL) | PLL Clock Oscillation Frequency ⁽⁴⁾ | | Topr = -40 to 105 ° C | 10 | 20 | MHz | | | | |
| | | | Topr = -40 to 125 ° C | 10 | 16 | MHz | | | | |
| f(BCLK) | CPU Operation Clock Frequency | | Topr = -40 to 105 ° C | 0 | 20 | MHz | | | | |
| | | | Topr = -40 to 125 ° C | 0 | 16 | MHz | | | | |
| t _{SU} (PLL) | Wait Time to Stabilize PLL Frequency Synthesizer | | V _{CC} = 5.0 V | | 20 | MHz | | | | |

NOTES:

1. Referenced to V_{CC} = 4.2 to 5.5 V at Topr = -40 to 125 ° C unless otherwise specified.
2. The mean output current is the mean value within 100ms.
3. The total I_{OL(peak)} for all ports must be 80 mA or less. The total I_{OH(peak)} for all ports must be -80 mA or less.
4. Relationship among main clock oscillation frequency, PLL clock oscillation frequency and supply voltage.

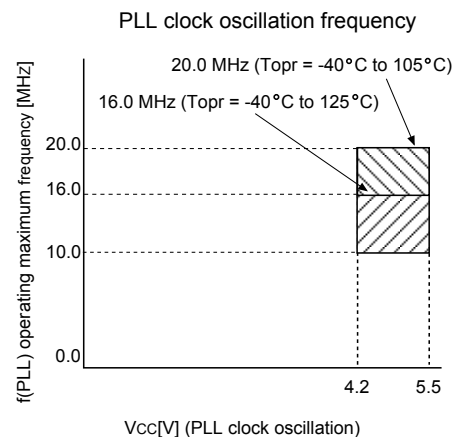
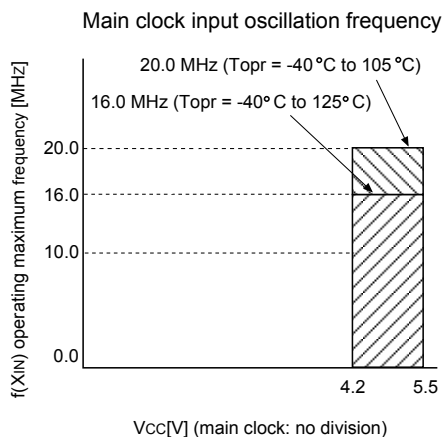


Table 21.80 A/D Conversion Characteristics (1)

| Symbol | Parameter | | Measurement Condition | Standard | | | Unit |
|------------|--|--------|---|----------|------|-----------|---------------|
| | | | | Min. | Typ. | Max. | |
| - | Resolution | | $V_{REF} = V_{CC}$ | | | 10 | Bits |
| INL | Integral Nonlinearity Error | 10 bit | $V_{REF} = V_{CC} = 5\text{ V}$ | | | ± 3 | LSB |
| | | 8 bit | $V_{REF} = V_{CC} = 5\text{ V}$ | | | ± 2 | LSB |
| - | Absolute Accuracy | 10 bit | $V_{REF} = V_{CC} = 5\text{ V}$ | | | ± 3 | LSB |
| | | 8 bit | $V_{REF} = V_{CC} = 5\text{ V}$ | | | ± 2 | LSB |
| DNL | Differential Nonlinearity Error | | | | | ± 1 | LSB |
| - | Offset Error | | | | | ± 3 | LSB |
| - | Gain Error | | | | | ± 3 | LSB |
| RLADDER | Resistor Ladder | | $V_{REF} = V_{CC}$ | 10 | | 40 | k Ω |
| t_{CONV} | 10-bit Conversion Time Sample & Hold Function Available | | $V_{REF} = V_{CC} = 5\text{ V}$, $\phi_{AD} = 10\text{ MHz}$ | 3.3 | | | μs |
| t_{CONV} | 8-bit Conversion Time Sample & Hold Function Available | | $V_{REF} = V_{CC} = 5\text{ V}$, $\phi_{AD} = 10\text{ MHz}$ | 2.8 | | | μs |
| V_{REF} | Reference Voltage | | | 2.0 | | V_{CC} | V |
| V_{IA} | Analog Input Voltage | | | 0 | | V_{REF} | V |

NOTES:

1. Referenced to $V_{CC} = AV_{CC} = V_{REF} = 4.2$ to 5.5 V , $V_{SS} = AV_{SS} = 0\text{ V}$ at $T_{opr} = -40$ to $125\text{ }^\circ\text{C}$ unless otherwise specified.
2. Keep ϕ_{AD} frequency at 10 MHz or less.
3. When sample & hold function is disabled, keep ϕ_{AD} frequency at 250kHz or more in addition to the limitation in Note 2.
When sample & hold function is enabled, keep ϕ_{AD} frequency at 1MHz or more in addition to the limitation in Note 2.
4. When sample & hold function is enabled, sampling time is $3/\phi_{AD}$ frequency.
When sample & hold function is disabled, sampling time is $2/\phi_{AD}$ frequency.

**Table 21.81 Flash Memory Version Electrical Characteristics ⁽¹⁾ for 100/1000 E/W cycle products
[Program Space and Data Space in U3; Program Space in U7]**

| Symbol | Parameter | Standard | | | Unit |
|-----------------|---|-----------------------------|---------------------|------|--------|
| | | Min. | Typ. ⁽²⁾ | Max. | |
| - | Program and Erase Endurance ⁽³⁾ | 100/1000 ^(4, 11) | | | cycles |
| - | Word Program Time (V _{CC} = 5.0 V, Topr = 25° C) | | 75 | 600 | μs |
| - | Block Erase Time (V _{CC} = 5.0 V, Topr = 25° C) | 2-Kbyte Block | 0.2 | 9 | s |
| | | 8-Kbyte Block | 0.4 | 9 | s |
| | | 16-Kbyte Block | 0.7 | 9 | s |
| | | 32-Kbyte Block | 1.2 | 9 | s |
| td(SR-ES) | Duration between Suspend Request and Erase Suspend | | | 8 | ms |
| t _{PS} | Wait Time to Stabilize Flash Memory Circuit | | | 15 | μs |
| - | Data Hold Time ⁽⁵⁾ | 20 | | | years |

**Table 21.82 Flash Memory Version Electrical Characteristics ⁽⁶⁾ for 10000 E/W cycle products
[Data Space in U7 ⁽⁷⁾]**

| Symbol | Parameter | Standard | | | Unit |
|-----------------|--|--------------------------|---------------------|------|--------|
| | | Min. | Typ. ⁽²⁾ | Max. | |
| - | Program and Erase Endurance ^(3, 8, 9) | 10000 ^(4, 10) | | | cycles |
| - | Word Program Time (V _{CC} = 5.0 V, Topr = 25° C) | | 100 | | μs |
| - | Block Erase Time (V _{CC} = 5.0V, Topr = 25° C) (2-Kbyte block) | | 0.3 | | s |
| td(SR-ES) | Duration between Suspend Request and Erase Suspend | | | 8 | ms |
| t _{PS} | Wait Time to Stabilize Flash Memory Circuit | | | 15 | μs |
| - | Data Hold Time ⁽⁵⁾ | 20 | | | years |

NOTES:

1. Referenced to V_{CC} = 4.2 to 5.5 V at Topr = 0 to 60° C (program space)/ Topr = -40 to 125° C(data space), unless otherwise specified.
2. V_{CC} = 5.0 V; TOPR = 25° C
3. Program and erase endurance is defined as number of program-erase cycles per block.
If program and erase endurance is *n* cycle (*n* = 100, 1000, 10000), each block can be erased and programmed *n* cycles.
For example, if a 2-Kbyte block A is erased after programming one-word data to each address 1,024 times, this counts as one program and erase endurance. Data cannot be programmed to the same address more than once without erasing the block. (rewrite prohibited).
4. Number of E/W cycles for which operation is guaranteed (1 to minimum value are guaranteed).
5. Topr = 55° C
6. Referenced to V_{CC} = 4.2 to 5.5 V at Topr = -40 to 125° C unless otherwise specified.
7. **Table 21.82** applies for data space in U7 when program and erase endurance is more than 1,000 cycles. Otherwise, use **Table 21.81**.
8. To reduce the number of program and erase endurance when working with systems requiring numerous rewrites, write to unused word addresses within the block instead of rewrite. Erase block only after all possible addresses are used. For example, an 8-word program can be written 128 times maximum before erase becomes necessary. Maintaining an equal number of times erasure between block A and block B will also improve efficiency. It is recommended to track the total number of erasure performed per block and to limit the number of erasure.
9. If an erase error is generated during block erase, execute the clear status register command and block erase command at least 3 times until an erase error is not generated.
10. When executing more than 100 times rewrites, set one wait state per block access by setting the FMR17 bit in the FMR1 register to 1 (wait state). When accessing to all other blocks and internal RAM, wait state can be set by the PM17 bit, regardless of the FMR17 bit setting value.
11. The program and erase endurance is 100 cycles for program space and data space in U3; 1,000 cycles for program space in U7.
12. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for further details on the E/W failure rate.

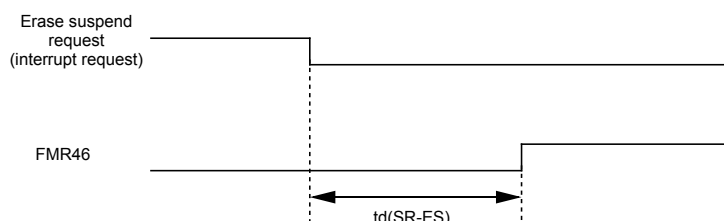
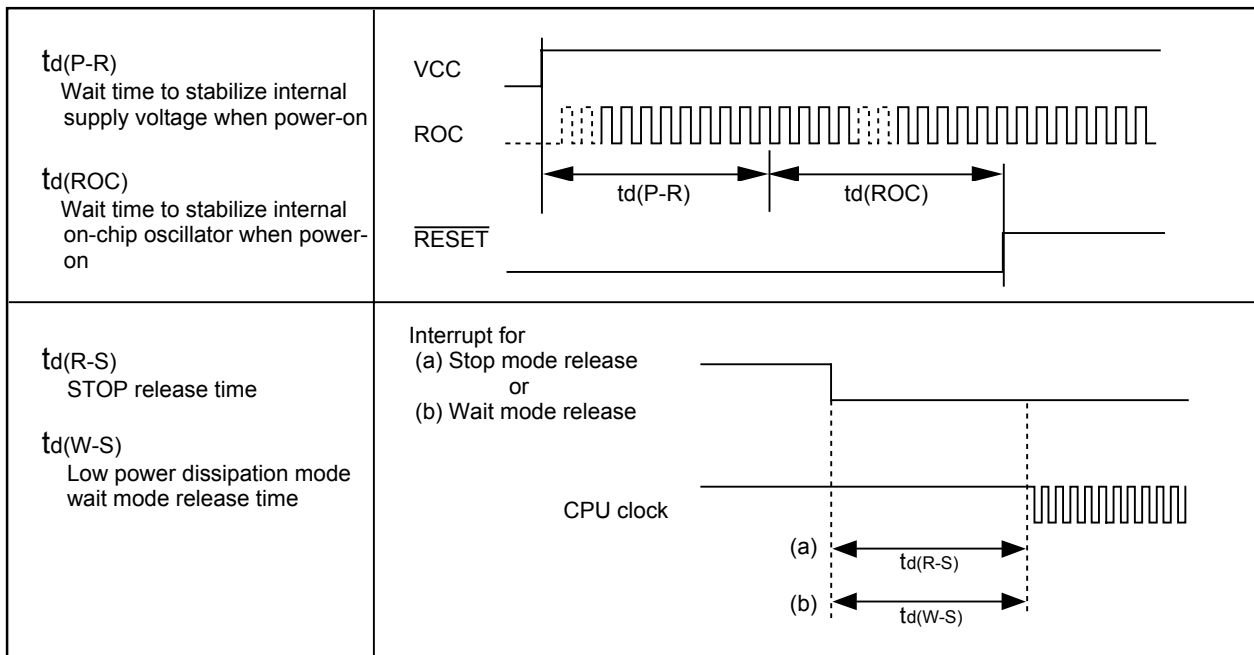


Table 21.83 Power Supply Circuit Timing Characteristics

| Symbol | Parameter | Measurement Condition | Standard | | | Unit |
|---------|--|------------------------------|----------|------|------|------|
| | | | Min. | Typ. | Max. | |
| td(P-R) | Wait Time to Stabilize Internal Supply Voltage when Power-on | V _{CC} =4.2 to 5.5V | | | 2 | ms |
| td(ROC) | Wait Time to Stabilize Internal On-chip Oscillator when Power-on | | | | 40 | μs |
| td(S-R) | STOP Release Time | | | | 150 | μs |
| td(E-A) | Low Power Dissipation Mode Wait Mode Release Time | | | | 150 | μs |



V_{CC} = 5V**Table 21.84 Electrical Characteristics (1)**

| Symbol | Parameter | | Condition | Standard | | | Unit |
|----------------------------------|---------------------------|---|---------------------------|---------------------------|----------------------|-----------------|------|
| | | | | Min. | Typ. | Max. | |
| V _{OH} | Output High ("H") Voltage | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | I _{OH} = -5 mA | V _{CC} -2.0 | | V _{CC} | V |
| V _{OH} | Output High ("H") Voltage | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | I _{OH} = -200 μA | V _{CC} -0.3 | | V _{CC} | V |
| V _{OH} | Output High ("H") Voltage | X _{OUT} | High Power | I _{OH} = -1 mA | V _{CC} -2.0 | V _{CC} | V |
| | | | Low Power | I _{OH} = -0.5 mA | V _{CC} -2.0 | V _{CC} | |
| | Output High ("H") Voltage | X _{COU} T | High Power | No load applied | | 2.5 | V |
| | | | Low Power | No load applied | | 1.6 | |
| V _{OL} | Output Low ("L") Voltage | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | I _{OL} = 5 mA | | | 2.0 | V |
| V _{OL} | Output Low ("L") Voltage | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | I _{OL} = 200 μA | | | 0.45 | V |
| V _{OL} | Output Low ("L") Voltage | X _{OUT} | High Power | I _{OL} = 1 mA | | 2.0 | V |
| | | | Low Power | I _{OL} = 0.5 mA | | 2.0 | |
| | Output Low ("L") Voltage | X _{COU} T | High Power | No load applied | | 0 | V |
| | | | Low Power | No load applied | | 0 | |
| V _{T+} -V _{T-} | Hysteresis | TA0 _{IN} -TA4 _{IN} , TB0 _{IN} -TB2 _{IN} , INT ₀ -INT ₅ , NMI, AD _{TRG} , CTS ₀ -CTS ₂ , SCL, SDA, CLK ₀ -CLK ₂ , TA2 _{OUT} -TA4 _{OUT} , KI ₀ -KI ₃ , RXD ₀ -RXD ₂ , SIN ₃ , SIN ₄ | | 0.2 | | 1.0 | V |
| V _{T+} -V _{T-} | Hysteresis | RESET | | 0.2 | | 2.5 | V |
| V _{T+} -V _{T-} | Hysteresis | X _{IN} | | 0.2 | | 0.8 | V |
| I _{IH} | Input High ("H") Current | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ X _{IN} , RESET, CNV _{SS} | V _I = 5 V | | | 5.0 | μA |
| I _{IL} | Input Low ("L") Current | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ X _{IN} , RESET, CNV _{SS} | V _I = 0 V | | | -5.0 | μA |
| R _{PULLUP} | Pull-up Resistance | P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P6 ₀ to P6 ₇ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₇ , P9 ₀ to P9 ₃ , P9 ₅ to P9 ₇ , P10 ₀ to P10 ₇ | V _I = 0 V | 30 | 50 | 170 | kΩ |
| R _{fXIN} | Feedback Resistance | X _{IN} | | | 1.5 | | MΩ |
| R _{fXCIN} | Feedback Resistance | X _{CIN} | | | 15 | | MΩ |
| V _{RAM} | RAM Standby Voltage | | In stop mode | 2.0 | | | V |

NOTE:

1. Referenced to V_{CC} = 4.2 to 5.5 V, V_{SS} = 0 V at T_{opr} = -40 to 105 ° C, f(BCLK) = 20 MHz / V_{CC} = 4.2 to 5.5 V, V_{SS} = 0 V at T_{opr} = -40 to 125 ° C, f(BCLK) = 16 MHz, unless otherwise specified.

$V_{CC} = 5V$ **Table 21.85 Electrical Characteristics (2) (1)**

| Symbol | Parameter | Measurement Condition | | | Standard | | | Unit |
|---------------------------------|--|---|------------------------|---|----------|------|------|------|
| | | | | | Min. | Typ. | Max. | |
| I _{CC} | Power Supply Current (V _{CC} =4.2 to 5.5V) | Output pins are left open and other pins are connected to V _{SS} | Mask ROM | f(BCLK) = 20 MHz, main clock, no division | | 18 | 25 | mA |
| | | | | f(BCLK) = 16 MHz, main clock, no division | | 14 | 20 | mA |
| | | | | On-chip oscillation, f _{2(ROC)} selected, f(BCLK) = 1 MHz | | 2 | | mA |
| | | | Flash memory | f(BCLK) = 20 MHz, main clock, no division | | 18 | 25 | mA |
| | | | | f(BCLK) = 16 MHz, main clock, no division | | 14 | 20 | mA |
| | | | | On-chip oscillation, f _{2(ROC)} selected, f(BCLK) = 1 MHz | | 2 | | mA |
| | | | Flash memory program | f(BCLK) = 10 MHz, V _{CC} = 5.0 V | | 11 | | mA |
| | | | Flash memory erase | f(BCLK) = 10 MHz, V _{CC} = 5.0 V | | 11 | | mA |
| | | | Mask ROM | f(BCLK) = 32 kHz, In low-power consumption mode, Program running on ROM ⁽³⁾ | | 25 | | μA |
| | | | | On-chip oscillation, f _{2(ROC)} selected, f(BCLK) = 1 MHz, In wait mode | | 50 | | μA |
| | | | Flash memory | f(BCLK) = 32 kHz, In low-power consumption mode, Program running on RAM ⁽³⁾ | | 25 | | μA |
| | | | | f(BCLK) = 32 kHz, In low-power consumption mode, Program running on flash memory ⁽³⁾ | | 450 | | μA |
| | | | | On-chip oscillation, f _{2(ROC)} selected, f(BCLK) = 1 MHz, In wait mode | | 50 | | μA |
| | | | Mask ROM, Flash memory | f(BCLK) = 32 kHz, In wait mode ⁽²⁾ , Oscillation capacity high | | 8.5 | | μA |
| | | | | f(BCLK) = 32 kHz, In wait mode ⁽²⁾ , Oscillation capacity low | | 3 | | μA |
| While clock stops, Topr = 25° C | | 0.8 | | 3 | μA | | | |

NOTES:

1. Referenced to V_{CC} = 4.2 to 5.5 V, V_{SS} = 0 V at Topr = -40 to 105 ° C, f(BCLK) = 20MHz / V_{CC} = 4.2 to 5.5 V, V_{SS} = 0V at Topr = -40 to 125 ° C, f(BCLK) = 16 MHz, unless otherwise specified.
2. With one timer operates, using fc32.
3. This indicates the memory in which the program to be executed exists.

$V_{CC} = 5V$ **Timing Requirements****($V_{CC}=5V$, $V_{SS}=0V$, at $T_{opr}=-40$ to $125^{\circ}C$ unless otherwise specified)****Table 21.86 External Clock Input (XIN input)**

| Symbol | Parameter | | Standard | | Unit |
|--------|---------------------------------------|-----------------------|----------|------|------|
| | | | Min. | Max. | |
| tc | External Clock Input Cycle Time | Topr=-40° C to 105° C | 50 | | ns |
| | | Topr=-40° C to 125° C | 62.5 | | ns |
| tw(H) | External Clock Input High ("H") Width | Topr=-40° C to 105° C | 20 | | ns |
| | | Topr=-40° C to 125° C | 25 | | ns |
| tw(L) | External Clock Input Low ("L") Width | Topr=-40° C to 105° C | 20 | | ns |
| | | Topr=-40° C to 125° C | 25 | | ns |
| tr | External Clock Rise Time | Topr=-40° C to 105° C | | 9 | ns |
| | | Topr=-40° C to 125° C | | 15 | ns |
| tf | External Clock Fall Time | Topr=-40° C to 105° C | | 9 | ns |
| | | Topr=-40° C to 125° C | | 15 | ns |

$V_{CC} = 5V$

Timing Requirements(V_{CC}=5V, V_{SS}=0V, at Topr=-40 to 125°C unless otherwise specified)**Table 21.87 Timer A Input (Counter Input in Event Counter Mode)**

| Symbol | Parameter | Standard | | Unit |
|---------------------|--|----------|------|------|
| | | Min. | Max. | |
| t _{C(TA)} | TA _{iIN} Input Cycle Time | 100 | | ns |
| t _{w(TAH)} | TA _{iIN} Input High ("H") Width | 40 | | ns |
| t _{w(TAL)} | TA _{iIN} Input Low ("L") Width | 40 | | ns |

Table 21.88 Timer A Input (Gating Input in Timer Mode)

| Symbol | Parameter | Standard | | Unit |
|---------------------|--|----------|------|------|
| | | Min. | Max. | |
| t _{C(TA)} | TA _{iIN} Input Cycle Time | 400 | | ns |
| t _{w(TAH)} | TA _{iIN} Input High ("H") Width | 200 | | ns |
| t _{w(TAL)} | TA _{iIN} Input Low ("L") Width | 200 | | ns |

Table 21.89 Timer A Input (External Trigger Input in One-shot Timer Mode)

| Symbol | Parameter | Standard | | Unit |
|---------------------|--|----------|------|------|
| | | Min. | Max. | |
| t _{C(TA)} | TA _{iIN} Input Cycle Time | 200 | | ns |
| t _{w(TAH)} | TA _{iIN} Input High ("H") Width | 100 | | ns |
| t _{w(TAL)} | TA _{iIN} Input Low ("L") Width | 100 | | ns |

Table 21.90 Timer A Input (External Trigger Input in Pulse Width Modulation Mode)

| Symbol | Parameter | Standard | | Unit |
|---------------------|--|----------|------|------|
| | | Min. | Max. | |
| t _{w(TAH)} | TA _{iIN} Input High ("H") Width | 100 | | ns |
| t _{w(TAL)} | TA _{iIN} Input Low ("L") Width | 100 | | ns |

Table 21.91 Timer A Input (Counter Increment/decrement Input in Event Counter Mode)

| Symbol | Parameter | Standard | | Unit |
|-------------------------|---|----------|------|------|
| | | Min. | Max. | |
| t _{C(UP)} | TA _{iOUT} Input Cycle Time | 2000 | | ns |
| t _{w(UPH)} | TA _{iOUT} Input High ("H") Width | 1000 | | ns |
| t _{w(UPL)} | TA _{iOUT} Input Low ("L") Width | 1000 | | ns |
| t _{su(UP-TIN)} | TA _{iOUT} Input Setup Time | 400 | | ns |
| t _{h(TIN-UP)} | TA _{iOUT} Input Hold Time | 400 | | ns |

Table 21.92 Timer A Input (Two-phase Pulse Input in Event Counter Mode)

| Symbol | Parameter | Standard | | Unit |
|---|-------------------------------------|----------|------|------|
| | | Min. | Max. | |
| t _{C(TA)} | TA _{iIN} Input Cycle Time | 800 | | ns |
| t _{su(TA_{iIN}-TA_{iOUT})} | TA _{iOUT} Input Setup Time | 200 | | ns |
| t _{su(TA_{iOUT}-TA_{iIN})} | TA _{iIN} Input Setup Time | 200 | | ns |

$V_{CC} = 5V$

Timing Requirements(V_{CC}=5V, V_{SS}=0V, at Topr=-40 to 125°C unless otherwise specified)**Table 21.93 Timer B Input (Counter Input in Event Counter Mode)**

| Symbol | Parameter | Standard | | Unit |
|---------------------|--|----------|------|------|
| | | Min. | Max. | |
| t _{c(TB)} | TBiIN Input Cycle Time (counted on one edge) | 100 | | ns |
| t _{w(TBH)} | TBiIN Input High ("H") Width (counted on one edge) | 40 | | ns |
| t _{w(TBL)} | TBiIN Input Low ("L") Width (counted on one edge) | 40 | | ns |
| t _{c(TB)} | TBiIN Input Cycle Time (counted on both edges) | 200 | | ns |
| t _{w(TBH)} | TBiIN Input High ("H") Width (counted on both edges) | 80 | | ns |
| t _{w(TBL)} | TBiIN Input Low ("L") Width (counted on both edges) | 80 | | ns |

Table 21.94 Timer B Input (Pulse Period Measurement Mode)

| Symbol | Parameter | Standard | | Unit |
|---------------------|------------------------------|----------|------|------|
| | | Min. | Max. | |
| t _{c(TB)} | TBiIN Input Cycle Time | 400 | | ns |
| t _{w(TBH)} | TBiIN Input High ("H") Width | 200 | | ns |
| t _{w(TBL)} | TBiIN Input Low ("L") Width | 200 | | ns |

Table 21.95 Timer B Input (Pulse Width Measurement Mode)

| Symbol | Parameter | Standard | | Unit |
|---------------------|------------------------------|----------|------|------|
| | | Min. | Max. | |
| t _{c(TB)} | TBiIN Input Cycle Time | 400 | | ns |
| t _{w(TBH)} | TBiIN Input High ("H") Width | 200 | | ns |
| t _{w(TBL)} | TBiIN Input Low ("L") Width | 200 | | ns |

Table 21.96 A/D Trigger Input

| Symbol | Parameter | Standard | | Unit |
|---------------------|--|----------|------|------|
| | | Min. | Max. | |
| t _{c(AD)} | \overline{ADTRG} Input Cycle Time (required for trigger) | 1000 | | ns |
| t _{w(ADL)} | \overline{ADTRG} Input Low ("L") Width | 125 | | ns |

Table 21.97 Serial I/O

| Symbol | Parameter | Standard | | Unit |
|----------------------|-----------------------------|----------|------|------|
| | | Min. | Max. | |
| t _{c(CK)} | CLKi Input Cycle Time | 200 | | ns |
| t _{w(CKH)} | CLKi Input High ("H") Width | 100 | | ns |
| t _{w(CKL)} | CLKi Input Low ("L") Width | 100 | | ns |
| t _{d(C-Q)} | TxDi Output Delay Time | | 80 | ns |
| t _{h(C-Q)} | TxDi Hold Time | 0 | | ns |
| t _{su(D-C)} | RxDi Input Setup Time | 70 | | ns |
| t _{h(C-Q)} | RxDi Input Hold Time | 90 | | ns |

Table 21.98 External Interrupt \overline{INTi} Input

| Symbol | Parameter | Standard | | Unit |
|---------------------|--|----------|------|------|
| | | Min. | Max. | |
| t _{w(INH)} | \overline{INTi} Input High ("H") Width | 250 | | ns |
| t _{w(INL)} | \overline{INTi} Input Low ("L") Width | 250 | | ns |

$V_{CC} = 5V$ **Timing Requirements****($V_{CC}=5V$, $V_{SS}=0V$, at $T_{op}=-40$ to $125^{\circ}C$ unless otherwise specified)****Table 21.99 Multi-master I²C Bus Line**

| Symbol | Parameter | Standard clock mode | | High-speed clock mode | | Unit |
|---------|---------------------------------------|---------------------|------|-----------------------|------|---------|
| | | Min. | Max. | Min. | Max. | |
| tBUF | Bus free time | 4.7 | | 1.3 | | μs |
| tHD;STA | The hold time in start condition | 4.0 | | 0.6 | | μs |
| tLOW | The hold time in SCL clock "0" status | 4.7 | | 1.3 | | μs |
| tR | SCL, SDA signals' rising time | | 1000 | $20+0.1C_b$ | 300 | ns |
| tHD;DAT | Data hold time | 0 | | 0 | 0.9 | μs |
| tHIGH | The hold time in SCL clock "1" status | 4.0 | | 0.6 | | μs |
| tF | SCL, SDA signals' falling time | | 300 | $20+0.1C_b$ | 300 | ns |
| tsu;DAT | Data setup time | 250 | | 100 | | ns |
| tsu;STA | The setup time in restart condition | 4.7 | | 0.6 | | μs |
| tsu;STO | Stop condition setup time | 4.0 | | 0.6 | | μs |

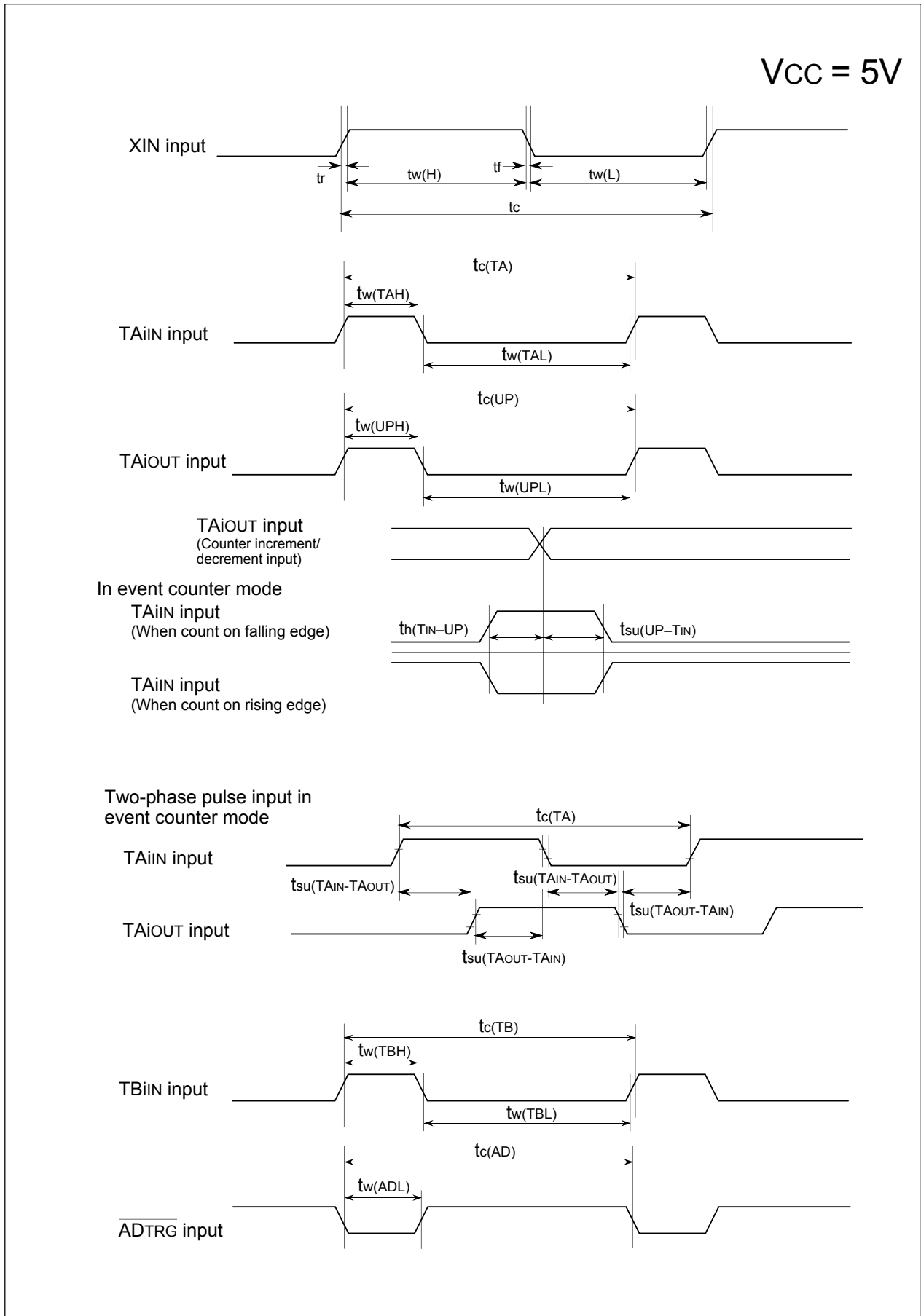


Figure 21.13 Timing Diagram (1)

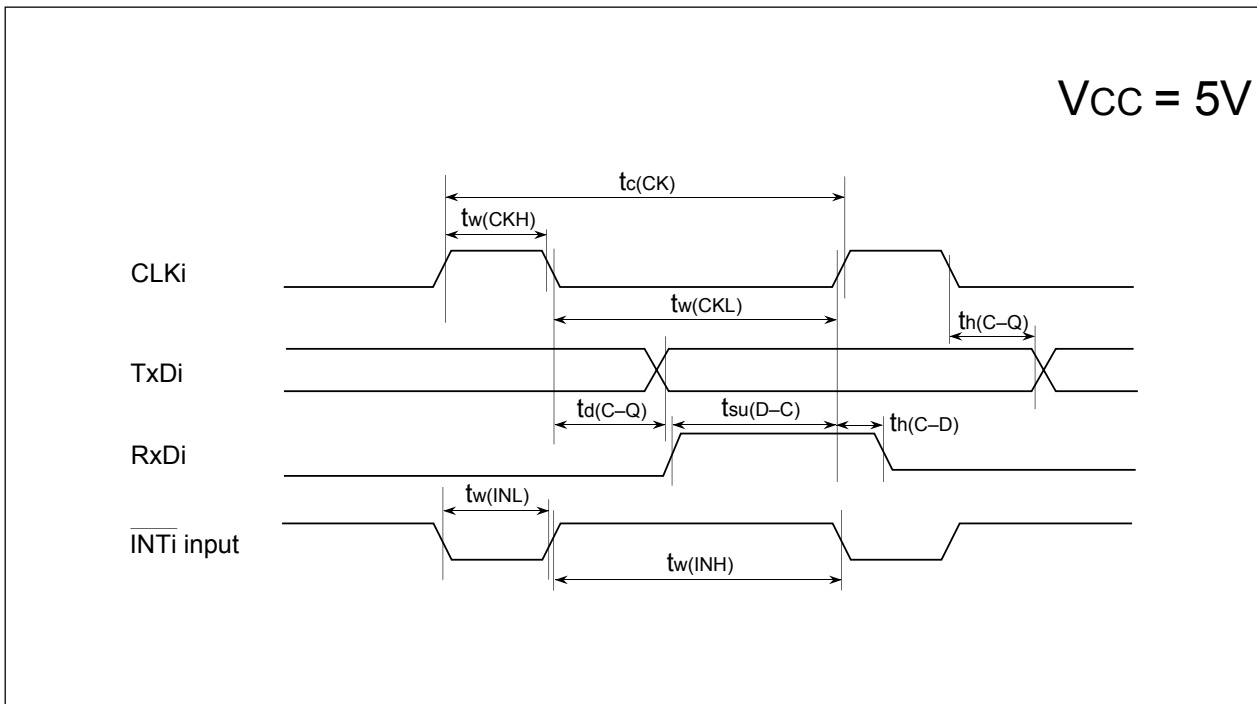


Figure 21.14 Timing Diagram (2)

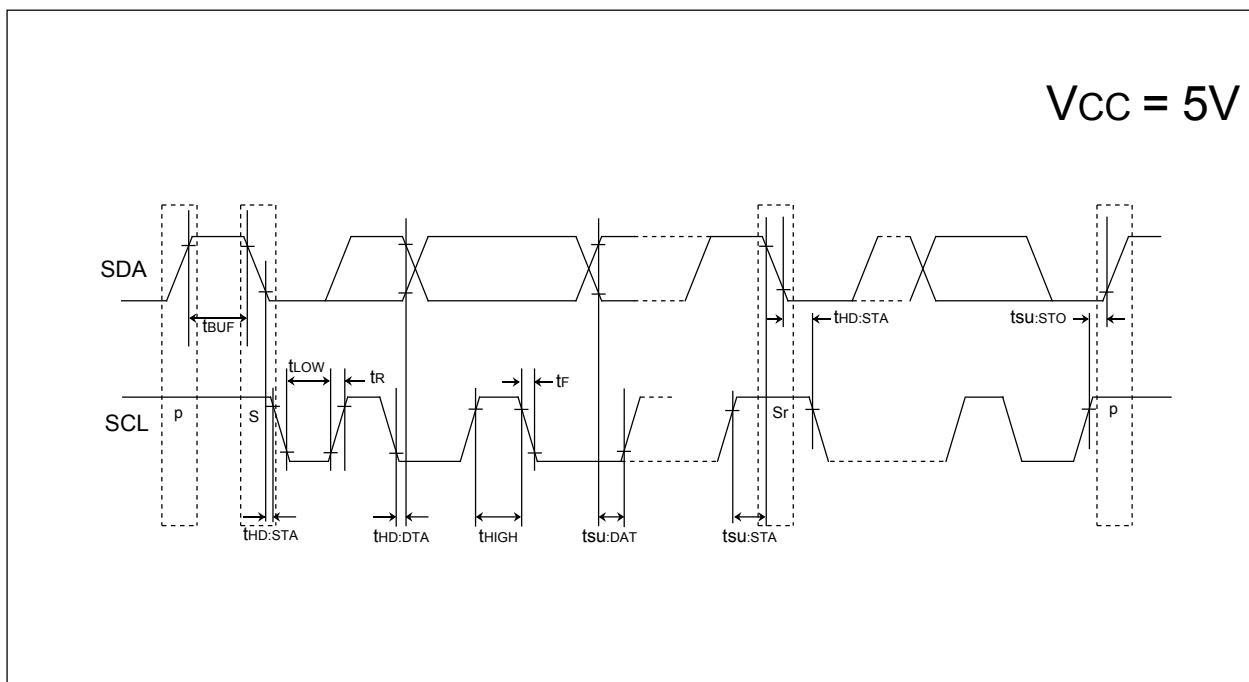


Figure 21.15 Timing Diagram (3)

22. Usage Notes

22.1 SFRs

22.1.1 For 80-Pin Package

Set the IFSR20 bit in the IFSR2A register to 0 after reset and set bits PACR2 to PACR0 in the PACR register to 0112.

22.1.2 For 64-Pin Package

Set the IFSR20bit in the IFSR2A register to 0 after reset and set bits PACR2 to PACR0 in the PACR register to 0102.

22.1.3 Register Setting

Immediate values should be set in the registers containing write-only bits. When establishing a new value by modifying a previous value, write the previous value into RAM as well as the register. Change the contents of the RAM and then transfer the new value to the register.

22.2 Clock Generation Circuit

22.2.1 PLL Frequency Synthesizer

Stabilize supply voltage so that the standard of the power supply ripple is met.

| Symbol | Parameter | Standard | | | Unit |
|-------------------|---|-----------------|------|------|------|
| | | Min. | Typ. | Max. | |
| $f_{(ripple)}$ | Power supply ripple allowable frequency(V_{CC}) | | | 10 | kHz |
| $V_{p-p(ripple)}$ | Power supply ripple allowable amplitude voltage | ($V_{CC}=5V$) | | 0.5 | V |
| | | ($V_{CC}=3V$) | | 0.3 | V |
| $V_{CC(DV/DT)}$ | Power supply ripple rising/falling gradient | ($V_{CC}=5V$) | | 0.3 | V/ms |
| | | ($V_{CC}=3V$) | | 0.3 | V/ms |

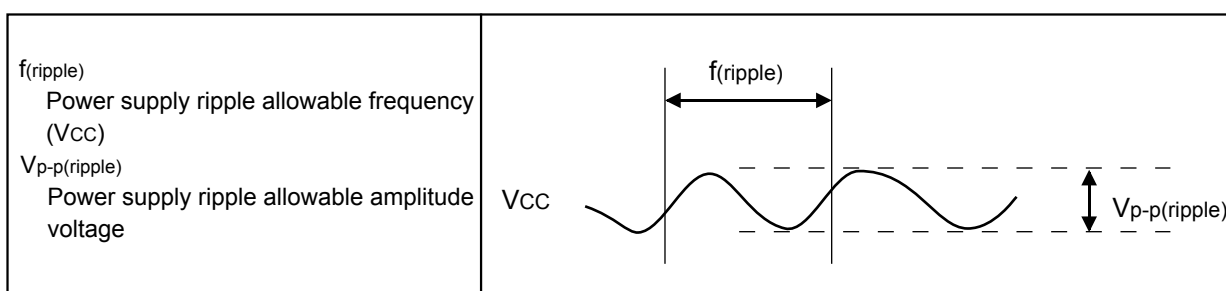


Figure 22.1 Voltage Fluctuation Timing

22.2.2 Power Control

1. When exiting stop mode by hardware reset, the device will startup using the on-chip oscillator.
2. Set the MR0 bit in the TAI_{MR} register(i=0 to 4) to 0 (pulse is not output) to use the timer A to exit stop mode.
3. When entering wait mode, insert a JMP.B instruction before a WAIT instruction. Do not excute any instructions which can generate a write to RAM between the JMP.B and WAIT instructions. Disable the DMA transfers, if a DMA transfer may occur between the JMP.B and WAIT instructions. After the WAIT instruction, insert at least 4 NOP instructions. When entering wait mode, the instruction queue reads ahead the instructions following WAIT, and depending on timing, some of these may execute before the MCU enters wait mode.

Program example when entering wait mode

```

Program Example:    JMP.B    L1    ; Insert JMP.B instruction before WAIT instruction
                   L1:
                   FSET    I    ;
                   WAIT    ; Enter wait mode
                   NOP     ; More than 4 NOP instructions
                   NOP
                   NOP
                   NOP

```

4. When entering stop mode, insert a JMP.B instruction immediately after executing an instruction which sets the CM10 bit in the CM1 register to 1, and then insert at least 4 NOP instructions. When entering stop mode, the instruction queue reads ahead the instructions following the instruction which sets the CM10 bit to 1 (all clock stops), and, some of these may execute before the MCU enters stop mode or before the interrupt routine for returning from stop mode.

Program example when entering stop mode

```

Program Example:    FSET    I
                   BSET    CM10 ; Enter stop mode
                   JMP.B   L2    ; Insert JMP.B instruction
                   L1:
                   NOP     ; More than 4 NOP instructions
                   NOP
                   NOP
                   NOP

```

5. Wait until the main clock oscillation stabilization time, before switching the CPU clock source to the main clock.

Similarly, wait until the sub clock oscillates stably before switching the CPU clock source to the sub clock.

6. Suggestions to reduce power consumption

(a) Ports

The processor retains the state of each I/O port even when it goes to wait mode or to stop mode. A current flows in active I/O ports. A dash current may flow through the input ports in high impedance state, if the input is floating. When entering wait mode or stop mode, set non-used ports to input and stabilize the potential.

(b) A/D converter

When A/D conversion is not performed, set the VCUT bit in ADiCON1 register to 0 (no Vref connection). When A/D conversion is performed, start the A/D conversion at least 1 μ s or longer after setting the VCUT bit to 1 (Vref connection).

(c) Stopping peripheral functions

Use the CM0 register CM02 bit to stop the unnecessary peripheral functions during wait mode.

However, because the peripheral function clock (fc32) generated from the sub-clock does not stop, this measure is not conducive to reducing the power consumption of the chip. If low speed mode or low power dissipation mode is to be changed to wait mode, set the CM02 bit to 0 (do not peripheral function clock stopped when in wait mode), before changing wait mode.

(d) Switching the oscillation-driving capacity

Set the driving capacity to "LOW" when oscillation is stable.

22.3 Protection

Set the PRC2 bit to 1 (write enabled) and then write to any address, and the PRC2 bit will be cleared to 0 (write protected). The registers protected by the PRC2 bit should be changed in the next instruction after setting the PRC2 bit to 1. Make sure no interrupts or DMA transfers will occur between the instruction in which the PRC2 bit is set to 1 and the next instruction.

22.4 Interrupts

22.4.1 Reading Address 00000₁₆

Do not read the address 00000₁₆ in a program. When a maskable interrupt request is accepted, the CPU reads interrupt information (interrupt number and interrupt request priority level) from the address 00000₁₆ during the interrupt sequence. At this time, the IR bit for the accepted interrupt is cleared to 0. If the address 00000₁₆ is read in a program, the IR bit for the interrupt which has the highest priority among the enabled interrupts is cleared to 0. This causes a problem that the interrupt is canceled, or an unexpected interrupt request is generated.

22.4.2 Setting the SP

Set any value in the SP(USP, ISP) before accepting an interrupt. The SP(USP, ISP) is cleared to 0000₁₆ after reset. Therefore, if an interrupt is accepted before setting any value in the SP(USP, ISP), the program may go out of control.

22.4.3 $\overline{\text{NMI}}$ Interrupt

1. The $\overline{\text{NMI}}$ interrupt is invalid after reset. The $\overline{\text{NMI}}$ interrupt becomes effective by setting the PM24 bit in the PM2 register to "1". Set the PM24 bit to "1" when a high-level signal ("H") is applied to the $\overline{\text{NMI}}$ pin. If the PM24 bit is set to "1" when a low-level signal ("L") is applied, $\overline{\text{NMI}}$ interrupt is generated. Once $\overline{\text{NMI}}$ interrupt is enabled, it will not be disabled unless a reset is applied.
2. The input level of the $\overline{\text{NMI}}$ pin can be read by accessing the P8_5 bit in the P8 register.
3. When selecting $\overline{\text{NMI}}$ function, stop mode cannot be entered into while input on the $\overline{\text{NMI}}$ pin is low. This is because while input on the $\overline{\text{NMI}}$ pin is low the CM1 register's CM10 bit is fixed to 0.
4. When selecting $\overline{\text{NMI}}$ function, do not go to wait mode while input on the $\overline{\text{NMI}}$ pin is low. This is because when input on the $\overline{\text{NMI}}$ pin goes low, the CPU stops but CPU clock remains active; therefore, the current consumption in the chip does not drop. In this case, normal condition is restored by an interrupt generated thereafter.
5. When selecting $\overline{\text{NMI}}$ function, the low and high level durations of the input signal to the $\overline{\text{NMI}}$ pin must each be 2 CPU clock cycles + 300 ns or more.
6. When using the $\overline{\text{NMI}}$ interrupt for exiting stop mode, set the NDDR register to FF₁₆ (disable digital debounce filter) before entering stop mode.

22.4.4 Changing the Interrupt Generate Factor

If the interrupt generate factor is changed, the IR bit in the interrupt control register for the changed interrupt may inadvertently be set to 1 (interrupt requested). If you changed the interrupt generate factor for an interrupt that needs to be used, be sure to clear the IR bit for that interrupt to 0 (interrupt not requested).

"Changing the interrupt generate factor" referred to here means any act of changing the source, polarity or timing of the interrupt assigned to each software interrupt number. Therefore, if a mode change of any peripheral function involves changing the generate factor, polarity or timing of an interrupt, be sure to clear the IR bit for that interrupt to 0 (interrupt not requested) after making such changes. Refer to the description of each peripheral function for details about the interrupts from peripheral functions.

Figure 22.2 shows the procedure for changing the interrupt generate factor.

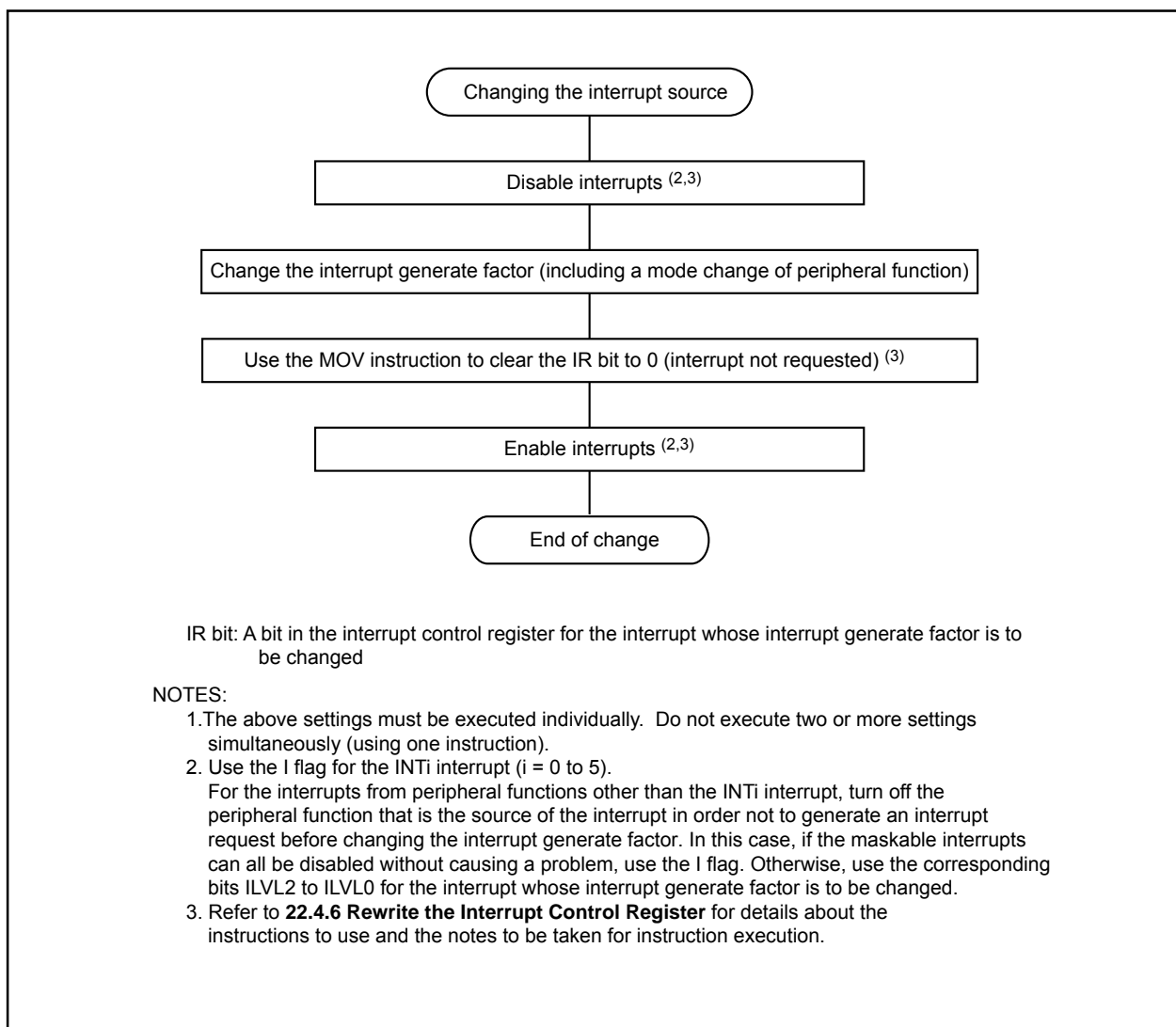


Figure 22.2 Procedure for Changing the Interrupt Generate Factor

22.4.5 INT Interrupt

1. Either an “L” level of at least $t_{w(INH)}$ or an “H” level of at least $t_{w(INL)}$ width is necessary for the signal input to pins INT₀ through INT₅ regardless of the CPU operation clock.
2. If the POL bit in registers INTOIC to INT5IC or bits IFSR7 to IFSR0 in the IFSR register are changed, the IR bit may inadvertently set to 1 (interrupt requested). Be sure to clear the IR bit to 0 (interrupt not requested) after changing any of those register bits.
3. When using the INT₅ interrupt for exiting stop mode, set the P17DDR register to FF₁₆ (disable digital debounce filter) before entering stop mode.

22.4.6 Rewrite the Interrupt Control Register

- (1) The interrupt control register for any interrupt should be modified in places where no requests for that interrupt may occur. Otherwise, disable the interrupt before rewriting the interrupt control register.
- (2) To rewrite the interrupt control register for any interrupt after disabling that interrupt, be careful with the instruction to be used.

Changing any bit other than the IR bit

If while executing an instruction, a request for an interrupt controlled by the register being modified occurs, the IR bit in the register may not be set to 1 (interrupt requested), with the result that the interrupt request is ignored. If such a situation presents a problem, use the instructions shown below to modify the register.

Usable instructions: AND, OR, BCLR, BSET

Changing the IR bit

Depending on the instruction used, the IR bit may not always be cleared to 0 (interrupt not requested). Therefore, be sure to use the MOV instruction to clear the IR bit.

- (3) When using the I flag to disable an interrupt, refer to the sample program fragments shown below as you set the I flag. (Refer to (2) for details about rewrite the interrupt control registers in the sample program fragments.)

Examples 1 through 3 show how to prevent the I flag from being set to 1 (interrupts enabled) before the interrupt control register is rewritten, due to the internal bus and the instruction queue buffer.

Example 1: Using the NOP instruction to keep the program waiting until the interrupt control register is modified

```
INT_SWITCH1:
  FCLR   I           ; Disable interrupts
  AND.B  #00h, 0055h ; Set the TA0IC register to 0016
  NOP
  NOP
  FSET   I           ; Enable interrupts
```

The number of NOP instruction is as follows.
 PM20 = 1 (1 wait) : 2, PM20 = 0 (2 waits): 3

Example 2: Using the dummy read to keep the FSET instruction waiting

```
INT_SWITCH2:
  FCLR   I           ; Disable interrupts
  AND.B  #00h, 0055h ; Set the TA0IC register to 0016
  MOV.W  MEM, R0     ; Dummy read
  FSET   I           ; Enable interrupts
```

Example 3: Using the POPC instruction to changing the I flag

```
INT_SWITCH3:
  PUSHC  FLG
  FCLR   I           ; Disable interrupts
  AND.B  #00h, 0055h ; Set the TA0IC register to 0016
  POPC   FLG        ; Enable interrupts
```

22.4.7 Watchdog Timer Interrupt

Initialize the watchdog timer after the watchdog timer interrupt occurs.

22.5 DMAC

22.5.1 Write to DMAE Bit in DMiCON Register

When both of the conditions below are met, follow the steps below.

(a) Conditions

- The DMAE bit is set to 1 again while it remains set (DMAi is in an active state).
- A DMA request may occur simultaneously when the DMAE bit is being written.

(b) Procedure

- (1) Write 1 to the DMAE bit and DMAS bit in DMiCON register simultaneously⁽¹⁾.
- (2) Make sure that the DMAi is in an initial state⁽²⁾ in a program.

If the DMAi is not in an initial state, the above steps should be repeated.

NOTES:

1. The DMAS bit remains unchanged even if 1 is written. However, if 0 is written to this bit, it is set to 0 (DMA not requested). In order to prevent the DMAS bit from being modified to 0, 1 should be written to the DMAS bit when 1 is written to the DMAE bit. In this way the state of the DMAS bit immediately before being written can be maintained.
Similarly, when writing to the DMAE bit with a read-modify-write instruction, 1 should be written to the DMAS bit in order to maintain a DMA request which is generated during execution.
2. Read the TCRi register to verify whether the DMAi is in an initial state. If the read value is equal to a value which was written to the TCRi register before DMA transfer start, the DMAi is in an initial state. (If a DMA request occurs after writing to the DMAE bit, the value written to the TCRi register is 1.) If the read value is a value in the middle of transfer, the DMAi is not in an initial state.

22.6 Timers

22.6.1 Timer A

22.6.1.1 Timer A (Timer Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAI_iMR (i = 0 to 4) register and the TAI register before setting the TAI_S bit in the TABSR register to 1 (count starts).
Always make sure the TAI_iMR register is modified while the TAI_S bit remains 0 (count stops) regardless whether after reset or not.
2. While counting is in progress, the counter value can be read out at any time by reading the TAI register. However, if the TAI register is read at the same time the counter is reloaded, the read value is always FFFF₁₆. If the TAI register is read after setting a value in it, but before the counter starts counting, the read value is the one that has been set in the register.
3. If a low-level signal is applied to the \overline{SD} pin when the IVPCR1 bit in the TB2SC register is set to 1 (three-phase output forcible cutoff by input on \overline{SD} pin enabled), the TA1_{OUT}, TA2_{OUT} and TA4_{OUT} pins go to a high-impedance state.

22.6.1.2 Timer A (Event Counter Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAI_iMR (i = 0 to 4) register, the TAI register, the UDF register, bits TAZIE, TA0TGL, and TA0TGH in the ONSF register and the TRGSR register before setting the TAI_S bit in the TABSR register to 1 (count starts).
Always make sure bits TAZIE, TA0TGL, and TA0TGH in the TAI_iMR register, the UDF register, the ONSF register, and the TRGSR register are modified while the TAI_S bit remains 0 (count stops) regardless whether after reset or not.
2. While counting is in progress, the counter value can be read out at any time by reading the TAI register. However, if the TAI register is read at the same time the counter is reloaded, the read value is always FFFF₁₆ when the timer counter underflows and 0000₁₆ when the timer counter overflows. If the TAI register is read after setting a value in it, but before the counter starts counting, the read value is the one that has been set in the register.
3. If a low-level signal is applied to the \overline{SD} pin when the IVPCR1 bit in the TB2SC register is set to 1 (three-phase output forcible cutoff by input on \overline{SD} pin enabled), the TA1_{OUT}, TA2_{OUT} and TA4_{OUT} pins go to a high-impedance state.

22.6.1.3 Timer A (One-shot Timer Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAI_{MR} (i = 0 to 4) register, the TAI register, bits TA0TGL and TA0TGH in the ONSF register and the TRGSR register before setting the TAI_S bit in the TABSR register to 1 (count starts).
Always make sure bits TA0TGL and TA0TGH in the TAI_{MR} register, the ONSF register, and the TRGSR register are modified while the TAI_S bit remains 0 (count stops) regardless whether after reset or not.
2. When setting TAI_S bit to 0 (count stop), the followings occur:
 - A counter stops counting and a content of reload register is reloaded.
 - TAI_{OUT} pin outputs "L".
 - After one cycle of the CPU clock, the IR bit in TAI_{IC} register is set to 1 (interrupt request).
3. Output in one-shot timer mode synchronizes with a count source internally generated. When the external trigger has been selected, a maximum delay of one cycle of the count source occurs between the trigger input to TAI_{IN} pin and output in one-shot timer mode.
4. The IR bit is set to 1 when timer operation mode is set with any of the following procedures:
 - Select one-shot timer mode after reset.
 - Change an operation mode from timer mode to one-shot timer mode.
 - Change an operation mode from event counter mode to one-shot timer mode.To use the timer A_i interrupt (the IR bit), set the IR bit to 0 after the changes listed above have been made.
5. When a trigger occurs while the timer is counting, the counter reloads the reload register value, and continues counting after a second trigger is generated and the counter is decremented once. To generate a trigger while counting, space more than one cycle of the timer count source from the first trigger and generate again.
6. When selecting the external trigger for the count start conditions in timer A one-shot timer mode, do generate an external trigger 300ns before the count value of timer A is set to 0000₁₆. The one-shot timer does not continue counting and may stop.
7. If a low-level signal is applied to the \overline{SD} pin when the IVPCR1 bit in the TB2SC register is set to 1 (three-phase output forcible cutoff by input on \overline{SD} pin enabled), the TA1_{OUT}, TA2_{OUT} and TA4_{OUT} pins go to a high-impedance state.

22.6.1.4 Timer A (Pulse Width Modulation Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using bits TA0TGL and TA0TGH in the TAI_iMR (i = 0 to 4) register, the TAI register, the ONSF register and the TRGSR register before setting the TAI_iS bit in the TABSR register to 1 (count starts).
Always make sure bits TA0TGL and TA0TGH in the TAI_iMR register, the ONSF register and the TRGSR register are modified while the TAI_iS bit remains 0 (count stops) regardless whether after reset or not.
2. The IR bit is set to 1 when setting a timer operation mode with any of the following procedures:
 - Select the PWM mode after reset.
 - Change an operation mode from timer mode to PWM mode.
 - Change an operation mode from event counter mode to PWM mode.To use the timer A_i interrupt (interrupt request bit), set the IR bit to 0 by program after the above listed changes have been made.
3. When setting TAI_iS register to 0 (count stop) during PWM pulse output, the following action occurs:
 - Stop counting.
 - When TAI_iOUT pin is output “H”, output level is set to “L” and the IR bit is set to 1.
 - When TAI_iOUT pin is output “L”, both output level and the IR bit remains unchanged.
4. If a low-level signal is applied to the \overline{SD} pin when the IVPCR1 bit in the TB2SC register is set to 1 (three-phase output forcible cutoff by input on \overline{SD} pin enabled), the TA1_{OUT}, TA2_{OUT} and TA4_{OUT} pins go to a high-impedance state.

22.6.2 Timer B

22.6.2.1 Timer B (Timer Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TBiMR ($i = 0$ to 2) register and TBi register before setting the TBiS bit in the TABSR register to 1 (count starts).

Always make sure the TBiMR register is modified while the TBiS bit remains 0 (count stops) regardless whether after reset or not.

2. The counter value can be read out at any time by reading the TBi register. However, if this register is read at the same time the counter is reloaded, the read value is always FFFF₁₆. If the TBi register is read after setting a value in it but before the counter starts counting, the read value is the one that has been set in the register.

22.6.2.2 Timer B (Event Counter Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TBiMR ($i = 0$ to 2) register and TBi register before setting the TBiS bit in the TABSR register to 1 (count starts).

Always make sure the TBiMR register is modified while the TBiS bit remains 0 (count stops) regardless whether after reset or not.

2. The counter value can be read out at any time by reading the TBi register. However, if this register is read at the same time the counter is reloaded, the read value is always FFFF₁₆. If the TBi register is read after setting a value in it but before the counter starts counting, the read value is the one that has been set in the register.

22.6.2.3 Timer B (Pulse Period/pulse Width Measurement Mode)

1. The timer remains idle after reset. Set the mode, count source, etc. using the TBiMR ($i = 0$ to 2) register before setting the TBiS bit in the TABSR or the TBSR register to 1 (count starts).

Always make sure the TBiMR register is modified while the TBiS bit remains 0 (count stops) regardless whether after reset or not. To clear the MR3 bit to 0 by writing to the TBiMR register while the TBiS bit is set to 1 (count starts), be sure to write the same value as previously written to bits TMOD0, TMOD1, MR0, MR1, TCK0, and TCK1 and a 0 to the MR2 bit.

2. The IR bit in TBiIC register ($i=0$ to 2) goes to 1 (interrupt request), when an effective edge of a measurement pulse is input or timer Bi is overflowed. The factor of interrupt request can be determined by use of the MR3 bit in TBiMR register within the interrupt routine.
3. If the source of interrupt cannot be identified by the MR3 bit such as when the measurement pulse input and a timer overflow occur at the same time, use another timer to count the number of times timer B has overflowed.
4. To set the MR3 bit to 0 (no overflow), set TBiMR register with setting the TBiS bit to 1 and counting the next count source after setting the MR3 bit to 1 (overflow).
5. Use the IR bit in TBiIC register to detect only overflows. Use the MR3 bit only to determine the interrupt factor within the interrupt routine.

6. When a count is started and the first effective edge is input, an undefined value is transferred to the reload register. At this time, timer Bi interrupt request is not generated.
7. A value of the counter is undefined at the beginning of a count. MR3 may be set to 1 and timer Bi interrupt request may be generated between a count start and an effective edge input.
8. For pulse width measurement, pulse widths are successively measured. Use program to check whether the measurement result is an "H" level width or an "L" level width.

22.6.3 Three-phase Motor Control Timer Function

When the IVPCR1 bit in the TB2SC register is set to 1 (three-phase output forced cutoff by SD pin input (high-impedance) enabled), the INV03 bit in the INVC0 register is set to 1 (three-phase motor control timer output enabled), and a low-level ("L") signal is applied to the \overline{SD} pin while a three-phase PWM signal is output, the MCU is forced to cutoff and pins U, \overline{U} , V, \overline{V} , W, and \overline{W} are placed in a high-impedance state and the INV03 bit is set to 0 (three-phase motor control timer output disabled).

To resume the three-phase PWM signal output from pins U, \overline{U} , V, \overline{V} , W, and \overline{W} , set the INV03 bit to 1 and the IVPCR1 bit to 0 (three-phase output forced cutoff disabled) after the \overline{SD} pin level becomes "H". Then set the IVPCR1 bit to 1 (three-phase output forced cutoff enabled) in order to enable the three-phase output forced cutoff function by input to the SD pin again.

The INV03 bit cannot be set to 1 while an "L" signal is input to the \overline{SD} pin. To set the INV03 bit to 1 after forcible cutoff, write 1 to the INV03 bit and read the bit to ensure that it is set to 1 by program. Then set the IVPCR1 bit to 1 after setting it to 0.

22.7 Timer S

22.7.1 Rewrite the G1IR Register

Bits in the G1IR register are not automatically set to 0 (no interrupt requested) even if a requested interrupt is acknowledged. Set each bit to 0 by program after the interrupt requests are verified.

The IC/OC interrupt is generated when any bit in the G1IR register is set to 1 (interrupt requested) after all the bits are set to 0. If conditions to generate an interrupt are met when the G1IR register holds the value other than 00₁₆, the IC/OC interrupt request will not be generated. In order to enable an IC/OC interrupt request again, clear the G1IR register to 00₁₆. Use the following instructions to set each bit in the G1IR register to 0.

Subject instructions: AND, BCL

Figure 22.3 shows an example of IC/OC interrupt i flow chart.

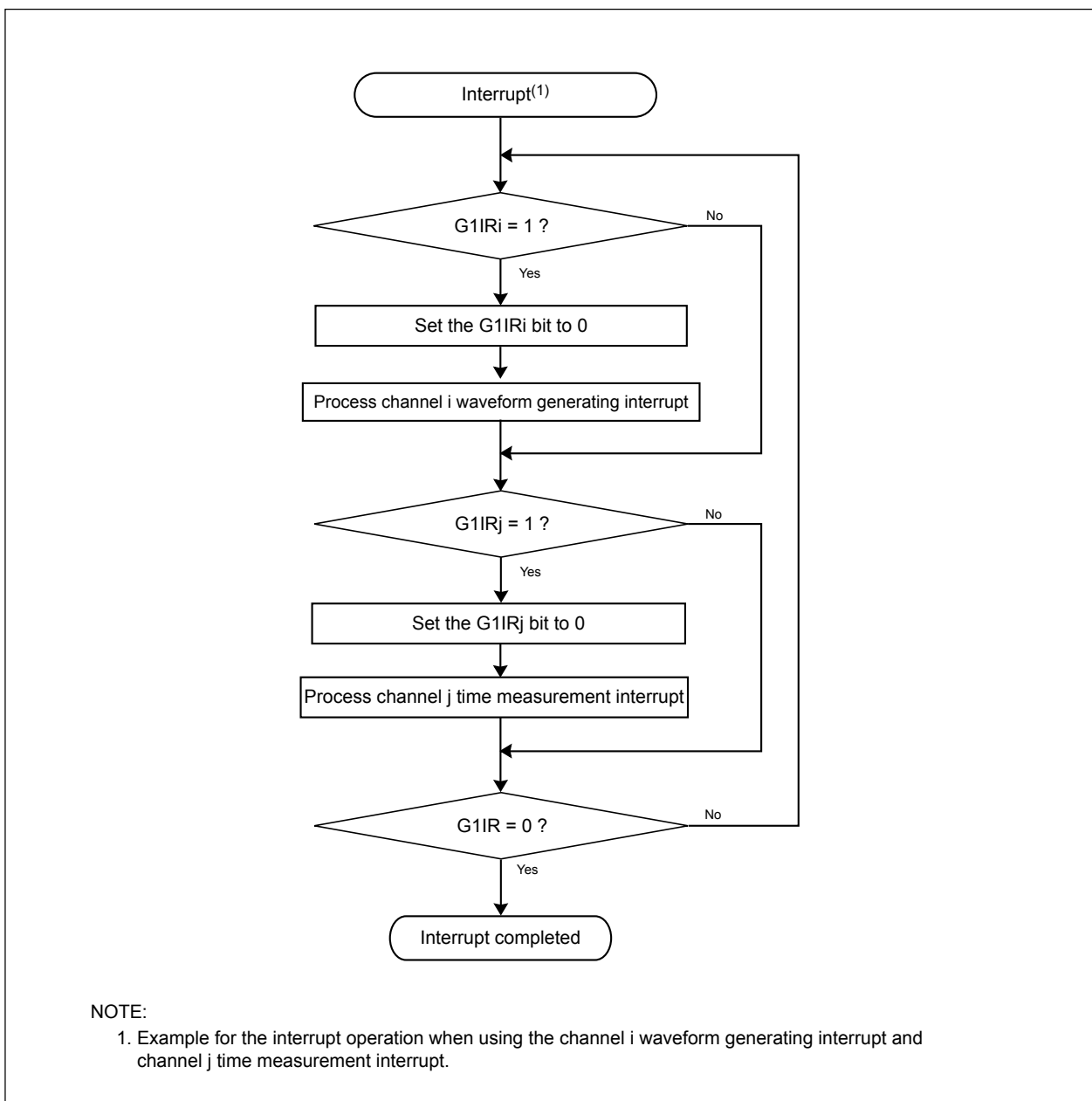


Figure 22.3 IC/OC Interrupt i Flow Chart

22.7.2 Rewrite the ICOCiIC Register

When the interrupt request to the ICOCiIC register is generated during the instruction process, the IR bit may not be set to 1 (interrupt requested) and the interrupt request may not be acknowledged. At that time, when the bit in the G1IR register is held to 1 (interrupt requested), the following IC/OC interrupt request will not be generated. When changing the ICOCiIC register setting, use the following instruction.

Subject instructions: AND, OR, BCLR, BSET

When initializing Timer S, change the ICOCiIC register setting with the request again after setting registers IOCiIC and G1IR to 00₁₆.

22.7.3 Waveform Generating Function

1. If the BTS bit in the G1BCR1 register is set to 0 (base timer is reset) when the waveform is generating and the base timer is stopped counting, the waveform output pin keeps the same output level. The output level will be changed when the base timer and the G1POj register match the setting value next time after the base timer starts counting again.
2. If the G1POCRj register is set when the waveform is generated, the same setting value of the IVL bit is applied to the waveform generating pin. Do not set the G1POCRj register when the waveform is generating.
3. When the RST1 bit in the G1BCR1 register is set to 1 (the base timer is reset by matching the G1PO0 register), the base timer is reset after two clock cycles of f_{BT1} when the base timer value matches the G1PO0 register value. A high-level ("H") signal is applied to the OUTC10 pin between the base timer value match to the base timer reset.

22.7.4 IC/OC Base Timer Interrupt

If the MCU is operated in the combination selected from **Table 22.1** for use when the RST4 bit in the G1BCR0 register is set to 1 (reset the base timer that matches the G1BTRR register) to reset the base timer, an IC/OC base timer interrupt request is generated twice.

Table 22.1 Uses of IT Bit in the G1BCR0 Register and G1BTRR Register

| IT Bit in the G1BCR0 Register | G1BTRR Register |
|--|---|
| 0 (bit 15 in the base timer overflows) | 07FFF ₁₆ to 0FFFE ₁₆ |
| 1 (bit 14 in the base timer overflows) | 03FFF ₁₆ to 0FFFE ₁₆ or 0BFFF ₁₆ to 0FFFE ₁₆ |

The second IC/OC base timer interrupt request is generated because the base timer overflow request is generated after one f_{BT1} clock cycle as soon as the base timer is reset.

One of the following conditions must be met in order not to generate the IC/OC base timer interrupt request twice:

- 1) When the RST4 bit is set to 1, set the G1BTRR register with a combination other than what is listed in **Table 22.1**.
- 2) Do not reset the base timer by matching the G1BTRR register. Reset the base timer by matching the G1P00 register. In other words, do not set the RST4 bit to 1 to reset the base timer. Set the RST1 bit in the G1BCR1 register to 1 (reset the base timer that matches the G1P00 register).

22.8 Serial I/O

22.8.1 Clock-Synchronous Serial I/O

22.8.1.1 Transmission/reception

1. With an external clock selected, and choosing the $\overline{\text{RTS}}$ function, the output level of the $\overline{\text{RTSi}}$ pin goes to "L" when the data-receivable status becomes ready, which informs the transmission side that the reception has become ready. The output level of the $\overline{\text{RTSi}}$ pin goes to "H" when reception starts. So if the $\overline{\text{RTSi}}$ pin is connected to the $\overline{\text{CTSi}}$ pin on the transmission side, the circuit can transmission and reception data with consistent timing. With the internal clock, the $\overline{\text{RTS}}$ function has no effect.
2. If a low-level signal is applied to the $\overline{\text{SD}}$ pin when the IVPCR1 bit in the TB2SC register is set to 1 (three-phase output forcible cutoff by input on $\overline{\text{SD}}$ pin enabled), the P7₃/RTS₂/TxD₁(when the U1MAP bit in PACR register is 1) and CLK₂ pins go to a high-impedance state.

22.8.1.2 Transmission

When an external clock is selected, the conditions must be met while if the CKPOL bit in the UiC0 register is set to 0 (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock), the external clock is in the high state; if the CKPOL bit in the UiC0 register is set to 1 (transmit data output at the rising edge and the receive data taken in at the falling edge of the transfer clock), the external clock is in the low state.

- The TE bit in UiC1 register is set to 1 (transmission enabled)
- The TI bit in UiC1 register is set to 0 (data present in UiTB register)
- If $\overline{\text{CTS}}$ function is selected, input on the $\overline{\text{CTSi}}$ pin is set to "L"

22.8.1.3 Reception

1. In operating the clock-synchronous serial I/O, operating a transmitter generates a shift clock. Fix settings for transmission even when using the device only for reception. Dummy data is output to the outside from the TxDi pin when receiving data.
2. When an internal clock is selected, set the TE bit in the UiC1 register (i = 0 to 2) to 1 (transmission enabled) and write dummy data to the UiTB register, and the shift clock will thereby be generated. When an external clock is selected, set the TE bit in the UiC1 register (i = 0 to 2) to 1 and write dummy data to the UiTB register, and the shift clock will be generated when the external clock is fed to the CLKi input pin.
3. When successively receiving data, if all bits of the next receive data are prepared in the UARTi receive register while the RE bit in the UiC1 register (i = 0 to 2) is set to 1 (data present in the UiRB register), an overrun error occurs and the UiRB register OER bit is set to 1 (overrun error occurred). In this case, because the content of the UiRB register is undefined, a corrective measure must be taken by programs on the transmit and receive sides so that the valid data before the overrun error occurred will be retransmitted. Note that when an overrun error occurred, the SiRIC register IR bit does not change state.
4. To receive data in succession, set dummy data in the lower-order byte of the UiTB register every time reception is made.
5. When an external clock is selected, make sure the external clock is in high state if the CKPOL bit is set to 0, and in low state if the CKPOL bit is set to 1 before the following conditions are met:
 - The RE bit in the UiC1 register is set to 1 (reception enabled)
 - The TE bit in the UiC1 register is set to 1 (transmission enabled)
 - The TI bit in the UiC1 register = 0 (data present in the UiTB register)

22.8.2 UART Mode

22.8.2.1 Special Mode 1 (I²C bus Mode)

When generating start, stop and restart conditions, set the STSPSEL bit in the U2SMR4 register to 0 and wait for more than half cycle of the transfer clock before setting each condition generate bit (STAREQ, RSTAREQ and STPREQ) from 0 to 1.

22.8.2.2 Special Mode 2

If a low-level signal is applied to the \overline{SD} pin when the IVPCR1 bit in the TB2SC register is set to 1 (three-phase output forcible cutoff by input on \overline{SD} pin enabled), the RTS2 and CLK2 pins go to a high-impedance state.

22.8.2.3 Special Mode 4 (SIM Mode)

A transmit interrupt request is generated by setting the U2C1 register U2IRS bit to 1 (transmission complete) and U2ERE bit to 1 (error signal output) after reset. Therefore, when using SIM mode, be sure to clear the IR bit to 0 (no interrupt request) after setting these bits.

22.8.3 SI/O3, SI/O4

The SOUT_i default value which is set to the SOUT_i pin by the SM_i7 bit approximately 10ns may be output when changing the SM_i3 bit from 0 (I/O port) to 1 (SOUT_i output and CLKfunction) while the SM_i2 bit in the SiC (i=3 and 4) to 0 (SOUT_i output) and the SM_i6 bit is set to 1 (internal clock). And then the SOUT_i pin is held high-impedance.

If the level which is output from the SOUT_i pin is a problem when changing the SM_i3 bit from 0 to 1, set the default value of the SOUT_i pin by the SM_i7 bit.

22.9 A/D Converter

1. Set registers ADCON0 (except bit 6), ADCON1, ADCON2 and ADTRGCON when A/D conversion is stopped (before a trigger occurs).
2. When the VCUT bit in ADCON1 register is changed from 0 (Vref not connected) to 1 (Vref connected), start A/D conversion after passing 1 μ s or longer.
3. To prevent noise-induced device malfunction or latchup, as well as to reduce conversion errors, insert capacitors between the AVCC, VREF, and analog input pins (AN_i, AN0_i, AN2_i(i=0 to 7), and AN3_i(i=0 to 2)) each and the AVSS pin. Similarly, insert a capacitor between the VCC1 pin and the VSS pin. **Figure 22.4** is an example connection of each pin.
4. Make sure the port direction bits for those pins that are used as analog inputs are set to 0 (input mode). Also, if the TGR bit in the ADCON0 register is set to 1 (external trigger), make sure the port direction bit for the $\overline{\text{ADTRG}}$ pin is set to 0 (input mode).
5. When using key input interrupts, do not use any of the four AN4 to AN7 pins as analog inputs. (A key input interrupt request is generated when the A/D input voltage goes low.)
6. The ϕ_{AD} frequency must be 10 MHz or less. Without sample-and-hold function, limit the ϕ_{AD} frequency to 250kHz or more. With the sample and hold function, limit the ϕ_{AD} frequency to 1MHz or more.
7. When changing an A/D operation mode, select analog input pin again in bits CH2 to CH0 in the ADCON0 register and bits SCAN1 to SCAN0 in the ADCON1 register.

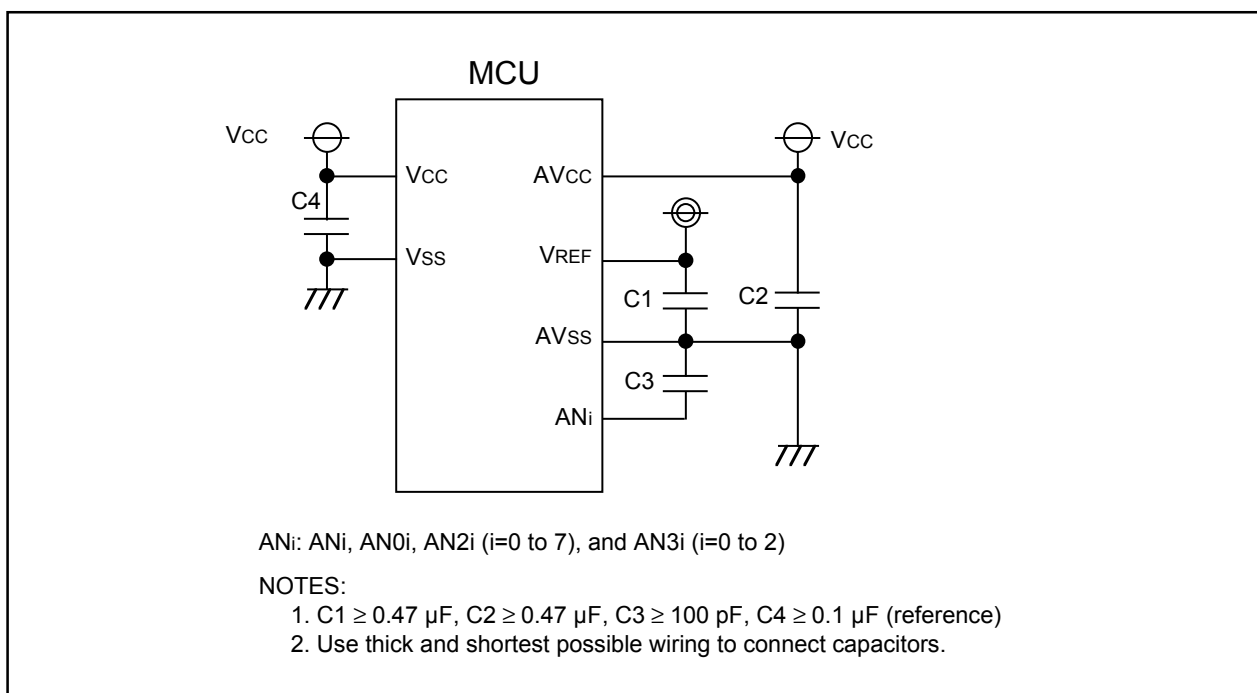


Figure 22.4 Use of capacitors to reduce noise

8. If the CPU reads the ADi register ($i = 0$ to 7) at the same time the conversion result is stored in the ADi register after completion of A/D conversion, an incorrect value may be stored in the ADi register. This problem occurs when a divide-by-n clock derived from the main clock or a subclock is selected for CPU clock.

- When operating in one-shot, single-sweep mode, simultaneous sample sweep mode, delayed trigger mode 0 or delayed trigger mode 1
Check to see that A/D conversion is completed before reading the target ADi register. (Check the ADIC register's IR bit to see if A/D conversion is completed.)
- When operating in repeat mode or repeat sweep mode 0 or 1
Use the main clock for CPU clock directly without dividing it.

9. If A/D conversion is forcibly terminated while in progress by setting the ADST bit in the ADCON0 register to 0 (A/D conversion halted), the conversion result of the A/D converter is undefined. The contents of ADi registers irrelevant to A/D conversion may also become undefined. If while A/D conversion is underway the ADST bit is cleared to 0 in a program, ignore the values of all ADi registers.

10. When setting the ADST bit in the ADCON register to 0 and terminating forcefully by a program in single sweep conversion mode, A/D delayed trigger mode 0 and A/D delayed trigger mode 1 during A/D converting operation, the A/D interrupt request may be generated. If this causes a problem, set the ADST bit to 0 after an interrupt is disabled.

22.10 Multi-Master I²C bus Interface

22.10.1 Writing to the S00 Register

When the start condition is not generated, the SCL pin may output the short low-signal ("L") by setting the S00 register. Set the register when the SCL pin outputs an "L" signal.

22.10.2 AL Flag

When the arbitration lost is generated and the AL flag in the S10 register is set to 1 (detected), the AL flag can be cleared to 0 (not detected) by writing a transmit data to the S00 register. The AL flag should be cleared at the timing when master generates the start condition to start a new transfer.

22.11 CAN Module

22.11.1 Reading C0STR Register

The CAN module on the M16C/29 Group updates the status of the C0STR register in a certain period. When the CPU and the CAN module access to the C0STR register at the same time, the CPU has the access priority; the access from the CAN module is disabled. Consequently, when the updating period of the CAN module matches the access period from the CPU, the status of the CAN module cannot be updated. (See **Figure 22.5**)

Accordingly, be careful about the following points so that the access period from the CPU should not match the updating period of the CAN module:

- (1) There should be a wait time of $3f_{CAN}$ or longer (see **Table 22.2**) before the CPU reads the C0STR register. (See **Figure 22.6**)
- (2) When the CPU polls the C0STR register, the polling period must be $3f_{CAN}$ or longer. (See **Figure 22.7**)

Table 22.2 CAN Module Status Updating Period

| $3f_{CAN}$ period = $3 \times X_{IN}$ (Original oscillation period) \times Division value of the CAN clock (CCLK) | |
|---|--|
| (Example 1) Condition X_{IN} 16 MHz CCLK: Divided by 1 | $3f_{CAN}$ period = $3 \times 62.5 \text{ ns} \times 1 = 187.5 \text{ ns}$ |
| (Example 2) Condition X_{IN} 16 MHz CCLK: Divided by 2 | $3f_{CAN}$ period = $3 \times 62.5 \text{ ns} \times 2 = 375 \text{ ns}$ |
| (Example 3) Condition X_{IN} 16 MHz CCLK: Divided by 4 | $3f_{CAN}$ period = $3 \times 62.5 \text{ ns} \times 4 = 750 \text{ ns}$ |
| (Example 4) Condition X_{IN} 16 MHz CCLK: Divided by 8 | $3f_{CAN}$ period = $3 \times 62.5 \text{ ns} \times 8 = 1.5 \mu\text{s}$ |
| (Example 5) Condition X_{IN} 16 MHz CCLK: Divided by 16 | $3f_{CAN}$ period = $3 \times 62.5 \text{ ns} \times 16 = 3 \mu\text{s}$ |

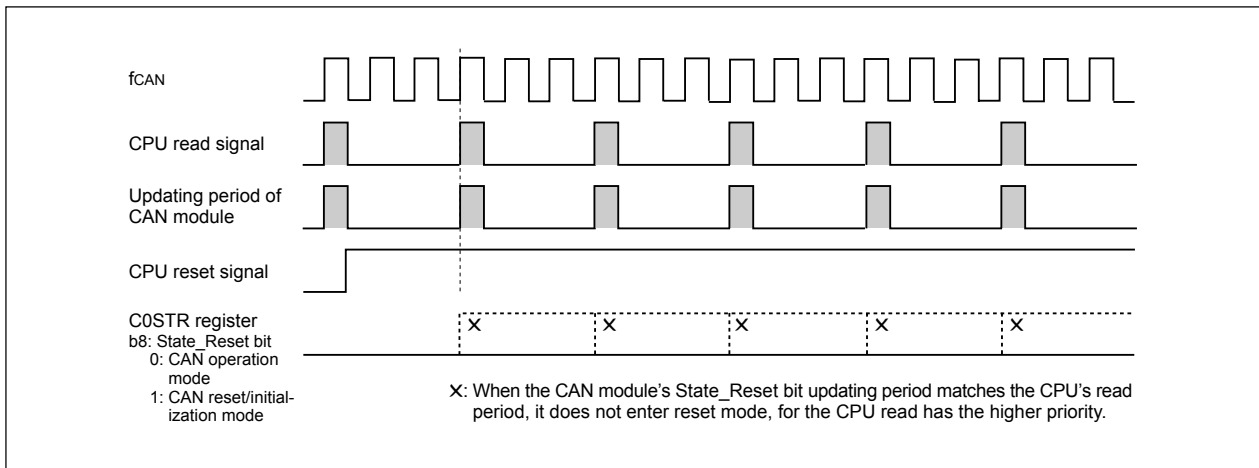


Figure 22.5 When Updating Period of CAN Module Matches Access Period from CPU

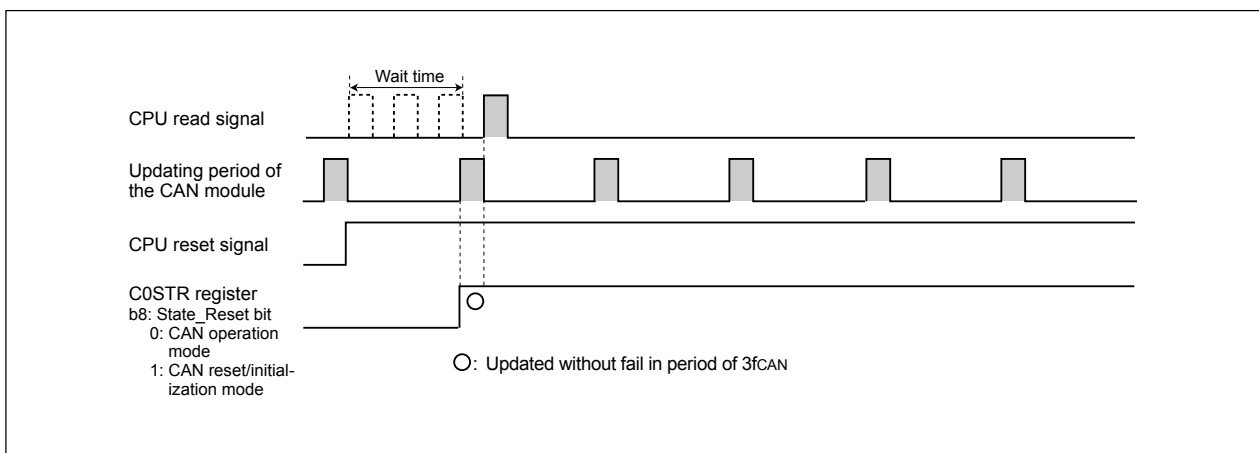


Figure 22.6 With a Wait Time of 3fCAN Before CPU Read

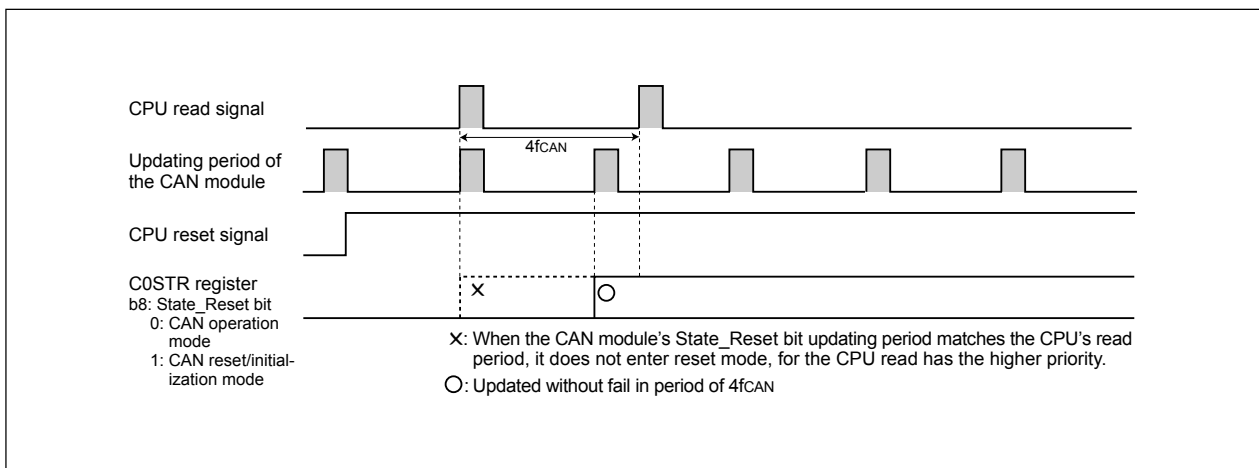


Figure 22.7 When Polling Period of CPU is 3fCAN or Longer

22.11.2 CAN Transceiver in Boot Mode

When programming the flash memory in boot mode via CAN bus, the operation mode of CAN transceiver should be set to “high-speed mode” or “normal operation mode”. If the operation mode is controlled by the MCU, CAN transceiver must be set the operation mode to “high-speed mode” or “normal operation mode” before programming the flash memory by changing the switch etc. **Tables 22.3 and 22.4** show pin connections of CAN transceiver.

Table 22.3 Pin Connections of CAN Transceiver (In case of PCA82C250: Philips product)

| | Standby mode | High-speed mode |
|-------------------|--------------|-----------------|
| Rs pin (Note 1) | “H” | “L” |
| CAN communication | impossible | possible |
| Connection | | |

Note 1: The pin which controls the operation mode of CAN transceiver.

Note 2: Connect to enabled port to control CAN transceiver.

Table 22.4 Pin Connections of CAN Transceiver (In case of PCA82C252: Philips product)

| | Sleep mode | Normal operation mode |
|-------------------|------------|-----------------------|
| STB pin (Note 1) | “L” | “H” |
| EN pin (Note 1) | “L” | “H” |
| CAN communication | impossible | possible |
| Connection | | |

Note 1: The pin which controls the operation mode of CAN transceiver.

Note 2: Connect to enabled port to control CAN transceiver.

22.12 Programmable I/O Ports

1. If a low-level signal is applied to the \overline{SD} pin when the IVPCR1 bit in the TB2SC register is set to 1 (three-phase output forcible cutoff by input on \overline{SD} pin enabled), the P72 to P75, P80 and P81 pins go to a high-impedance state.

2. The input threshold voltage of pins differs between programmable input/output ports and peripheral functions.

Therefore, if any pin is shared by a programmable input/output port and a peripheral function and the input level at this pin is outside the range of recommended operating conditions V_{IH} and V_{IL} (neither "high" nor "low"), the input level may be determined differently depending on which side—the programmable input/output port or the peripheral function—is currently selected.

3. When the SM32 bit in the S3C register is set to 1, the P32 pin goes to high-impedance state. When the SM42 bit in the S4C register is set to 1, the P96 pin goes to high-impedance state.

4. When the INV03 bit in the INVC0 register is 1 (three-phase motor control timer output enabled), an "L" input on the P85 $\overline{NMI}/\overline{SD}$ pin, has the following effect.

- When the TB2SC register IVPCR1 bit is set to 1 (three-phase output forcible cutoff by input on \overline{SD} pin enabled), the U/ \overline{U} / V/ \overline{V} / W/ \overline{W} pins go to a high-impedance state.
- When the TB2SC register IVPCR1 bit is set to 0 (three-phase output forcible cutoff by input on \overline{SD} pin disabled), the U/ \overline{U} / V/ \overline{V} / W/ \overline{W} pins go to a normal port.

Therefore, the P85 pin can not be used as programmable I/O port when the INV03 bit is set to 1.

When the \overline{SD} function isn't used, set to 0 (Input) in PD85 and pullup to H in the P85 $\overline{NMI}/\overline{SD}$ pin from outside.

22.13 Electric Characteristic Differences Between Mask ROM and Flash Memory Version

Flash memory version and mask ROM version may have different characteristics, operating margin, noise tolerated dose, noise width dose in electrical characteristics due to internal ROM, different layout pattern, etc. When switching to the mask ROM version, conduct equivalent tests as system evaluation tests conducted in the flash memory version.

22.14 Mask ROM Version

22.14.1 Internal ROM Area

In the masked ROM version, do not write to internal ROM area. Writing to the area may increase power consumption.

22.14.2 Reserved Bit

The b3 to b0 in addresses 0FFFFFF₁₆ are reserved bits. Set these bits to 1112.

22.15 Flash Memory Version

22.15.1 Functions to Inhibit Rewriting Flash Memory Rewrite

ID codes are stored in addresses 0FFFFDF₁₆, 0FFFE3₁₆, 0FFFE3₁₆, 0FFFEF₁₆, 0FFFF3₁₆, 0FFFF7₁₆, and 0FFFFB₁₆. If wrong data are written to these addresses, the flash memory cannot be read or written in standard serial I/O mode.

The ROMCP register is mapped in address 0FFFFFF₁₆. If wrong data is written to this address, the flash memory cannot be read or written in parallel I/O mode.

In the flash memory version of MCU, these addresses are allocated to the vector addresses ("H") of fixed vectors. The b3 to b0 in address 0FFFFFF₁₆ are reserved bits. Set these bits to 11112.

22.15.2 Stop Mode

When the MCU enters stop mode, execute the instruction which sets the CM10 bit to 1 (stop mode) after setting the FMR01 bit to 0 (CPU rewrite mode disabled) and disabling the DMA transfer.

22.15.3 Wait Mode

When the MCU enters wait mode, execute the WAIT instruction after setting the FMR01 bit to 0 (CPU rewrite mode disabled).

22.15.4 Low Power Dissipation Mode, On-Chip Oscillator Low Power Dissipation Mode

If the CM05 bit is set to 1 (main clock stop), the following commands must not be executed.

- Program
- Block erase

22.15.5 Writing Command and Data

Write the command code and data at even addresses.

22.15.6 Program Command

Write xx40₁₆ 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 even address as the write address specified in the second bus cycle.

22.15.7 Operation Speed

When CPU clock source is main clock, before entering CPU rewrite mode (EW mode 0 or 1), select 10 MHz or less for BCLK using the CM06 bit in the CM0 register and bits CM17 to CM16 in the CM1 register. Also, when CPU clock is f₃(ROC) on-chip oscillator clock, before entering CPU rewrite mode (EW mode 0 or 1), set the ROCR3 to ROCR2 bits in the ROCR register to "divided by 4" or "divide by 8". On both cases, set the PM17 bit in the PM1 register to 1 (with wait state).

22.15.8 Instructions Inhibited Against Use

The following instructions cannot be used in EW mode 0 because the flash memory's internal data is referenced: UND instruction, INTO instruction, JMPS instruction, JSRS instruction, and BRK instruction

22.15.9 Interrupts

EW Mode 0

- Any interrupt which has a vector in the variable vector table can be used providing that its vector is transferred into the RAM area.
- The $\overline{\text{NMI}}$ and watchdog timer interrupts can be used because the FMR0 register and FMR1 register are initialized when one of those interrupts occurs. The jump addresses for those interrupt service routines should be set in the fixed vector table.

Because the rewrite operation is halted when a $\overline{\text{NMI}}$ or watchdog timer interrupt occurs, the rewrite program must be executed again after exiting the interrupt service routine.

- The address match interrupt cannot be used because the flash memory's internal data is referenced.

EW Mode 1

- Make sure that any interrupt which has a vector in the variable vector table or address match interrupt will not be accepted during the auto program period or auto erase period with erase-suspend function disabled.
- The $\overline{\text{NMI}}$ interrupt can be used because the FMR0 register and FMR1 register are initialized when this interrupt occurs. The jump address for the interrupt service routine should be set in the fixed vector table.

Because the rewrite operation is halted when a $\overline{\text{NMI}}$ interrupt occurs, the rewrite program must be executed again after exiting the interrupt service routine.

22.15.10 How to Access

To set the FMR01, FMR02, FMR11 or FMR16 bit to 1, set the subject bit to 1 immediately after setting to 0. Do not generate an interrupt or a DMA transfer between the instruction to set the bit to 0 and the instruction to set the bit to 1. Set the bit when the PM24 bit is set to 1 ($\overline{\text{NMI}}$ function) and an high-level ("H") signal is applied to the $\overline{\text{NMI}}$ pin.

22.15.11 Writing in the User ROM Area

EW Mode 0

- If the power supply voltage drops while rewriting any block in which the rewrite control program is stored, a problem may occur that the rewrite control program is not correctly rewritten and, consequently, the flash memory becomes unable to be rewritten thereafter. In this case, standard serial I/O or parallel I/O mode should be used.

EW Mode 1

- Avoid rewriting any block in which the rewrite control program is stored.

22.15.12 DMA Transfer

In EW mode 1, make sure that no DMA transfers will occur while the FMR00 bit in the FMR0 register is set to 0 (during the auto program or auto erase period).

22.15.13 Regarding Programming/Erase Times and Execution Time

As the number of programming/erase times increases, so does the execution time for software commands (Program, and Block Erase).

The software commands are aborted by hardware reset 1, brown-out detection reset (hardware reset 2), $\overline{\text{NMI}}$ interrupt, and watchdog timer interrupt. If a software command is aborted by such reset or interrupt, the affected block must be erased before reexecuting the aborted command.

22.15.14 Definition of Programming/Erase Times

"Number of programs and erasure" refers to the number of erasure per block.

If the number of program and erasure is n ($n=100, 1,000, 10,000$) each block can be erased n times.

For example, if a 2K byte block A is erased after writing 1 word data 1024 times, each to a different address, this is counted as one program and erasure. However, data cannot be written to the same address more than once without erasing the block. (Rewrite prohibited)

22.15.15 Flash Memory Version Electrical Characteristics 10,000 E/W cycle products (Normal: U7, U9; T-ver./V-ver.: U7)

When Block A or B E/W cycles exceed 100, set the FMR17 bit in the FMR1 register to 1 (1 wait) to select one wait state per block access for products U7 and U9. When the FMR17 bit is set to 1, one wait state is inserted per access to Block A or B - regardless of the value of the PM17 bit. Wait state insertion during access to all other blocks, as well as to internal RAM, is controlled by the PM17 bit - regardless of the setting of the FMR17 bit.

To use the limited number of erasure efficiently, write to unused address within the block instead of rewrite. Erase block only after all possible address are used. For example, an 8-word program can be written 128 times before erase becomes necessary.

Maintaining an equal number of erasure between Block A and B will also improve efficiency.

We recommend keeping track of the number of times erasure is used.

22.15.16 Boot Mode

An undefined value is sometimes output in the I/O port until the internal power supply becomes stable when "H" is applied to the CNVss pin and "L" is applied to the $\overline{\text{RESET}}$ pin.

When setting the CNVss pin to "H", the following procedure is required:

- (1) Apply an "L" signal to the $\overline{\text{RESET}}$ pin and the CNVss pin.
- (2) Bring Vcc to more than 2.7V, and wait at least 2 msec. (Internal power supply stable waiting time)
- (3) Apply an "H" signal to the CNVss pin.
- (4) Apply an "H" signal to the $\overline{\text{RESET}}$ pin.

When the CNVss pin is "H" and RESET pin is "L", P67 pin is connected to the pull-up resistor.

22.16 Noise

Connect a bypass capacitor (approximately $0.1\mu\text{F}$) across the VCC and VSS pins using the shortest and thicker possible wiring. **Figure 22.8** shows the bypass capacitor connection.

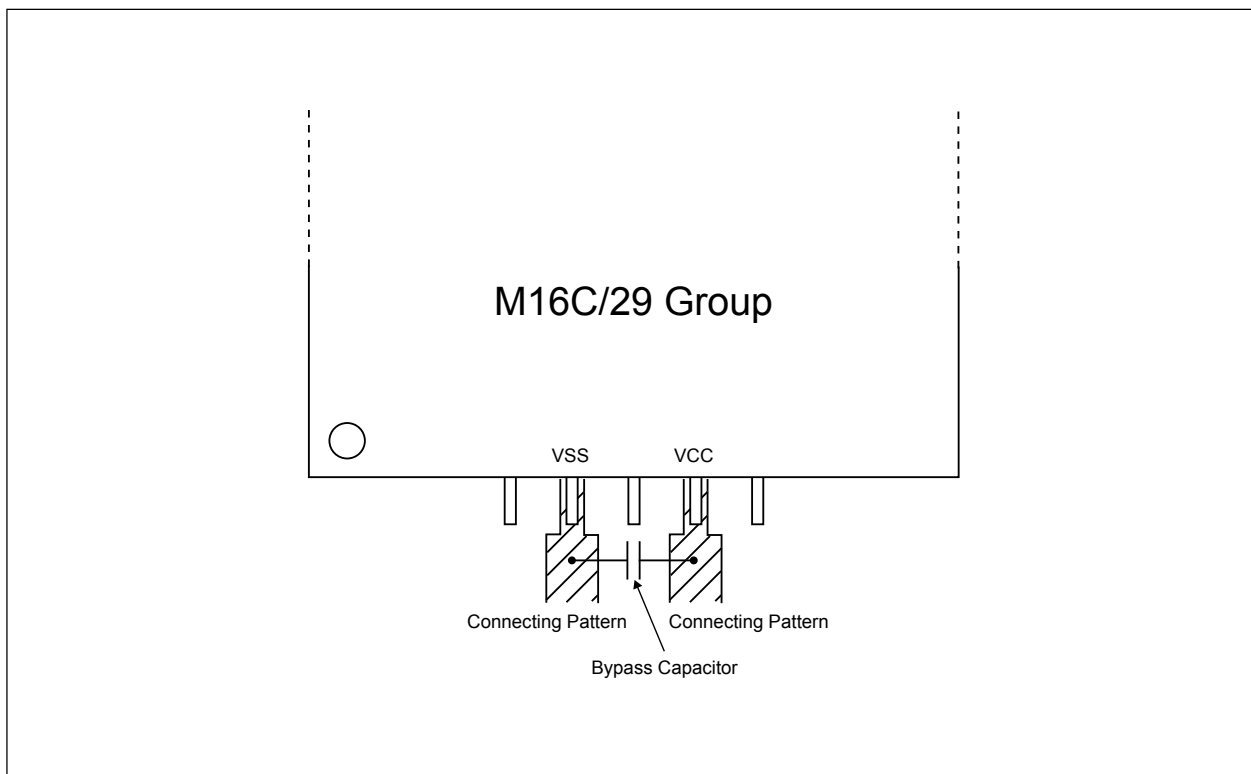


Figure 22.8 Bypass Capacitor Connection

22.17 Instruction for a Device Use

When handling a device, extra attention is necessary to prevent it from crashing during the electrostatic discharge period.

Appendix 1. Package Dimensions

| | | | |
|---|-------------------------------|---|--------------------|
| JEITA Package Code P-LQFP64-10x10-0.50 | RENEASAS Code PLQP0064KB-A | Previous Code 64P6Q-A / FP-64K / FP-64KV | MASS[Typ.] 0.3g |
|---|-------------------------------|---|--------------------|

NOTE)
1. DIMENSIONS **1* AND **2* DO NOT INCLUDE MOLD FLASH.
2. DIMENSION **3* DOES NOT INCLUDE TRIM OFFSET.

| Reference Symbol | Dimension in Millimeters | | |
|------------------|--------------------------|-------|------|
| | Min | Nom | Max |
| D | 9.9 | 10.0 | 10.1 |
| E | 9.9 | 10.0 | 10.1 |
| A ₂ | — | 1.4 | — |
| H _D | 11.8 | 12.0 | 12.2 |
| H _E | 11.8 | 12.0 | 12.2 |
| A | — | — | 1.7 |
| A ₁ | 0.05 | 0.1 | 0.15 |
| b _p | 0.15 | 0.20 | 0.25 |
| b ₁ | — | 0.18 | — |
| c | 0.09 | 0.145 | 0.20 |
| c ₁ | — | 0.125 | — |
| θ | 0° | — | 8° |
| ⌀ | — | 0.5 | — |
| x | — | — | 0.08 |
| y | — | — | 0.08 |
| Z _D | — | 1.25 | — |
| Z _E | — | 1.25 | — |
| L | 0.35 | 0.5 | 0.65 |
| L ₁ | — | 1.0 | — |

| | | | |
|---|-------------------------------|--------------------------|--------------------|
| JEITA Package Code P-LQFP80-12x12-0.50 | RENEASAS Code PLQP0080KB-A | Previous Code 80P6Q-A | MASS[Typ.] 0.5g |
|---|-------------------------------|--------------------------|--------------------|

NOTE)
1. DIMENSIONS **1* AND **2* DO NOT INCLUDE MOLD FLASH.
2. DIMENSION **3* DOES NOT INCLUDE TRIM OFFSET.

| Reference Symbol | Dimension in Millimeters | | |
|------------------|--------------------------|-------|------|
| | Min | Nom | Max |
| D | 11.9 | 12.0 | 12.1 |
| E | 11.9 | 12.0 | 12.1 |
| A ₂ | — | 1.4 | — |
| H _D | 13.8 | 14.0 | 14.2 |
| H _E | 13.8 | 14.0 | 14.2 |
| A | — | — | 1.7 |
| A ₁ | 0 | 0.1 | 0.2 |
| b _p | 0.15 | 0.20 | 0.25 |
| b ₁ | — | 0.18 | — |
| c | 0.09 | 0.145 | 0.20 |
| c ₁ | — | 0.125 | — |
| θ | 0° | — | 10° |
| ⌀ | — | 0.5 | — |
| x | — | — | 0.08 |
| y | — | — | 0.08 |
| Z _D | — | 1.25 | — |
| Z _E | — | 1.25 | — |
| L | 0.3 | 0.5 | 0.7 |
| L ₁ | — | 1.0 | — |

Appendix 2. Functional Comparison

Appendix 2.1 Difference between M16C/28 Group and M16C/29 Group (Normal-ver.) ⁽¹⁾

| Item | Description | M16C/28(Normal-ver.) | M16C/29(Normal-ver.) |
|---------------------------------|---|--|--|
| Clock Generation Circuit | Clock output function (function of b1 to b0 bits in the CM0 register) | Not available (reserved bit) | Available (clock output function select bit) |
| Protection | Function of the PRC0 bit | Enable to set the CM0, CM1, CM2, POOCR, PLC0 and PCLKR registers | Enable to set the CM0, CM1, CM2, POOCR, PLC0, PCLKR and CCLKR registers |
| Interrupt | The IFSR20 bit setting in the IFSR2A register | Set to 1 | Set to 0 |
| | The b1 bit in the IFSR2A register | Not available (reseved bit) | Interrupt cause switching bit (0: A/D conversion, 1:key input) |
| | The b2 bit in the IFSR2A register | Not available (reseved bit) | Interrupt cause switching bit (0: CAN0 wake-up/ error) |
| | Interrupt cause in the Interrupt number 13 | Key input interrupt | CAN0 error |
| | Interrupt cause in the Interrupt number 14 | Key input interrupt | A/D, key input interrupt |
| Three-phase Motor Control Timer | Three-phase port switching function (function of 0358 ₁₆) | Not available (reserved register) | Available (port function select register) |
| A/D | Number of A/D input pin | 24 channels (excluding AN ₃₀ to AN ₃₂) | 27 channels (including AN ₃₀ to AN ₃₂) |
| | Delayed trigger mode 0 | Not available in the 1st chip version and chip version A | Available |
| | Delayed trigger mode 1 | Not available in the 1st chip version and chip version A | Available |
| CAN module | compatible to 2.0B | Not available (all related registers are reserved registers) | Available (1 channel) |
| CRC Calculation | Available (compatible to CRC-CCITT and CRC-16 methods) | Not available (all related registers are reserved registers) | Available (1 circuit) |
| Pin Function | 2 pins (80-pin/85-pin package), 62 pins (64-pin package) | P9 ₃ /AN2 ₄ | P9 ₃ /AN2 ₄ /CTX |
| | 3 pins (80-pin/85-pin package), 64 pins (64-pin package) | P9 ₂ /TB2 _{IN} | P9 ₂ /AN3 ₂ /TB2 _{IN} /CRX |
| | 4 pins (80-pin/85-pin package), 1 pin (64-pin package) | P9 ₁ /TB1 _{IN} | P9 ₁ /AN3 ₁ /TB1 _{IN} |
| | 5 pins (80-pin/85-pin package), 2 pins (64-pin package) | P9 ₀ /TB0 _{IN} | P9 ₀ /AN3 ₀ /TB0 _{IN} /CLK _{OUT} |
| Flash Memory | P9 ₃ in standard serial I/O mode | I (other than 128 Kbyte version) I/O (128 Kbyte version) | CTX output |

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

NOTE:

1. Since the M16C/28 group uses the common emulator used in the M16C/29 group, all the functions are available for M16C/28. When evaluating M16C/28 group, do not access to the SFR which is not built-in the M16C/28 gorup. Refer to hardware manual for details and electrical characteristics.

Appendix 2.2 Difference between M16C/28 and M16C/29 Group (T-ver./V-ver.) (1)

| Item | Description | M16C/28(T-ver./V-ver.) | M16C/29(T-ver./V-ver.) |
|--------------|---|--|---|
| Protection | Function of the PRC0 bit | Enable to set the CM0, CM1, CM2, POOCR, PLC0 and PCLKR registers | Enable to set the CM0, CM1, CM2, POOCR, PLC0, PCLKR and CCLKR registers |
| Interrupt | The IFSR20 bit setting in the IFSR2A register | Set to 1 | Set to 0 |
| | The b1 bit in the IFSR2A register | Not available (reserved bit) | Interrupt cause switching bit (0: A/D conversion, 1:key input) |
| | The b2 bit in the IFSR2A register | Not available (reserved bit) | Interrupt cause switching bit (0: CAN0 wake-up/ error) |
| | Interrupt cause in the Interrupt number 13 | Key input interrupt | CAN0 error |
| | Interrupt cause in the Interrupt number 14 | Key input interrupt | A/D, key input interrupt |
| CAN module | compatible to 2.0B | Not available (all related registers are reserved registers) | Available (1 channel) |
| Pin Function | 2 pins (80-pin/85-pin package), 62 pins (64-pin package) | P93/AN24 | P93/AN24/CTX |
| | 3 pins (80-pin/85-pin package), 64 pins (64-pin package) | P92/TB2IN | P92/AN32/TB2IN/CRX |

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

NOTE:

1. Since the M16C/28 group uses the common emulator used in the M16C/29 group, all the functions are available for M16C/28. When evaluating M16C/28 group, do not access to the SFR which is not built-in the M16C/28 group. Refer to hardware manual for details and electrical characteristics.

Register Index

A

AD0 to AD7 226
 ADCON0 to ADCON2 224
 ADIC 76
 ADSTAT0 226
 ADTRGCON 225
 AIER 88

B

BCNIC 76
 BTIC 76

C

C01ERRIC 76
 C01WKIC 76
 C0AFS 299
 C0CONR 297
 C0CTLR 293
 C0ICR 296
 C0IDR 296
 C0MCTLj 292
 C0RECIC 76
 C0RECR 298
 C0SSTR 295
 C0STR 294
 C0TECR 298
 C0TRMIC 76
 C0TSR 299
 CCLKR 53
 CM0 49
 CM1 50
 CM2 51
 CPSRF 105, 118
 CRCD 314
 CRCIN 314
 CRCMR 314
 CRCSAR 314

D

D4INT 40
 DAR0 95
 DAR1 95
 DM0CON 94

DM0IC 76
 DM0SL 93
 DM1CON 94
 DM1IC 76
 DM1SL 94
 DTT 129

F

FMR0 341
 FMR1 341
 FMR4 342

G

G1BCR0 142
 G1BCR1 143
 G1BT 142
 G1BTRR 144
 G1DV 143
 G1FE 148
 G1FS 148
 G1IE0 150
 G1IE1 150
 G1IR 149
 G1PO0 to G1PO7 147
 G1POCR0 to G1POCR7 146
 G1TM0 to G1TM7 146
 G1TMCR0 to G1TMCR7 145
 G1TPR6 to G1TPR7 145

I

ICOC0IC 76
 ICOC1IC 76
 ICTB2 129, 130
 IDB0 129
 IDB1 129
 IFSR 77, 85
 IFSR2A 77
 IICIC 76
 INT0IC to INT2IC 76
 INT3IC 76
 INT4IC 76
 INT5IC 76
 INVC0 127
 INVC1 128

K

KUPIC 76

N

NDDR 327

O

ONSF 105

P

P0 to P3 324

P17DDR 327

P6 to P10 324

PACR 177, 326

PCLKR 52

PCR 326

PD0 to PD3 323

PD6 to PD10 323

PDRF 137

PFCR 139

PLC0 53

PM0 44

PM1 44

PM2 45, 52

PRCR 69

PUR0 to PUR2 325

R

RMAD0 88

RMAD1 88

ROCR 50

ROMCP 336

S

S00 258

S0D0 257

S0RIC to S2RIC 76

S0TIC to S2TIC 76

S10 260

S1D0 259

S20 258

S2D0 263

S31C 76

S3BRG 218

S3C 218

S3D0 261

S3TRR 218

S4BRG 218

S4C 218

S4D0 262

S4IC 76

S4TRR 218

SAR0 95

SAR1 95

SCLDAIC 76

T

TA0 to TA4 104

TA0IC to TA4IC 76

TA0MR to TA4MR 103

TA11 130

TA1MR 133

TA2 130

TA21 130

TA2MR 133

TA4 130

TA41 130

TA4MR 133

TABSR 104, 118, 132

TB0 to TB2 118

TB0IC to TB2IC 76

TB0MR to TB2MR 117

TB2 132

TB2MR 133

TB2SC 131, 227

TCR0 95

TCR1 95

TPRC 139

TRGSR 105, 132

U

U0BRG to U2BRG 174

U0C0 to U2C0 176

U0C1 to U2C1 177

U0MR to U2MR 175

U0RB to U2RB 174

U0TB to U2TB 174

U2SMR 178

U2SMR2 178

U2SMR3 179

U2SMR4 179

UCON 176

UDF 104

VVCR1 **39**VCR2 **39****W**WDC **90**WDTS **90**

REVISION HISTORY

M16C/29 Hardware Manual

| Rev. | Date | Description | |
|------|--|-------------|--|
| | | Page | Summary |
| 0.70 | Mar/ 29/Y04 | 1 | “1. Overview” and “1.1. Application” are partly revised. |
| | | 2, 3 | Table 1.2.1 and 1.2.2 are partly revised. |
| | | 8, 9 | Figure 1.5.1 and 1.5.2 are partly revised. |
| | | 10 | Table 1.6.1 is revised. |
| | | 22 | Figure 4.8 is partly revised. |
| | | 28 | Section “5.5 Voltage Detection Circuit” and Figure 5.5.2 are partly revised. |
| | | 30 | Figure 5.5.3 is partly revised. |
| | | 31 | Figure 5.5.4 is partly revised. |
| | | 32 | Section “5.5.1 Voltage Detection Interrupt” and “5.5.1.1.1 Limitations of Stop Mode” are partly revised. |
| | | 36 | Figure 7.1 is partly revised. |
| | | 37 | Figure 7.2 is partly revised. |
| | | 38 | Figure 7.3 is partly revised. |
| | | 39 | Figure 7.5 is partly revised. |
| | | 40 | Figure 7.6 is partly revised. |
| | | 41 | “CCLKR register” of Figure 7.7 is partly revised. |
| | | 42 | Section “7.1 Main clock” is partly revised. |
| | | 45 | Figure 7.4.1 is partly revised. |
| | | 46 | Section “7.5 CPU Clock and Peripheral Function Clock” and “7.5.2 Peripheral Function Clock” are partly revised. |
| | | 54 | Section “7.7 System Clock Protective Function” and “7.8 Oscillation Stop and Re-oscillation Detect Function” are partly revised. |
| | | 57 | Figure 8.1 is partly revised. |
| | | 64 | Figure 9.3.1 is partly revised. |
| | | 65 | IFSR2A register in Figure 9.3.2 is partly revised. |
| | | 66 | Section “9.3.2 IR Bit” is partly revised. |
| 67 | Section “9.4 Interrupt Sequence” is partly revised. | | |
| 68 | Section “9.4.1 Interrupt Response Time” and Figure 9.4.1.1 are partly revised. | | |
| 73 | Section “9.6 INT Interrupt” is partly revised. | | |
| 74 | Section “9.9 CAN0 Wake-up Interrupt” is partly revised. | | |
| 94 | “Divide ratio” of Table 12.1.1.1 is partly revised. | | |
| 102 | “8-bit PWM” of Table 12.1.4.1 is partly revised. | | |
| 106 | “Timer Bi register” in Figure 12.2.3 is partly revised. | | |
| 111 | Section “12.2.4 A-D Trigger mode” and Table 12.2.4.1 are partly revised. | | |
| 112 | Figure 12.2.4.2 is partly revised. | | |
| 115 | Figure 12.3.2 is partly revised. | | |
| 117 | “Timer B2 interrupt occurrences frequency set counter” in Figure 12.3.4 is partly revised. | | |
| 119 | Figure 12.3.6 is partly revised. | | |

REVISION HISTORY

M16C/29 Hardware Manual

| Rev. | Date | Description | |
|------|------|-------------|--|
| | | Page | Summary |
| | | 122 | “Figure 12.3.9 PFCR register and TPRC register” is deleted. |
| | | 125 | Figure 12.3.1.2.1 and the section 12.3.1.2.4 are partly revised. |
| | | 126 | Section “Three-phase/Port Output Switch Function” and “Figure 12.3.2.1 PFCR register and TPRC register” are added. |
| | | 166 | “UART 2 special mode register 2” in Figure 14.1.8 is partly revised. |
| | | 167 | “UART 2 special mode register 3” in Figure 14.1.9 is partly revised. |
| | | 210 | Note 1 in Table 15.1.1.1 is deleted. |
| | | 213 | Figure 15.4 is partly revised. |
| | | 214 | Figure 15.5 is partly revised. |
| | | 219 | Section “15.1.3 Single Sweep mode” is partly revised. |
| | | 221 | Section “15.1.4 Repeat Sweep mode 0” is partly revised. |
| | | 223 | Section “15.1.5 Repeat Sweep mode 1” is partly revised. |
| | | 225 | Section “15.1.6 Simultaneous Sample Sweep Mode”, Table 15.1.6.1, and Figure 15.1.6.1 are partly revised. |
| | | 228 | Section “15.1.7 Delayed Trigger Mode 0” and Table 15.1.7.1 are partly revised. |
| | | 229 | Figure 15.1.7.1 is partly revised. |
| | | 230, 231 | Figure 15.1.7.2 and 15.1.7.3 are partly revised. |
| | | 232 | Figure 15.1.7.3 is deleted. |
| | | 235 | Section “15.1.8 Delayed Trigger Mode 1” and Table 15.1.8.1 are partly revised. |
| | | 241 | Figure 15.5.1 is partly revised. |
| | | 276 to 300 | Chapter “17. CAN Module” is revised. |
| | | 301 | Chapter “18. CRC Calculation Circuit” is partly revised. |
| | | 303 | Figure 18.3 is partly revised. |
| | | 304 | Chapter “19. Programmable I/O ports” is partly revised. |
| | | 305 | Section “19.5 Pin Assignment Control Register” is partly revised. |
| | | 313 | “Pull-up control register” in Figure 19.3.1 is partly revised. |
| | | 320 | Table 20.4 and 20.5 and Note 6 and 10 are partly revised. |
| | | 321 | Note 3 in Table 20.6 is added. |
| | | 342 | Table 20.43 and 20.44 and Note 10 are partly revised. |
| | | 343 | Note 3 in Table 20.45 is added. |
| | | 360 to 372 | Section “20.3 V version” is deleted. |
| | | 373 | Table 21.1 is partly revised. |
| | | 282 | Section “•FMR01 Bit”, “•FMR02 Bit” and “•FMSTP Bit” are partly revised. |
| | | 383 | Section “•FMR16 Bit”, “•FMR17 Bit” and “FMR41 Bit” are partly revised. |
| | | 384 | Figure 21.5.1 is revised. |
| | | 387 | Figure 21.5.1.3 is partly revised. |
| | | 392 | Section “21.4.2 EW1 Mode” is partly revised. |
| | | | Section “21.6.4 How to Access” is partly revised. |
| | | | Section “21.7.5. Block Erase” is partly revised. |

REVISION HISTORY

M16C/29 Hardware Manual

| Rev. | Date | Description | |
|------|--------------|-------------|---|
| | | Page | Summary |
| | | 399 | Figure 21.9.1 is partly revised. |
| | | 400 | Figure 21.9.2 is partly revised. |
| | | 403 | Section "21.10.1 ROM Code Protect Function" is partly revised. |
| | | 404 | Section "21.11.1 ROM Code Protect Function" is partly revised. |
| | | | Table 21.11.1 is revised. |
| | | 405 | Figure 21.11.1 is revised. |
| | | 406 | Figure 21.11.2 is revised. |
| | | 407 | Figure 21.11.3 is revised. |
| 0.71 | April/15/Y04 | B-1 to B-3 | "Quick Reference to Page Classified by Address" are revised. |
| | | B-4, B-5 | "Quick Reference to Page Classified by Address" are partly revised. |
| | | 2,3 | Table 1.2.1 and Table 1.2.2 is partly revised. |
| | | 6,7 | Table 1.4.1 to 1.4.3 is partly revised. |
| | | 14 | Not e2 in Figure 3.1 is added. |
| | | 15 to 20 | Figure 4.1 to Figure 4.6 are revised. |
| | | 21, 22, 25 | Figure 4.7, Figure 4.8 and Figure 4.11 are partly revised. |
| | | 29 | Section "5.5 Voltage Detection Circuit" is partly revised. |
| | | 33 | Figure 5.5.1.1.2.1 is partly revised. |
| | | 34 | Figure 6.2 is partly revised. |
| | | 40 | The PM2 register in Figure 7.6 is partly revised. |
| | | 64 | Figure 9.3.1 is partly revised. |
| | | 65 | The IFSR2A register in Figure 9.3.2 is partly revised. |
| | | 112 | Figure 12.2.4.2 is partly revised. |
| | | 119 | Figure 12.3.6 is partly revised. |
| | | 126 | Section "12.3.2 Three-phase/Port Output Switch Function" is revised. Figure "12.3.2.1. Usage Example of Three-phse/Port output switch function" is added. |
| | | 130 | Figure 13.2 is partly revised. |
| | | 134 | Figure 13.6 is partly revised. |
| | | 137 | Figure 13.10 is partly revised. |
| | | 162 | "UARTi receive buffer register" in Figure 14.1.4 is partly revised. |
| | | 170 | Table 14.1.1.2 is partly revised. |
| | | 177 | Table 14.1.2.2 is partly revised. |
| | | 184 | Figure 14.1.3.1 is partly revised. |
| | | 214 | Figure 15.5 is partly revised. |
| | | 230, 231 | Figure 15.1.7.2 and Figure 15.1.7.3 are partly revised. |
| | | 233 | Figure 15.1.7.5 is partly revised. |
| | | 235 | Section "15.1.8 Delayed Trigger Mode 1" is partly revised. |
| | | 236, 237 | Figure 15.1.8.2 and Figure 15.1.8.3 are partly revised. |
| | | 240 | Section "15.3 Sample and Hold" and Figure 15.5.1 are partly revised. |
| | | 244 | Figure 16.2 is partly revised. |
| | | 321 | Table 20.4 and Table 20.5 are partly revised. |
| | | 342 | Table 20.43 and Table 20.44 are partly revised. |

REVISION HISTORY

M16C/29 Hardware Manual

| Rev. | Date | Description | |
|------|------------|-------------|--|
| | | Page | Summary |
| | | 360 | Table 21.1 is partly revised. |
| | | 368 | Section "21.4.2 EW1 Mode" is partly revised. |
| 0.80 | Sep/03/Y04 | 2,3 | Table 1.2.1 and Table 1.2.2 are partly revised. |
| | | 6,7 | Table 1.4.1 to Table 1.4.3 are partly revised. |
| | | 7 | Figure 1.4.1 is partly revised. |
| | | 8,9 | Figure 1.5.1 and Figure 1.5.2 are partly revised. |
| | | 21 | Figure 4.7 is partly revised. |
| | | 24 | Figure 4.10 is partly revised. |
| | | 26 | Section "5.1.2 Hardware Reset 2" is partly revised. |
| | | 29 to 34 | Section "5.5 Voltage Detection Circuit" is revised. |
| | | 80 | Section "10.2 Cold start / Warm start" is added. |
| | | 322 | Table 20.2 is partly revised. |
| | | 323 | Table 20.3 is partly revised. |
| | | 325 | Table 20.6 and Table 20.7 are partly revised. |
| | | 327 | Table 20.9 is partly revised. |
| | | 331 | Title of Table 20.23 is partly revised. |
| | | 335 | Table 20.25 is partly revised. |
| | | 339 | Title of Table 20.39 is partly revised. |
| | | 343 | Table 20.41 is partly revised. |
| | | 344 | Table 20.42 is partly revised. |
| | | 346 | "Low Voltage Detection Circuit Electrical Characteristics" is deleted. Table 20.45 is partly revised. |
| | | 348 | Table 20.47 is partly revised. |
| | | 352 | Title of Table 20.61 is partly revised. |
| | | 356 | Table 20.63 is partly revised. |
| | | 360 | Title of Table 20.77 is partly revised. |
| | | 398 | 64P6Q-A package is revised. |
| 1.00 | Nov/01/Y04 | All pages | Words standardized (on-chip oscillator, A/D) |
| | | 2, 3 | Table 1.2.1 and Table 1.2.2 are partly revised. |
| | | 8, 9 | Table 1.4.4 to 1.4.6 and figure 1.4.2 to 1.4.6 are added. |
| | | 28 | "5.1.2 Hardware Reset 2" is partly revised. |
| | | 29 | "5.4 Oscillation Stop Detection Reset" is partly revised. |
| | | 38 | Table 7.1 is partly revised. |
| | | 41 | Note 6 in Figure 7.3 is partly revised. b7 to b4 bit in Figure 7.4 is revised. |
| | | 42 | Figure 7.5 is partly revised. |
| | | 43 | "PCLKR register" in Figure 7.6 is partly revised. |
| | | 50 | "7.6.1 Normal Operation Mode" is partly revised. |
| | | 51 | Note 1 in Table 7.6.1.1 is partly revised. |
| | | 57 | "7.8 Oscillation Stop and Re-oscillation Detect Function" is partly revised. |

REVISION HISTORY

M16C/29 Hardware Manual

| Rev. | Date | Description | |
|------|----------|-------------|--|
| | | Page | Summary |
| | | 66 | “9.3 Interrupt Control” is partly revised. |
| | | 76 | “9.6 \overline{INT} Interrupt” and “9.7 \overline{NMI} Interrupt” are partly revised. |
| | | 77 | “9.8 Key Input Interrupt” and “9.9 CAN0 Wake-up Interrupt” are partly revised. |
| | | 80 | “10. Watchdog Timer” is partly revised. |
| | | 80, 81 | “10.1 Count source protective mode” is partly revised. |
| | | 81 | Note 2 in Figure 10.2 is revised. |
| | | 118 | Figure 12.3.1 is partly revised. |
| | | 121 | “Three-phase output buffer register” in Figure 12.3.4 is partly revised. |
| | | 133 to 138 | Figure 13.1 to 13.6 are partly revised. |
| | | 141 | “Function enable register” in Figure 13.9 is partly revised. |
| | | 150 | Table 13.4.1 is partly revised. |
| | | 161 | “13.6 I/O Port Function Select” is partly revised. |
| | | 198 | Figure 14.1.4.1 is partly revised. |
| | | 209 | Figure 14.2.1 is partly revised. |
| | | 210 | Figure 14.2.2 is partly revised. |
| | | 214 | “Integral Nonlinearity Error” in Table 15.1 is partly revised. |
| | | 253,254 | Figure 16.6 and Figure 16.7 are partly revised. |
| | | 261 | “16.5.4 Bit 3: Arbitration lost detection flag” is partly revised. |
| | | 266 | “16.6.5 I2C system clock select bits” and Table 16.6 are partly revised. |
| | | 275 | “9)” in “16.13.2 Example of Slave Receive” is revised. |
| | | 296 | “17.3 Configuration of the CAN Module System Clock” is partly revised. |
| | | 306 | “18.1 CRC snoop” is partly revised. |
| | | 337 | Table 20.25 is partly revised. |
| | | 368 | “21.1 Flash Memory Performance” is partly revised. |
| | | 367,368 | “21.2 Memory Map” is partly revised. |
| | | 372 | “21.4 CPU Rewrite Mode” is partly revised. |
| | | 373 | “21.4.1 EW0 Mode” and “21.4.2 EW1 Mode” are partly revised. |
| | | 374 | “FMR01 Bit” is partly revised. |
| | | 375 | “FMR17 Bit” is partly revised. |
| | | 383 | “21.7.4 Program Command (40 ₁₆)” is partly revised. |
| | | 390 | Table 21.9.1 and Note 2 are partly revised. |
| | | 391,392 | Figure 21.9.1 and Figure 21.9.2 are partly revised. |
| | | 393,394 | Figure 21.9.2.1 and Figure 21.9.2.2 are partly revised. |
| | | 396 | Table 21.11.1 and Note 1 are partly revised. |
| | | 397,398 | Figure 21.11.1 and Figure 21.11.2 are partly revised. |
| | | 399 | Figure 21.11.3 is partly revised. |
| 1.10 | 10/10/06 | All Pages | Package code changed: 80P6Q-A to PLQP0080KB-A, 64P6Q-A to PLQP0064KB-A Words standardized: Low voltage detection, CPU clock, MCU, SDA ₂ , SCL ₂ |

| Rev. | Date | Description | |
|------|------|-------------|---|
| | | Page | Summary |
| | | 2 | Overview • Table 1.1 and 1.2 Performance Outline Voltage detection circuit are modified, note 3 is modified |
| | | 4 - 5 | • Figure 1.1 and 1.2 Block Diagrams are updated |
| | | 6 - 7 | • Table 1.3 to 1.5 Product Lists are updated |
| | | 8 | • Figure 1.3 Product Numbering System is modified |
| | | 9 | • Tables 1.6 to 1.8 Product Code B3, B7, D3, D5, D7, D9 are deleted |
| | | 13 - 17 | • Tables 1.9 to 1.11 Product Code Mask ROM versions are newly added |
| | | 18 | • Table 1.9 and 1.10 Pin Characteristics for 80-, and 64-pin Packages are added • Table 1.11 Pin Description Tables are modified |
| | | 23 | Memory • Figure 3.1 Memory Map 48Kbyte memory size is deleted |
| | | 24 - 34 | Special Function Register • Table 4.1 to 4.11 SFR Information values after reset |
| | | 24 | • Table 4.1 SFR Information(1) Note 3 is deleted |
| | | 35 | Reset • 5.1.2 Hardware Reset 2 Note is modified, description is modified |
| | | 38 | • 5.5 Voltage Dection Circuit modified • Figure 5.4 Voltage Detection Circuit Block modified, WDC5 bit circuit deleted |
| | | 44 | Processor Mode • Figure 6.1 PM1 Register Note 2 information partially added |
| | | 45 | • Figure 6.2 PM2 Register added |
| | | 46 | • Figure 6.3 Bus Block Diagram and Table 6.1 Accessible Area and Bus Cycle added |
| | | 47 | Clock Generation Circuit • Table 7.1 Clock Generation Circuit Specifications Oscillation stop, restart function modified |
| | | 48 | • Figure 7.1 Clock Generation Circuit Upper portion of figure is modified |
| | | 50 | • Figure 7.4 ROCR Register Bit conents are modified |
| | | 52 | • Figure 7.6 PCLKR Register and PM2 Register Note 2 is modified |
| | | 54 | • Figure 7.8 Examples of Main Clock Connection Circuit is modified |
| | | 55 | • Figure 7.9 Examples of Sub Clock Connection Circuit is modified • 7.5.2 Peripheral Function Clock (f1, f2, f8, f32, f1SIO, f2SIO, f8SIO, f32SIO, fAD, fc32, fCAN0) revised |
| | | 59 | • 7.6.1 Normal Operation Mode Information is modified |
| | | 60 | • Table 7.4 Setting Clock Related Bit and Modes Multi-master I2C bus interrupt and Timer S interrupt added |
| | | 61 | • Table 7.5 Pin Status in Wait Mode newly added • Table 7.6 Interrupts to Exit Wait Mode modified |

REVISION HISTORY

M16C/29 Hardware Manual

| Rev. | Date | Description | |
|------|------|-------------|--|
| | | Page | Summary |
| | | 63 | • Figure 7.11 State Transition to Stop Mode and Wait Mode modified, Note 7 is added |
| | | 64 | • Figure 7.12 State Transition in Normal Mode modified, note 5 deleted, note 6 and 7 are simplified |
| | | 65 | • Table 7.7 Allowed Transition and Setting note 2 partially modified, table contents are partially modified |
| | | 68 | • Figure 7.13 Procedure to Switch Clock Source From On-chip Oscillator Clock to Main Clock is modified |
| | | | Interrupt |
| | | 70 | • Note is newly added |
| | | 73 | • Table 9.1 Fixed Vector Tables Note 2 is added |
| | | | Watchdog Timer |
| | | 89 | • Additional information of the WDTS register is inserted |
| | | 90 | • Figure 10.1 Watchdog Timer Block Diagram modified |
| | | - | • Figure 10.2 WDC Register and WDTS Register All notes are deleted |
| | | - | • 10.2 Cold Start/Warm Start Section is deleted |
| | | | DMAC |
| | | 96 | • Note is added |
| | | | Timer |
| | | 105 | • Figure 12.6 TRGSR Register Note 2 added |
| | | 117 | • 12.2 Timer B Description of A/D trigger mode modified |
| | | | • Figure 12.15 Timer B Block Diagram "A/D trigger mode" is added |
| | | 123 | • 12.2.4 A/D Trigger Mode Description modified |
| | | 129 | • Figure 12.28 IDB0 Register, IDB1 Register, DTT Register, and ICTB2 Register Information of bit 7 and 6 modified |
| | | 131 | • Figure 12.30 TB2SC Register Note 4 added, contents modified |
| | | 133 | • Figure 12.32 TA1MR Register, TA2MR Register, TA4MR Register MR0 bit is modified |
| | | 134 | • Figure 12.33 Triangular Wave Modulation Operation Description modified |
| | | 135 | • Figure 12.34 Sawtooth Wave Modulation Operation Description modified |
| | | 139 | • Figure 12.38 TPRC Register Bit map is modified |
| | | | Timer S |
| | | 142 | • Figure 13.2 G1BT and G1BCR0 Registers Function of G1BT register modified, note 3 is added, function of bits 5 to 3 modified, description partially modified |
| | | 143 | • Figure 13.3 G1BCR1 Register Note 1 is partially added |
| | | 146 | • Figure 13.6 G1TM0 to G1TM7 Registers Note 3 and 4 are added |
| | | 151-166 | • Table 13.2, 13.5, 13.8, 13.9 and 13.10 Output wave form and Selectable function are modified |
| | | 155 | • Figure 13.15 Base Timer Reset Operation by Base Timer Reset Register Base timer overflow request line is added, base timer interrupt line is modified, |

| Rev. | Date | Description | |
|------|------|-------------|--|
| | | Page | Summary |
| | | | note 1 is added |
| | | 160 | • Figure 13.21 Prescaler Function and Gate Function Note 1 modified |
| | | 166 | • Table 13.10 SR Waveform Output Mode Specifications Specification modified |
| | | 167 | • Figure 13.24 Set/Reset Waveform Output Mode Description for (1) Free-running operation modified, register names modified |
| | | 168 | • Table 13.11 Pin Setting for Time Measurement and Waveform Generating Functions Description of port direction modified |
| | | | Serial I/O |
| | | 170 | • Note is modified |
| | | 171 | • Figure 14.1 Block Diagram of UARTi (i = 0 to 2) PLL clock is added to the upper portion of diagram |
| | | 174 | • Figure 14.4 U0TB to U2TB, U0RB to U2RB, U0BRG to U2BRG Registers Note 2 is modified, note 3 is newly added |
| | | 175 | • Figure 14.5 U0MR Register, U1MR Register Bit map is modified |
| | | 176 | • Figure 14.6 U0C0 Register Note 3 modified, Note 4 to 7 are added |
| | | | • Figure 14.6 U2C0 Register Note 2 is added |
| | | 177 | • Figure 14.7 PACR Register added |
| | | 180 | • Table 14.1 Clock Synchronous Serial I/O Mode Specifications Select function modified, note 2 modified |
| | | 182 | • Table 14.3 Pin Functions Note 1 added |
| | | | • Table 14.4 P64 Pin Functions Note 1 added |
| | | 183 | • Figure 14.10 Typical transmit/receive timings in clock synchronous serial I/O mode Example of receive timing: figure modified |
| | | 184 | • 14.1.1.1 Counter Measure for Communication Error Occurs newly added |
| | | 185 | • 14.1.1.2 CLK Polarity Select Function Newly added |
| | | 186 | • Figure 14.14 Transfer Clock Output From Multiple Pins Note 2 added |
| | | 187 | • 14.1.1.7 CTS/RTS separate function (UART0) modified |
| | | | • Figure 14.15 CTS/RTS Separate Function Usage Note 1 added |
| | | 188 | • Table 14.5 UART Mode Specifications Select function modified, note 1 modified |
| | | 190 | • Table 14.7 I/O Pin Functions in UART Mode Note 1 added |
| | | | • Table 14.8 P64 Pin Functions in UART Mode Note 2 added |
| | | 192 | • Figure 14.17 Receive Operation \overline{RTSi} line is modified |
| | | | • 14.1.2.1 Bit Rates newly added |
| | | | • Table 14.9 Example of Bit Rates and Settings newly added |
| | | 193 | • 14.1.2.2 Counter Measure for Communication Error newly added |
| | | 195 | • 14.1.2.6 CTS/RTS Separate Function (UART0) P70 pin is added |
| | | | • Figure 14.21 CTS/RTS Separate Function Note 1 added |
| | | 196 | • Table 14.10 I²C mode Specifications Note 2 modified |

| Rev. | Date | Description | |
|------|------|-------------|---|
| | | Page | Summary |
| | | 214 | • Figure 14.31 Transmit and Received Timing in SIM Mode partially modified |
| | | 217 | • 14.2 SI/O3 and SI/O4 Note added |
| | | 218 | • Figure 14.36 S3C and S4C Registers Note 5 is added |
| | | | • Figure 14.36 S3BRG and S4BRG Registers Note 3 is added |
| | | 220 | • Figure 14.38 Polarity of Transfer Clock figure modified |
| | | 221 | • 14.2.3 Functions for Setting an SOUTi Initial Value Description modified |
| | | | A/D Converter |
| | | 222 | • Note added |
| | | | • Table 15.1 A/D Converter Performance Integral Nonlinearity Error modified |
| | | 224 | • Figure 15.2 ADCON2 Registers b2-b1 function modified |
| | | 227 | • Figure 15.5 TB2SC Register Reserved bit map modified |
| | | 229 | • Figure 15.7 ADCON0 to ADCON2 Registers in One-shot Mode ADCON2 register: b2-b1 function modified |
| | | 231 | • Figure 15.9 ADCON0 to ADCON2 Registers in Repeat Mode ADCON2 register: b2-b1 function modified |
| | | 233 | • Figure 15.11 ADCON0 to ADCON2 Registers in Single Sweep Mode ADCON2 register: b2-b1 function modified |
| | | 235 | • Figure 15.13 ADCON0 to ADCON2 Registers in Repeat Sweep Mode 0 ADCON2 register: b2-b1 function modified |
| | | 237 | • Figure 15.15 ADCON0 to ADCON2 Registers in Repeat Sweep Mode 1 ADCON2 register: b2-b1 function modified |
| | | 239 | • Figure 15.17 ADCON0 to ADCON2 Registers in Simultaneous Sample Sweep Mode ADCON2 register: b2-b1 function modified |
| | | 241 | • Table 15.10 Delayed Trigger Mode 1 Specifications Note 1 is modified |
| | | 245 | • Figure 15.22 ADCON0 to ADCON2 Registers in Delayed Trigger Mode 0 ADCON2 register: b2-b1 function modified |
| | | 251 | • Figure 15.27 ADCON0 to ADCON2 Registers in Delayed Trigger Mode 1 ADCON2 register: b2-b1 function modified |
| | | 254 | • 15.5 Analog Input Pin and External Sensor Equivalent Circuit Example is deleted |
| | | | • 15.5 Output Impedance of Sensor under A/D Conversion is added |
| | | | • Figure 15.29 Analog Input Pin and External Sensor Equivalent Circuit Note 1 is added |
| | | - | • Precaution of Using A/D Converter deleted |
| | | | Multi-master I²C bus INTERFACE |
| | | 255 | • Table 16.1 Multi-master I²C bus Interface Functions I/O pin added |
| | | 256 | • Figure 16.1 Block Diagram of Multi-master I²C bus Interface Bit name and register name are modified |
| | | 257 | • Figure 16.2 S0D0 Register Bit map is modified |

| Rev. | Date | Description | |
|------|------|-------------|--|
| | | Page | Summary |
| | | 258 | • Figure 16.3 S00 Register Note is modified |
| | | 259 | • Figure 16.4 S1D0 Register Reserved bit map modified |
| | | 260 | • Figure 16.5 S10 Register b7-b6 modified |
| | | 262 | • Figure 16.7 S4D0 Register Bit reserved map is modified |
| | | 269 | • 16.5.1 Bit 0: Last Receive Bit (LRB) modified |
| | | | • 16.5.2 Bit 1: General call detection flag (ADR0) modified, note 1 modified |
| | | | • 16.5.3 Bit 2: Slave address comparison flag (AAS) modified |
| | | 270 | • 16.5.5 Bit 4: I²C Bus Interface Interrupt Request Bit (PIN) modified |
| | | | • 16.5.6 Bit 5: Bus Busy Flag (BB) Bit names are modified |
| | | 271 | • 16.5.8 Bit 7: Communication Mode Select bit (MST) modified |
| | | 276 | • 16.7.1 Bit0: Time-Out Detection Function Enable Bit (TOE) is modified |
| | | | • 16.7.5 Bit7: STOP Condition Detection Interrupt Request Bit (SCPIN) is modified |
| | | 279 | • 16.11 Stop Condition Generation Method Description added |
| | | 282 | • 16.13 Address Data Communication modified |
| | | | CAN Module |
| | | 292 | • Figure 17.6 C0MCTLj Register RspLock bit's name changed, note 2 revised |
| | | 293 | • Figure 17.7 C0CTRL Register Note 4 added, functions partially modified |
| | | 294 | • Figure 17.8 C0STR Register Note 1 deleted, functions partially modified |
| | | 298 | • Figure 17.13 C0RECR Register Note 2 deleted, note 1 partially modified |
| | | | • Figure 17.14 C0TECR Register Note 1 modified, note is relocated |
| | | 299 | • Figure 17.15 C0TSR Register Note 1 modified |
| | | 300 | • Figure 17.17 Transition Between Operational Modes Partially modified |
| | | 301 | • 17.2.3 CAN Sleep Mode Partially deleted |
| | | 304 | • Table 17.2 Example of Bit-Rate 24-MHz is deleted |
| | | 308 | • 17.8 Time Stamp Counter and Time Stamp Function Partially deleted |
| | | 310 | • Figure 17.25 Timing of Receive Data Frame Sequence IF to IFS |
| | | 311 | • Figure 17.26 Timing of Transmit Sequence IF to IFS |
| | | | CRC Calculation Circuit |
| | | 313 | • 18.1 CRC Snoop Description partially added |
| | | | Programmable I/O Ports |
| | | 316 | • Note added |
| | | | • 19.3 Pull-up Control Register 0 to 2 Description partially added |
| | | 317 | • 19.6 Digital Debounce Function Filter width formula modified |
| | | 318-321 | • Figure 19.1 I/O Ports (1) to Figure 19.4 I/O Ports (4) are modified |
| | | 326 | • Figure 19.10 PACR Register Note 1 is modified |
| | | 327 | • Figure 19.11 NDDR and P17DDR Register Functions modified, notes are added |
| | | 328 | • Figure 19.12 Functioning of Digital Debounce Filter modified, procedure note modified |

| Rev. | Date | Description | |
|------|------|-------------|--|
| | | Page | Summary |
| | | 329 | <ul style="list-style-type: none"> • Table 19.1 Unassigned Pin Handling in Single-chip Mode Note 5 added |
| | | 330 | <ul style="list-style-type: none"> • Flash Memory Version • 20.1 Flash Memory Performance Description partially deleted • Table 20.14 Flash Memory Version Specifications Note 3 added |
| | | 331 | <ul style="list-style-type: none"> • 20.1.1 Boot Mode added |
| | | 332 | <ul style="list-style-type: none"> • 20.2 Memory Map Description is modified |
| | | 335 | <ul style="list-style-type: none"> • 20.3.1 ROM Code Protect Function Description is modified |
| | | 336 | <ul style="list-style-type: none"> • Figure 20.4 ROMCP Address is modified |
| | | 337 | <ul style="list-style-type: none"> • Table 20.3 EW Mode 0 and EW Mode 1 Note 2 is modified |
| | | 339 | <ul style="list-style-type: none"> • 20.5.1 Flash Memory Control Register 0 FMR01 Bit and FMR02 Bit: description is modified |
| | | 340 | <ul style="list-style-type: none"> • 20.5.2 Flash Memory Control Register 1 (FMR1) FMR6 Bit is modified, FMR17 Bit is modified |
| | | 341 | <ul style="list-style-type: none"> • Figure 20.6 FMR0 and FMR1 Registers FMR0 register: note 3 modified, value after reset modified; FMR1 register: note 3 modified, reserved bit map modified |
| | | 342 | <ul style="list-style-type: none"> • Figure 20.7 FMR4 Register Note 2 is modified |
| | | 345 | <ul style="list-style-type: none"> • 20.6.3 Interrupts EW1 mode modified • 20.6.4 How to Access FMR16 bit is added |
| | | 346 | <ul style="list-style-type: none"> • 20.6.9 Stop Mode modified |
| | | 352 | <ul style="list-style-type: none"> • Table 20.7 Errors and FMR0 Register Status Register name modified |
| | | 355 | <ul style="list-style-type: none"> • Table 20.8 Pin Functions Pin settings are partially modified |
| | | | <ul style="list-style-type: none"> • Electrical Characteristics • V version is newly added |
| | | 366 | <ul style="list-style-type: none"> • Table 21.1 Absolute Maximum Ratings Parameters of Pd and Topr are modified |
| | | 367 | <ul style="list-style-type: none"> • Table 21.2 Recommended Operating Conditions VIH and VIL are modified |
| | | 368 | <ul style="list-style-type: none"> • Table 21.3 A/D Conversion Characteristics tsAMP deleted, note 4 added |
| | | 369 | <ul style="list-style-type: none"> • Table 21.4 Flash Memory Version Electrical Characteristics: Standard values of Program and Erase Endurance cycle modified, tps added • Table 21.5 Flash Memory Version Electrical Characteristics: tps added, data hold time added, note 1, 3, 8 modified, note 11 and 12 added |
| | | 370 | <ul style="list-style-type: none"> • Table 21.6 Low Voltage Detection Circuit Electrical Characteristics Note 4 added • Table 21.7 Power Supply Circuit from Timing CharacteristicsL Note 2 & 3 are deleted, figure modified |
| | | 372 | <ul style="list-style-type: none"> • Table 21.9 Electrical Characteristics(2) Note 5 is added |
| | | 380 | <ul style="list-style-type: none"> • Table 21.25 Electrical Characteristics(2) Note 5 is added |
| | | 387 | <ul style="list-style-type: none"> • Table 21.40 Absolute Maximum Ratings Parameters of Pd and Topr are modified |

| Rev. | Date | Description | |
|------|------|-------------|---|
| | | Page | Summary |
| | | 388 | • Table 21.41 Recommended Operating Conditions V_{IH} and V_{IL} are modified |
| | | 389 | • Table 21.42 A/D Conversion Characteristics t_{SAMP} deleted, note 4 added |
| | | 390 | • Table 21.43 Flash Memory Version Electrical Characteristics: Standard values of Program and Erase Endurance cycle modified, t_{ps} added |
| | | | • Table 21.44 Flash Memory Version Electrical Characteristics: t_{ps} added, data hold time added, note 1, 3, 8 modified, note 11 and 12 added |
| | | 391 | • Table 21.45 Power Supply Circuit from Timing Characteristics Note 2 & 3 are deleted, $t_d(S-R)$ and $t_d(E-A)$ are deleted, figure modified |
| | | 393 | • Table 21.47 Electrical Characteristics(2) Note 4 is added |
| | | 401 | • Table 21.63 Electrical Characteristics(2) Note 4 is added |
| | | | Precautions |
| | | 422 | • 22.2.1 PLL Frequency Synthesizer modified |
| | | 423 | • 22.2.2 Power Control Subsection sequence modified, 2., 3. and 4. information modified |
| | | 425 | • 22.4.3 NMI Interrupt 2. information partially deleted, 6. information added |
| | | 426 | • 22.4.5 INT Interrupt 3. information added |
| | | 427 | • 22.4.6 Rewrite the Interrupt Control Register Example 1 is modified |
| | | 431 | • 22.6.1.3 Timer A (One-shot Timer Mode) 6. information added |
| | | 434 | • 22.6.3 Three-phase Motor Control Timer Function newly added |
| | | 435 | • 22.7.1 Rewrite the G1 IR Register description modified |
| | | | • Figure 22.3 IC/OC Interrupt Flow Chart newly added |
| | | 436 | • 22.7.2 Rewrite the ICOCiC Register newly added |
| | | | • 22.7.3 Waveform Generating Function newly added |
| | | | • 22.7.4 IC/OC Base Timer Interrupt newly added |
| | | 438 | • 22.8.2.1 Special Mode (I²C bus Mode) added |
| | | | • 22.8.2.3 SI/O3, SI/O4 added |
| | | 441 | • 22.10 Multi-master I²C bus Interface added |
| | | 445 | • 22.12 Programmable I/O Ports 2. and 3. information modified |
| | | 447 | • 20.14 Mask ROM Version is added |
| | | 448 | • 22.15.1 Functions to Inhibit Rewriting Flash Memory Rewrite modified |
| | | | • 22.15.2 Stop Mode modified |
| | | | • 22.16.4 Low Power Dissipation Mode, On-chip Oscillator Low Power Dissipation Mode modified |
| | | | • 22.15.7 Operating Speed modified |
| | | 449 | • 22.15.9 Interrupts modified |
| | | | • 22.15.13 Regarding Programming/Erasure Times and Execution Time modified |
| | | 450 | • 22.15.14 Definition of Programming/Erasure Times added |
| | | | • 22.15.15 Flash Memory version Electrical Characteristics 10,000 E/W cycle |

REVISION HISTORY

M16C/29 Hardware Manual

| Rev. | Date | Description | |
|------|--------------|-------------|--|
| | | Page | Summary |
| | | 451 | products (U7, U9) added |
| | | 452 | • 22.15.16 Boot Mode added |
| | | | • 22.16 Noise added |
| | | | • 20.17 Instruction fo Device Use added |
| | | 453 | Appendix 1. Package Dimensions |
| | | | • Dimensions are updated |
| | | 454-455 | Appendix 2. Functional Comparison added |
| 1.11 | Dec.11,2006 | 54 | Clock Generation Circuit |
| | | | • Figure 7.8 Examples of Main Clock Connection Circuit Note 2 added |
| | | 88 | Interrupts |
| | | | • Table 9.6 PC Value Saved in Stack Area When Address Match Interrupt Request Is Acknowledged Table contents partially modified, note added |
| | | 198 | Serial I/O |
| | | | • Table 14.11 Registers to Be Used and Settings in I²C bus Mode Note Relocated |
| | | 297 | CAN Module |
| | | | • Figure 17.12 C0CONR Register Note 2 modified |
| | | 372 | Electrical Characteristics |
| | | | • Table 21.9 Electrical Characteristics (2) Mask ROM data added, value partially changed: 420 to 450 |
| | | 379 | • Table 21.24 Electrical Characteristics Note 1 modified |
| | | 380 | • Table 21.25 Electrical Characteristics Mask ROM data added, note 1 modified |
| | | 391 | • Table 21.45 Power Supply Circuit Timing Characteristics figure for td(P-R) and td(ROC) added, Mask ROM data added |
| | | 393 | • Table 21.47 Electrical Characteristics (2) Mask ROM data added, value partially changed |
| | | 400 | • Table 21.62 Electrical Characteristics Note 1 modified |
| | | 401 | • Table 21.63 Electrical Characteristics Mask ROM data added, note 1 modified |
| | | 412 | • Table 21.83 Power Supply Circuit Timing Characteristics figure for td(P-R) and td(ROC) added, Mask ROM data added |
| | | 414 | • Table 21.85 Electrical Characteristics (2) Mask ROM data added, value partially changed |
| | | - | Usage Notes |
| | | 421 | • Table numbers and Figure numbers are revised |
| | | 447 | • 22.1.3 Register Setting added |
| | | | • 22.14.1 Internal ROM Area Description added |
| 1.12 | Mar.30, 2007 | 1 | Overview |
| | | 2, 3 | • 1.1 Features modified |
| | | | • note on trademark modified |

REVISION HISTORY

M16C/29 Hardware Manual

| Rev. | Date | Description | |
|------|------|---|---|
| | | Page | Summary |
| | | 9 19, 20 | <ul style="list-style-type: none"> • Tables 1.6 to 1.8 Product Codes modified • Table 1.14 Pin Description pin description on I/O ports modified |
| | | 37 | <p>Reset</p> <ul style="list-style-type: none"> • Figure 5.2 Reset Sequence Vcc and ROC timings modified |
| | | 45 | <p>Processor Mode</p> <ul style="list-style-type: none"> • Figure 6.2 PM2 Register Description on notes 5 and 6 modified |
| | | 52 64 | <p>Clock Generation Circuit</p> <ul style="list-style-type: none"> • Figure 7.6 PM2 Register Description on notes 5 and 6 modified • Figure 7.12 State Transition in Normal Mode note 2 modified |
| | | 69 | <p>Protection</p> <ul style="list-style-type: none"> • Description on protection modified • Figure 8.1 PRCR Register note 1 modified |
| | | 88 | <p>Interrupts</p> <ul style="list-style-type: none"> • Table 9.6 PC Value Saved in Stack Area When Address Match Interrupt Request I Acknowledged instruction modified |
| | | 90 | <p>Watchdog Timer</p> <ul style="list-style-type: none"> • Figure 10.2 WDTS Register modified • 10.1 Count Source Protective Mode description modified |
| | | 129 | <p>Timer</p> <ul style="list-style-type: none"> • Figure 12.28 ICTB2 Register modified |
| | | 256 | <p>Multi-Master I²C bus Interface</p> <ul style="list-style-type: none"> • Figure 16.1 Block Diagram of Multi-Master I²C bus Interface modified |
| | | 335 340 341 343 | <p>Flash Memory Version</p> <ul style="list-style-type: none"> • 20.3.1 ROM Code Protect Function register name modified • 20.5.2 Flash Memory Control Register 1 description on FMR17 bit modified • Figure 20.6 FMR1 Register note 2 modified • Figure 20.9 Setting and Resetting of EW Mode 1 modified |
| | | 369 370 372 380 390 391 393 | <p>Electrical Characteristics</p> <ul style="list-style-type: none"> • Table 21.5 Flash Memory Version Electrical Characteristics note 10 modified • Timing figure for td(P-R) and td(ROC) modified • Table 21.9 Electrical Characteristics parameter and measurement condition modified, note 5 deleted • Table 21.25 Electrical Characteristics measurement condition modified, note 5 deleted • Tables 21.43 and 44 Flash Memory Version Electrical Characteristics note 10 modified • Timing figure for td(P-R) and td(ROC) modified • Table 21.47 Electrical Characteristics parameter and condition modified, note |

REVISION HISTORY

M16C/29 Hardware Manual

| Rev. | Date | Description | |
|------|------|-------------|--|
| | | Page | Summary |
| | | 401 | 4 deleted • Table 21.63 Elctrical Characteristics measurement condition modified, note 4 deleted |
| | | 411 | • Tables 21.81 and 21.82 Flash Memory Version Electrical Characteristics note 10 modified |
| | | 412 | • Timing figure for td(P-R) and td(ROC) modified |
| | | 414 | • Table 21.85 Electrical Characteristics measurment condition modified, note 4 deleted |
| | | | Usage Notes |
| | | 439 | • Figure 22.4 Use of Capacitors to Reduce Noise note 1 modified |
| | | 449 | • 22.15.10 How to Access description modified |
| | | 450 | • 22.15.15 Flash Memory Version Electrical Characteristics 10,000 E/W Cycle Products description modified |
| | | | |

**RENESAS 16-BIT SINGLE-CHIP MICROCOMPUTER
HARDWARE MANUAL
M16C/29 Group**

**Publication Data : Rev.0.70 Mar. 29, 2004
Rev.1.12 Mar. 30, 2007**

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

M16C/29 Group Hardware Manual



Renesas Electronics Corporation

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

REJ09B0101-0112