

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/6N Group (M16C/6N4)

Hardware Manual

Renesas MCU
M16C Family / M16C/60 Series

All information contained in this material, including products and product specifications at the time of publication of this material, 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>).

Keep safety first in your circuit designs!

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

Notes regarding these materials

- These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corporation product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corporation or a third party.
- Renesas Technology Corporation assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
- All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor for the latest product information before purchasing a product listed herein.

The information described here may contain technical inaccuracies or typographical errors. Renesas Technology Corporation assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.

Please also pay attention to information published by Renesas Technology Corporation by various means, including the Renesas Technology Corporation Semiconductor home page (<http://www.renesas.com>).

- When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
- Renesas Technology Corporation semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
- The prior written approval of Renesas Technology Corporation is necessary to reprint or reproduce in whole or in part these materials.
- If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
- Please contact Renesas Technology Corporation for further details on these materials or the products contained therein.

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 type number, confirm that the change will not lead to problems.

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

Blank page

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/6N Group (M16C/6N4). Make sure to refer to the latest versions of these documents. The newest versions of the documents listed may be obtained from the Renesas Technology Web site.

| Document Type | Description | Document Title | Document No. |
|--------------------------|--|--|-----------------------------------|
| Datasheet | Hardware overview and electrical characteristics | M16C/6N Group (M16C/6N4) Datasheet | REJ03B0003 |
| 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/6N Group (M16C/6N4) Hardware Manual | This hardware manual (REJ09B0009) |
| 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 “b” is appended to numeric values given in binary format. However, nothing is appended to the values of single bits. The indication “h” is appended to numeric values given in hexadecimal format. Nothing is appended to numeric values given in decimal format.

Examples Binary: 11b

Hexadecimal: EFA0h

Decimal: 1234

3. Register Notation

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

XXX Register

| Bit Symbol | Bit Name | Function | RW |
|------------|---|---|------------------|
| XXX0 | XXX bits | b1b0 0 0: XXX 0 1: XXX | RW ^{*2} |
| XXX1 | | 1 0: Do not set a value 1 1: XXX | RW |
| – (b2) | Nothing is assigned. If necessary, set to 0, When read, the content is undefined. | | – ^{*3} |
| – (b4-b3) | Reserved bits | Set to 0 | WO ^{*4} |
| XXX5 | XXX bits | Function varies depending on operating mode | RW |
| XXX6 | | | RW |
| XXX7 | XXX bit | 0: XXX 1: XXX | RO |

^{*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 a value
- Operation is not guaranteed when a value is set.
- Function varies depending on 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 |

Table of Contents

| | |
|---|-----------|
| SFR Page Reference | B-1 |
| 1. Overview | 1 |
| 1.1 Applications | 1 |
| 1.2 Performance Overview | 2 |
| 1.3 Block Diagram | 3 |
| 1.4 Product Information | 4 |
| 1.5 Pin Assignments | 5 |
| 1.6 Pin Functions | 9 |
| 2. Central Processing Unit (CPU) | 12 |
| 2.1 Data Registers (R0, R1, R2, and R3) | 12 |
| 2.2 Address Registers (A0 and A1) | 12 |
| 2.3 Frame Base Register (FB) | 13 |
| 2.4 Interrupt Table Register (INTB) | 13 |
| 2.5 Program Counter (PC) | 13 |
| 2.6 User Stack Pointer (USP), Interrupt Stack Pointer (ISP) | 13 |
| 2.7 Static Base Register (SB) | 13 |
| 2.8 Flag Register (FLG) | 13 |
| 2.8.1 Carry Flag (C Flag) | 13 |
| 2.8.2 Debug Flag (D Flag) | 13 |
| 2.8.3 Zero Flag (Z Flag) | 13 |
| 2.8.4 Sign Flag (S Flag) | 13 |
| 2.8.5 Register Bank Select Flag (B Flag) | 13 |
| 2.8.6 Overflow Flag (O Flag) | 13 |
| 2.8.7 Interrupt Enable Flag (I Flag) | 13 |
| 2.8.8 Stack Pointer Select Flag (U Flag) | 13 |
| 2.8.9 Processor Interrupt Priority Level (IPL) | 13 |
| 2.8.10 Reserved Area | 13 |
| 3. Memory | 14 |
| 4. Special Function Registers (SFRs) | 15 |
| 5. Resets | 31 |
| 5.1 Hardware Reset | 31 |
| 5.1.1 Reset on a Stable Supply Voltage | 31 |
| 5.1.2 Power-on Reset | 31 |
| 5.2 Software Reset | 33 |
| 5.3 Watchdog Timer Reset | 33 |
| 5.4 Oscillation Stop Detection Reset | 33 |
| 5.5 Internal Space | 33 |
| 6. Processor Mode | 34 |
| 6.1 Types of Processor Mode | 34 |
| 6.2 Setting Processor Modes | 34 |
| 7. Bus | 40 |
| 7.1 Bus Mode | 40 |
| 7.1.1 Separate Bus | 40 |
| 7.1.2 Multiplexed Bus | 40 |

| | |
|--|----|
| 7.2 Bus Control | 41 |
| 7.2.1 Address Bus | 41 |
| 7.2.2 Data Bus | 41 |
| 7.2.3 Chip Select Signal | 41 |
| 7.2.4 Read and Write Signals | 43 |
| 7.2.5 ALE Signal | 43 |
| 7.2.6 $\overline{\text{RDY}}$ Signal | 44 |
| 7.2.7 $\overline{\text{HOLD}}$ Signal | 45 |
| 7.2.8 BCLK Output | 45 |
| 7.2.9 External Bus Status when Internal Area Accessed | 47 |
| 7.2.10 Software Wait | 47 |
| 8. Clock Generation Circuit | 51 |
| 8.1 Types of Clock Generation Circuit | 51 |
| 8.1.1 Main Clock | 58 |
| 8.1.2 Sub Clock | 59 |
| 8.1.3 On-chip Oscillator Clock | 60 |
| 8.1.4 PLL Clock | 60 |
| 8.2 CPU Clock and Peripheral Function Clock | 62 |
| 8.2.1 CPU Clock and BCLK | 62 |
| 8.2.2 Peripheral Function Clock | 62 |
| 8.3 Clock Output Function | 62 |
| 8.4 Power Control | 63 |
| 8.4.1 Normal Operating Mode | 63 |
| 8.4.2 Wait Mode | 65 |
| 8.4.3 Stop Mode | 67 |
| 8.5 Oscillation Stop and Re-oscillation Detection Function | 72 |
| 8.5.1 Operation when CM27 Bit = 0 (Oscillation Stop Detection Reset) | 72 |
| 8.5.2 Operation when CM27 Bit = 1 (Oscillation Stop, Re-oscillation Detection Interrupt) | 72 |
| 8.5.3 How to Use Oscillation Stop and Re-oscillation Detection Function | 73 |
| 9. Protection | 74 |
| 10. Interrupts | 75 |
| 10.1 Type of Interrupts | 75 |
| 10.2 Software Interrupts | 76 |
| 10.2.1 Undefined Instruction Interrupt | 76 |
| 10.2.2 Overflow Interrupt | 76 |
| 10.2.3 BRK Interrupt | 76 |
| 10.2.4 INT Instruction Interrupt | 76 |
| 10.3 Hardware Interrupts | 77 |
| 10.3.1 Special Interrupts | 77 |
| 10.3.2 Peripheral Function Interrupts | 77 |
| 10.4 Interrupts and Interrupt Vector | 78 |
| 10.4.1 Fixed Vector Tables | 78 |
| 10.4.2 Relocatable Vector Tables | 79 |
| 10.5 Interrupt Control | 80 |
| 10.5.1 I Flag | 82 |
| 10.5.2 IR Bit | 82 |
| 10.5.3 Bits ILVL2 to ILVL0 and IPL | 82 |

| | | |
|---------|---|-----|
| 10.5.4 | Interrupt Sequence | 83 |
| 10.5.5 | Interrupt Response Time | 84 |
| 10.5.6 | Variation of IPL when Interrupt Request is Accepted | 84 |
| 10.5.7 | Saving Registers | 85 |
| 10.5.8 | Returning from Interrupt Routine | 86 |
| 10.5.9 | Interrupt Priority | 86 |
| 10.5.10 | Interrupt Priority Level Select Circuit | 86 |
| 10.6 | $\overline{\text{INT}}$ Interrupt | 88 |
| 10.7 | $\overline{\text{NMI}}$ Interrupt | 90 |
| 10.8 | Key Input Interrupt | 90 |
| 10.9 | CAN0/1 Wake-up Interrupt | 90 |
| 10.10 | Address Match Interrupt | 91 |
| 11. | Watchdog Timer | 93 |
| 11.1 | Count Source Protective Mode | 94 |
| 12. | DMAC | 95 |
| 12.1 | Transfer Cycle | 100 |
| 12.1.1 | Effect of Source and Destination Addresses | 100 |
| 12.1.2 | Effect of BYTE Pin Level | 100 |
| 12.1.3 | Effect of Software Wait | 100 |
| 12.1.4 | Effect of $\overline{\text{RDY}}$ Signal | 100 |
| 12.2 | DMA Transfer Cycles | 102 |
| 12.3 | DMA Enable | 103 |
| 12.4 | DMA Request | 103 |
| 12.5 | Channel Priority and DMA Transfer Timing | 104 |
| 13. | Timers | 105 |
| 13.1 | Timer A | 107 |
| 13.1.1 | Timer Mode | 111 |
| 13.1.2 | Event Counter Mode | 112 |
| 13.1.3 | One-shot Timer Mode | 117 |
| 13.1.4 | Pulse Width Modulation (PWM) Mode | 119 |
| 13.2 | Timer B | 122 |
| 13.2.1 | Timer Mode | 125 |
| 13.2.2 | Event Counter Mode | 126 |
| 13.2.3 | Pulse Period and Pulse Width Measurement Mode | 127 |
| 14. | Three-Phase Motor Control Timer Function | 130 |
| 15. | Serial Interface | 141 |
| 15.1 | UARTi | 141 |
| 15.1.1 | Clock Synchronous Serial I/O Mode | 151 |
| 15.1.2 | Clock Asynchronous Serial I/O (UART) Mode | 159 |
| 15.1.3 | Special Mode 1 (I ² C Mode) | 167 |
| 15.1.4 | Special Mode 2 | 176 |
| 15.1.5 | Special Mode 3 (IE Mode) | 181 |
| 15.1.6 | Special Mode 4 (SIM Mode) (UART2) | 183 |
| 15.2 | SI/O3 | 188 |
| 15.2.1 | SI/O3 Operation Timing | 191 |
| 15.2.2 | CLK Polarity Selection | 191 |
| 15.2.3 | Functions for Setting SOUT3 Initial Value | 192 |

| | |
|--|-----|
| 16. A/D Converter | 193 |
| 16.1 Mode Description | 197 |
| 16.1.1 One-shot Mode | 197 |
| 16.1.2 Repeat Mode | 199 |
| 16.1.3 Single Sweep Mode | 201 |
| 16.1.4 Repeat Sweep Mode 0 | 203 |
| 16.1.5 Repeat Sweep Mode 1 | 205 |
| 16.2 Function | 207 |
| 16.2.1 Resolution Select Function | 207 |
| 16.2.2 Sample and Hold | 207 |
| 16.2.3 Extended Analog Input Pins | 207 |
| 16.2.4 External Operation Amplifier (Op-Amp) Connection Mode | 207 |
| 16.2.5 Current Consumption Reducing Function | 208 |
| 16.2.6 Output Impedance of Sensor under A/D Conversion | 208 |
| 17. D/A Converter | 210 |
| 18. CRC Calculation | 212 |
| 19. CAN Module | 214 |
| 19.1 CAN Module-Related Registers | 215 |
| 19.1.1 CANi Message Box | 215 |
| 19.1.2 Acceptance Mask Registers | 215 |
| 19.1.3 CAN SFR Registers | 215 |
| 19.2 CANi Message Box | 216 |
| 19.3 Acceptance Mask Registers | 218 |
| 19.4 CAN SFR Registers | 219 |
| 19.5 Operational Modes | 225 |
| 19.5.1 CAN Reset/Initialization Mode | 225 |
| 19.5.2 CAN Operation Mode | 226 |
| 19.5.3 CAN Sleep Mode | 226 |
| 19.5.4 CAN Interface Sleep Mode | 226 |
| 19.5.5 Bus Off State | 227 |
| 19.6 CAN Module System Clock Configuration | 228 |
| 19.7 Bit Timing Configuration | 228 |
| 19.8 Bit-rate | 229 |
| 19.8.1 Calculation of Bit-rate | 229 |
| 19.9 Acceptance Filtering Function and Masking Function | 230 |
| 19.10 Acceptance Filter Support Unit (ASU) | 231 |
| 19.11 Basic CAN Mode | 232 |
| 19.12 Return from Bus Off Function | 233 |
| 19.13 Time Stamp Counter and Time Stamp Function | 233 |
| 19.14 Listen-Only Mode | 233 |
| 19.15 Reception and Transmission | 234 |
| 19.15.1 Reception | 235 |
| 19.15.2 Transmission | 236 |
| 19.16 CAN Interrupt | 237 |

| | |
|---|-----|
| 20. Programmable I/O Ports | 238 |
| 20.1 PDi Register | 238 |
| 20.2 Pi Register | 238 |
| 20.3 PURj Register | 238 |
| 20.4 PCR Register | 239 |
| 21. Flash Memory Version | 251 |
| 21.1 Memory Map | 252 |
| 21.1.1 Boot Mode | 252 |
| 21.2 Functions to Prevent Flash Memory from Rewriting | 253 |
| 21.2.1 ROM Code Protect Function | 253 |
| 21.2.2 ID Code Check Function | 253 |
| 21.3 CPU Rewrite Mode | 255 |
| 21.3.1 EW0 Mode | 256 |
| 21.3.2 EW1 Mode | 256 |
| 21.3.3 Registers FMR0 and FMR1 | 257 |
| 21.3.4 Notes on CPU Rewrite Mode | 262 |
| 21.3.5 Software Commands | 264 |
| 21.3.6 Data Protect Function | 269 |
| 21.3.7 Status Register (SRD Register) | 269 |
| 21.3.8 Full Status Check | 271 |
| 21.4 Standard Serial I/O Mode | 273 |
| 21.4.1 ID Code Check Function | 273 |
| 21.4.2 Example of Circuit Application in Standard Serial I/O Mode | 277 |
| 21.5 Parallel I/O Mode | 278 |
| 21.5.1 User ROM and Boot ROM Areas | 278 |
| 21.5.2 ROM Code Protect Function | 278 |
| 21.6 CAN I/O Mode | 279 |
| 21.6.1 ID Code Check Function | 279 |
| 21.6.2 Example of Circuit Application in CAN I/O Mode | 282 |
| 21.7 Electrical Characteristics | 283 |
| 21.7.1 Electrical Characteristics (T/V-ver.) | 283 |
| 21.7.2 Electrical Characteristics (Normal-ver.) | 284 |
| 22. Electrical Characteristics | 285 |
| 22.1 Electrical Characteristics (T/V-ver.) | 285 |
| 22.2 Electrical Characteristics (Normal-ver.) | 306 |
| 23. Usage Notes | 342 |
| 23.1 SFRs | 342 |
| 23.2 External Bus | 343 |
| 23.3 PLL Frequency Synthesizer | 344 |
| 23.4 Power Control | 345 |
| 23.5 Protection | 347 |
| 23.6 Interrupts | 348 |
| 23.6.1 Reading Address 00000h | 348 |
| 23.6.2 Setting SP | 348 |
| 23.6.3 NMI Interrupt | 348 |
| 23.6.4 Changing Interrupt Source | 349 |
| 23.6.5 INT Interrupt | 349 |
| 23.6.6 Rewrite Interrupt Control Register | 350 |
| 23.6.7 Watchdog Timer Interrupt | 350 |

| | |
|--|---------|
| 23.7 DMAC | 351 |
| 23.7.1 Write to DMAE Bit in DMiCON Register | 351 |
| 23.8 Timers | 352 |
| 23.8.1 Timer A | 352 |
| 23.8.2 Timer B | 355 |
| 23.9 Serial Interface | 357 |
| 23.9.1 Clock Synchronous Serial I/O Mode | 357 |
| 23.9.2 Special Modes | 358 |
| 23.9.3 SI/O3 | 359 |
| 23.10 A/D Converter | 360 |
| 23.11 CAN Module | 362 |
| 23.11.1 Reading CiSTR Register | 362 |
| 23.11.2 Performing CAN Configuration | 364 |
| 23.11.3 Suggestions to Reduce Power Consumption | 365 |
| 23.11.4 CAN Transceiver in Boot Mode | 366 |
| 23.12 Programmable I/O Ports | 367 |
| 23.13 Electrical Characteristic Differences between Mask ROM and Flash Memory Version MCUs | 368 |
| 23.14 Mask ROM Version | 369 |
| 23.15 Flash Memory Version | 370 |
| 23.15.1 Functions to Prevent Flash Memory from Rewriting | 370 |
| 23.15.2 Stop Mode | 370 |
| 23.15.3 Wait Mode | 370 |
| 23.15.4 Low Power Dissipation Mode and On-Chip Oscillator Low Power Dissipation Mode | 370 |
| 23.15.5 Writing Command and Data | 370 |
| 23.15.6 Program Command | 370 |
| 23.15.7 Lock Bit Program Command | 370 |
| 23.15.8 Operating Speed | 370 |
| 23.15.9 Prohibited Instructions | 371 |
| 23.15.10 Interrupts | 371 |
| 23.15.11 How to Access | 371 |
| 23.15.12 Rewriting in User ROM Area | 371 |
| 23.15.13 DMA Transfer | 371 |
| 23.16 Flash Memory Programming Using Boot Program | 372 |
| 23.16.1 Programming Using Serial I/O Mode | 372 |
| 23.16.2 Programming Using CAN I/O Mode | 372 |
| 23.17 Noise | 373 |
| Appendix 1. Package Dimensions | 374 |
| Register Index | 375 |

| |
|---|
| <p>Specifications written in this manual are believed to be accurate, but are not guaranteed to be entirely free of error. Specifications in this manual may be changed for functional or performance improvements. Please make sure your manual is the latest edition.</p> |
|---|

| Address | Register | Symbol | Page |
|---------|--------------------------------------|--------|------|
| 0080h | CAN0 Message Box 2: Identifier / DLC | | |
| 0081h | | | |
| 0082h | | | |
| 0083h | | | |
| 0084h | | | |
| 0085h | CAN0 Message Box 2: Data Field | | |
| 0086h | | | |
| 0087h | | | |
| 0088h | | | |
| 0089h | | | |
| 008Ah | | | |
| 008Bh | | | |
| 008Ch | | | |
| 008Dh | | | |
| 008Eh | CAN0 Message Box 2: Time Stamp | | |
| 008Fh | | | |
| 0090h | CAN0 Message Box 3: Identifier / DLC | | |
| 0091h | | | |
| 0092h | | | |
| 0093h | | | |
| 0094h | | | |
| 0095h | | | |
| 0096h | CAN0 Message Box 3: Data Field | | |
| 0097h | | | |
| 0098h | | | |
| 0099h | | | |
| 009Ah | | | |
| 009Bh | | | |
| 009Ch | | | |
| 009Dh | | | |
| 009Eh | CAN0 Message Box 3: Time Stamp | | |
| 009Fh | | | |
| 00A0h | CAN0 Message Box 4: Identifier / DLC | | |
| 00A1h | | | |
| 00A2h | | | |
| 00A3h | | | |
| 00A4h | | | |
| 00A5h | | | |
| 00A6h | CAN0 Message Box 4: Data Field | | |
| 00A7h | | | |
| 00A8h | | | |
| 00A9h | | | |
| 00AAh | | | |
| 00ABh | | | |
| 00ACh | | | |
| 00ADh | | | |
| 00AEh | CAN0 Message Box 4: Time Stamp | | |
| 00AFh | | | |
| 00B0h | CAN0 Message Box 5: Identifier / DLC | | |
| 00B1h | | | |
| 00B2h | | | |
| 00B3h | | | |
| 00B4h | | | |
| 00B5h | | | |
| 00B6h | CAN0 Message Box 5: Data Field | | |
| 00B7h | | | |
| 00B8h | | | |
| 00B9h | | | |
| 00BAh | | | |
| 00BBh | | | |
| 00BCh | | | |
| 00BDh | | | |
| 00BEh | CAN0 Message Box 5: Time Stamp | | |
| 00BFh | | | |

216
217

| Address | Register | Symbol | Page |
|---------|--------------------------------------|--------|------|
| 00C0h | CAN0 Message Box 6: Identifier / DLC | | |
| 00C1h | | | |
| 00C2h | | | |
| 00C3h | | | |
| 00C4h | | | |
| 00C5h | CAN0 Message Box 6: Data Field | | |
| 00C6h | | | |
| 00C7h | | | |
| 00C8h | | | |
| 00C9h | | | |
| 00CAh | | | |
| 00CBh | | | |
| 00CCh | | | |
| 00CDh | | | |
| 00CEh | CAN0 Message Box 6: Time Stamp | | |
| 00CFh | | | |
| 00D0h | CAN0 Message Box 7: Identifier / DLC | | |
| 00D1h | | | |
| 00D2h | | | |
| 00D3h | | | |
| 00D4h | | | |
| 00D5h | | | |
| 00D6h | CAN0 Message Box 7: Data Field | | |
| 00D7h | | | |
| 00D8h | | | |
| 00D9h | | | |
| 00DAh | | | |
| 00DBh | | | |
| 00DCh | | | |
| 00DDh | | | |
| 00DEh | CAN0 Message Box 7: Time Stamp | | |
| 00DFh | | | |
| 00E0h | CAN0 Message Box 8: Identifier / DLC | | |
| 00E1h | | | |
| 00E2h | | | |
| 00E3h | | | |
| 00E4h | | | |
| 00E5h | | | |
| 00E6h | CAN0 Message Box 8: Data Field | | |
| 00E7h | | | |
| 00E8h | | | |
| 00E9h | | | |
| 00EAh | | | |
| 00EBh | | | |
| 00ECh | | | |
| 00EDh | | | |
| 00EEh | CAN0 Message Box 8: Time Stamp | | |
| 00EFh | | | |
| 00F0h | CAN0 Message Box 9: Identifier / DLC | | |
| 00F1h | | | |
| 00F2h | | | |
| 00F3h | | | |
| 00F4h | | | |
| 00F5h | | | |
| 00F6h | CAN0 Message Box 9: Data Field | | |
| 00F7h | | | |
| 00F8h | | | |
| 00F9h | | | |
| 00FAh | | | |
| 00FBh | | | |
| 00FCh | | | |
| 00FDh | | | |
| 00FEh | CAN0 Message Box 9: Time Stamp | | |
| 00FFh | | | |

216
217

| Address | Register | Symbol | Page |
|---------|---------------------------------------|--------|------|
| 0100h | CAN0 Message Box 10: Identifier / DLC | | |
| 0101h | | | |
| 0102h | | | |
| 0103h | | | |
| 0104h | | | |
| 0105h | | | |
| 0106h | CAN0 Message Box 10: Data Field | | |
| 0107h | | | |
| 0108h | | | |
| 0109h | | | |
| 010Ah | | | |
| 010Bh | | | |
| 010Ch | | | |
| 010Dh | | | |
| 010Eh | CAN0 Message Box 10: Time Stamp | | |
| 010Fh | | | |
| 0110h | CAN0 Message Box 11: Identifier / DLC | | |
| 0111h | | | |
| 0112h | | | |
| 0113h | | | |
| 0114h | | | |
| 0115h | | | |
| 0116h | CAN0 Message Box 11: Data Field | | |
| 0117h | | | |
| 0118h | | | |
| 0119h | | | |
| 011Ah | | | |
| 011Bh | | | |
| 011Ch | | | |
| 011Dh | | | |
| 011Eh | CAN0 Message Box 11: Time Stamp | | 216 |
| 011Fh | | | |
| 0120h | CAN0 Message Box 12: Identifier / DLC | | 217 |
| 0121h | | | |
| 0122h | | | |
| 0123h | | | |
| 0124h | | | |
| 0125h | | | |
| 0126h | CAN0 Message Box 12: Data Field | | |
| 0127h | | | |
| 0128h | | | |
| 0129h | | | |
| 012Ah | | | |
| 012Bh | | | |
| 012Ch | | | |
| 012Dh | | | |
| 012Eh | CAN0 Message Box 12: Time Stamp | | |
| 012Fh | | | |
| 0130h | CAN0 Message Box 13: Identifier / DLC | | |
| 0131h | | | |
| 0132h | | | |
| 0133h | | | |
| 0134h | | | |
| 0135h | | | |
| 0136h | CAN0 Message Box 13: Data Field | | |
| 0137h | | | |
| 0138h | | | |
| 0139h | | | |
| 013Ah | | | |
| 013Bh | | | |
| 013Ch | | | |
| 013Dh | | | |
| 013Eh | CAN0 Message Box 13: Time Stamp | | |
| 013Fh | | | |

| Address | Register | Symbol | Page |
|---------|--------------------------------------|--------|------|
| 0140h | CAN0 Message Box 14: Identifier /DLC | | |
| 0141h | | | |
| 0142h | | | |
| 0143h | | | |
| 0144h | | | |
| 0145h | | | |
| 0146h | CAN0 Message Box 14: Data Field | | |
| 0147h | | | |
| 0148h | | | |
| 0149h | | | |
| 014Ah | | | |
| 014Bh | | | |
| 014Ch | | | |
| 014Dh | | | |
| 014Eh | CAN0 Message Box 14: Time Stamp | | 216 |
| 014Fh | | | |
| 0150h | CAN0 Message Box 15: Identifier /DLC | | 217 |
| 0151h | | | |
| 0152h | | | |
| 0153h | | | |
| 0154h | | | |
| 0155h | | | |
| 0156h | CAN0 Message Box 15: Data Field | | |
| 0157h | | | |
| 0158h | | | |
| 0159h | | | |
| 015Ah | | | |
| 015Bh | | | |
| 015Ch | | | |
| 015Dh | | | |
| 015Eh | CAN0 Message Box 15: Time Stamp | | |
| 015Fh | | | |
| 0160h | CAN0 Global Mask Register | | 218 |
| 0161h | | | |
| 0162h | | | |
| 0163h | | | |
| 0164h | | | |
| 0165h | | | |
| 0166h | CAN0 Local Mask A Register | | 218 |
| 0167h | | | |
| 0168h | | | |
| 0169h | | | |
| 016Ah | | | |
| 016Bh | | | |
| 016Ch | CAN0 Local Mask B Register | | 218 |
| 016Dh | | | |
| 016Eh | | | |
| 016Fh | | | |
| 0170h | | | |
| 0171h | | | |
| 0172h | | | |
| 0173h | | | |
| 0174h | | | |
| 0175h | | | |
| 0176h | | | |
| 0177h | | | |
| 0178h | | | |
| 0179h | | | |
| 017Ah | | | |
| 017Bh | | | |
| 017Ch | | | |
| 017Dh | | | |
| 017Eh | | | |
| 017Fh | | | |

Blank spaces are reserved. No access is allowed.

| Address | Register | Symbol | Page |
|---------|---|--------|------|
| 0180h | | | |
| 0181h | | | |
| 0182h | | | |
| 0183h | | | |
| 0184h | | | |
| 0185h | | | |
| 0186h | | | |
| 0187h | | | |
| 0188h | | | |
| 0189h | | | |
| 018Ah | | | |
| 018Bh | | | |
| 018Ch | | | |
| 018Dh | | | |
| 018Eh | | | |
| 018Fh | | | |
| 0190h | | | |
| 0191h | | | |
| 0192h | | | |
| 0193h | | | |
| 0194h | | | |
| 0195h | | | |
| 0196h | | | |
| 0197h | | | |
| 0198h | | | |
| 0199h | | | |
| 019Ah | | | |
| 019Bh | | | |
| 019Ch | | | |
| 019Dh | | | |
| 019Eh | | | |
| 019Fh | | | |
| 01A0h | | | |
| 01A1h | | | |
| 01A2h | | | |
| 01A3h | | | |
| 01A4h | | | |
| 01A5h | | | |
| 01A6h | | | |
| 01A7h | | | |
| 01A8h | | | |
| 01A9h | | | |
| 01AAh | | | |
| 01ABh | | | |
| 01ACh | | | |
| 01ADh | | | |
| 01AEh | | | |
| 01AFh | | | |
| 01B0h | | | |
| 01B1h | | | |
| 01B2h | | | |
| 01B3h | | | |
| 01B4h | | | |
| 01B5h | Flash Memory Control Register 1 | FMR1 | 257 |
| 01B6h | | | |
| 01B7h | Flash Memory Control Register 0 | FMR0 | 257 |
| 01B8h | | | |
| 01B9h | Address Match Interrupt Register 2 | RMAD2 | 92 |
| 01BAh | | | |
| 01BBh | Address Match Interrupt Enable Register 2 | AIER2 | 92 |
| 01BCh | | | |
| 01BDh | Address Match Interrupt Register 3 | RMAD3 | 92 |
| 01BEh | | | |
| 01BFh | | | |

| Address | Register | Symbol | Page |
|---------|---|--------|------|
| 01C0h | Timer B3, B4, B5 Count Start Flag | TBSR | 124 |
| 01C1h | | | |
| 01C2h | | | |
| 01C3h | Timer A1-1 Register | TA11 | 135 |
| 01C4h | | | |
| 01C5h | Timer A2-1 Register | TA21 | 135 |
| 01C6h | | | |
| 01C7h | Timer A4-1 Register | TA41 | 135 |
| 01C8h | Three-Phase PWM Control Register 0 | INVC0 | 132 |
| 01C9h | Three-Phase PWM Control Register 1 | INVC1 | 133 |
| 01CAh | Three-Phase Output Buffer Register 0 | IDB0 | 134 |
| 01CBh | Three-Phase Output Buffer Register 1 | IDB1 | 134 |
| 01CCh | Dead Time Timer | DTT | 134 |
| 01CDh | Timer B2 Interrupt Generation Frequency Set Counter | ICTB2 | 136 |
| 01CEh | | | |
| 01CFh | | | |
| 01D0h | | | |
| 01D1h | Timer B3 Register | TB3 | 123 |
| 01D2h | | | |
| 01D3h | Timer B4 Register | TB4 | 123 |
| 01D4h | | | |
| 01D5h | Timer B5 Register | TB5 | 123 |
| 01D6h | | | |
| 01D7h | | | |
| 01D8h | | | |
| 01D9h | | | |
| 01DAh | | | |
| 01DBh | Timer B3 Mode Register | TB3MR | 123 |
| 01DCh | Timer B4 Mode Register | TB4MR | 125 |
| 01DDh | Timer B5 Mode Register | TB5MR | 126 |
| 01DEh | Interrupt Source Select Register 0 | IFSR0 | 128 |
| 01DFh | Interrupt Source Select Register 1 | IFSR1 | 89 |
| 01E0h | SI/O3 Transmit/Receive Register | S3TRR | 89 |
| 01E1h | | | |
| 01E2h | SI/O3 Control Register | S3C | 189 |
| 01E3h | SI/O3 Bit Rate Register | S3BRG | 189 |
| 01E4h | | | |
| 01E5h | | | |
| 01E6h | | | |
| 01E7h | | | |
| 01E8h | | | |
| 01E9h | | | |
| 01EAh | | | |
| 01EBh | | | |
| 01ECh | UART0 Special Mode Register 4 | U0SMR4 | 150 |
| 01EDh | UART0 Special Mode Register 3 | U0SMR3 | 149 |
| 01EEh | UART0 Special Mode Register 2 | U0SMR2 | 149 |
| 01EFh | UART0 Special Mode Register | U0SMR | 148 |
| 01F0h | UART1 Special Mode Register 4 | U1SMR4 | 150 |
| 01F1h | UART1 Special Mode Register 3 | U1SMR3 | 149 |
| 01F2h | UART1 Special Mode Register 2 | U1SMR2 | 149 |
| 01F3h | UART1 Special Mode Register | U1SMR | 148 |
| 01F4h | UART2 Special Mode Register 4 | U2SMR4 | 150 |
| 01F5h | UART2 Special Mode Register 3 | U2SMR3 | 149 |
| 01F6h | UART2 Special Mode Register 2 | U2SMR2 | 149 |
| 01F7h | UART2 Special Mode Register | U2SMR | 148 |
| 01F8h | UART2 Transmit/Receive Mode Register | U2MR | 146 |
| 01F9h | UART2 Bit Rate Register | U2BRG | 145 |
| 01FAh | | | |
| 01FBh | UART2 Transmit Buffer Register | U2TB | 145 |
| 01FCh | UART2 Transmit/Receive Control Register 0 | U2C0 | 146 |
| 01FDh | UART2 Transmit/Receive Control Register 1 | U2C1 | 147 |
| 01FEh | | | |
| 01FFh | UART2 Receive Buffer Register | U2RB | 145 |

Blank spaces are reserved. No access is allowed.

| Address | Register | Symbol | Page |
|---------|------------------------------------|----------|------|
| 0200h | CAN0 Message Control Register 0 | C0MCTL0 | 219 |
| 0201h | CAN0 Message Control Register 1 | C0MCTL1 | |
| 0202h | CAN0 Message Control Register 2 | C0MCTL2 | |
| 0203h | CAN0 Message Control Register 3 | C0MCTL3 | |
| 0204h | CAN0 Message Control Register 4 | C0MCTL4 | |
| 0205h | CAN0 Message Control Register 5 | C0MCTL5 | |
| 0206h | CAN0 Message Control Register 6 | C0MCTL6 | |
| 0207h | CAN0 Message Control Register 7 | C0MCTL7 | |
| 0208h | CAN0 Message Control Register 8 | C0MCTL8 | |
| 0209h | CAN0 Message Control Register 9 | C0MCTL9 | |
| 020Ah | CAN0 Message Control Register 10 | C0MCTL10 | |
| 020Bh | CAN0 Message Control Register 11 | C0MCTL11 | |
| 020Ch | CAN0 Message Control Register 12 | C0MCTL12 | |
| 020Dh | CAN0 Message Control Register 13 | C0MCTL13 | |
| 020Eh | CAN0 Message Control Register 14 | C0MCTL14 | |
| 020Fh | CAN0 Message Control Register 15 | C0MCTL15 | |
| 0210h | CAN0 Control Register | C0CTLR | 220 |
| 0211h | | | |
| 0212h | CAN0 Status Register | C0STR | 221 |
| 0213h | | | |
| 0214h | CAN0 Slot Status Register | C0SSTR | 222 |
| 0215h | | | |
| 0216h | CAN0 Interrupt Control Register | C0ICR | 222 |
| 0217h | | | |
| 0218h | CAN0 Extended ID Register | C0IDR | 222 |
| 0219h | | | |
| 021Ah | CAN0 Configuration Register | C0CONR | 223 |
| 021Bh | | | |
| 021Ch | CAN0 Receive Error Count Register | C0RECR | 224 |
| 021Dh | CAN0 Transmit Error Count Register | C0TECR | 224 |
| 021Eh | CAN0 Time Stamp Register | C0TSR | 224 |
| 021Fh | | | |
| 0220h | CAN1 Message Control Register 0 | C1MCTL0 | 219 |
| 0221h | CAN1 Message Control Register 1 | C1MCTL1 | |
| 0222h | CAN1 Message Control Register 2 | C1MCTL2 | |
| 0223h | CAN1 Message Control Register 3 | C1MCTL3 | |
| 0224h | CAN1 Message Control Register 4 | C1MCTL4 | |
| 0225h | CAN1 Message Control Register 5 | C1MCTL5 | |
| 0226h | CAN1 Message Control Register 6 | C1MCTL6 | |
| 0227h | CAN1 Message Control Register 7 | C1MCTL7 | |
| 0228h | CAN1 Message Control Register 8 | C1MCTL8 | |
| 0229h | CAN1 Message Control Register 9 | C1MCTL9 | |
| 022Ah | CAN1 Message Control Register 10 | C1MCTL10 | |
| 022Bh | CAN1 Message Control Register 11 | C1MCTL11 | |
| 022Ch | CAN1 Message Control Register 12 | C1MCTL12 | |
| 022Dh | CAN1 Message Control Register 13 | C1MCTL13 | |
| 022Eh | CAN1 Message Control Register 14 | C1MCTL14 | |
| 022Fh | CAN1 Message Control Register 15 | C1MCTL15 | |
| 0230h | CAN1 Control Register | C1CTLR | 220 |
| 0231h | | | |
| 0232h | CAN1 Status Register | C1STR | 221 |
| 0233h | | | |
| 0234h | CAN1 Slot Status Register | C1SSTR | 222 |
| 0235h | | | |
| 0236h | CAN1 Interrupt Control Register | C1ICR | 222 |
| 0237h | | | |
| 0238h | CAN1 Extended ID Register | C1IDR | 222 |
| 0239h | | | |
| 023Ah | CAN1 Configuration Register | C1CONR | 223 |
| 023Bh | | | |
| 023Ch | CAN1 Receive Error Count Register | C1RECR | 224 |
| 023Dh | CAN1 Transmit Error Count Register | C1TECR | 224 |
| 023Eh | CAN1 Time Stamp Register | C1TSR | 224 |
| 023Fh | | | |

| Address | Register | Symbol | Page | | | |
|---------|---|--------|------|--------------------------------------|--|-----|
| 0240h | | | | | | |
| 0241h | | | | | | |
| 0242h | CAN0 Acceptance Filter Support Register | C0AFS | 224 | | | |
| 0243h | | | | | | |
| 0244h | CAN1 Acceptance Filter Support Register | C1AFS | 224 | | | |
| 0245h | | | | | | |
| 0246h | | | | | | |
| 0247h | | | | | | |
| 0248h | | | | | | |
| 0249h | | | | | | |
| 024Ah | | | | | | |
| 024Bh | | | | | | |
| 024Ch | | | | | | |
| 024Dh | | | | | | |
| 024Eh | | | | | | |
| 024Fh | | | | | | |
| 0250h | | | | | | |
| 0251h | | | | | | |
| 0252h | | | | | | |
| 0253h | | | | | | |
| 0254h | | | | | | |
| 0255h | | | | | | |
| 0256h | | | | | | |
| 0257h | | | | | | |
| 0258h | | | | | | |
| 0259h | | | | | | |
| 025Ah | | | | | | |
| 025Bh | | | | | | |
| 025Ch | | | | | | |
| 025Dh | | | | | | |
| 025Eh | Peripheral Clock Select Register | PCLKR | 56 | | | |
| 025Fh | CAN0/1 Clock Select Register | CCLKR | 56 | | | |
| 0260h | CAN1 Message Box 0: Identifier / DLC | | | | | |
| 0261h | | | | | | |
| 0262h | | | | | | |
| 0263h | | | | | | |
| 0264h | | | | | | |
| 0265h | | | | | | |
| 0266h | CAN1 Message Box 0: Data Field | | | | | |
| 0267h | | | | | | |
| 0268h | | | | | | |
| 0269h | | | | | | |
| 026Ah | | | | | | |
| 026Bh | | | | | | |
| 026Ch | CAN1 Message Box 0:Time Stamp | | 216 | | | |
| 026Dh | | | | | | |
| 026Eh | | | | | | |
| 026Fh | | | | | | |
| 0270h | | | | CAN1 Message Box 1: Identifier / DLC | | 217 |
| 0271h | | | | | | |
| 0272h | | | | | | |
| 0273h | | | | | | |
| 0274h | | | | | | |
| 0275h | | | | | | |
| 0276h | CAN1 Message Box 1: Data Field | | | | | |
| 0277h | | | | | | |
| 0278h | | | | | | |
| 0279h | | | | | | |
| 027Ah | | | | | | |
| 027Bh | | | | | | |
| 027Ch | CAN1 Message Box 1:Time Stamp | | | | | |
| 027Dh | | | | | | |
| 027Eh | | | | | | |
| 027Fh | | | | | | |

Blank spaces are reserved. No access is allowed.

| Address | Register | Symbol | Page |
|---------|--------------------------------------|--------|------|
| 0280h | CAN1 Message Box 2: Identifier / DLC | | |
| 0281h | | | |
| 0282h | | | |
| 0283h | | | |
| 0284h | | | |
| 0285h | CAN1 Message Box 2: Data Field | | |
| 0286h | | | |
| 0287h | | | |
| 0288h | | | |
| 0289h | | | |
| 028Ah | | | |
| 028Bh | CAN1 Message Box 2: Time Stamp | | |
| 028Ch | | | |
| 028Dh | CAN1 Message Box 3: Identifier / DLC | | |
| 028Eh | | | |
| 028Fh | | | |
| 0290h | | | |
| 0291h | | | |
| 0292h | | | |
| 0293h | CAN1 Message Box 3: Data Field | | |
| 0294h | | | |
| 0295h | | | |
| 0296h | | | |
| 0297h | | | |
| 0298h | | | |
| 0299h | CAN1 Message Box 3: Time Stamp | | |
| 029Ah | | | |
| 029Bh | CAN1 Message Box 4: Identifier / DLC | | |
| 029Ch | | | |
| 029Dh | | | |
| 029Eh | | | |
| 029Fh | | | |
| 02A0h | CAN1 Message Box 4: Data Field | | |
| 02A1h | | | |
| 02A2h | | | |
| 02A3h | | | |
| 02A4h | | | |
| 02A5h | | | |
| 02A6h | CAN1 Message Box 4: Time Stamp | | |
| 02A7h | | | |
| 02A8h | | | |
| 02A9h | | | |
| 02AAh | | | |
| 02ABh | CAN1 Message Box 5: Identifier / DLC | | |
| 02ACh | | | |
| 02ADh | | | |
| 02AEh | | | |
| 02AFh | | | |
| 02B0h | CAN1 Message Box 5: Data Field | | |
| 02B1h | | | |
| 02B2h | | | |
| 02B3h | | | |
| 02B4h | | | |
| 02B5h | | | |
| 02B6h | CAN1 Message Box 5: Time Stamp | | |
| 02B7h | | | |
| 02B8h | | | |
| 02B9h | | | |
| 02BAh | | | |
| 02BBh | | | |
| 02BCh | | | |
| 02BDh | | | |
| 02BEh | | | |
| 02BFh | | | |

216
217

| Address | Register | Symbol | Page |
|---------|--------------------------------------|--------|------|
| 02C0h | CAN1 Message Box 6: Identifier / DLC | | |
| 02C1h | | | |
| 02C2h | | | |
| 02C3h | | | |
| 02C4h | | | |
| 02C5h | CAN1 Message Box 6: Data Field | | |
| 02C6h | | | |
| 02C7h | | | |
| 02C8h | | | |
| 02C9h | | | |
| 02CAh | | | |
| 02CBh | CAN1 Message Box 6: Time Stamp | | |
| 02CCh | | | |
| 02CDh | CAN1 Message Box 7: Identifier / DLC | | |
| 02CEh | | | |
| 02CFh | | | |
| 02D0h | | | |
| 02D1h | | | |
| 02D2h | | | |
| 02D3h | CAN1 Message Box 7: Data Field | | |
| 02D4h | | | |
| 02D5h | | | |
| 02D6h | | | |
| 02D7h | | | |
| 02D8h | | | |
| 02D9h | CAN1 Message Box 7: Time Stamp | | |
| 02DAh | | | |
| 02DBh | CAN1 Message Box 8: Identifier / DLC | | |
| 02DCh | | | |
| 02DDh | | | |
| 02DEh | | | |
| 02DFh | | | |
| 02E0h | CAN1 Message Box 8: Data Field | | |
| 02E1h | | | |
| 02E2h | | | |
| 02E3h | | | |
| 02E4h | | | |
| 02E5h | | | |
| 02E6h | CAN1 Message Box 8: Time Stamp | | |
| 02E7h | | | |
| 02E8h | | | |
| 02E9h | | | |
| 02EAh | | | |
| 02EBh | CAN1 Message Box 9: Identifier / DLC | | |
| 02ECh | | | |
| 02EDh | | | |
| 02EEh | | | |
| 02EFh | | | |
| 02F0h | CAN1 Message Box 9: Data Field | | |
| 02F1h | | | |
| 02F2h | | | |
| 02F3h | | | |
| 02F4h | | | |
| 02F5h | | | |
| 02F6h | CAN1 Message Box 9: Time Stamp | | |
| 02F7h | | | |
| 02F8h | | | |
| 02F9h | | | |
| 02FAh | | | |
| 02FBh | | | |
| 02FCh | | | |
| 02FDh | | | |
| 02FEh | | | |
| 02FFh | | | |

216
217

| Address | Register | Symbol | Page |
|---------|---------------------------------------|--------|------|
| 0300h | CAN1 Message Box 10: Identifier / DLC | | |
| 0301h | | | |
| 0302h | | | |
| 0303h | | | |
| 0304h | | | |
| 0305h | | | |
| 0306h | CAN1 Message Box 10: Data Field | | |
| 0307h | | | |
| 0308h | | | |
| 0309h | | | |
| 030Ah | | | |
| 030Bh | | | |
| 030Ch | | | |
| 030Dh | | | |
| 030Eh | CAN1 Message Box 10: Time Stamp | | |
| 030Fh | | | |
| 0310h | CAN1 Message Box 11: Identifier / DLC | | |
| 0311h | | | |
| 0312h | | | |
| 0313h | | | |
| 0314h | | | |
| 0315h | | | |
| 0316h | CAN1 Message Box 11: Data Field | | |
| 0317h | | | |
| 0318h | | | |
| 0319h | | | |
| 031Ah | | | |
| 031Bh | | | |
| 031Ch | | | |
| 031Dh | | | |
| 031Eh | CAN1 Message Box 11: Time Stamp | | |
| 031Fh | | | |
| 0320h | CAN1 Message Box 12: Identifier / DLC | | |
| 0321h | | | |
| 0322h | | | |
| 0323h | | | |
| 0324h | | | |
| 0325h | | | |
| 0326h | CAN1 Message Box 12: Data Field | | |
| 0327h | | | |
| 0328h | | | |
| 0329h | | | |
| 032Ah | | | |
| 032Bh | | | |
| 032Ch | | | |
| 032Dh | | | |
| 032Eh | CAN1 Message Box 12: Time Stamp | | |
| 032Fh | | | |
| 0330h | CAN1 Message Box 13: Identifier / DLC | | |
| 0331h | | | |
| 0332h | | | |
| 0333h | | | |
| 0334h | | | |
| 0335h | | | |
| 0336h | CAN1 Message Box 13: Data Field | | |
| 0337h | | | |
| 0338h | | | |
| 0339h | | | |
| 033Ah | | | |
| 033Bh | | | |
| 033Ch | | | |
| 033Dh | | | |
| 033Eh | CAN1 Message Box 13: Time Stamp | | |
| 033Fh | | | |

216
217

| Address | Register | Symbol | Page |
|---------|---------------------------------------|--------|------|
| 0340h | CAN1 Message Box 14: Identifier / DLC | | |
| 0341h | | | |
| 0342h | | | |
| 0343h | | | |
| 0344h | | | |
| 0345h | | | |
| 0346h | CAN1 Message Box 14: Data Field | | |
| 0347h | | | |
| 0348h | | | |
| 0349h | | | |
| 034Ah | | | |
| 034Bh | | | |
| 034Ch | | | |
| 034Dh | | | |
| 034Eh | CAN1 Message Box 14: Time Stamp | | |
| 034Fh | | | |
| 0350h | CAN1 Message Box 15: Identifier / DLC | | |
| 0351h | | | |
| 0352h | | | |
| 0353h | | | |
| 0354h | | | |
| 0355h | | | |
| 0356h | CAN1 Message Box 15: Data Field | | |
| 0357h | | | |
| 0358h | | | |
| 0359h | | | |
| 035Ah | | | |
| 035Bh | | | |
| 035Ch | | | |
| 035Dh | | | |
| 035Eh | CAN1 Message Box 15: Time Stamp | | |
| 035Fh | | | |
| 0360h | CAN1 Global Mask Register | C1GMR | 218 |
| 0361h | | | |
| 0362h | | | |
| 0363h | | | |
| 0364h | | | |
| 0365h | | | |
| 0366h | CAN1 Local Mask A Register | C1LMAR | 218 |
| 0367h | | | |
| 0368h | | | |
| 0369h | | | |
| 036Ah | | | |
| 036Bh | | | |
| 036Ch | CAN1 Local Mask B Register | C1LMBR | 218 |
| 036Dh | | | |
| 036Eh | | | |
| 036Fh | | | |
| 0370h | | | |
| 0371h | | | |
| 0372h | | | |
| 0373h | | | |
| 0374h | | | |
| 0375h | | | |
| 0376h | | | |
| 0377h | | | |
| 0378h | | | |
| 0379h | | | |
| 037Ah | | | |
| 037Bh | | | |
| 037Ch | | | |
| 037Dh | | | |
| 037Eh | | | |
| 037Fh | | | |

216
217

Blank spaces are reserved. No access is allowed.

| Address | Register | Symbol | Page |
|---------|---|--------|-------------|
| 0380h | Count Start Flag | TABSR | 109,124,137 |
| 0381h | Clock Prescaler Reset Flag | CPSRF | 110,124 |
| 0382h | One-Shot Start Flag | ONSF | 110 |
| 0383h | Trigger Select Register | TRGSR | 110,137 |
| 0384h | Up/Down Flag | UDF | 109 |
| 0385h | | | |
| 0386h | Timer A0 Register | TA0 | 108 |
| 0387h | | | |
| 0388h | Timer A1 Register | TA1 | 108 |
| 0389h | | | 135 |
| 038Ah | Timer A2 Register | TA2 | 108 |
| 038Bh | | | 135 |
| 038Ch | Timer A3 Register | TA3 | 108 |
| 038Dh | | | |
| 038Eh | Timer A4 Register | TA4 | 108 |
| 038Fh | | | 135 |
| 0390h | Timer B0 Register | TB0 | 123 |
| 0391h | | | |
| 0392h | Timer B1 Register | TB1 | 123 |
| 0393h | | | |
| 0394h | Timer B2 Register | TB2 | 123 |
| 0395h | | | 135 |
| 0396h | Timer A0 Mode Register | TA0MR | 108 |
| 0397h | Timer A1 Mode Register | TA1MR | 111 |
| 0398h | Timer A2 Mode Register | TA2MR | 113 |
| 0399h | Timer A3 Mode Register | TA3MR | 118 |
| 039Ah | Timer A4 Mode Register | TA4MR | 120 |
| 039Bh | Timer B0 Mode Register | TB0MR | 123,125 |
| 039Ch | Timer B1 Mode Register | TB1MR | 126,128 |
| 039Dh | Timer B2 Mode Register | TB2MR | 138 |
| 039Eh | Timer B2 Special Mode Register | TB2SC | 136 |
| 039Fh | | | |
| 03A0h | UART0 Transmit/Receive Mode Register | U0MR | 146 |
| 03A1h | UART0 Bit Rate Register | U0BRG | 145 |
| 03A2h | UART0 Transmit Buffer Register | U0TB | 145 |
| 03A3h | | | |
| 03A4h | UART0 Transmit/Receive Control Register 0 | U0C0 | 146 |
| 03A5h | UART0 Transmit/Receive Control Register 1 | U0C1 | 147 |
| 03A6h | UART0 Receive Buffer Register | U0RB | 145 |
| 03A7h | | | |
| 03A8h | UART1 Transmit/Receive Mode Register | U1MR | 146 |
| 03A9h | UART1 Bit Rate Register | U1BRG | 145 |
| 03AAh | UART1 Transmit Buffer Register | U1TB | 145 |
| 03ABh | | | |
| 03ACh | UART1 Transmit/Receive Control Register 0 | U1C0 | 146 |
| 03ADh | UART1 Transmit/Receive Control Register 1 | U1C1 | 147 |
| 03AEh | UART1 Receive Buffer Register | U1RB | 145 |
| 03AFh | | | |
| 03B0h | UART Transmit/Receive Control Register 2 | UCON | 148 |
| 03B1h | | | |
| 03B2h | | | |
| 03B3h | | | |
| 03B4h | | | |
| 03B5h | | | |
| 03B6h | | | |
| 03B7h | | | |
| 03B8h | DMA0 Request Source Select Register | DM0SL | 97 |
| 03B9h | | | |
| 03BAh | DMA1 Request Source Select Register | DM1SL | 98 |
| 03BBh | | | |
| 03BCh | CRC Data Register | CRCD | 212 |
| 03BDh | | | |
| 03BEh | CRC Input Register | CRCIN | 212 |
| 03BFh | | | |

| Address | Register | Symbol | Page |
|---------|-----------------------------|--------|-------------|
| 03C0h | A/D Register 0 | AD0 | 196 |
| 03C1h | | | |
| 03C2h | A/D Register 1 | AD1 | |
| 03C3h | | | |
| 03C4h | A/D Register 2 | AD2 | |
| 03C5h | | | |
| 03C6h | A/D Register 3 | AD3 | |
| 03C7h | | | |
| 03C8h | A/D Register 4 | AD4 | |
| 03C9h | | | |
| 03CAh | A/D Register 5 | AD5 | |
| 03CBh | | | |
| 03CCh | A/D Register 6 | AD6 | |
| 03CDh | | | |
| 03CEh | A/D Register 7 | AD7 | |
| 03CFh | | | |
| 03D0h | | | |
| 03D1h | | | |
| 03D2h | | | |
| 03D3h | | | |
| 03D4h | A/D Control Register 2 | ADCON2 | 196 |
| 03D5h | | | |
| 03D6h | A/D Control Register 0 | ADCON0 | 195,198,200 |
| 03D7h | A/D Control Register 1 | ADCON1 | 202,204,206 |
| 03D8h | D/A Register 0 | DA0 | 211 |
| 03D9h | | | |
| 03DAh | D/A Register 1 | DA1 | 211 |
| 03DBh | | | |
| 03DCh | D/A Control Register | DACON | 211 |
| 03DDh | | | |
| 03DEh | | | |
| 03DFh | | | |
| 03E0h | Port P0 Register | P0 | 246 |
| 03E1h | Port P1 Register | P1 | 246 |
| 03E2h | Port P0 Direction Register | PD0 | 245 |
| 03E3h | Port P1 Direction Register | PD1 | 245 |
| 03E4h | Port P2 Register | P2 | 246 |
| 03E5h | Port P3 Register | P3 | 246 |
| 03E6h | Port P2 Direction Register | PD2 | 245 |
| 03E7h | Port P3 Direction Register | PD3 | 245 |
| 03E8h | Port P4 Register | P4 | 246 |
| 03E9h | Port P5 Register | P5 | 246 |
| 03EAh | Port P4 Direction Register | PD4 | 245 |
| 03EBh | Port P5 Direction Register | PD5 | 245 |
| 03ECh | Port P6 Register | P6 | 246 |
| 03EDh | Port P7 Register | P7 | 246 |
| 03EEh | Port P6 Direction Register | PD6 | 245 |
| 03EFh | Port P7 Direction Register | PD7 | 245 |
| 03F0h | Port P8 Register | P8 | 246 |
| 03F1h | Port P9 Register | P9 | 246 |
| 03F2h | Port P8 Direction Register | PD8 | 245 |
| 03F3h | Port P9 Direction Register | PD9 | 245 |
| 03F4h | Port P10 Register | P10 | 246 |
| 03F5h | | | |
| 03F6h | Port P10 Direction Register | PD10 | 245 |
| 03F7h | | | |
| 03F8h | | | |
| 03F9h | | | |
| 03FAh | | | |
| 03FBh | | | |
| 03FCh | Pull-up Control Register 0 | PUR0 | 247 |
| 03FDh | Pull-up Control Register 1 | PUR1 | 247 |
| 03FEh | Pull-up Control Register 2 | PUR2 | 247 |
| 03FFh | Port Control Register | PCR | 248 |

Blank spaces are reserved. No access is allowed.

M16C/6N Group (M16C/6N4)

Renesas MCU

1. Overview

The M16C/6N Group (M16C/6N4) of MCUs are built using the high-performance silicon gate CMOS process using the M16C/60 Series CPU core and are packaged in 100-pin plastic molded QFP and LQFP. These MCUs operate using sophisticated instructions featuring a high level of instruction efficiency. With 1 Mbyte of address space, they are capable of executing instructions at high speed. Being equipped with two CAN (Controller Area Network) modules in the M16C/6N Group (M16C/6N4), the MCU is suited to drive automotive and industrial control systems. The CAN modules comply with the 2.0B specification. In addition, this MCU contains a multiplier and DMAC which combined with fast instruction processing capability, makes it suitable for control of various OA, communication, and industrial equipment which requires high-speed arithmetic/logic operations.

1.1 Applications

- Automotive, industrial control systems and other automobile, other (T/V-ver. product)
- Car audio and industrial control systems, other (Normal-ver. product)

1.2 Performance Overview

Table 1.1 lists the Functions and Specifications for M16C/6N Group (M16C/6N4).

Table 1.1 Functions and Specifications for M16C/6N Group (M16C/6N4)

| Item | | Specification | | |
|-------------------------------|------------------------------------|--|---|---|
| | | Normal-ver. | T/V-ver. | |
| CPU | Number of fundamental instructions | 91 instructions | | |
| | Minimum instruction execution time | 41.7 ns (f(BCLK) = 24 MHz, 1/1 prescaler, without software wait) | 50.0 ns (f(BCLK) = 20 MHz, 1/1 prescaler, without software wait) | |
| | Operating mode | Single-chip, memory expansion, and microprocessor modes | | |
| | Address space | 1 Mbyte | | |
| | Memory capacity | Refer to Table 1.2 Product Information | | |
| Peripheral Function | Ports | Input/Output: 87 pins, Input: 1 pin | | |
| | Multifunction timers | Timer A: 16 bits × 5 channels Timer B: 16 bits × 6 channels Three-phase motor control circuit | | |
| | Serial interfaces | 3 channels Clock synchronous, UART, I ² C-bus ⁽¹⁾ , IEBus ⁽²⁾ 1 channel Clock synchronous | | |
| | A/D converter | 10-bit A/D converter: 1 circuit, 26 channels | | |
| | D/A converter | 8 bits × 2 channels | | |
| | DMAC | 2 channels | | |
| | CRC calculation circuit | CRC-CCITT | | |
| | CAN module | 2 channels with 2.0B specification | | |
| | Watchdog timer | 15 bits × 1 channel (with prescaler) | | |
| | Interrupts | Internal: 31 sources, External: 9 sources Software: 4 sources, Priority levels: 7 levels | | |
| | Clock generation circuits | 4 circuits • Main clock oscillation circuit (*) • Sub clock oscillation circuit (*) • On-chip oscillator • PLL frequency synthesizer (*) Equipped with on-chip feedback resistor | | |
| | Oscillation-stopped detector | Main clock oscillation stop and re-oscillation detection function | | |
| | Electrical Characteristics | Supply voltage | VCC = 3.0 to 5.5 V (f(BCLK) = 24 MHz, 1/1 prescaler, without software wait) | VCC = 4.2 to 5.5 V (f(BCLK) = 20 MHz, 1/1 prescaler, without software wait) |
| Consumption current | | Mask ROM | 20 mA (f(BCLK) = 24 MHz, PLL operation, no division) | 18 mA (f(BCLK) = 20 MHz, PLL operation, no division) |
| | | Flash memory | 22 mA (f(BCLK) = 24 MHz, PLL operation, no division) | 20 mA (f(BCLK) = 20 MHz, PLL operation, no division) |
| | | Mask ROM Flash memory | 3 μA (f(BCLK) = 32 kHz, Wait mode, Oscillation capacity Low) 0.8 μA (Stop mode, Topr = 25°C) | |
| Flash Memory Version | Programming and erasure voltage | 3.0 ± 0.3 V or 5.0 ± 0.5 V | 5.0 ± 0.5 V | |
| | Programming and erasure endurance | 100 times | | |
| I/O Characteristics | I/O withstand voltage | 5.0 V | | |
| | Output current | 5 mA | | |
| Operating Ambient Temperature | -40 to 85°C | T version: -40 to 85°C V version: -40 to 125°C (option) | | |
| Device Configuration | CMOS high-performance silicon gate | | | |
| Package | 100-pin molded-plastic QFP, LQFP | | | |

NOTES:

1. I²C-bus is a trademark of Koninklijke Philips Electronics N.V.
2. IEBus is a trademark of NEC Electronics Corporation.

option: All options are on request basis.

1.3 Block Diagram

Figure 1.1 shows a Block Diagram.

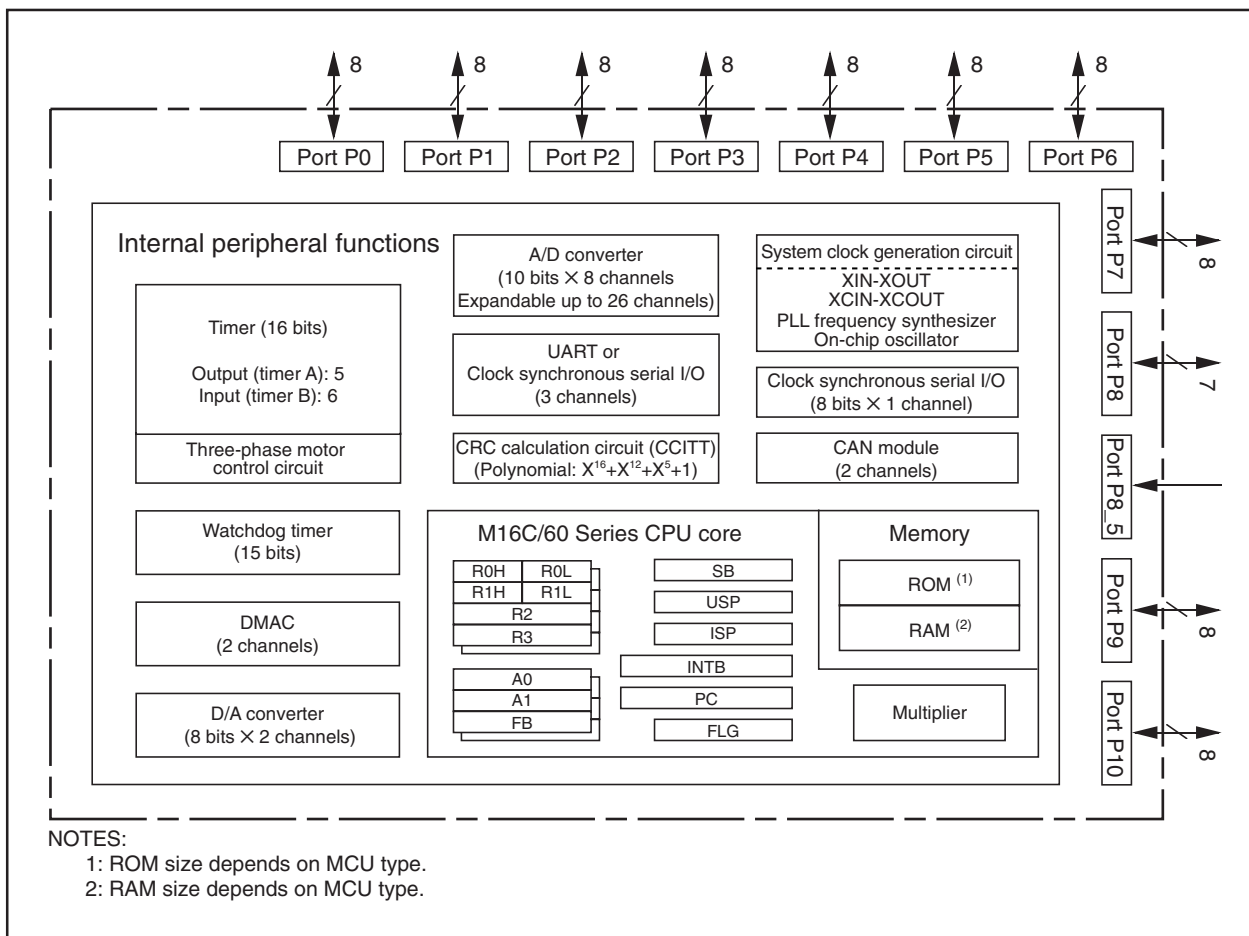


Figure 1.1 Block Diagram

1.4 Product Information

Table 1.2 lists the Product Information and Figure 1.2 shows the Type Number, Memory Size, and Packages.

Table 1.2 Product Information

As of Apr. 2006

| Type No. | ROM Capacity | RAM Capacity | Package Type ⁽²⁾ | Remarks | | |
|-----------------|------------------|--------------|-----------------------------|-------------------------------------|-------------|-------------|
| M306N4FCFP | 128 K + 4 Kbytes | 5 Kbytes | PRQP0100JB-A | Flash memory version ⁽¹⁾ | Normal-ver. | |
| M306N4FCGP | | | PLQP0100KB-A | | | |
| M306N4FGFP | 256 K + 4 Kbytes | 10 Kbytes | PRQP0100JB-A | | | |
| M306N4FGGP | | | PLQP0100KB-A | | | |
| M306N4FCTFP | 128 K + 4 Kbytes | 5 Kbytes | PRQP0100JB-A | | | T-ver. |
| M306N4FCTGP | | | PLQP0100KB-A | | | |
| M306N4FGTFP | 256 K + 4 Kbytes | 10 Kbytes | PRQP0100JB-A | | | |
| M306N4FGTGP | | | PLQP0100KB-A | | | |
| M306N4FCVFP | 128 K + 4 Kbytes | 5 Kbytes | PRQP0100JB-A | | V-ver. | |
| M306N4FCVGP | | | PLQP0100KB-A | | | |
| M306N4FGVFP | 256 K + 4 Kbytes | 10 Kbytes | PRQP0100JB-A | | | |
| M306N4FGVGP | | | PLQP0100KB-A | | | |
| M306N4MC-XXXGP | 128 Kbytes | 5 Kbytes | PLQP0100KB-A | Mask ROM version | | Normal-ver. |
| M306N4MG-XXXGP | 256 Kbytes | 10 Kbytes | PLQP0100KB-A | | | |
| M306N4MCT-XXXFP | 128 Kbytes | 5 Kbytes | PRQP0100JB-A | | T-ver. | |
| M306N4MCT-XXXGP | | | PLQP0100KB-A | | | |
| M306N4MGT-XXXFP | 256 Kbytes | 10 Kbytes | PRQP0100JB-A | | | |
| M306N4MGT-XXXGP | | | PLQP0100KB-A | | | |
| M306N4MCV-XXXFP | 128 Kbytes | 5 Kbytes | PRQP0100JB-A | | | V-ver. |
| M306N4MCV-XXXGP | | | PLQP0100KB-A | | | |
| M306N4MGV-XXXFP | 256 Kbytes | 10 Kbytes | PRQP0100JB-A | | | |
| M306N4MGV-XXXGP | | | PLQP0100KB-A | | | |

NOTES:

- Data flash memory provides an additional 4 Kbytes of ROM capacity (block A).
- The correspondence between new and old package types is as follows.
 PRQP0100JB-A: 100P6S-A
 PLQP0100KB-A: 100P6Q-A

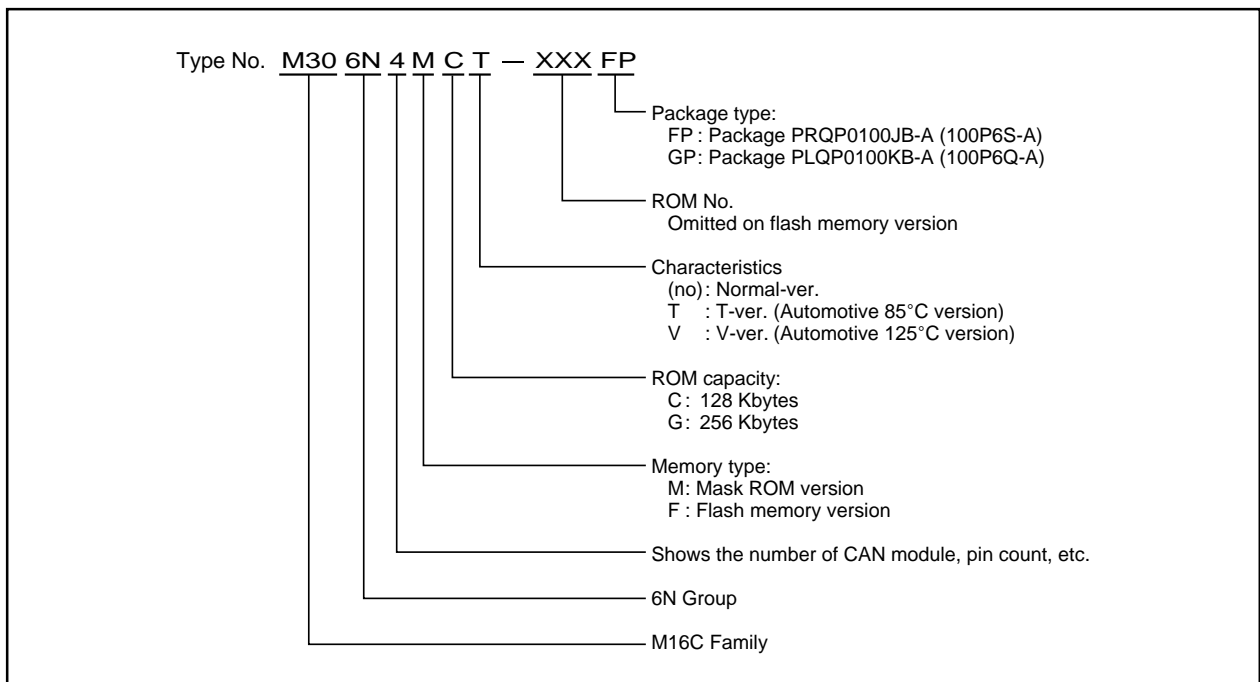


Figure 1.2 Type Number, Memory Size, and Package

1.5 Pin Assignments

Figures 1.3 and 1.4 show the Pin Assignment (Top View). Tables 1.3 and 1.4 list the List of Pin Names.

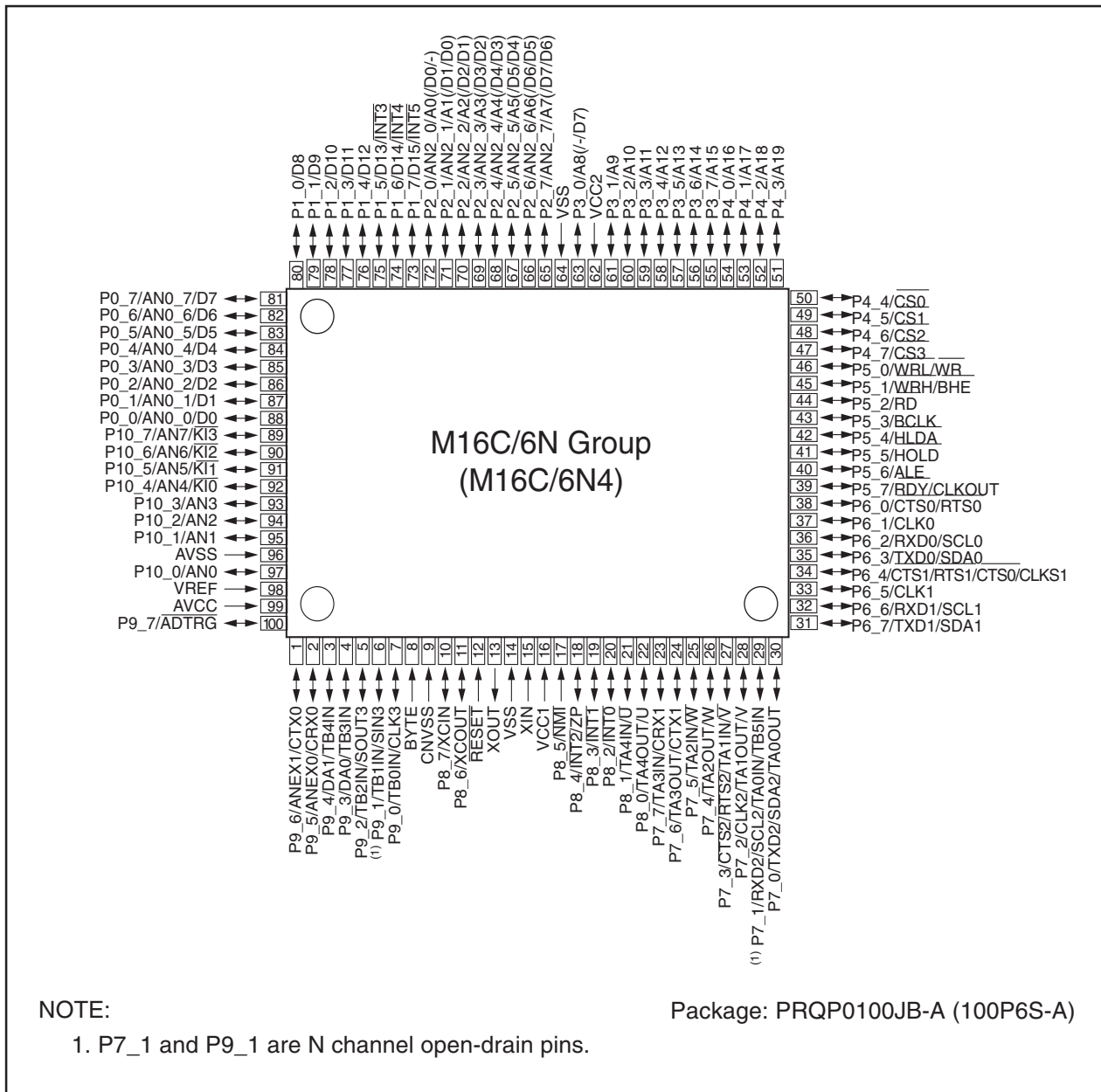


Figure 1.3 Pin Assignments (Top View) (1)

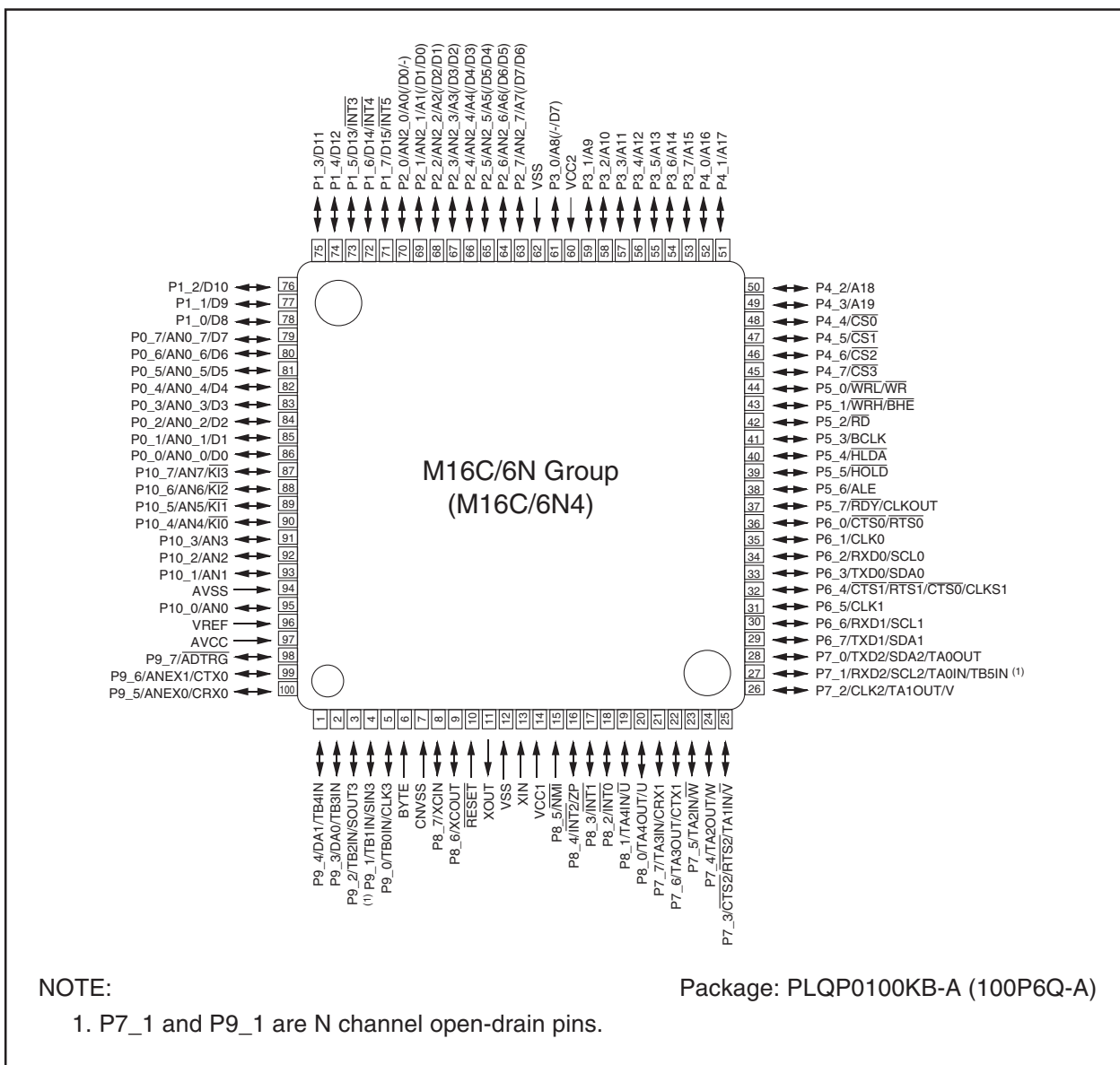


Figure 1.4 Pin Assignments (Top View) (2)

Table 1.3 List of Pin Names (1)

| Pin No. | Control Pin | Port | Interrupt Pin | Timer Pin | UART Pin | Analog Pin | CAN Module Pin | Bus Control Pin |
|---------|-------------|-------|---------------|-------------------|----------------------|------------|----------------|---------------------|
| FP | GP | | | | | | | |
| 1 | 99 | | P9_6 | | | ANEX1 | CTX0 | |
| 2 | 100 | | P9_5 | | | ANEX0 | CRX0 | |
| 3 | 1 | | P9_4 | TB4IN | | DA1 | | |
| 4 | 2 | | P9_3 | TB3IN | | DA0 | | |
| 5 | 3 | | P9_2 | TB2IN | SOUT3 | | | |
| 6 | 4 | | P9_1 | TB1IN | SIN3 | | | |
| 7 | 5 | | P9_0 | TB0IN | CLK3 | | | |
| 8 | 6 | BYTE | | | | | | |
| 9 | 7 | CNVSS | | | | | | |
| 10 | 8 | XCIN | P8_7 | | | | | |
| 11 | 9 | XCOUT | P8_6 | | | | | |
| 12 | 10 | RESET | | | | | | |
| 13 | 11 | XOUT | | | | | | |
| 14 | 12 | VSS | | | | | | |
| 15 | 13 | XIN | | | | | | |
| 16 | 14 | VCC1 | | | | | | |
| 17 | 15 | | P8_5 | NMI | | | | |
| 18 | 16 | | P8_4 | INT2 | ZP | | | |
| 19 | 17 | | P8_3 | INT1 | | | | |
| 20 | 18 | | P8_2 | INT0 | | | | |
| 21 | 19 | | P8_1 | TA4IN/ \bar{U} | | | | |
| 22 | 20 | | P8_0 | TA4OUT/U | | | | |
| 23 | 21 | | P7_7 | TA3IN | | | CRX1 | |
| 24 | 22 | | P7_6 | TA3OUT | | | CTX1 | |
| 25 | 23 | | P7_5 | TA2IN/ \bar{W} | | | | |
| 26 | 24 | | P7_4 | TA2OUT/ \bar{W} | | | | |
| 27 | 25 | | P7_3 | TA1IN/ \bar{V} | CTS2/RTS2 | | | |
| 28 | 26 | | P7_2 | TA1OUT/ \bar{V} | CLK2 | | | |
| 29 | 27 | | P7_1 | TA0IN/TB5IN | RXD2/SCL2 | | | |
| 30 | 28 | | P7_0 | TA0OUT | TXD2/SDA2 | | | |
| 31 | 29 | | P6_7 | | TXD1/SDA1 | | | |
| 32 | 30 | | P6_6 | | RXD1/SCL1 | | | |
| 33 | 31 | | P6_5 | | CLK1 | | | |
| 34 | 32 | | P6_4 | | CTS1/RTS1/CTS0/CLKS1 | | | |
| 35 | 33 | | P6_3 | | TXD0/SDA0 | | | |
| 36 | 34 | | P6_2 | | RXD0/SCL0 | | | |
| 37 | 35 | | P6_1 | | CLK0 | | | |
| 38 | 36 | | P6_0 | | CTS0/RTS0 | | | |
| 39 | 37 | | P5_7 | | | | | \bar{RDY} /CLKOUT |
| 40 | 38 | | P5_6 | | | | | ALE |
| 41 | 39 | | P5_5 | | | | | HOLD |
| 42 | 40 | | P5_4 | | | | | HLDA |
| 43 | 41 | | P5_3 | | | | | BCLK |
| 44 | 42 | | P5_2 | | | | | \bar{RD} |
| 45 | 43 | | P5_1 | | | | | WRH/BHE |
| 46 | 44 | | P5_0 | | | | | WRL/WR |
| 47 | 45 | | P4_7 | | | | | CS3 |
| 48 | 46 | | P4_6 | | | | | CS2 |
| 49 | 47 | | P4_5 | | | | | CS1 |
| 50 | 48 | | P4_4 | | | | | CS0 |

FP: PRQP0100JB-A (100P6S-A), GP: PLQP0100KB-A (100P6Q-A)

Table 1.4 List of Pin Names (2)

| Pin No. | | Control Pin | Port | Interrupt Pin | Timer Pin | UART Pin | Analog Pin | CAN Module Pin | Bus Control Pin |
|---------|----|-------------|-------|---------------|-----------|----------|------------|----------------|-----------------|
| FP | GP | | | | | | | | |
| 51 | 49 | | P4_3 | | | | | | A19 |
| 52 | 50 | | P4_2 | | | | | | A18 |
| 53 | 51 | | P4_1 | | | | | | A17 |
| 54 | 52 | | P4_0 | | | | | | A16 |
| 55 | 53 | | P3_7 | | | | | | A15 |
| 56 | 54 | | P3_6 | | | | | | A14 |
| 57 | 55 | | P3_5 | | | | | | A13 |
| 58 | 56 | | P3_4 | | | | | | A12 |
| 59 | 57 | | P3_3 | | | | | | A11 |
| 60 | 58 | | P3_2 | | | | | | A10 |
| 61 | 59 | | P3_1 | | | | | | A9 |
| 62 | 60 | VCC2 | | | | | | | |
| 63 | 61 | | P3_0 | | | | | | A8(/-/D7) |
| 64 | 62 | VSS | | | | | | | |
| 65 | 63 | | P2_7 | | | | AN2_7 | | A7(/D7/D6) |
| 66 | 64 | | P2_6 | | | | AN2_6 | | A6(/D6/D5) |
| 67 | 65 | | P2_5 | | | | AN2_5 | | A5(/D5/D4) |
| 68 | 66 | | P2_4 | | | | AN2_4 | | A4(/D4/D3) |
| 69 | 67 | | P2_3 | | | | AN2_3 | | A3(/D3/D2) |
| 70 | 68 | | P2_2 | | | | AN2_2 | | A2(/D2/D1) |
| 71 | 69 | | P2_1 | | | | AN2_1 | | A1(/D1/D0) |
| 72 | 70 | | P2_0 | | | | AN2_0 | | A0(/D0/-) |
| 73 | 71 | | P1_7 | INT5 | | | | | D15 |
| 74 | 72 | | P1_6 | INT4 | | | | | D14 |
| 75 | 73 | | P1_5 | INT3 | | | | | D13 |
| 76 | 74 | | P1_4 | | | | | | D12 |
| 77 | 75 | | P1_3 | | | | | | D11 |
| 78 | 76 | | P1_2 | | | | | | D10 |
| 79 | 77 | | P1_1 | | | | | | D9 |
| 80 | 78 | | P1_0 | | | | | | D8 |
| 81 | 79 | | P0_7 | | | | AN0_7 | | D7 |
| 82 | 80 | | P0_6 | | | | AN0_6 | | D6 |
| 83 | 81 | | P0_5 | | | | AN0_5 | | D5 |
| 84 | 82 | | P0_4 | | | | AN0_4 | | D4 |
| 85 | 83 | | P0_3 | | | | AN0_3 | | D3 |
| 86 | 84 | | P0_2 | | | | AN0_2 | | D2 |
| 87 | 85 | | P0_1 | | | | AN0_1 | | D1 |
| 88 | 86 | | P0_0 | | | | AN0_0 | | D0 |
| 89 | 87 | | P10_7 | KI3 | | | AN7 | | |
| 90 | 88 | | P10_6 | KI2 | | | AN6 | | |
| 91 | 89 | | P10_5 | KI1 | | | AN5 | | |
| 92 | 90 | | P10_4 | KI0 | | | AN4 | | |
| 93 | 91 | | P10_3 | | | | AN3 | | |
| 94 | 92 | | P10_2 | | | | AN2 | | |
| 95 | 93 | | P10_1 | | | | AN1 | | |
| 96 | 94 | AVSS | | | | | | | |
| 97 | 95 | | P10_0 | | | | AN0 | | |
| 98 | 96 | VREF | | | | | | | |
| 99 | 97 | AVCC | | | | | | | |
| 100 | 98 | | P9_7 | | | | ADTRG | | |

FP: PRQP0100JB-A (100P6S-A), GP: PLQP0100KB-A (100P6Q-A)

1.6 Pin Functions

Tables 1.5 to 1.7 list the Pin Functions.

Table 1.5 Pin Functions (1)

| Signal Name | Pin Name | I/O Type | Description |
|--------------------------------------|-------------------------|--|---|
| Power supply input | VCC1, VCC2, VSS | I | Apply 4.2 to 5.5 V (T/V-ver.), 3.0 to 5.5 V (Normal-ver.) to the VCC1 and VCC2 pins and 0 V to the VSS pin. The VCC apply condition is that VCC2 = VCC1 ⁽¹⁾ . |
| Analog power supply input | AVCC, AVSS | I | Applies the power supply for the A/D converter. Connect the AVCC pin to VCC1. Connect the AVSS pin to VSS. |
| Reset input | RESET | I | The MCU is in a reset state when applying "L" to the this pin. |
| CNVSS | CNVSS | I | Switches processor mode. Connect this pin to VSS to when after a reset to start up in single-chip mode. Connect this pin to VCC1 to start up in microprocessor mode. |
| External data bus width select input | BYTE | I | Switches the data bus in external memory space. The data bus is 16-bit long when the this pin is held "L" and 8-bit long when the this pin is held "H". Set it to either one. Connect this pin to VSS when single-chip mode. |
| Bus control pins | D0 to D7 | I/O | Inputs and outputs data (D0 to D7) when these pins are set as the separate bus. |
| | D8 to D15 | I/O | Inputs and outputs data (D8 to D15) when external 16-bit data bus is set as the separate bus. |
| | A0 to A19 | O | Output address bits (A0 to A19). |
| | A0/D0 to A7/D7 | I/O | Input and output data (D0 to D7) and output address bits (A0 to A7) by time-sharing when external 8-bit data bus are set as the multiplexed bus. |
| | A1/D0 to A8/D7 | I/O | Input and output data (D0 to D7) and output address bits (A1 to A8) by time-sharing when external 16-bit data bus are set as the multiplexed bus. |
| | CS0 to CS3 | O | Output CS0 to CS3 signals. CS0 to CS3 are chip-select signals to specify an external space. |
| | WRL/WR WRH/BHE RD | O | Output WRL, WRH, (WR, BHE), RD signals. WRL and WRH or BHE, and WR can be switched by program. <ul style="list-style-type: none"> • WRL, WRH, and RD are selected The WRL signal becomes "L" by writing data to an even address in an external memory space. The WRH signal becomes "L" by writing data to an odd address in an external memory space. The RD pin signal becomes "L" by reading data in an external memory space. • WR, BHE, and RD are selected The WR signal becomes "L" by writing data in an external memory space. The RD signal becomes "L" by reading data in an external memory space. The BHE signal becomes "L" by accessing an odd address. Select WR, BHE, and RD for an external 8-bit data bus. |
| | ALE | O | ALE is a signal to latch the address. |
| | HOLD | I | While the HOLD pin is held "L", the MCU is placed in a hold state. |
| | HLDA | O | In a hold state, HLDA outputs a "L" signal. |
| RDY | I | While applying a "L" signal to the RDY pin, the MCU is placed in a wait state. | |

I: Input O: Output I/O: Input/Output

NOTE:

1. In this manual, hereafter, VCC refers to VCC1 unless otherwise noted.

Table 1.6 Pin Functions (2)

| Signal Name | Pin Name | I/O Type | Description |
|----------------------------------|--|----------|--|
| 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 ⁽¹⁾ . |
| Main clock output | XOUT | O | To use the external clock, input the clock from XIN and leave XOUT open. |
| Sub clock input | XCIN | I | I/O pins for a sub clock oscillation circuit. Connect a crystal oscillator between XCIN and XCOU ⁽¹⁾ . |
| Sub clock output | XCOU | O | To use the external clock, input the clock from XCIN and leave XCOU open. |
| BCLK output | BCLK | O | Outputs the BCLK signal. |
| Clock output | CLKOUT | O | The clock of the same cycle as f _C , f ₈ , or f ₃₂ is output. |
| INT interrupt input | INT0 to INT5 | I | Input pins for the INT interrupt. |
| NMI interrupt input | NMI | I | Input pin for the NMI interrupt. |
| Key input interrupt input | KI0 to KI3 | I | Input pins for the key input interrupt. |
| Timer A | TA0OUT to TA4OUT | I/O | These are timer A0 to timer A4 I/O pins. |
| | TA0IN to TA4IN | I | These are timer A0 to timer A4 input pins. |
| | ZP | I | Input pin for the Z-phase. |
| Timer B | TB0IN to TB5IN | I | These are timer B0 to timer B5 input pins. |
| Three-phase motor control output | U, U, V, V, W, W | O | These are Three-phase motor control output pins. |
| Serial interface | CTS0 to CTS2 | I | These are transmit control input pins. |
| | RTS0 to RTS2 | O | These are receive control output pins. |
| | CLK0 to CLK3 | I/O | These are transfer clock I/O pins. |
| | RXD0 to RXD2 | I | These are serial data input pins. |
| | SIN3 | I | These are serial data input pins. |
| | TXD0 to TXD2 | O | These are serial data output pins. |
| | SOUT3 | O | These are serial data output pins. |
| | CLKS1 | O | This is output pin for transfer clock output from multiple pins function. |
| I ² C mode | SDA0 to SDA2 | I/O | These are serial data I/O pins. |
| | SCL0 to SCL2 | I/O | These are transfer clock I/O pins. (however, SCL2 for the N-channel open drain output.) |
| Reference voltage input | VREF | I | Applies the reference voltage for the A/D converter and D/A converter. |
| A/D converter | AN0 to AN7 AN0_0 to AN0_7 AN2_0 to AN2_7 | I | Analog input pins for the A/D converter. |
| | ADTRG | I | This is an A/D trigger input pin. |
| | ANEX0 | I/O | This is the extended analog input pin for the A/D converter, and is the output in external op-amp connection mode. |
| | ANEX1 | I | This is the extended analog input pin for the A/D converter. |
| D/A converter | DA0, DA1 | O | These are the output pins for the D/A converter. |
| CAN module | CRX0, CRX1 | I | These are the input pins for the CAN module. |
| | CTX0, CTX1 | O | These are the output pins for the CAN module. |

I: Input O: Output I/O: Input/Output

NOTE:

1. Ask the oscillator maker the oscillation characteristic.

Table 1.7 Pin Functions (3)

| Signal Name | Pin Name | I/O Type | Description |
|-------------|--|----------|--|
| I/O port | P0_0 to P0_7 P1_0 to P1_7 P2_0 to P2_7 P3_0 to P3_7 P4_0 to P4_7 P5_0 to P5_7 P6_0 to P6_7 P7_0 to P7_7 P8_0 to P8_4 P8_6, P8_7 P9_0 to P9_7 P10_0 to P10_7 | I/O | 8-bit I/O ports in CMOS, having a direction register to select an input or output. Each pin is set as an input port or output port. An input port can be set for a pull-up or for no pull-up in 4-bit unit by program. (however, P7_1 and P9_1 for the N-channel open drain output.) |
| Input port | P8_5 | I | Input pin for the NMI interrupt. Pin states can be read by the P8_5 bit in the P8 register. |

I: Input O: Output I/O: Input/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 configure a register bank. There are two register banks.

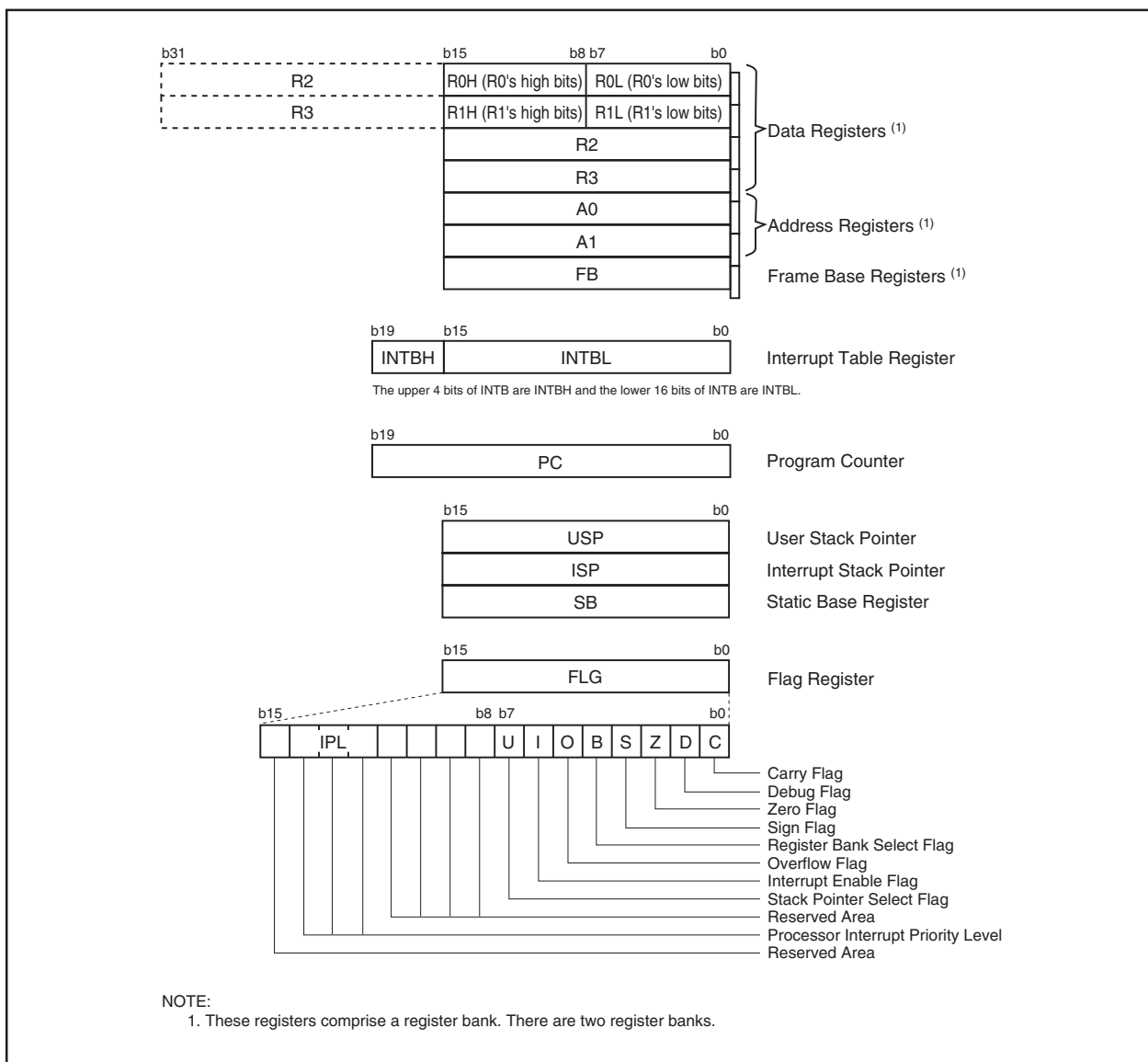


Figure 2.1 CPU Registers

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 analogous to R2R0.

2.2 Address Registers (A0 and A1)

The A0 register 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, 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), 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)

This flag is used exclusively for debugging purpose. During normal use, 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 set 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 set 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 request 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 shows a Memory Map. The address space extends the 1 Mbyte from address 00000h to FFFFFh. The internal ROM is allocated in a lower address direction beginning with address FFFFFh. For example, a 128-Kbyte internal ROM is allocated to the addresses from E0000h to FFFFFh.

As for the flash memory version, 4-Kbyte space (block A) exists in 0F000h to 0FFFFh. 4-Kbyte space is mainly for storing data. In addition to storing data, 4-Kbyte space also can store programs.

The fixed interrupt vector table is allocated to the addresses from FFFDCh to FFFFFh. Therefore, store the start address of each interrupt routine here.

The internal RAM is allocated in an upper address direction beginning with address 00400h. For example, a 5-Kbyte internal RAM is allocated to the addresses from 00400h to 017FFh. In addition to storing data, the internal RAM also stores the stack used when calling subroutines and when interrupts are generated.

The Special Function Registers (SFRs) are allocated to the addresses from 00000h to 003FFh. Peripheral function control registers are located here. Of the SFR, any area which has no functions allocated is reserved for future use and cannot be accessed by user.

The special page vector table is allocated to the addresses from FFE00h to FFFDBh. This vector is used by the JMPS or JSRS instruction. For details, refer to **M16C/60, M16C/20, M16C/Tiny Series Software Manual**.

In memory expansion and microprocessor modes, some areas are reserved for future use and cannot be used by users.

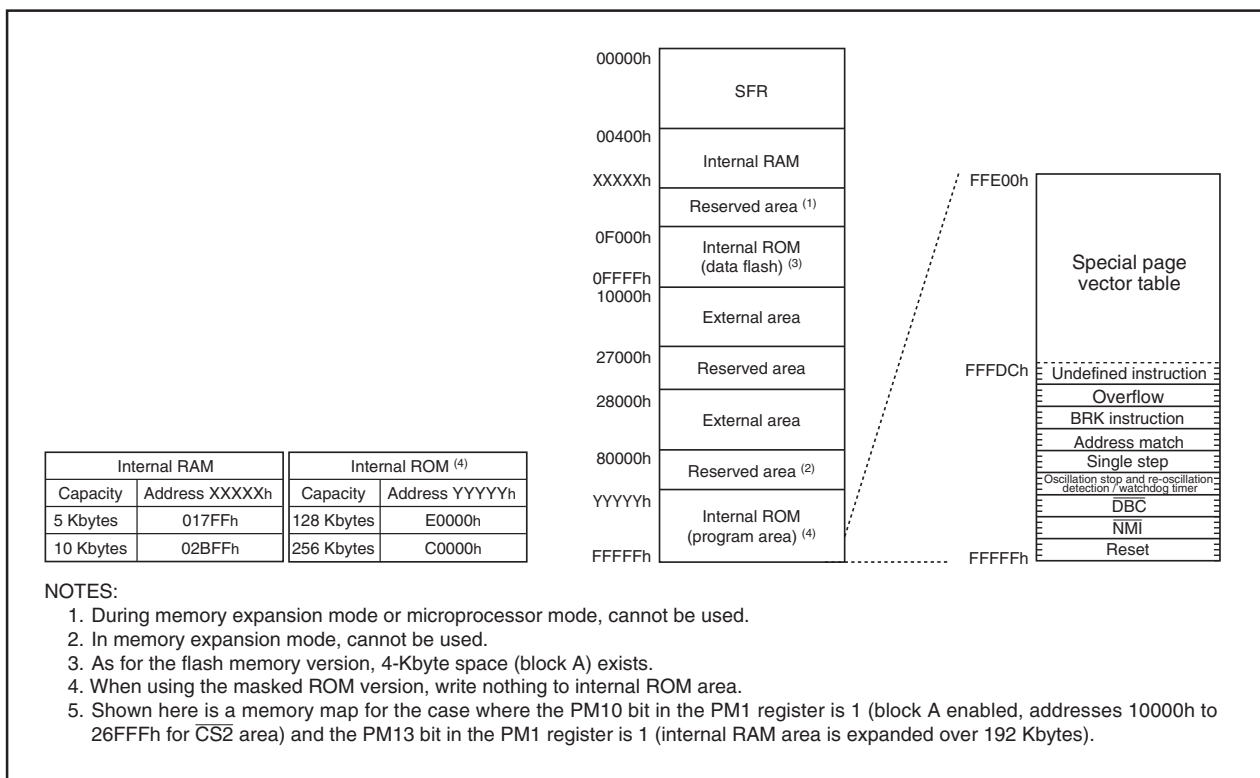


Figure 3.1 Memory Map

4. Special Function Registers (SFRs)

An SFR (Special Function Register) is a control register for a peripheral function.

Tables 4.1 to 4.16 list the SFR Information.

Table 4.1 SFR Information (1) ⁽³⁾

| Address | Register | Symbol | After Reset |
|---------|--|--------|---|
| 0000h | | | |
| 0001h | | | |
| 0002h | | | |
| 0003h | | | |
| 0004h | Processor Mode Register 0 ⁽¹⁾ | PM0 | 0000000b (CNVSS pin is "L") 00000011b (CNVSS pin is "H") |
| 0005h | Processor Mode Register 1 | PM1 | 00001000b |
| 0006h | System Clock Control Register 0 | CM0 | 01001000b |
| 0007h | System Clock Control Register 1 | CM1 | 00100000b |
| 0008h | Chip Select Control Register | CSR | 00000001b |
| 0009h | Address Match Interrupt Enable Register | AIER | XXXXXX00b |
| 000Ah | Protect Register | PRCR | XX000000b |
| 000Bh | | | |
| 000Ch | Oscillation Stop Detection Register ⁽²⁾ | CM2 | 0X000000b |
| 000Dh | | | |
| 000Eh | Watchdog Timer Start Register | WDTS | XXh |
| 000Fh | Watchdog Timer Control Register | WDC | 00XXXXXXb |
| 0010h | | | 00h |
| 0011h | Address Match Interrupt Register 0 | RMAD0 | 00h |
| 0012h | | | X0h |
| 0013h | | | |
| 0014h | | | 00h |
| 0015h | Address Match Interrupt Register 1 | RMAD1 | 00h |
| 0016h | | | X0h |
| 0017h | | | |
| 0018h | | | |
| 0019h | | | |
| 001Ah | | | |
| 001Bh | Chip Select Expansion Control Register | CSE | 00h |
| 001Ch | PLL Control Register 0 | PLC0 | 0001X010b |
| 001Dh | | | |
| 001Eh | Processor Mode Register 2 | PM2 | XXX00000b |
| 001Fh | | | |
| 0020h | | | XXh |
| 0021h | DMA0 Source Pointer | SAR0 | XXh |
| 0022h | | | XXh |
| 0023h | | | |
| 0024h | | | XXh |
| 0025h | DMA0 Destination Pointer | DAR0 | XXh |
| 0026h | | | XXh |
| 0027h | | | |
| 0028h | DMA0 Transfer Counter | TCR0 | XXh |
| 0029h | | | XXh |
| 002Ah | | | |
| 002Bh | | | |
| 002Ch | DMA0 Control Register | DM0CON | 00000X00b |
| 002Dh | | | |
| 002Eh | | | |
| 002Fh | | | |
| 0030h | | | XXh |
| 0031h | DMA1 Source Pointer | SAR1 | XXh |
| 0032h | | | XXh |
| 0033h | | | |
| 0034h | | | XXh |
| 0035h | DMA1 Destination Pointer | DAR1 | XXh |
| 0036h | | | XXh |
| 0037h | | | |
| 0038h | DMA1 Transfer Counter | TCR1 | XXh |
| 0039h | | | XXh |
| 003Ah | | | |
| 003Bh | | | |
| 003Ch | DMA1 Control Register | DM1CON | 00000X00b |
| 003Dh | | | |
| 003Eh | | | |
| 003Fh | | | |

X: Undefined

NOTES:

1. Bits PM00 and PM01 in the PM0 register do not change at software reset, watchdog timer reset and oscillation stop detection reset.
2. Bits CM20, CM21, and CM27 in the CM2 register do not change at oscillation stop detection reset.
3. Blank spaces are reserved. No access is allowed.

Table 4.2 SFR Information (2) ⁽¹⁾

| Address | Register | Symbol | After Reset |
|---------|--|----------|-------------|
| 0040h | | | |
| 0041h | CAN0/1 Wake-up Interrupt Control Register | C01WKIC | XXXXX000b |
| 0042h | CAN0 Successful Reception Interrupt Control Register | C0RECIC | XXXXX000b |
| 0043h | CAN0 Successful Transmission Interrupt Control Register | C0TRMIC | XXXXX000b |
| 0044h | INT3 Interrupt Control Register | INT3IC | XX00X000b |
| 0045h | Timer B5 Interrupt Control Register | TB5IC | XXXXX000b |
| 0046h | Timer B4 Interrupt Control Register | TB4IC | XXXXX000b |
| | UART1 Bus Collision Detection Interrupt Control Register | U1BCNIC | |
| 0047h | Timer B3 Interrupt Control Register | TB3IC | XXXXX000b |
| | UART0 Bus Collision Detection Interrupt Control Register | U0BCNIC | |
| 0048h | CAN1 Successful Reception Interrupt Control Register | C1RECIC | XX00X000b |
| | INT5 Interrupt Control Register | INT5IC | |
| 0049h | CAN1 Successful Transmission Interrupt Control Register | C1TRMIC | XX00X000b |
| | SI/O3 Interrupt Control Register | S3IC | |
| | INT4 Interrupt Control Register | INT4IC | |
| 004Ah | UART2 Bus Collision Detection Interrupt Control Register | U2BCNIC | XXXXX000b |
| 004Bh | DMA0 Interrupt Control Register | DM0IC | XXXXX000b |
| 004Ch | DMA1 Interrupt Control Register | DM1IC | XXXXX000b |
| 004Dh | CAN0/1 Error Interrupt Control Register | C01ERRIC | XXXXX000b |
| 004Eh | A/D Conversion Interrupt Control Register | ADIC | XXXXX000b |
| | Key Input Interrupt Control Register | KUPIC | |
| 004Fh | UART2 Transmit Interrupt Control Register | S2TIC | XXXXX000b |
| 0050h | UART2 Receive Interrupt Control Register | S2RIC | XXXXX000b |
| 0051h | UART0 Transmit Interrupt Control Register | S0TIC | XXXXX000b |
| 0052h | UART0 Receive Interrupt Control Register | S0RIC | XXXXX000b |
| 0053h | UART1 Transmit Interrupt Control Register | S1TIC | XXXXX000b |
| 0054h | UART1 Receive Interrupt Control Register | S1RIC | XXXXX000b |
| 0055h | Timer A0 Interrupt Control Register | TA0IC | XXXXX000b |
| 0056h | Timer A1 Interrupt Control Register | TA1IC | XXXXX000b |
| 0057h | Timer A2 Interrupt Control Register | TA2IC | XXXXX000b |
| 0058h | Timer A3 Interrupt Control Register | TA3IC | XXXXX000b |
| 0059h | Timer A4 Interrupt Control Register | TA4IC | XXXXX000b |
| 005Ah | Timer B0 Interrupt Control Register | TB0IC | XXXXX000b |
| 005Bh | Timer B1 Interrupt Control Register | TB1IC | XXXXX000b |
| 005Ch | Timer B2 Interrupt Control Register | TB2IC | XXXXX000b |
| 005Dh | INT0 Interrupt Control Register | INT0IC | XX00X000b |
| 005Eh | INT1 Interrupt Control Register | INT1IC | XX00X000b |
| 005Fh | INT2 Interrupt Control Register | INT2IC | XX00X000b |
| 0060h | | | XXh |
| 0061h | | | XXh |
| 0062h | CAN0 Message Box 0: Identifier / DLC | | XXh |
| 0063h | | | XXh |
| 0064h | | | XXh |
| 0065h | | | XXh |
| 0066h | | | XXh |
| 0067h | CAN0 Message Box 0: Data Field | | XXh |
| 0068h | | | XXh |
| 0069h | | | XXh |
| 006Ah | | | XXh |
| 006Bh | | | XXh |
| 006Ch | | | XXh |
| 006Dh | | | XXh |
| 006Eh | CAN0 Message Box 0: Time Stamp | | XXh |
| 006Fh | | | XXh |
| 0070h | | | XXh |
| 0071h | CAN0 Message Box 1: Identifier / DLC | | XXh |
| 0072h | | | XXh |
| 0073h | | | XXh |
| 0074h | | | XXh |
| 0075h | | | XXh |
| 0076h | CAN0 Message Box 1: Data Field | | XXh |
| 0077h | | | XXh |
| 0078h | | | XXh |
| 0079h | | | XXh |
| 007Ah | | | XXh |
| 007Bh | | | XXh |
| 007Ch | | | XXh |
| 007Dh | | XXh | |
| 007Eh | CAN0 Message Box 1: Time Stamp | | XXh |
| 007Fh | | | XXh |

X: Undefined

NOTE:

- Blank space is reserved. No access is allowed.

Table 4.3 SFR Information (3)

| Address | Register | Symbol | After Reset | | |
|---------|--------------------------------------|--------|--------------------------------------|--|-----|
| 0080h | CAN0 Message Box 2: Identifier / DLC | | XXh | | |
| 0081h | | | XXh | | |
| 0082h | | | XXh | | |
| 0083h | | | XXh | | |
| 0084h | | | XXh | | |
| 0085h | | | XXh | | |
| 0086h | CAN0 Message Box 2: Data Field | | XXh | | |
| 0087h | | | XXh | | |
| 0088h | | | XXh | | |
| 0089h | | | XXh | | |
| 008Ah | | | XXh | | |
| 008Bh | | | XXh | | |
| 008Ch | CAN0 Message Box 2: Time Stamp | | XXh | | |
| 008Dh | | | XXh | | |
| 008Eh | | | XXh | | |
| 008Fh | | | XXh | | |
| 0090h | | | CAN0 Message Box 3: Identifier / DLC | | XXh |
| 0091h | | | | | XXh |
| 0092h | XXh | | | | |
| 0093h | XXh | | | | |
| 0094h | XXh | | | | |
| 0095h | XXh | | | | |
| 0096h | CAN0 Message Box 3: Data Field | | XXh | | |
| 0097h | | | XXh | | |
| 0098h | | | XXh | | |
| 0099h | | | XXh | | |
| 009Ah | | | XXh | | |
| 009Bh | | | XXh | | |
| 009Ch | CAN0 Message Box 3: Time Stamp | | XXh | | |
| 009Dh | | | XXh | | |
| 009Eh | | | XXh | | |
| 009Fh | | | XXh | | |
| 00A0h | | | CAN0 Message Box 4: Identifier / DLC | | XXh |
| 00A1h | | | | | XXh |
| 00A2h | XXh | | | | |
| 00A3h | XXh | | | | |
| 00A4h | XXh | | | | |
| 00A5h | XXh | | | | |
| 00A6h | CAN0 Message Box 4: Data Field | | XXh | | |
| 00A7h | | | XXh | | |
| 00A8h | | | XXh | | |
| 00A9h | | | XXh | | |
| 00AAh | | | XXh | | |
| 00ABh | | | XXh | | |
| 00ACh | CAN0 Message Box 4: Time Stamp | | XXh | | |
| 00ADh | | | XXh | | |
| 00AEh | | | XXh | | |
| 00AFh | | | XXh | | |
| 00B0h | | | CAN0 Message Box 5: Identifier / DLC | | XXh |
| 00B1h | | | | | XXh |
| 00B2h | XXh | | | | |
| 00B3h | XXh | | | | |
| 00B4h | XXh | | | | |
| 00B5h | XXh | | | | |
| 00B6h | CAN0 Message Box 5: Data Field | | XXh | | |
| 00B7h | | | XXh | | |
| 00B8h | | | XXh | | |
| 00B9h | | | XXh | | |
| 00BAh | | | XXh | | |
| 00BBh | | | XXh | | |
| 00BCh | CAN0 Message Box 5: Time Stamp | | XXh | | |
| 00BDh | | | XXh | | |
| 00BEh | | | XXh | | |
| 00BFh | | | XXh | | |

X: Undefined

Table 4.4 SFR Information (4)

| Address | Register | Symbol | After Reset | | |
|---------|--------------------------------------|--------|--------------------------------------|--|-----|
| 00C0h | CAN0 Message Box 6: Identifier / DLC | | XXh | | |
| 00C1h | | | XXh | | |
| 00C2h | | | XXh | | |
| 00C3h | | | XXh | | |
| 00C4h | | | XXh | | |
| 00C5h | | | XXh | | |
| 00C6h | CAN0 Message Box 6: Data Field | | XXh | | |
| 00C7h | | | XXh | | |
| 00C8h | | | XXh | | |
| 00C9h | | | XXh | | |
| 00CAh | | | XXh | | |
| 00CBh | | | XXh | | |
| 00CCh | CAN0 Message Box 6: Time Stamp | | XXh | | |
| 00CDh | | | XXh | | |
| 00CEh | | | XXh | | |
| 00CFh | | | XXh | | |
| 00D0h | | | CAN0 Message Box 7: Identifier / DLC | | XXh |
| 00D1h | | | | | XXh |
| 00D2h | XXh | | | | |
| 00D3h | XXh | | | | |
| 00D4h | XXh | | | | |
| 00D5h | XXh | | | | |
| 00D6h | CAN0 Message Box 7: Data Field | | XXh | | |
| 00D7h | | | XXh | | |
| 00D8h | | | XXh | | |
| 00D9h | | | XXh | | |
| 00DAh | | | XXh | | |
| 00DBh | | | XXh | | |
| 00DCh | CAN0 Message Box 7: Time Stamp | | XXh | | |
| 00DDh | | | XXh | | |
| 00DEh | | | XXh | | |
| 00DFh | | | XXh | | |
| 00E0h | | | CAN0 Message Box 8: Identifier / DLC | | XXh |
| 00E1h | | | | | XXh |
| 00E2h | XXh | | | | |
| 00E3h | XXh | | | | |
| 00E4h | XXh | | | | |
| 00E5h | XXh | | | | |
| 00E6h | CAN0 Message Box 8: Data Field | | XXh | | |
| 00E7h | | | XXh | | |
| 00E8h | | | XXh | | |
| 00E9h | | | XXh | | |
| 00EAh | | | XXh | | |
| 00EBh | | | XXh | | |
| 00ECh | CAN0 Message Box 8: Time Stamp | | XXh | | |
| 00EDh | | | XXh | | |
| 00EEh | | | XXh | | |
| 00EFh | | | XXh | | |
| 00F0h | | | CAN0 Message Box 9: Identifier / DLC | | XXh |
| 00F1h | | | | | XXh |
| 00F2h | XXh | | | | |
| 00F3h | XXh | | | | |
| 00F4h | XXh | | | | |
| 00F5h | XXh | | | | |
| 00F6h | CAN0 Message Box 9: Data Field | | XXh | | |
| 00F7h | | | XXh | | |
| 00F8h | | | XXh | | |
| 00F9h | | | XXh | | |
| 00FAh | | | XXh | | |
| 00FBh | | | XXh | | |
| 00FCh | CAN0 Message Box 9: Time Stamp | | XXh | | |
| 00FDh | | | XXh | | |
| 00FEh | | | XXh | | |
| 00FFh | | | XXh | | |

X: Undefined

Table 4.5 SFR Information (5)

| Address | Register | Symbol | After Reset | | |
|---------|---------------------------------------|--------|---------------------------------------|--|-----|
| 0100h | CAN0 Message Box 10: Identifier / DLC | | XXh | | |
| 0101h | | | XXh | | |
| 0102h | | | XXh | | |
| 0103h | | | XXh | | |
| 0104h | | | XXh | | |
| 0105h | | | XXh | | |
| 0106h | CAN0 Message Box 10: Data Field | | XXh | | |
| 0107h | | | XXh | | |
| 0108h | | | XXh | | |
| 0109h | | | XXh | | |
| 010Ah | | | XXh | | |
| 010Bh | | | XXh | | |
| 010Ch | CAN0 Message Box 10: Time Stamp | | XXh | | |
| 010Dh | | | XXh | | |
| 010Eh | | | XXh | | |
| 010Fh | | | XXh | | |
| 0110h | | | CAN0 Message Box 11: Identifier / DLC | | XXh |
| 0111h | | | | | XXh |
| 0112h | XXh | | | | |
| 0113h | XXh | | | | |
| 0114h | XXh | | | | |
| 0115h | XXh | | | | |
| 0116h | CAN0 Message Box 11: Data Field | | XXh | | |
| 0117h | | | XXh | | |
| 0118h | | | XXh | | |
| 0119h | | | XXh | | |
| 011Ah | | | XXh | | |
| 011Bh | | | XXh | | |
| 011Ch | CAN0 Message Box 11: Time Stamp | | XXh | | |
| 011Dh | | | XXh | | |
| 011Eh | | | XXh | | |
| 011Fh | | | XXh | | |
| 0120h | | | CAN0 Message Box 12: Identifier / DLC | | XXh |
| 0121h | | | | | XXh |
| 0122h | XXh | | | | |
| 0123h | XXh | | | | |
| 0124h | XXh | | | | |
| 0125h | XXh | | | | |
| 0126h | CAN0 Message Box 12: Data Field | | XXh | | |
| 0127h | | | XXh | | |
| 0128h | | | XXh | | |
| 0129h | | | XXh | | |
| 012Ah | | | XXh | | |
| 012Bh | | | XXh | | |
| 012Ch | CAN0 Message Box 12: Time Stamp | | XXh | | |
| 012Dh | | | XXh | | |
| 012Eh | | | XXh | | |
| 012Fh | | | XXh | | |
| 0130h | | | CAN0 Message Box 13: Identifier / DLC | | XXh |
| 0131h | | | | | XXh |
| 0132h | XXh | | | | |
| 0133h | XXh | | | | |
| 0134h | XXh | | | | |
| 0135h | XXh | | | | |
| 0136h | CAN0 Message Box 13: Data Field | | XXh | | |
| 0137h | | | XXh | | |
| 0138h | | | XXh | | |
| 0139h | | | XXh | | |
| 013Ah | | | XXh | | |
| 013Bh | | | XXh | | |
| 013Ch | CAN0 Message Box 13: Time Stamp | | XXh | | |
| 013Dh | | | XXh | | |
| 013Eh | | | XXh | | |
| 013Fh | | | XXh | | |

X: Undefined

Table 4.6 SFR Information (6) ⁽¹⁾

| Address | Register | Symbol | After Reset | | |
|---------|--------------------------------------|--------|--------------------------------------|-------|-----|
| 0140h | CAN0 Message Box 14: Identifier /DLC | | XXh | | |
| 0141h | | | XXh | | |
| 0142h | | | XXh | | |
| 0143h | | | XXh | | |
| 0144h | | | XXh | | |
| 0145h | | | XXh | | |
| 0146h | CAN0 Message Box 14: Data Field | | XXh | | |
| 0147h | | | XXh | | |
| 0148h | | | XXh | | |
| 0149h | | | XXh | | |
| 014Ah | | | XXh | | |
| 014Bh | | | XXh | | |
| 014Ch | CAN0 Message Box 14: Time Stamp | | XXh | | |
| 014Dh | | | XXh | | |
| 014Eh | | | XXh | | |
| 014Fh | | | XXh | | |
| 0150h | | | CAN0 Message Box 15: Identifier /DLC | | XXh |
| 0151h | | | | | XXh |
| 0152h | XXh | | | | |
| 0153h | XXh | | | | |
| 0154h | XXh | | | | |
| 0155h | XXh | | | | |
| 0156h | CAN0 Message Box 15: Data Field | | XXh | | |
| 0157h | | | XXh | | |
| 0158h | | | XXh | | |
| 0159h | | | XXh | | |
| 015Ah | | | XXh | | |
| 015Bh | | | XXh | | |
| 015Ch | CAN0 Message Box 15: Time Stamp | | XXh | | |
| 015Dh | | | XXh | | |
| 015Eh | | | XXh | | |
| 015Fh | | | XXh | | |
| 0160h | | | CAN0 Global Mask Register | C0GMR | XXh |
| 0161h | | | | | XXh |
| 0162h | XXh | | | | |
| 0163h | XXh | | | | |
| 0164h | XXh | | | | |
| 0165h | XXh | | | | |
| 0166h | CAN0 Local Mask A Register | C0LMAR | XXh | | |
| 0167h | | | XXh | | |
| 0168h | | | XXh | | |
| 0169h | | | XXh | | |
| 016Ah | | | XXh | | |
| 016Bh | | | XXh | | |
| 016Ch | CAN0 Local Mask B Register | C0LMBR | XXh | | |
| 016Dh | | | XXh | | |
| 016Eh | | | XXh | | |
| 016Fh | | | XXh | | |
| 0170h | | | XXh | | |
| 0171h | | | XXh | | |
| 0172h | | | | | |
| 0173h | | | | | |
| 0174h | | | | | |
| 0175h | | | | | |
| 0176h | | | | | |
| 0177h | | | | | |
| 0178h | | | | | |
| 0179h | | | | | |
| 017Ah | | | | | |
| 017Bh | | | | | |
| 017Ch | | | | | |
| 017Dh | | | | | |
| 017Eh | | | | | |
| 017Fh | | | | | |

X: Undefined

NOTE:

- Blank spaces are reserved. No access is allowed.

Table 4.7 SFR Information (7) ⁽²⁾

| Address | Register | Symbol | After Reset |
|---------|--|--------|-------------|
| 0180h | | | |
| 0181h | | | |
| 0182h | | | |
| 0183h | | | |
| 0184h | | | |
| 0185h | | | |
| 0186h | | | |
| 0187h | | | |
| 0188h | | | |
| 0189h | | | |
| 018Ah | | | |
| 018Bh | | | |
| 018Ch | | | |
| 018Dh | | | |
| 018Eh | | | |
| 018Fh | | | |
| 0190h | | | |
| 0191h | | | |
| 0192h | | | |
| 0193h | | | |
| 0194h | | | |
| 0195h | | | |
| 0196h | | | |
| 0197h | | | |
| 0198h | | | |
| 0199h | | | |
| 019Ah | | | |
| 019Bh | | | |
| 019Ch | | | |
| 019Dh | | | |
| 019Eh | | | |
| 019Fh | | | |
| 01A0h | | | |
| 01A1h | | | |
| 01A2h | | | |
| 01A3h | | | |
| 01A4h | | | |
| 01A5h | | | |
| 01A6h | | | |
| 01A7h | | | |
| 01A8h | | | |
| 01A9h | | | |
| 01AAh | | | |
| 01ABh | | | |
| 01ACh | | | |
| 01ADh | | | |
| 01AEh | | | |
| 01AFh | | | |
| 01B0h | | | |
| 01B1h | | | |
| 01B2h | | | |
| 01B3h | | | |
| 01B4h | | | |
| 01B5h | Flash Memory Control Register 1 ⁽¹⁾ | FMR1 | 0X00XX0Xb |
| 01B6h | | | |
| 01B7h | Flash Memory Control Register 0 ⁽¹⁾ | FMR0 | 00000001b |
| 01B8h | | | 00h |
| 01B9h | Address Match Interrupt Register 2 | RMAD2 | 00h |
| 01BAh | | | X0h |
| 01BBh | Address Match Interrupt Enable Register 2 | AIER2 | XXXXXX00b |
| 01BCh | | | 00h |
| 01BDh | Address Match Interrupt Register 3 | RMAD3 | 00h |
| 01BEh | | | X0h |
| 01BFh | | | |

X: Undefined

NOTES:

1. These registers are included in the flash memory version. Cannot be accessed by users in the mask ROM version.
2. Blank spaces are reserved. No access is allowed.

Table 4.8 SFR Information (8) ⁽¹⁾

| Address | Register | Symbol | After Reset |
|---------|---|--------|-------------|
| 01C0h | Timer B3, B4, B5 Count Start Flag | TBSR | 000XXXXXb |
| 01C1h | | | |
| 01C2h | | | XXh |
| 01C3h | Timer A1-1 Register | TA11 | XXh |
| 01C4h | | | XXh |
| 01C5h | Timer A2-1 Register | TA21 | XXh |
| 01C6h | | | XXh |
| 01C7h | Timer A4-1 Register | TA41 | XXh |
| 01C8h | Three-Phase PWM Control Register 0 | INVC0 | 00h |
| 01C9h | Three-Phase PWM Control Register 1 | INVC1 | 00h |
| 01CAh | Three-Phase Output Buffer Register 0 | IDB0 | 00111111b |
| 01CBh | Three-Phase Output Buffer Register 1 | IDB1 | 00111111b |
| 01CCh | Dead Time Timer | DTT | XXh |
| 01CDh | Timer B2 Interrupt Generation Frequency Set Counter | ICTB2 | XXh |
| 01CEh | | | |
| 01CFh | | | |
| 01D0h | | | XXh |
| 01D1h | Timer B3 Register | TB3 | XXh |
| 01D2h | | | XXh |
| 01D3h | Timer B4 Register | TB4 | XXh |
| 01D4h | | | XXh |
| 01D5h | Timer B5 Register | TB5 | XXh |
| 01D6h | | | |
| 01D7h | | | |
| 01D8h | | | |
| 01D9h | | | |
| 01DAh | | | |
| 01DBh | Timer B3 Mode Register | TB3MR | 00XX0000b |
| 01DCh | Timer B4 Mode Register | TB4MR | 00XX0000b |
| 01DDh | Timer B5 Mode Register | TB5MR | 00XX0000b |
| 01DEh | Interrupt Source Select Register 0 | IFSR0 | 00XX0000b |
| 01DFh | Interrupt Source Select Register 1 | IFSR1 | 00h |
| 01E0h | SI/O3 Transmit/Receive Register | S3TRR | XXh |
| 01E1h | | | |
| 01E2h | SI/O3 Control Register | S3C | 01000000b |
| 01E3h | SI/O3 Bit Rate Register | S3BRG | XXh |
| 01E4h | | | |
| 01E5h | | | |
| 01E6h | | | |
| 01E7h | | | |
| 01E8h | | | |
| 01E9h | | | |
| 01EAh | | | |
| 01EBh | | | |
| 01ECh | UART0 Special Mode Register 4 | U0SMR4 | 00h |
| 01EDh | UART0 Special Mode Register 3 | U0SMR3 | 00X0X0Xb |
| 01EEh | UART0 Special Mode Register 2 | U0SMR2 | X0000000b |
| 01EFh | UART0 Special Mode Register | U0SMR | X0000000b |
| 01F0h | UART1 Special Mode Register 4 | U1SMR4 | 00h |
| 01F1h | UART1 Special Mode Register 3 | U1SMR3 | 00X0X0Xb |
| 01F2h | UART1 Special Mode Register 2 | U1SMR2 | X0000000b |
| 01F3h | UART1 Special Mode Register | U1SMR | X0000000b |
| 01F4h | UART2 Special Mode Register 4 | U2SMR4 | 00h |
| 01F5h | UART2 Special Mode Register 3 | U2SMR3 | 00X0X0Xb |
| 01F6h | UART2 Special Mode Register 2 | U2SMR2 | X0000000b |
| 01F7h | UART2 Special Mode Register | U2SMR | X0000000b |
| 01F8h | UART2 Transmit/Receive Mode Register | U2MR | 00h |
| 01F9h | UART2 Bit Rate Register | U2BRG | XXh |
| 01FAh | | | XXh |
| 01FBh | UART2 Transmit Buffer Register | U2TB | XXh |
| 01FCh | UART2 Transmit/Receive Control Register 0 | U2C0 | 00001000b |
| 01FDh | UART2 Transmit/Receive Control Register 1 | U2C1 | 00000010b |
| 01FEh | | | XXh |
| 01FFh | UART2 Receive Buffer Register | U2RB | XXh |

X: Undefined

NOTE:

- Blank spaces are reserved. No access is allowed.

Table 4.9 SFR Information (9)

| Address | Register | Symbol | After Reset |
|---------|------------------------------------|----------|-------------|
| 0200h | CAN0 Message Control Register 0 | COMCTL0 | 00h |
| 0201h | CAN0 Message Control Register 1 | COMCTL1 | 00h |
| 0202h | CAN0 Message Control Register 2 | COMCTL2 | 00h |
| 0203h | CAN0 Message Control Register 3 | COMCTL3 | 00h |
| 0204h | CAN0 Message Control Register 4 | COMCTL4 | 00h |
| 0205h | CAN0 Message Control Register 5 | COMCTL5 | 00h |
| 0206h | CAN0 Message Control Register 6 | COMCTL6 | 00h |
| 0207h | CAN0 Message Control Register 7 | COMCTL7 | 00h |
| 0208h | CAN0 Message Control Register 8 | COMCTL8 | 00h |
| 0209h | CAN0 Message Control Register 9 | COMCTL9 | 00h |
| 020Ah | CAN0 Message Control Register 10 | COMCTL10 | 00h |
| 020Bh | CAN0 Message Control Register 11 | COMCTL11 | 00h |
| 020Ch | CAN0 Message Control Register 12 | COMCTL12 | 00h |
| 020Dh | CAN0 Message Control Register 13 | COMCTL13 | 00h |
| 020Eh | CAN0 Message Control Register 14 | COMCTL14 | 00h |
| 020Fh | CAN0 Message Control Register 15 | COMCTL15 | 00h |
| 0210h | CAN0 Control Register | C0CTLR | X0000001b |
| 0211h | | | XX0X0000b |
| 0212h | CAN0 Status Register | C0STR | 00h |
| 0213h | | | X0000001b |
| 0214h | CAN0 Slot Status Register | C0SSTR | 00h |
| 0215h | | | 00h |
| 0216h | CAN0 Interrupt Control Register | C0ICR | 00h |
| 0217h | | | 00h |
| 0218h | CAN0 Extended ID Register | C0IDR | 00h |
| 0219h | | | 00h |
| 021Ah | CAN0 Configuration Register | C0CONR | XXh |
| 021Bh | | | XXh |
| 021Ch | CAN0 Receive Error Count Register | C0RECR | 00h |
| 021Dh | CAN0 Transmit Error Count Register | C0TECR | 00h |
| 021Eh | CAN0 Time Stamp Register | C0TSR | 00h |
| 021Fh | | | 00h |
| 0220h | CAN1 Message Control Register 0 | C1MCTL0 | 00h |
| 0221h | CAN1 Message Control Register 1 | C1MCTL1 | 00h |
| 0222h | CAN1 Message Control Register 2 | C1MCTL2 | 00h |
| 0223h | CAN1 Message Control Register 3 | C1MCTL3 | 00h |
| 0224h | CAN1 Message Control Register 4 | C1MCTL4 | 00h |
| 0225h | CAN1 Message Control Register 5 | C1MCTL5 | 00h |
| 0226h | CAN1 Message Control Register 6 | C1MCTL6 | 00h |
| 0227h | CAN1 Message Control Register 7 | C1MCTL7 | 00h |
| 0228h | CAN1 Message Control Register 8 | C1MCTL8 | 00h |
| 0229h | CAN1 Message Control Register 9 | C1MCTL9 | 00h |
| 022Ah | CAN1 Message Control Register 10 | C1MCTL10 | 00h |
| 022Bh | CAN1 Message Control Register 11 | C1MCTL11 | 00h |
| 022Ch | CAN1 Message Control Register 12 | C1MCTL12 | 00h |
| 022Dh | CAN1 Message Control Register 13 | C1MCTL13 | 00h |
| 022Eh | CAN1 Message Control Register 14 | C1MCTL14 | 00h |
| 022Fh | CAN1 Message Control Register 15 | C1MCTL15 | 00h |
| 0230h | CAN1 Control Register | C1CTLR | X0000001b |
| 0231h | | | XX0X0000b |
| 0232h | CAN1 Status Register | C1STR | 00h |
| 0233h | | | X0000001b |
| 0234h | CAN1 Slot Status Register | C1SSTR | 00h |
| 0235h | | | 00h |
| 0236h | CAN1 Interrupt Control Register | C1ICR | 00h |
| 0237h | | | 00h |
| 0238h | CAN1 Extended ID Register | C1IDR | 00h |
| 0239h | | | 00h |
| 023Ah | CAN1 Configuration Register | C1CONR | XXh |
| 023Bh | | | XXh |
| 023Ch | CAN1 Receive Error Count Register | C1RECR | 00h |
| 023Dh | CAN1 Transmit Error Count Register | C1TECR | 00h |
| 023Eh | CAN1 Time Stamp Register | C1TSR | 00h |
| 023Fh | | | 00h |

X: Undefined

Table 4.10 SFR Information (10) ⁽¹⁾

| Address | Register | Symbol | After Reset |
|---------|---|--------|-------------|
| 0240h | | | |
| 0241h | | | |
| 0242h | | | |
| 0243h | CAN0 Acceptance Filter Support Register | C0AFS | XXh |
| 0244h | | | XXh |
| 0245h | CAN1 Acceptance Filter Support Register | C1AFS | XXh |
| 0246h | | | |
| 0247h | | | |
| 0248h | | | |
| 0249h | | | |
| 024Ah | | | |
| 024Bh | | | |
| 024Ch | | | |
| 024Dh | | | |
| 024Eh | | | |
| 024Fh | | | |
| 0250h | | | |
| 0251h | | | |
| 0252h | | | |
| 0253h | | | |
| 0254h | | | |
| 0255h | | | |
| 0256h | | | |
| 0257h | | | |
| 0258h | | | |
| 0259h | | | |
| 025Ah | | | |
| 025Bh | | | |
| 025Ch | | | |
| 025Dh | | | |
| 025Eh | Peripheral Clock Select Register | PCLKR | 00h |
| 025Fh | CAN0/1 Clock Select Register | CCLKR | 00h |
| 0260h | | | XXh |
| 0261h | | | XXh |
| 0262h | CAN1 Message Box 0: Identifier / DLC | | XXh |
| 0263h | | | XXh |
| 0264h | | | XXh |
| 0265h | | | XXh |
| 0266h | | | XXh |
| 0267h | | | XXh |
| 0268h | | | XXh |
| 0269h | CAN1 Message Box 0: Data Field | | XXh |
| 026Ah | | | XXh |
| 026Bh | | | XXh |
| 026Ch | | | XXh |
| 026Dh | | | XXh |
| 026Eh | CAN1 Message Box 0:Time Stamp | | XXh |
| 026Fh | | | XXh |
| 0270h | | | XXh |
| 0271h | | | XXh |
| 0272h | CAN1 Message Box 1: Identifier / DLC | | XXh |
| 0273h | | | XXh |
| 0274h | | | XXh |
| 0275h | | | XXh |
| 0276h | | | XXh |
| 0277h | | | XXh |
| 0278h | | | XXh |
| 0279h | CAN1 Message Box 1: Data Field | | XXh |
| 027Ah | | | XXh |
| 027Bh | | | XXh |
| 027Ch | | | XXh |
| 027Dh | | | XXh |
| 027Eh | CAN1 Message Box 1:Time Stamp | | XXh |
| 027Fh | | | XXh |

X: Undefined

NOTE:

- Blank spaces are reserved. No access is allowed.

Table 4.11 SFR Information (11)

| Address | Register | Symbol | After Reset | | |
|---------|--------------------------------------|--------|--------------------------------------|--|-----|
| 0280h | CAN1 Message Box 2: Identifier / DLC | | XXh | | |
| 0281h | | | XXh | | |
| 0282h | | | XXh | | |
| 0283h | | | XXh | | |
| 0284h | | | XXh | | |
| 0285h | | | XXh | | |
| 0286h | CAN1 Message Box 2: Data Field | | XXh | | |
| 0287h | | | XXh | | |
| 0288h | | | XXh | | |
| 0289h | | | XXh | | |
| 028Ah | | | XXh | | |
| 028Bh | | | XXh | | |
| 028Ch | CAN1 Message Box 2: Time Stamp | | XXh | | |
| 028Dh | | | XXh | | |
| 028Eh | | | XXh | | |
| 028Fh | | | XXh | | |
| 0290h | | | CAN1 Message Box 3: Identifier / DLC | | XXh |
| 0291h | | | | | XXh |
| 0292h | XXh | | | | |
| 0293h | XXh | | | | |
| 0294h | XXh | | | | |
| 0295h | XXh | | | | |
| 0296h | CAN1 Message Box 3: Data Field | | XXh | | |
| 0297h | | | XXh | | |
| 0298h | | | XXh | | |
| 0299h | | | XXh | | |
| 029Ah | | | XXh | | |
| 029Bh | | | XXh | | |
| 029Ch | CAN1 Message Box 3: Time Stamp | | XXh | | |
| 029Dh | | | XXh | | |
| 029Eh | | | XXh | | |
| 029Fh | | | XXh | | |
| 02A0h | | | CAN1 Message Box 4: Identifier / DLC | | XXh |
| 02A1h | | | | | XXh |
| 02A2h | XXh | | | | |
| 02A3h | XXh | | | | |
| 02A4h | XXh | | | | |
| 02A5h | XXh | | | | |
| 02A6h | CAN1 Message Box 4: Data Field | | XXh | | |
| 02A7h | | | XXh | | |
| 02A8h | | | XXh | | |
| 02A9h | | | XXh | | |
| 02AAh | | | XXh | | |
| 02ABh | | | XXh | | |
| 02ACh | CAN1 Message Box 4: Time Stamp | | XXh | | |
| 02ADh | | | XXh | | |
| 02AEh | | | XXh | | |
| 02AFh | | | XXh | | |
| 02B0h | | | CAN1 Message Box 5: Identifier / DLC | | XXh |
| 02B1h | | | | | XXh |
| 02B2h | XXh | | | | |
| 02B3h | XXh | | | | |
| 02B4h | XXh | | | | |
| 02B5h | XXh | | | | |
| 02B6h | CAN1 Message Box 5: Data Field | | XXh | | |
| 02B7h | | | XXh | | |
| 02B8h | | | XXh | | |
| 02B9h | | | XXh | | |
| 02BAh | | | XXh | | |
| 02BBh | | | XXh | | |
| 02BCh | CAN1 Message Box 5: Time Stamp | | XXh | | |
| 02BDh | | | XXh | | |
| 02BEh | | | XXh | | |
| 02BFh | | | XXh | | |

X: Undefined

Table 4.12 SFR Information (12)

| Address | Register | Symbol | After Reset | | |
|---------|--------------------------------------|--------|--------------------------------------|--|-----|
| 02C0h | CAN1 Message Box 6: Identifier / DLC | | XXh | | |
| 02C1h | | | XXh | | |
| 02C2h | | | XXh | | |
| 02C3h | | | XXh | | |
| 02C4h | | | XXh | | |
| 02C5h | | | XXh | | |
| 02C6h | CAN1 Message Box 6: Data Field | | XXh | | |
| 02C7h | | | XXh | | |
| 02C8h | | | XXh | | |
| 02C9h | | | XXh | | |
| 02CAh | | | XXh | | |
| 02CBh | | | XXh | | |
| 02CCh | CAN1 Message Box 6: Time Stamp | | XXh | | |
| 02CDh | | | XXh | | |
| 02CEh | | | XXh | | |
| 02CFh | | | XXh | | |
| 02D0h | | | CAN1 Message Box 7: Identifier / DLC | | XXh |
| 02D1h | | | | | XXh |
| 02D2h | XXh | | | | |
| 02D3h | XXh | | | | |
| 02D4h | XXh | | | | |
| 02D5h | XXh | | | | |
| 02D6h | CAN1 Message Box 7: Data Field | | XXh | | |
| 02D7h | | | XXh | | |
| 02D8h | | | XXh | | |
| 02D9h | | | XXh | | |
| 02DAh | | | XXh | | |
| 02DBh | | | XXh | | |
| 02DCh | CAN1 Message Box 7: Time Stamp | | XXh | | |
| 02DDh | | | XXh | | |
| 02DEh | | | XXh | | |
| 02DFh | | | XXh | | |
| 02E0h | | | CAN1 Message Box 8: Identifier / DLC | | XXh |
| 02E1h | | | | | XXh |
| 02E2h | XXh | | | | |
| 02E3h | XXh | | | | |
| 02E4h | XXh | | | | |
| 02E5h | XXh | | | | |
| 02E6h | CAN1 Message Box 8: Data Field | | XXh | | |
| 02E7h | | | XXh | | |
| 02E8h | | | XXh | | |
| 02E9h | | | XXh | | |
| 02EAh | | | XXh | | |
| 02EBh | | | XXh | | |
| 02ECh | CAN1 Message Box 8: Time Stamp | | XXh | | |
| 02EDh | | | XXh | | |
| 02EEh | | | XXh | | |
| 02EFh | | | XXh | | |
| 02F0h | | | CAN1 Message Box 9: Identifier / DLC | | XXh |
| 02F1h | | | | | XXh |
| 02F2h | XXh | | | | |
| 02F3h | XXh | | | | |
| 02F4h | XXh | | | | |
| 02F5h | XXh | | | | |
| 02F6h | CAN1 Message Box 9: Data Field | | XXh | | |
| 02F7h | | | XXh | | |
| 02F8h | | | XXh | | |
| 02F9h | | | XXh | | |
| 02FAh | | | XXh | | |
| 02FBh | | | XXh | | |
| 02FCh | CAN1 Message Box 9: Time Stamp | | XXh | | |
| 02FDh | | | XXh | | |
| 02FEh | | | XXh | | |
| 02FFh | | | XXh | | |

X: Undefined

Table 4.13 SFR Information (13)

| Address | Register | Symbol | After Reset | | |
|---------|---------------------------------------|--------|---------------------------------------|--|-----|
| 0300h | CAN1 Message Box 10: Identifier / DLC | | XXh | | |
| 0301h | | | XXh | | |
| 0302h | | | XXh | | |
| 0303h | | | XXh | | |
| 0304h | | | XXh | | |
| 0305h | | | XXh | | |
| 0306h | CAN1 Message Box 10: Data Field | | XXh | | |
| 0307h | | | XXh | | |
| 0308h | | | XXh | | |
| 0309h | | | XXh | | |
| 030Ah | | | XXh | | |
| 030Bh | | | XXh | | |
| 030Ch | CAN1 Message Box 10: Time Stamp | | XXh | | |
| 030Dh | | | XXh | | |
| 030Eh | | | XXh | | |
| 030Fh | | | XXh | | |
| 0310h | | | CAN1 Message Box 11: Identifier / DLC | | XXh |
| 0311h | | | | | XXh |
| 0312h | XXh | | | | |
| 0313h | XXh | | | | |
| 0314h | XXh | | | | |
| 0315h | XXh | | | | |
| 0316h | CAN1 Message Box 11: Data Field | | XXh | | |
| 0317h | | | XXh | | |
| 0318h | | | XXh | | |
| 0319h | | | XXh | | |
| 031Ah | | | XXh | | |
| 031Bh | | | XXh | | |
| 031Ch | CAN1 Message Box 11: Time Stamp | | XXh | | |
| 031Dh | | | XXh | | |
| 031Eh | | | XXh | | |
| 031Fh | | | XXh | | |
| 0320h | | | CAN1 Message Box 12: Identifier / DLC | | XXh |
| 0321h | | | | | XXh |
| 0322h | XXh | | | | |
| 0323h | XXh | | | | |
| 0324h | XXh | | | | |
| 0325h | XXh | | | | |
| 0326h | CAN1 Message Box 12: Data Field | | XXh | | |
| 0327h | | | XXh | | |
| 0328h | | | XXh | | |
| 0329h | | | XXh | | |
| 032Ah | | | XXh | | |
| 032Bh | | | XXh | | |
| 032Ch | CAN1 Message Box 12: Time Stamp | | XXh | | |
| 032Dh | | | XXh | | |
| 032Eh | | | XXh | | |
| 032Fh | | | XXh | | |
| 0330h | | | CAN1 Message Box 13: Identifier / DLC | | XXh |
| 0331h | | | | | XXh |
| 0332h | XXh | | | | |
| 0333h | XXh | | | | |
| 0334h | XXh | | | | |
| 0335h | XXh | | | | |
| 0336h | CAN1 Message Box 13: Data Field | | XXh | | |
| 0337h | | | XXh | | |
| 0338h | | | XXh | | |
| 0339h | | | XXh | | |
| 033Ah | | | XXh | | |
| 033Bh | | | XXh | | |
| 033Ch | CAN1 Message Box 13: Time Stamp | | XXh | | |
| 033Dh | | | XXh | | |
| 033Eh | | | XXh | | |
| 033Fh | | | XXh | | |

X: Undefined

Table 4.14 SFR Information (14) ⁽¹⁾

| Address | Register | Symbol | After Reset | | |
|---------|---------------------------------------|--------|---------------------------------------|-------|-----|
| 0340h | CAN1 Message Box 14: Identifier / DLC | | XXh | | |
| 0341h | | | XXh | | |
| 0342h | | | XXh | | |
| 0343h | | | XXh | | |
| 0344h | | | XXh | | |
| 0345h | | | XXh | | |
| 0346h | CAN1 Message Box 14: Data Field | | XXh | | |
| 0347h | | | XXh | | |
| 0348h | | | XXh | | |
| 0349h | | | XXh | | |
| 034Ah | | | XXh | | |
| 034Bh | | | XXh | | |
| 034Ch | CAN1 Message Box 14: Time Stamp | | XXh | | |
| 034Dh | | | XXh | | |
| 034Eh | | | XXh | | |
| 034Fh | | | XXh | | |
| 0350h | | | CAN1 Message Box 15: Identifier / DLC | | XXh |
| 0351h | | | | | XXh |
| 0352h | XXh | | | | |
| 0353h | XXh | | | | |
| 0354h | XXh | | | | |
| 0355h | XXh | | | | |
| 0356h | CAN1 Message Box 15: Data Field | | XXh | | |
| 0357h | | | XXh | | |
| 0358h | | | XXh | | |
| 0359h | | | XXh | | |
| 035Ah | | | XXh | | |
| 035Bh | | | XXh | | |
| 035Ch | CAN1 Message Box 15: Time Stamp | | XXh | | |
| 035Dh | | | XXh | | |
| 035Eh | | | XXh | | |
| 035Fh | | | XXh | | |
| 0360h | | | CAN1 Global Mask Register | C1GMR | XXh |
| 0361h | | | | | XXh |
| 0362h | XXh | | | | |
| 0363h | XXh | | | | |
| 0364h | XXh | | | | |
| 0365h | XXh | | | | |
| 0366h | CAN1 Local Mask A Register | C1LMAR | XXh | | |
| 0367h | | | XXh | | |
| 0368h | | | XXh | | |
| 0369h | | | XXh | | |
| 036Ah | | | XXh | | |
| 036Bh | | | XXh | | |
| 036Ch | CAN1 Local Mask B Register | C1LMBR | XXh | | |
| 036Dh | | | XXh | | |
| 036Eh | | | XXh | | |
| 036Fh | | | XXh | | |
| 0370h | | | XXh | | |
| 0371h | | | XXh | | |
| 0372h | | | | | |
| 0373h | | | | | |
| 0374h | | | | | |
| 0375h | | | | | |
| 0376h | | | | | |
| 0377h | | | | | |
| 0378h | | | | | |
| 0379h | | | | | |
| 037Ah | | | | | |
| 037Bh | | | | | |
| 037Ch | | | | | |
| 037Dh | | | | | |
| 037Eh | | | | | |
| 037Fh | | | | | |

X: Undefined

NOTE:

- Blank spaces are reserved. No access is allowed.

Table 4.15 SFR Information (15) ⁽²⁾

| Address | Register | Symbol | After Reset |
|---------|---|--------|--------------------|
| 0380h | Count Start Flag | TABSR | 00h |
| 0381h | Clock Prescaler Reset Flag | CPSRF | 0XXXXXXb |
| 0382h | One-Shot Start Flag | ONSF | 00h |
| 0383h | Trigger Select Register | TRGSR | 00h |
| 0384h | Up/Down Flag | UDF | 00h ⁽¹⁾ |
| 0385h | | | |
| 0386h | Timer A0 Register | TA0 | XXh |
| 0387h | | | XXh |
| 0388h | Timer A1 Register | TA1 | XXh |
| 0389h | | | XXh |
| 038Ah | Timer A2 Register | TA2 | XXh |
| 038Bh | | | XXh |
| 038Ch | Timer A3 Register | TA3 | XXh |
| 038Dh | | | XXh |
| 038Eh | Timer A4 Register | TA4 | XXh |
| 038Fh | | | XXh |
| 0390h | Timer B0 Register | TB0 | XXh |
| 0391h | | | XXh |
| 0392h | Timer B1 Register | TB1 | XXh |
| 0393h | | | XXh |
| 0394h | Timer B2 Register | TB2 | XXh |
| 0395h | | | XXh |
| 0396h | Timer A0 Mode Register | TA0MR | 00h |
| 0397h | Timer A1 Mode Register | TA1MR | 00h |
| 0398h | Timer A2 Mode Register | TA2MR | 00h |
| 0399h | Timer A3 Mode Register | TA3MR | 00h |
| 039Ah | Timer A4 Mode Register | TA4MR | 00h |
| 039Bh | Timer B0 Mode Register | TB0MR | 00XX0000b |
| 039Ch | Timer B1 Mode Register | TB1MR | 00XX0000b |
| 039Dh | Timer B2 Mode Register | TB2MR | 00XX0000b |
| 039Eh | Timer B2 Special Mode Register | TB2SC | XXXXXX00b |
| 039Fh | | | |
| 03A0h | UART0 Transmit/Receive Mode Register | U0MR | 00h |
| 03A1h | UART0 Bit Rate Register | U0BRG | XXh |
| 03A2h | UART0 Transmit Buffer Register | U0TB | XXh |
| 03A3h | | | XXh |
| 03A4h | UART0 Transmit/Receive Control Register 0 | U0C0 | 00001000b |
| 03A5h | UART0 Transmit/Receive Control Register 1 | U0C1 | 00XX0010b |
| 03A6h | UART0 Receive Buffer Register | U0RB | XXh |
| 03A7h | | | XXh |
| 03A8h | UART1 Transmit/Receive Mode Register | U1MR | 00h |
| 03A9h | UART1 Bit Rate Register | U1BRG | XXh |
| 03AAh | UART1 Transmit Buffer Register | U1TB | XXh |
| 03ABh | | | XXh |
| 03ACh | UART1 Transmit/Receive Control Register 0 | U1C0 | 00001000b |
| 03ADh | UART1 Transmit/Receive Control Register 1 | U1C1 | 00XX0010b |
| 03AEh | UART1 Receive Buffer Register | U1RB | XXh |
| 03AFh | | | XXh |
| 03B0h | UART Transmit/Receive Control Register 2 | UCON | X0000000b |
| 03B1h | | | |
| 03B2h | | | |
| 03B3h | | | |
| 03B4h | | | |
| 03B5h | | | |
| 03B6h | | | |
| 03B7h | | | |
| 03B8h | DMA0 Request Source Select Register | DM0SL | 00h |
| 03B9h | | | |
| 03BAh | DMA1 Request Source Select Register | DM1SL | 00h |
| 03BBh | | | |
| 03BCh | CRC Data Register | CRCD | XXh |
| 03BDh | | | XXh |
| 03BEh | CRC Input Register | CRCIN | XXh |
| 03BFh | | | |

X: Undefined

NOTES:

1. Bits TA2P to TA4P in the UDF register are set to 0 after reset. However, the contents in these bits are undefined when read.
2. Blank spaces are reserved. No access is allowed.

Table 4.16 SFR Information (16) ⁽²⁾

| Address | Register | Symbol | After Reset |
|---------|-----------------------------|--------|---------------------------------------|
| 03C0h | A/D Register 0 | AD0 | XXh |
| 03C1h | | | XXh |
| 03C2h | A/D Register 1 | AD1 | XXh |
| 03C3h | | | XXh |
| 03C4h | A/D Register 2 | AD2 | XXh |
| 03C5h | | | XXh |
| 03C6h | A/D Register 3 | AD3 | XXh |
| 03C7h | | | XXh |
| 03C8h | A/D Register 4 | AD4 | XXh |
| 03C9h | | | XXh |
| 03CAh | A/D Register 5 | AD5 | XXh |
| 03CBh | | | XXh |
| 03CCh | A/D Register 6 | AD6 | XXh |
| 03CDh | | | XXh |
| 03CEh | A/D Register 7 | AD7 | XXh |
| 03CFh | | | XXh |
| 03D0h | | | |
| 03D1h | | | |
| 03D2h | | | |
| 03D3h | | | |
| 03D4h | A/D Control Register 2 | ADCON2 | 00h |
| 03D5h | | | |
| 03D6h | A/D Control Register 0 | ADCON0 | 00000XXXb |
| 03D7h | A/D Control Register 1 | ADCON1 | 00h |
| 03D8h | D/A Register 0 | DA0 | 00h |
| 03D9h | | | |
| 03DAh | D/A Register 1 | DA1 | 00h |
| 03DBh | | | |
| 03DCh | D/A Control Register | DACON | 00h |
| 03DDh | | | |
| 03DEh | | | |
| 03DFh | | | |
| 03E0h | Port P0 Register | P0 | XXh |
| 03E1h | Port P1 Register | P1 | XXh |
| 03E2h | Port P0 Direction Register | PD0 | 00h |
| 03E3h | Port P1 Direction Register | PD1 | 00h |
| 03E4h | Port P2 Register | P2 | XXh |
| 03E5h | Port P3 Register | P3 | XXh |
| 03E6h | Port P2 Direction Register | PD2 | 00h |
| 03E7h | Port P3 Direction Register | PD3 | 00h |
| 03E8h | Port P4 Register | P4 | XXh |
| 03E9h | Port P5 Register | P5 | XXh |
| 03EAh | Port P4 Direction Register | PD4 | 00h |
| 03EBh | Port P5 Direction Register | PD5 | 00h |
| 03ECh | Port P6 Register | P6 | XXh |
| 03EDh | Port P7 Register | P7 | XXh |
| 03EEh | Port P6 Direction Register | PD6 | 00h |
| 03EFh | Port P7 Direction Register | PD7 | 00h |
| 03F0h | Port P8 Register | P8 | XXh |
| 03F1h | Port P9 Register | P9 | XXh |
| 03F2h | Port P8 Direction Register | PD8 | 00X00000b |
| 03F3h | Port P9 Direction Register | PD9 | 00h |
| 03F4h | Port P10 Register | P10 | XXh |
| 03F5h | | | |
| 03F6h | Port P10 Direction Register | PD10 | 00h |
| 03F7h | | | |
| 03F8h | | | |
| 03F9h | | | |
| 03FAh | | | |
| 03FBh | | | |
| 03FCh | Pull-up Control Register 0 | PUR0 | 00h |
| 03FDh | Pull-up Control Register 1 | PUR1 | 00000000b ⁽¹⁾ 00000010b |
| 03FEh | Pull-up Control Register 2 | PUR2 | 00h |
| 03FFh | Port Control Register | PCR | 00h |

X: Undefined

NOTES:

1. At hardware reset, the register is as follows:

- 00000000b where "L" is input to the CNVSS pin
- 00000010b where "H" is input to the CNVSS pin

At software reset, watchdog timer reset and oscillation stop detection reset, the register is as follows:

- 00000000b where bits PM01 to PM00 in the PM0 register are 00b (single-chip mode)
- 00000010b where bits PM01 to PM00 in the PM0 register are 01b (memory expansion mode) or 11b (microprocessor mode)

2. Blank spaces are reserved. No access is allowed.

5. Resets

Hardware reset, software reset, watchdog timer reset, and oscillation stop detection reset are available to reset the MCU.

5.1 Hardware Reset

The MCU resets pins, the CPU and SFR by setting the $\overline{\text{RESET}}$ pin. If the supply voltage meets the recommended operating conditions, the MCU resets all pins when an “L” signal is applied to the $\overline{\text{RESET}}$ pin (see **Table 5.1 Pin Status When $\overline{\text{RESET}}$ Pin Level is “L”**). The oscillation circuit is also reset and the main clock starts oscillation. The MCU resets the CPU and SFR when the signal applied to the $\overline{\text{RESET}}$ pin changes low (“L”) to high (“H”). The MCU executes the program in an 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 internal RAM is in an undefined state.

Figure 5.1 shows an Example Reset Circuit. Figure 5.2 shows a Reset Sequence. Table 5.1 lists the Pin States when $\overline{\text{RESET}}$ Pin Level is “L”.

5.1.1 Reset on a Stable Supply Voltage

- (1) Apply “L” to the $\overline{\text{RESET}}$ pin
- (2) Apply 20 or more clock cycles to the XIN pin
- (3) Apply “H” to the $\overline{\text{RESET}}$ pin

5.1.2 Power-on Reset

- (1) Apply “L” to the $\overline{\text{RESET}}$ pin
- (2) Raise the supply voltage to the recommended operating level
- (3) Insert $t_d(\text{P-R})$ ms as wait time for the internal voltage to stabilize
- (4) Apply 20 or more clock cycles to the XIN pin
- (5) Apply “H” to the $\overline{\text{RESET}}$ pin

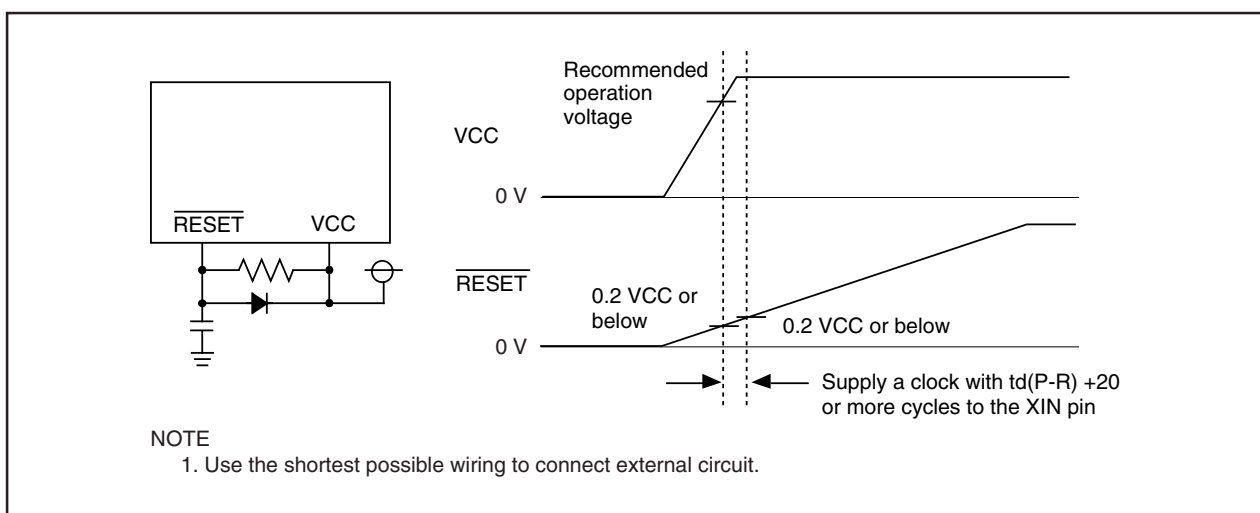


Figure 5.1 Example Reset Circuit

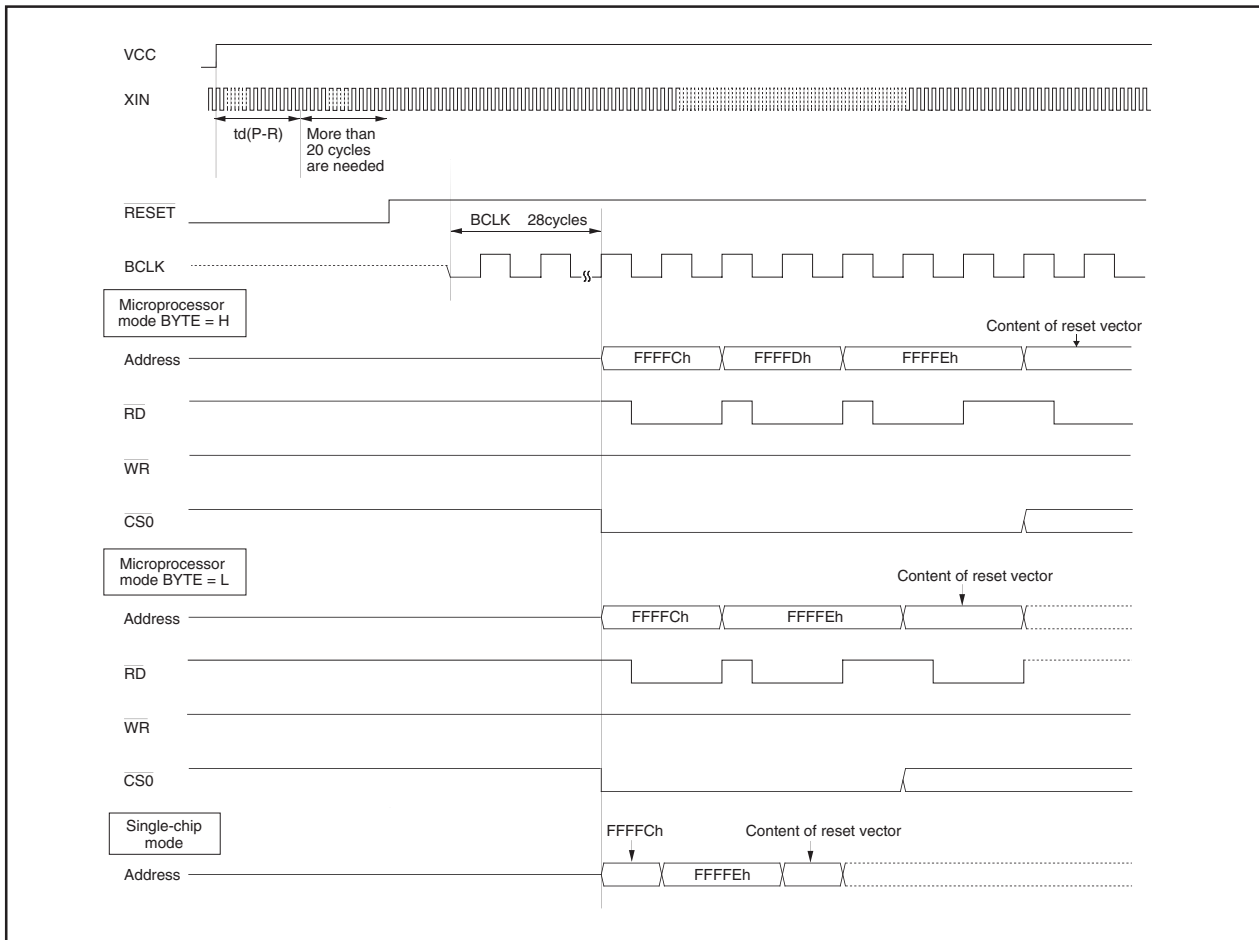


Figure 5.2 Reset Sequence

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

| Pin Name | Status | | |
|---|-------------|--|--|
| | CNVSS = VSS | CNVSS = VCC ⁽¹⁾ | |
| | | BYTE = VSS | BYTE = VCC |
| P0 | Input port | Data input | Data input |
| P1 | Input port | Data input | Input port |
| P2, P3, P4_0 to P4_3 | Input port | Address output (undefined) | Address output (undefined) |
| P4_4 | Input port | CS0 output (“H” is output) | CS0 output (“H” is output) |
| P4_5 to P4_7 | Input port | Input port (Pulled high) | Input port (Pulled high) |
| P5_0 | Input port | WR output (“H” is output) | WR output (“H” is output) |
| P5_1 | Input port | BHE output (undefined) | BHE output (undefined) |
| P5_2 | Input port | RD output (“H” is output) | RD output (“H” is output) |
| P5_3 | Input port | BCLK output | BCLK output |
| P5_4 | Input port | HLDA output (The output value depends on the input to the HOLD pin) | HLDA output (The output value depends on the input to the HOLD pin) |
| P5_5 | Input port | HOLD input | HOLD input |
| P5_6 | Input port | ALE output (“L” is output) | ALE output (“L” is output) |
| P5_7 | Input port | RDY input | RDY input |
| P6, P7, P8_0 to P8_4, P8_6, P8_7, P9, P10 | Input port | Input port | Input port |

NOTE:

1. Shown here is the valid pin state when the internal power supply voltage has stabilized after power-on. When CNVSS = VCC, the pin state is undefined until the internal power supply voltage stabilizes.

5.2 Software Reset

The MCU resets pins, the CPU and SFR when the PM03 bit in the PM0 register is set to 1 (MCU reset). Then the MCU executes the program in an address determined by the reset vector. Set the PM03 bit to 1 while the main clock is selected as the CPU clock and the main clock oscillation is stable. In the software reset, the MCU does not reset a part of the SFR. Refer to **4. Special Function Registers (SFRs)** for details. Processor mode remains unchanged since bits PM01 to PM00 in the PM0 register are not reset.

5.3 Watchdog Timer Reset

The MCU resets pins, the CPU and SFR when the PM12 bit in the PM1 register is set to 1 (reset when watchdog timer underflows) and the watchdog timer underflows. Then the MCU executes the program in an address determined by the reset vector. In the watchdog timer reset, the MCU does not reset a part of the SFR. Refer to **4. Special Function Registers (SFRs)** for details. Processor mode remains unchanged since bits PM01 to PM00 in the PM0 register are not reset.

5.4 Oscillation Stop Detection Reset

The MCU resets and stops pins, the CPU and SFR when the CM27 bit in the CM2 register is 0 (reset at oscillation stop, re-oscillation detection), if it detects main clock oscillation circuit stop. Refer to **8.5 Oscillation Stop and Re-Oscillation Detection Function** for details. In the oscillation stop detection reset, the MCU does not reset a part of the SFR. Refer to **4. Special Function Registers (SFRs)** for details. Processor mode remains unchanged since bits PM01 to PM00 in the PM0 register are not reset.

5.5 Internal Space

Figure 5.3 shows CPU Register Status After Reset. Refer to **4. Special Function Registers (SFRs)** for SFR states after reset.

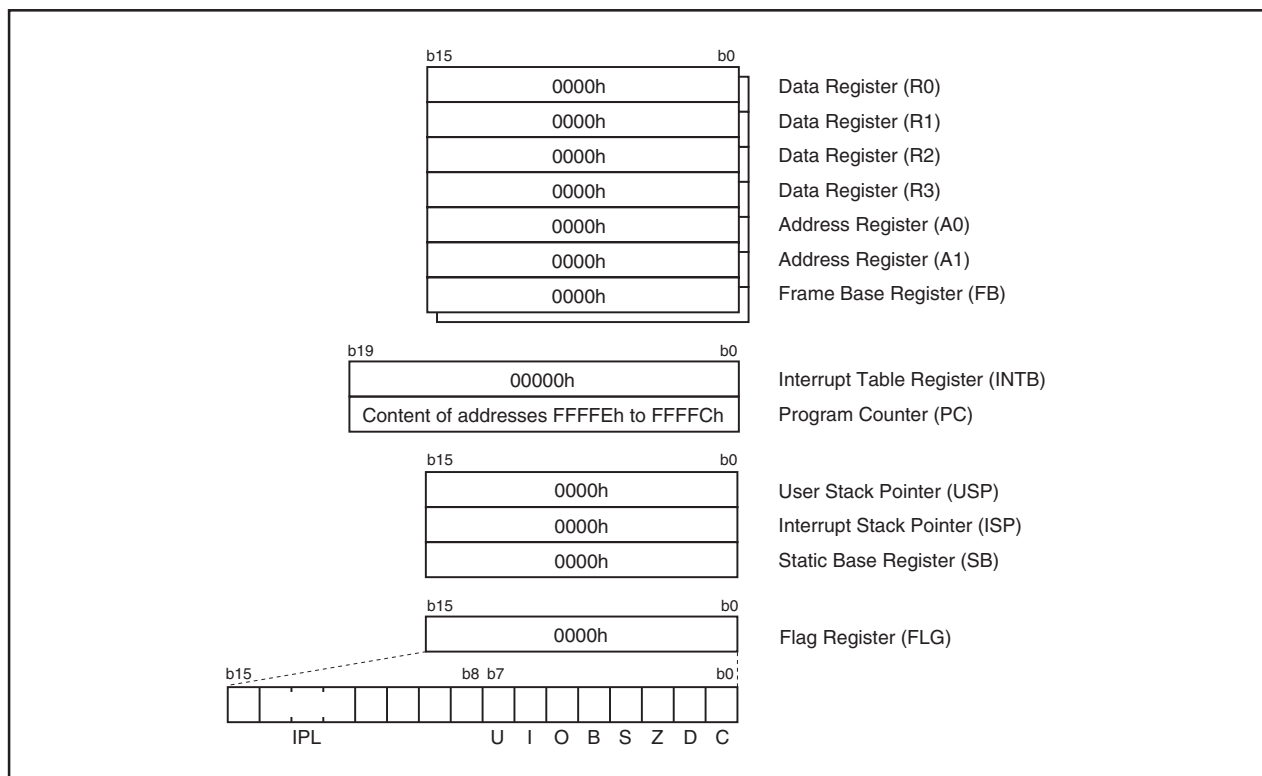


Figure 5.3 CPU Register Status After Reset

6. Processor Mode

6.1 Types of Processor Mode

Three processor modes are available to choose from: single-chip mode, memory expansion mode, and microprocessor mode. Table 6.1 shows the Features of Processor Modes.

Table 6.1 Features of Processor Modes

| Processor Mode | Access Space | Pins Which are Assigned I/O Ports |
|-----------------------|---|--|
| Single-chip mode | SFR, internal RAM, internal ROM | All pins are I/O ports or peripheral function I/O pins |
| Memory expansion mode | SFR, internal RAM, internal ROM, external area ⁽¹⁾ | Some pins serve as bus control pins ⁽¹⁾ |
| Microprocessor mode | SFR, internal RAM, external area ⁽¹⁾ | Some pins serve as bus control pins ⁽¹⁾ |

NOTE:

1. Refer to 7. Bus.

6.2 Setting Processor Modes

Processor mode is set by using the CNVSS pin and bits PM01 to PM00 in the PM0 register.

Table 6.2 shows the Processor Mode after Hardware Reset. Table 6.3 shows Bits PM01 to PM00 Set Values and Processor Modes.

Table 6.2 Processor Mode after Hardware Reset

| CNVSS Pin Input Level | Processor Mode |
|------------------------|---------------------|
| VSS | Single-chip mode |
| VCC ^{(1) (2)} | Microprocessor mode |

NOTES:

1. If the MCU is reset in hardware by applying VCC to the CNVSS pin, the internal ROM cannot be accessed regardless of bits PM01 to PM00.
2. The multiplexed bus cannot be assigned to the entire \overline{CS} space.

Table 6.3 Bits PM01 to PM00 Set Values and Processor Modes

| Bits PM01 to PM00 | Processor Mode |
|-------------------|-----------------------|
| 00b | Single-chip mode |
| 01b | Memory expansion mode |
| 10b | Do not set a value |
| 11b | Microprocessor mode |

Rewriting bits PM01 to PM00 places the MCU in the corresponding processor mode regardless of whether the input level on the CNVSS pin is "H" or "L". Note, however, that bits PM01 to PM00 cannot be rewritten to 01b (memory expansion mode) or 11b (microprocessor mode) at the same time bits PM07 to PM02 are rewritten. Note also that these bits cannot be rewritten to enter microprocessor mode in the internal ROM, nor can they be rewritten to exit microprocessor mode in areas overlapping the internal ROM.

If the MCU is reset in hardware by applying VCC to the CNVSS pin (hardware reset), the internal ROM cannot be accessed regardless of bits PM01 to PM00.

Figures 6.1 and 6.2 show the PM0 Register and PM1 Register. Figure 6.3 shows the Memory Map in Single-chip Mode. Figures 6.4 to 6.7 show the Memory Map and \overline{CS} Area in Memory Expansion Mode and Microprocessor Mode.

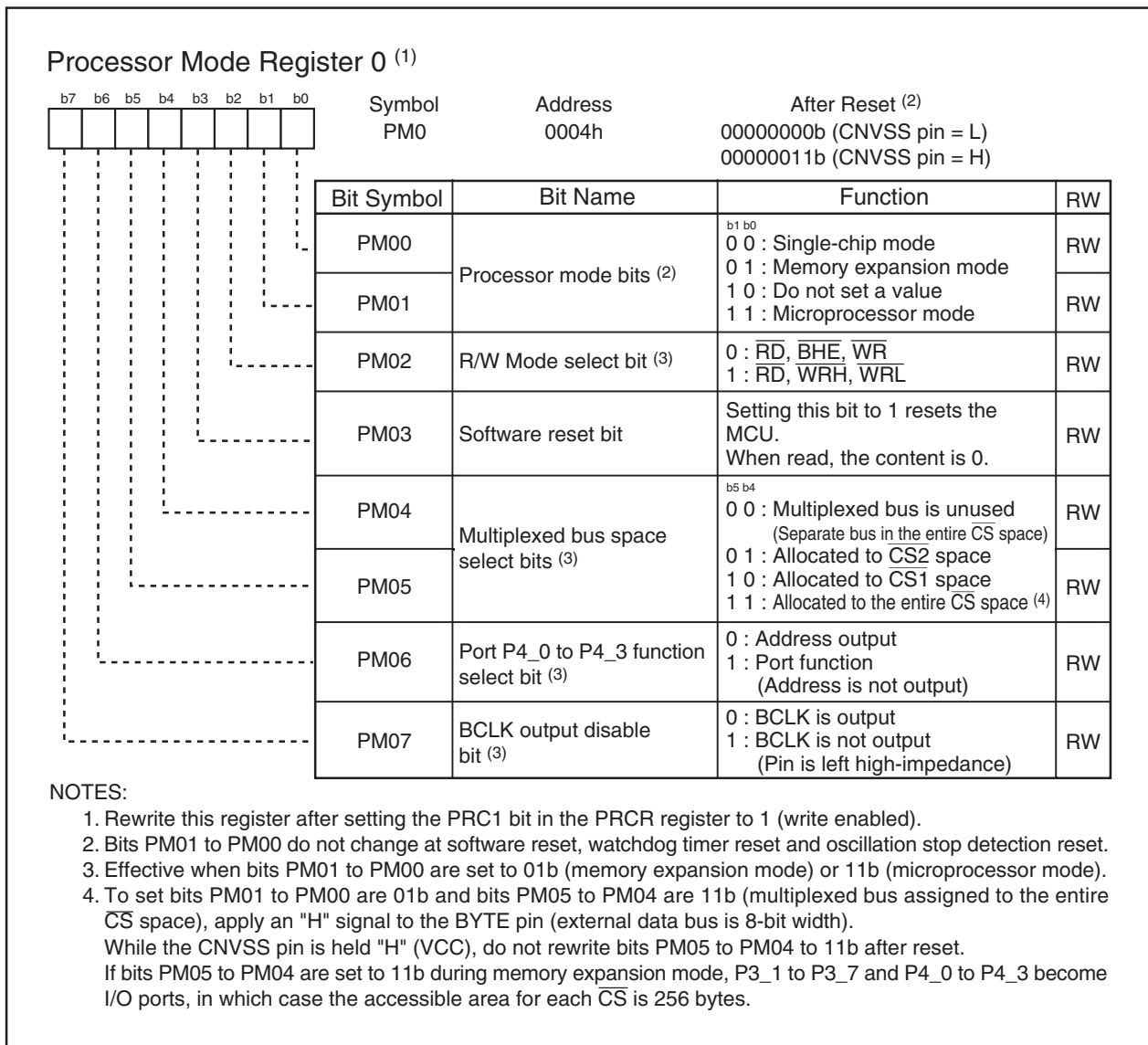


Figure 6.1 PM0 Register

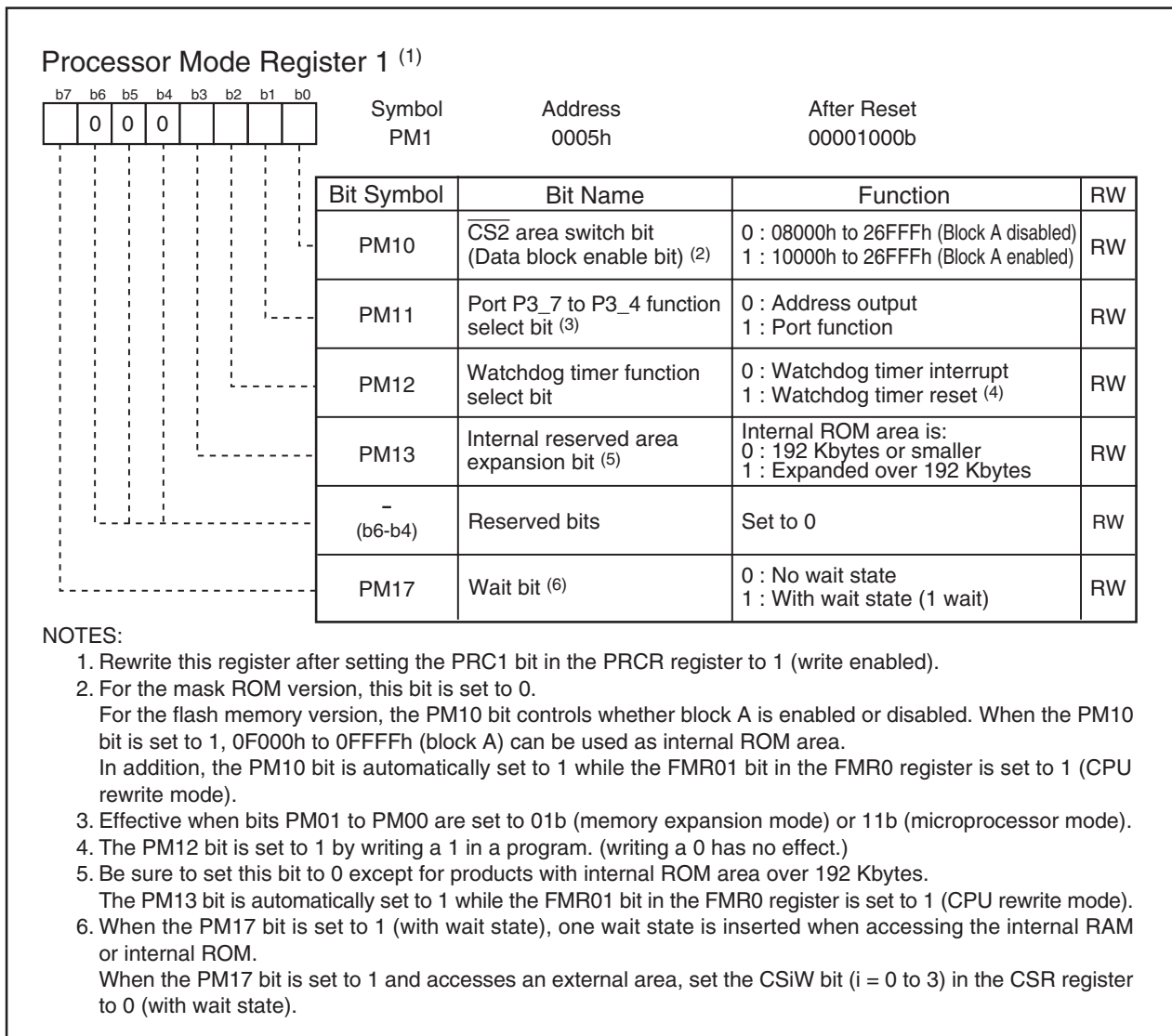


Figure 6.2 PM1 Register

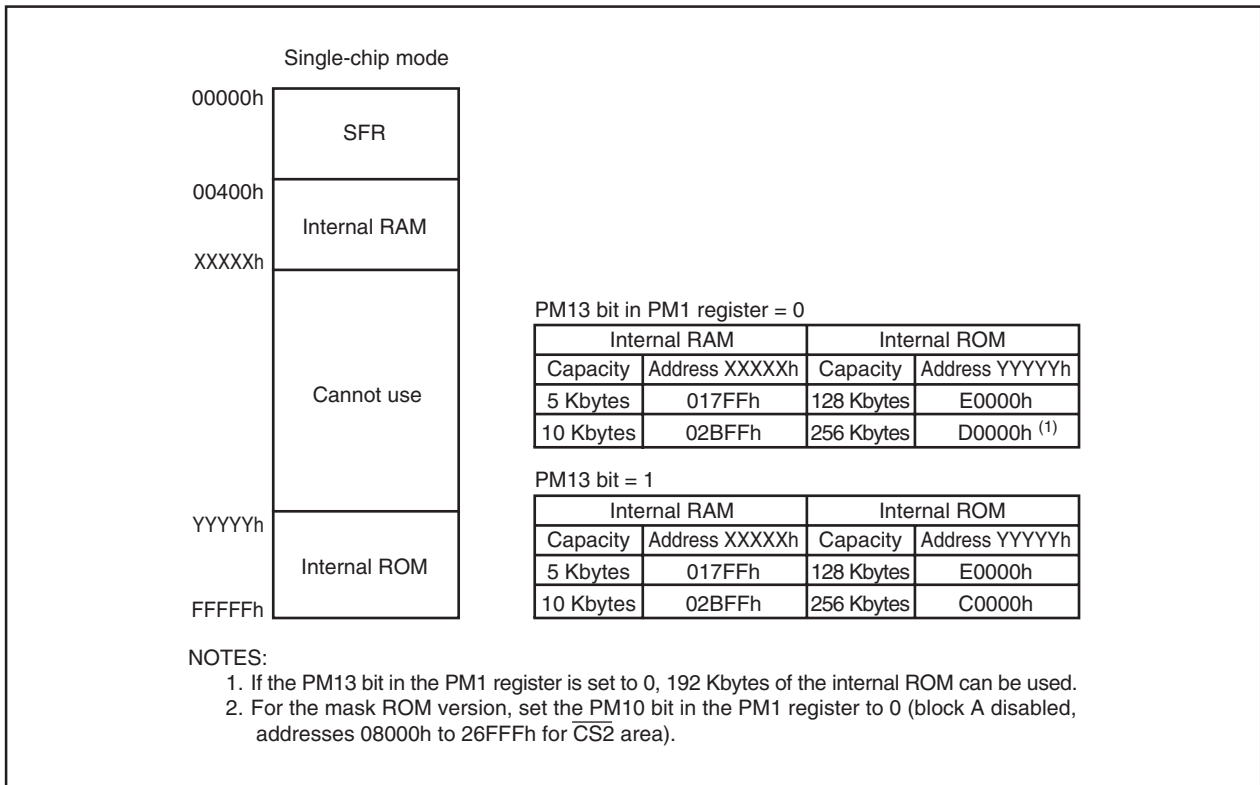


Figure 6.3 Memory Map in Single-chip Mode

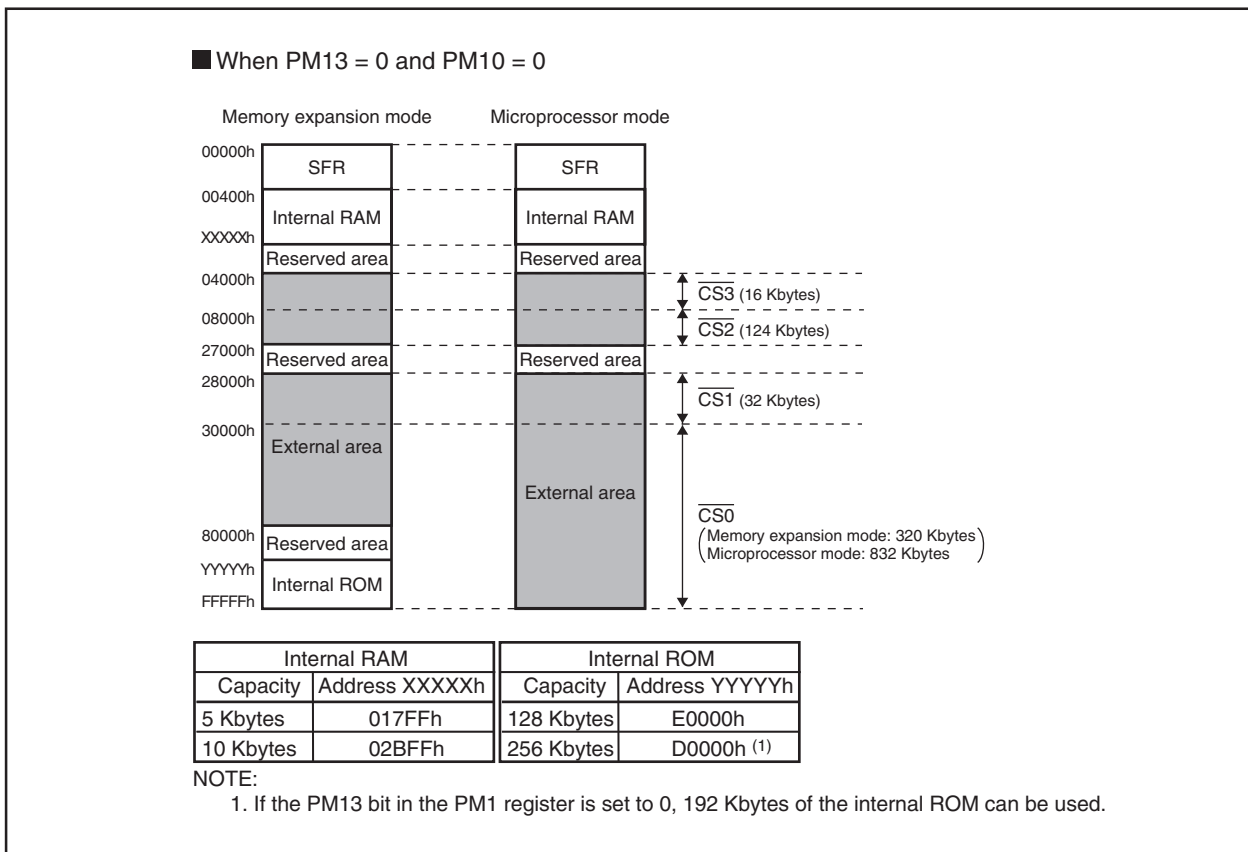


Figure 6.4 Memory Map and CS Area in Memory Expansion Mode and Microprocessor Mode (1)

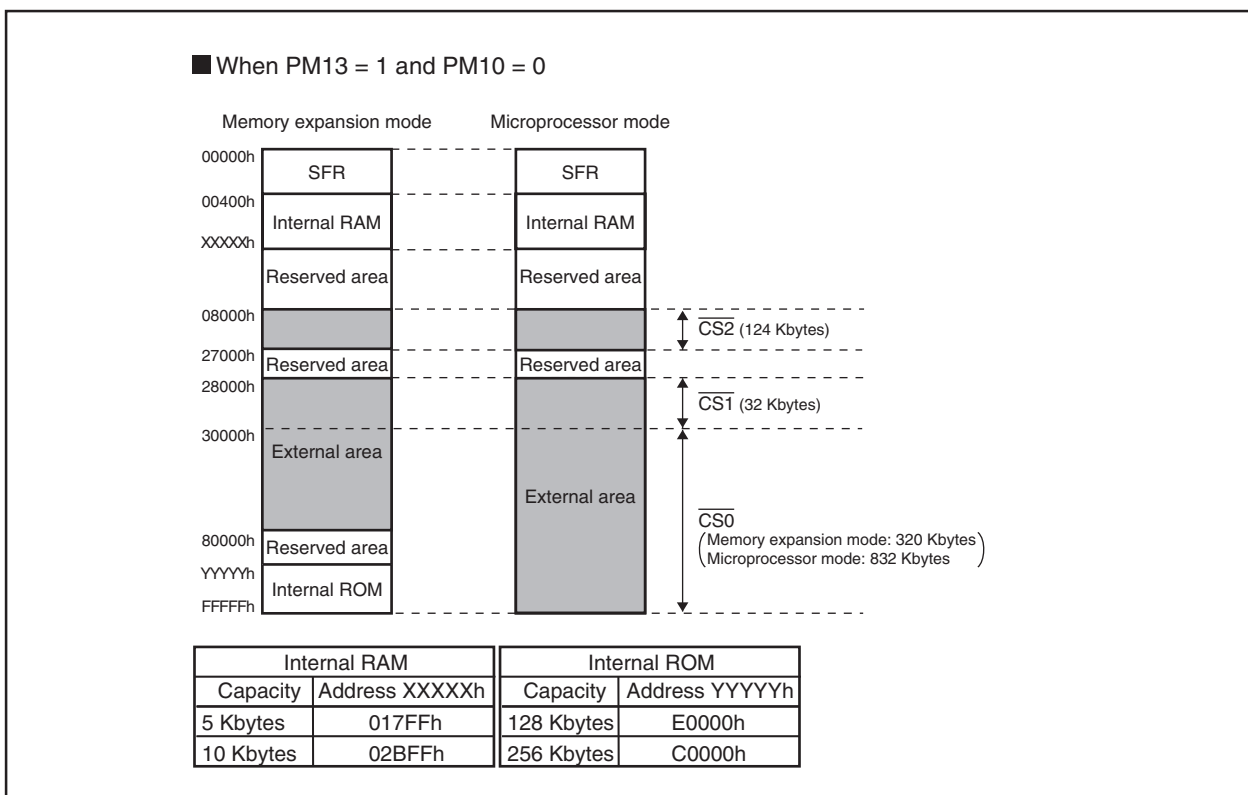


Figure 6.5 Memory Map and CS Area in Memory Expansion Mode and Microprocessor Mode (2)

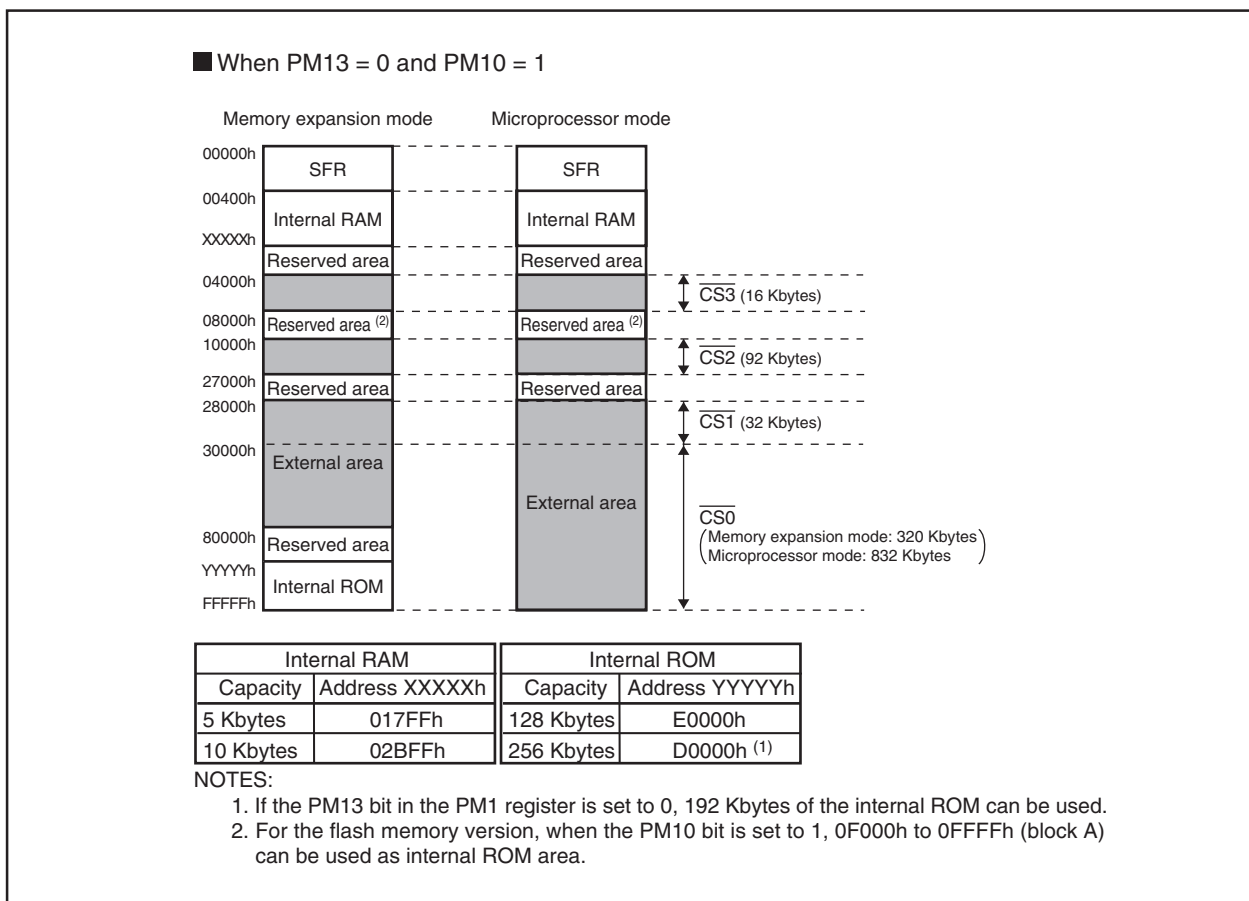


Figure 6.6 Memory Map and CS Area in Memory Expansion Mode and Microprocessor Mode (3)

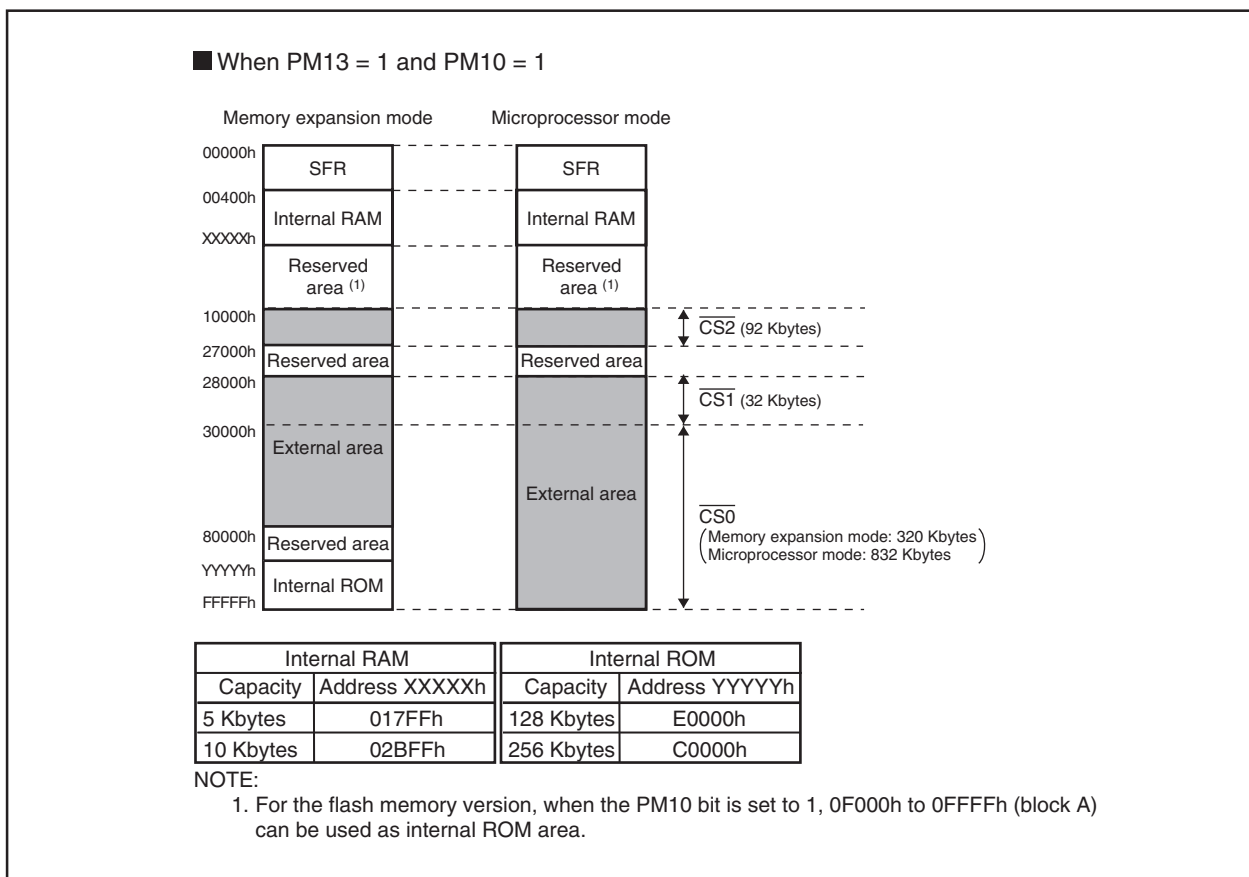


Figure 6.7 Memory Map and CS Area in Memory Expansion Mode and Microprocessor Mode (4)

7. Bus

During memory expansion or microprocessor mode, some pins serve as the bus control pins to perform data input/output to and from external devices. These bus control pins include A0 to A19, D0 to D15, $\overline{CS0}$ to $\overline{CS3}$, RD, WRL/WR, WRH/BHE, ALE, RDY, HOLD, HLDA, and BCLK.

7.1 Bus Mode

The bus mode, either multiplexed or separate, can be selected using bits PM05 to PM04 in the PM0 register.

7.1.1 Separate Bus

In this bus mode, data and address are separate.

7.1.2 Multiplexed Bus

In this bus mode, data and address are multiplexed.

7.1.2.1 When the input level on BYTE pin is high (8-bit data bus)

D0 to D7 and A0 to A7 are multiplexed.



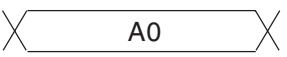

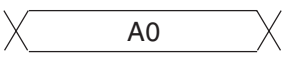
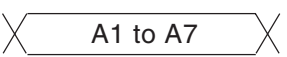


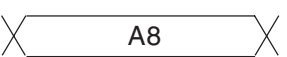
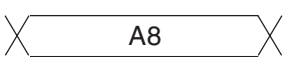

7.1.2.2 When the input level on BYTE pin is low (16-bit data bus)

D0 to D7 and A1 to A8 are multiplexed. D8 to D15 are not multiplexed. Do not use D8 to D15.

External devices connecting to a multiplexed bus are allocated to only the even addresses of the MCU. Odd addresses cannot be accessed.

Table 7.1 shows the Difference between Separate Bus and Multiplexed Bus.

Table 7.1 Difference between Separate Bus and Multiplexed Bus

| Pin Name ⁽¹⁾ | Separate Bus | Multiplexed Bus | |
|---|---|--|---|
| | | BYTE = H | BYTE = L |
| P0_0 to P0_7/D0 to D7 |  | (NOTE 2) | (NOTE 2) |
| P1_0 to P1_7/D8 to D15 |  | I/O Port P1_0 to P1_7 | (NOTE 2) |
| P2_0/A0(/D0/-) |  |  |  |
| P2_1 to P2_7/A1 to A7 (/D1 to D7/D0 to D6) |  |  |  |
| P3_0/A8(/-/D7) |  |  |  |

NOTES :

1. See **Table 7.6 Pin Functions for Each Processor Mode** for bus control signals other than the above.
2. It changes with a setup of bits PM05 to PM04 in the PM0 register, and area to access. See **Table 7.6 Pin Functions for Each Processor Mode** for details.

7.2 Bus Control

The following describes the signals needed for accessing external devices and the functionality of software wait.

7.2.1 Address Bus

The address bus consists of 20 lines, A0 to A19. The address bus width can be chosen to be 12, 16 or 20 bits by using the PM06 bit in the PM0 register and the PM11 bit in the PM1 register. Table 7.2 shows Bits PM06 and PM11 Set Values and Address Bus Widths.

When processor mode is changed from single-chip mode to memory expansion mode, the address bus is undefined until any external area is accessed.

Table 7.2 Bits PM06 and PM11 Set Value and Address Bus Width

| Set Value ⁽¹⁾ | Pin Function | Address Bus Width |
|--------------------------|--------------|-------------------|
| PM11 = 1 | P3_4 to P3_7 | 12 bits |
| PM06 = 1 | P4_0 to P4_3 | |
| PM11 = 0 | A12 to A15 | 16 bits |
| PM06 = 1 | P4_0 to P4_3 | |
| PM11 = 0 | A12 to A15 | 20 bits |
| PM06 = 0 | A16 to A19 | |

NOTE:

- No values other than those shown above can be set.

7.2.2 Data Bus

When input on the BYTE pin is high (data bus is an 8-bit width), 8 lines D0 to D7 comprise the data bus; when input on the BYTE pin is low (data bus is a 16-bit width), 16 lines D0 to D15 comprise the data bus. Do not change the input level on the BYTE pin while in operation.

7.2.3 Chip Select Signal

The chip select (hereafter referred to as the \overline{CS}) signals are output from the \overline{CS}_i ($i = 0$ to 3) pins. These pins can be chosen to function as I/O ports or as \overline{CS} by using the CS_i bit in the CSR register. Figure 7.1 shows the CSR Register.

During 1 Mbyte mode, the external area can be separated into up to 4 by the \overline{CS}_i signal which is output from the \overline{CS}_i pin.

Figure 7.2 shows the Example of Address Bus and \overline{CS}_i Signal Output.

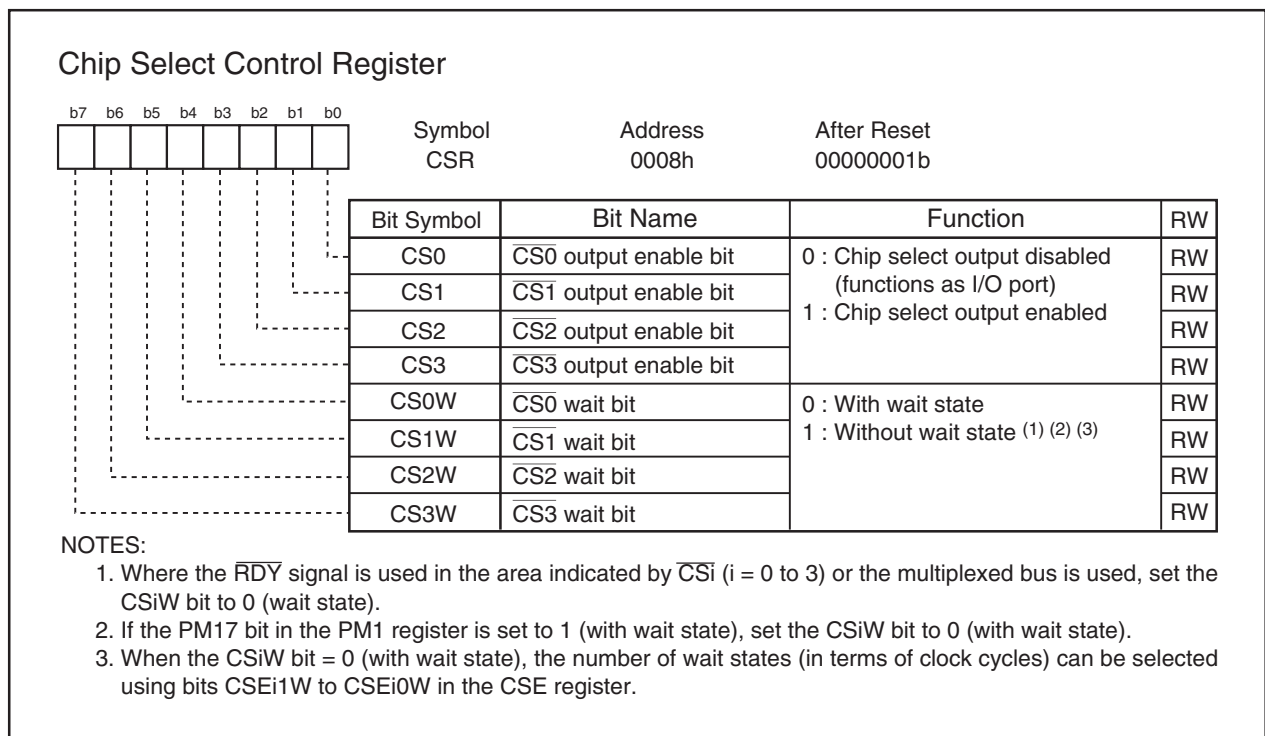
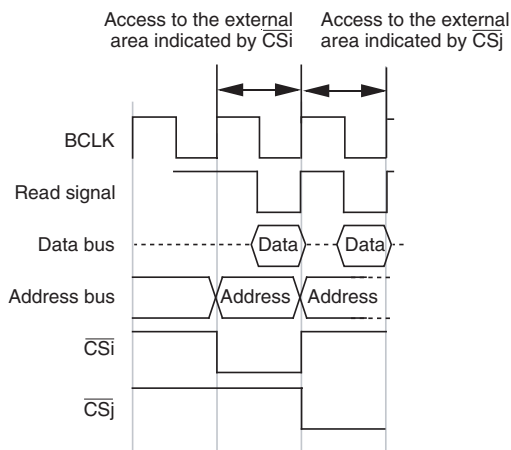


Figure 7.1 CSR Register

Example 1

To access the external area indicated by \overline{CSj} in the next cycle after accessing the external area indicated by \overline{CSi} .

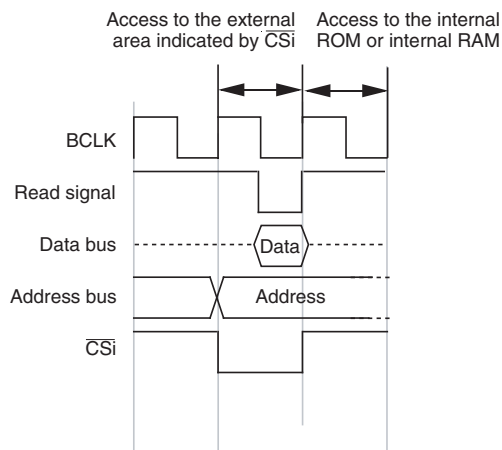
The address bus and the chip select signal both change state between these two cycles.



Example 2

To access the internal ROM or internal RAM in the next cycle after accessing the external area indicated by \overline{CSi} .

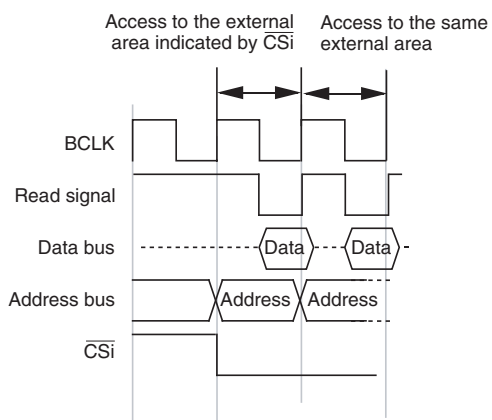
The chip select signal changes state but the address bus does not change state.



Example 3

To access the external area indicated by \overline{CSi} in the next cycle after accessing the external area indicated by the same \overline{CSi} .

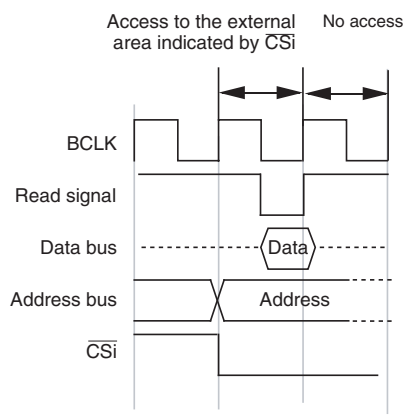
The address bus changes state but the chip select signal does not change state.



Example 4

Not to access any area (nor instruction prefetch generated) in the next cycle after accessing the external area indicated by \overline{CSi} .

Neither the address bus nor the chip select signal changes state between these two cycles.



NOTE:

1. These examples show the address bus and chip select signal when accessing areas in two successive cycles. The chip select bus cycle may be extended more than two cycles depending on a combination of these examples.

Shown above is the case where separate bus is selected and the area is accessed for read without wait states. $i = 0$ to 3 , $j = 0$ to 3 (not including i , however)

Figure 7.2 Example of Address Bus and CSi Signal Output

7.2.4 Read and Write Signals

When the data bus is 16-bit width, the read and write signals can be chosen to be a combination of \overline{RD} , \overline{WR} , and \overline{BHE} or a combination of \overline{RD} , \overline{WRL} , and \overline{WRH} by using the PM02 bit in the PM0 register. When the data bus is 8-bit width, use a combination of \overline{RD} , \overline{WR} , and \overline{BHE} .

Table 7.3 shows the Operation of \overline{RD} , \overline{WRL} , and \overline{WRH} Signals. Table 7.4 shows the Operation of \overline{RD} , \overline{WR} , and \overline{BHE} Signals.

Table 7.3 Operation of \overline{RD} , \overline{WRL} , and \overline{WRH} Signals

| Data Bus Width | RD | WRL | WRH | Status of External Data Bus |
|------------------------------------|----|-----|-----|---|
| 16 bits (BYTE pin input = L) | L | H | H | Read data |
| | H | L | H | Write 1 byte of data to an even address |
| | H | H | L | Write 1 byte of data to an odd address |
| | H | L | L | Write data to both even and odd addresses |

Table 7.4 Operation of \overline{RD} , \overline{WR} , and \overline{BHE} Signals

| Data Bus Width | RD | WR | BHE | A0 | Status of External Data Bus |
|------------------------------------|----|----|----------|--------|--|
| 16 bits (BYTE pin input = L) | H | L | L | H | Write 1 byte of data to an odd address |
| | L | H | L | H | Read 1 byte of data from an odd address |
| | H | L | H | L | Write 1 byte of data to an even address |
| | L | H | H | L | Read 1 byte of data from an even address |
| | H | L | L | L | Write data to both even and odd addresses |
| | L | H | L | L | Read data from both even and odd addresses |
| 8 bits (BYTE pin input = H) | H | L | Not used | H to L | Write 1 byte of data |
| | L | H | Not used | H to L | Read 1 byte of data |

7.2.5 ALE Signal

The ALE signal latches the address when accessing the multiplexed bus space. Latch the address when the ALE signal falls.

Figure 7.3 shows the ALE Signal, Address Bus and Data Bus.

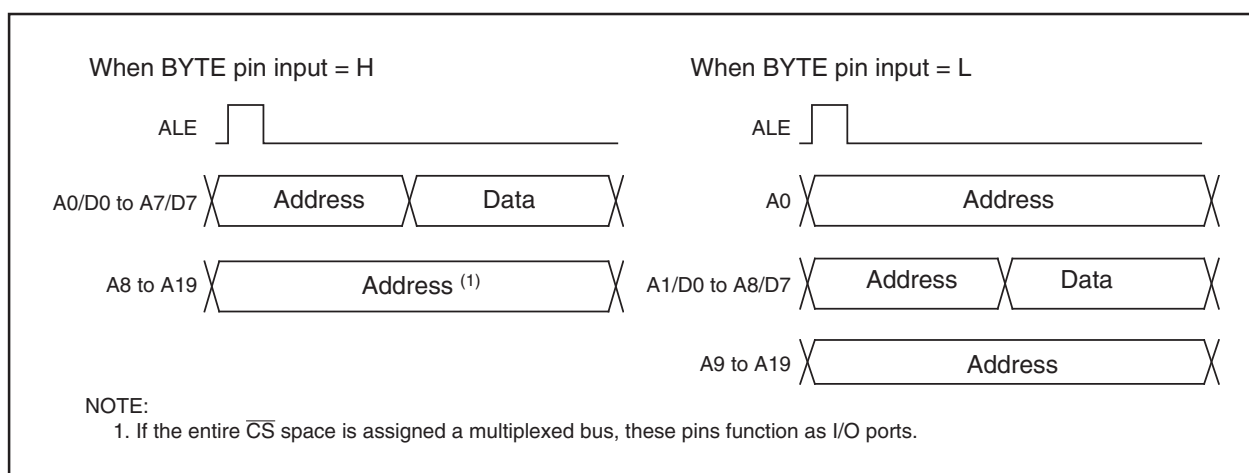


Figure 7.3 ALE Signal, Address Bus, and Data Bus

7.2.6 RDY Signal

This signal is provided for accessing external devices which need to be accessed at low speed. If input on the RDY pin is asserted low at the last falling edge of BCLK of the bus cycle, one wait state is inserted in the bus cycle. While in a wait state, the following signals retain the state in which they were when the RDY signal was acknowledged.

A0 to A19, D0 to D15, CS0 to CS3, RD, WRL, WRH, WR, BHE, ALE, HLDA

Then, when the input on the RDY pin is detected high at the falling edge of BCLK, the remaining bus cycle is executed. Figure 7.4 shows an Example in which Wait State was Inserted into Read Cycle by RDY Signal. To use the RDY signal, set the corresponding bit (bits CS3W to CS0W) in the CSR register to 0 (with wait state). When not using the RDY signal, the RDY pin must be pulled-up.

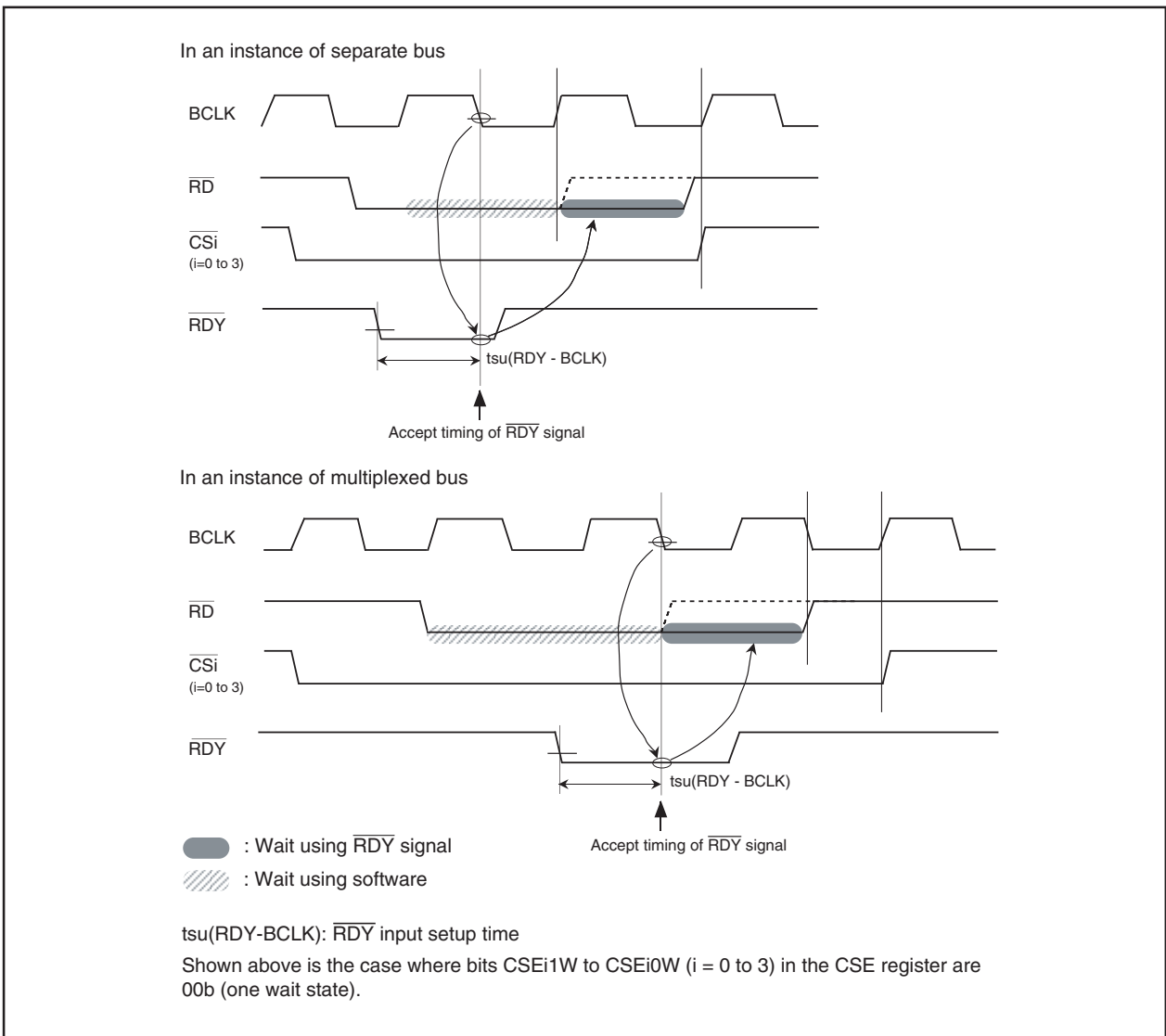


Figure 7.4 Example in which Wait State was Inserted into Read Cycle by RDY Signal

7.2.7 $\overline{\text{HOLD}}$ Signal

This signal is used to transfer control of the bus from the CPU or DMAC to an external circuit. When the input on $\overline{\text{HOLD}}$ pin is pulled low, the MCU is placed in a hold state after the bus access then in process finishes. The MCU remains in a hold state while the $\overline{\text{HOLD}}$ pin is held low, during which time the $\overline{\text{HLDA}}$ pin outputs a low-level signal.

Table 7.5 shows the MCU Status in Hold State.

Bus-using priorities are given to $\overline{\text{HOLD}}$, DMAC, and CPU in order of decreasing precedence (see **Figure 7.5 Bus-using Priorities**). However, if the CPU is accessing an odd address in word units, the DMAC cannot gain control of the bus during two separate accesses.

| |
|---|
| $\overline{\text{HOLD}} > \text{DMAC} > \text{CPU}$ |
|---|

Figure 7.5 Bus-using Priorities

Table 7.5 MCU Status in Hold State

| Item | | Status |
|---|-------------------------------|---|
| BCLK | | Output |
| A0 to A19, D0 to D15, CS0 to CS3, RD, WRL, WRH, WR, BHE | | High-impedance |
| I/O ports | P0, P1, P3, P4 ⁽¹⁾ | High-impedance |
| | P6 to P10 | Maintains status when hold signal is received |
| $\overline{\text{HLDA}}$ | | Output "L" |
| Internal peripheral circuits | | ON (but watchdog timer stops ⁽²⁾) |
| ALE signal | | Undefined |

NOTES:

1. When I/O port function is selected.
2. The watchdog timer does not stop when the PM22 bit in the PM2 register is set to 1 (the count source for the watchdog timer is the on-chip oscillator clock).

7.2.8 BCLK Output

If the PM07 bit in the PM0 register is set to 0 (output enable), a clock with the same frequency as that of the CPU clock is output as BCLK from the BCLK pin. Refer to **8.2 CPU Clock and Peripheral Function Clock**.

Table 7.6 shows the Pin Functions for Each Processor Mode.

Table 7.6 Pin Functions for Each Processor Mode

| Processor Mode | Memory Expansion Mode or Microprocessor Mode | | | | Memory Expansion Mode |
|----------------------------|--|-------------------------|--|--------------------------------------|---|
| Bits PM05 to PM04 | 00b (separate bus) | | 01b ($\overline{\text{CS2}}$ is for multiplexed bus and others are for separate bus) 10b ($\overline{\text{CS1}}$ is for multiplexed bus and others are for separate bus) | | 11b (multiplexed bus for the entire space) ⁽¹⁾ |
| Data bus width BYTE pin | 8 bits “H” | 16 bits “L” | 8 bits “H” | 16 bits “L” | 8 bits “H” |
| P0_0 to P0_7 | D0 to D7 | | D0 to D7 ⁽⁴⁾ | | I/O ports |
| P1_0 to P1_7 | I/O ports | D8 to D15 | I/O ports | D8 to D15 ⁽⁴⁾ | I/O ports |
| P2_0 | A0 | | A0/D0 ⁽²⁾ | A0 | A0/D0 |
| P2_1 to P2_7 | A1 to A7 | | A1 to A7 /D1 to D7 ⁽²⁾ | A1 to A7 /D0 to D6 ⁽²⁾ | A1 to A7/D1 to D7 |
| P3_0 | A8 | | | A8/D7 ⁽²⁾ | A8 |
| P3_1 to P3_3 | A9 to A11 | | | | I/O ports |
| P3_4 to P3_7 | PM11 = 0 | A12 to A15 | | | I/O ports |
| | PM11 = 1 | I/O ports | | | |
| P4_0 to P4_3 | PM06 = 0 | A16 to A19 | | | I/O ports |
| | PM06 = 1 | I/O ports | | | |
| P4_4 | CS0 = 0 | I/O ports | | | |
| | CS0 = 1 | $\overline{\text{CS0}}$ | | | |
| P4_5 | CS1 = 0 | I/O ports | | | |
| | CS1 = 1 | CS1 | | | |
| P4_6 | CS2 = 0 | I/O ports | | | |
| | CS2 = 1 | CS2 | | | |
| P4_7 | CS3 = 0 | I/O ports | | | |
| | CS3 = 1 | CS3 | | | |
| P5_0 | PM02 = 0 | WR | | | |
| | PM02 = 1 | – ⁽³⁾ | $\overline{\text{WRL}}$ | – ⁽³⁾ | $\overline{\text{WRL}}$ |
| P5_1 | PM02 = 0 | BHE | | | |
| | PM02 = 1 | – ⁽³⁾ | $\overline{\text{WRH}}$ | – ⁽³⁾ | $\overline{\text{WRH}}$ |
| P5_2 | RD | | | | |
| P5_3 | BCLK | | | | |
| P5_4 | HLDA | | | | |
| P5_5 | HOLD | | | | |
| P5_6 | ALE | | | | |
| P5_7 | RDY | | | | |

I/O ports: Function as I/O ports or peripheral function I/O pins.

NOTES:

- For setting bits PM01 to PM00 to 01b (memory expansion mode) and bits PM05 to PM04 to 11b (multiplexed bus assigned to the entire $\overline{\text{CS}}$ space), apply “H” to the BYTE pin (external data bus is an 8-bit width). While the CNVSS pin is held “H” (VCC), do not rewrite bits PM05 to PM04 to 11b after reset. If bits PM05 to PM04 are set to 11b during memory expansion mode, P3_1 to P3_7 and P4_0 to P4_3 become I/O ports, in which case the accessible area for each $\overline{\text{CS}}$ is 256 bytes.
- In separate bus mode, these pins serve as the address bus.
- If the data bus is 8-bit width, make sure the PM02 bit is set to 0 ($\overline{\text{RD}}$, $\overline{\text{BHE}}$, $\overline{\text{WR}}$).
- When accessing the area that uses a multiplexed bus, these pins output an undefined value during a write.

7.2.9 External Bus Status when Internal Area Accessed

Table 7.7 shows the External Bus Status When Internal Area Accessed.

Table 7.7 External Bus Status When Internal Area Accessed

| Item | SFR Accessed | Internal ROM, Internal RAM Accessed |
|------------------|-------------------------|---|
| A0 to A19 | Address output | Maintain status before accessed address of external area or SFR |
| D0 to D15 | When read | High-impedance |
| | When write | Output data |
| RD, WR, WRL, WRH | RD, WR, WRL, WRH output | Output "H" |
| BHE | BHE output | Maintain status before accessed status of external area or SFR |
| CS0 to CS3 | Output "H" | Output "H" |
| ALE | Output "L" | Output "L" |

7.2.10 Software Wait

Software wait states can be inserted by using the PM17 bit in the PM1 register, bits CS0W to CS3W in the CSR register, and the CSE register. The SFR area is unaffected by these control bits. This area is always accessed in 2 BCLK or 3 BCLK cycles as determined by the PM20 bit in the PM2 register. See **Table 7.8 Bit and Bus Cycle Related to Software Wait** for details.

To use the \overline{RDY} signal, set the corresponding bit of bits CS3W to CS0W to 0 (with wait state).

Figure 7.6 shows the CSE Register. Table 7.8 shows the Software Wait Related Bits and Bus Cycles. Figures 7.7 and 7.8 show the Typical Bus Timings Using Software Wait.

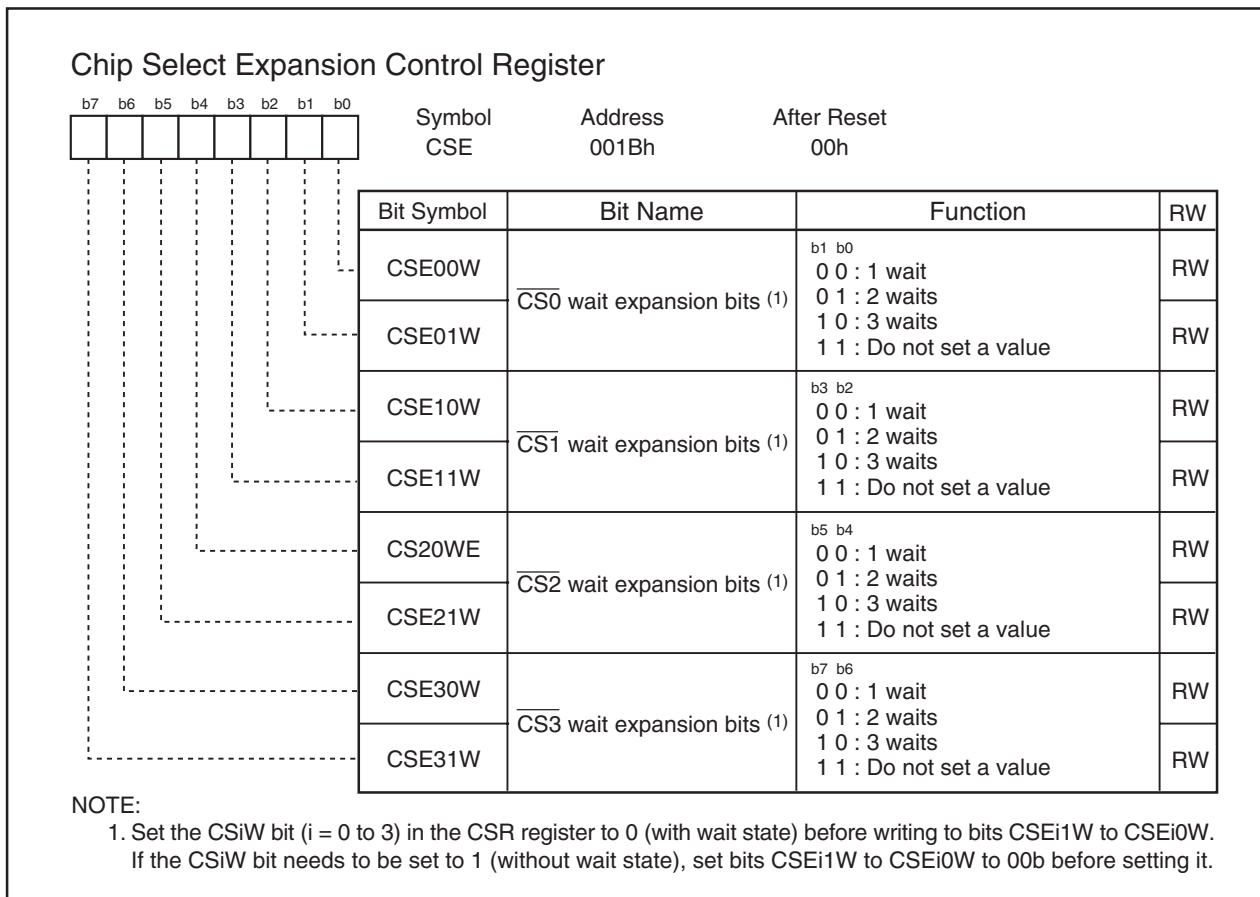


Figure 7.6 CSE Register

Table 7.8 Software Wait Related Bits and Bus Cycles

| Area | Bus Mode | PM2 Register PM20 Bit | PM1 Register PM17 Bit ⁽⁵⁾ | CSR Register CS3W Bit ⁽¹⁾ CS2W Bit ⁽¹⁾ CS1W Bit ⁽¹⁾ CS0W Bit ⁽¹⁾ | CSE Register Bits CS31W to CS30W Bits CS21W to CS20W Bits CS11W to CS10W Bits CS01W to CS00W | Software Wait | Bus Cycle |
|----------------------|-----------------------------------|--------------------------|---|--|--|------------------|--|
| SFR | – | 0 | – | – | – | – | 3 BCLK cycles ⁽⁴⁾ |
| | – | 1 | – | – | – | – | 2 BCLK cycles ⁽⁴⁾ |
| Internal ROM, RAM | – | – | 0 | – | – | No wait | 1 BCLK cycle ⁽³⁾ |
| | – | – | 1 | – | – | 1 wait | 2 BCLK cycles |
| External area | Separate bus | – | 0 | 1 | 00b | No wait | 1 BCLK cycle (read) 2 BCLK cycles (write) |
| | | – | – | 0 | 00b | 1 wait | 2 BCLK cycles ⁽³⁾ |
| | | – | – | 0 | 01b | 2 waits | 3 BCLK cycles |
| | | – | – | 0 | 10b | 3 waits | 4 BCLK cycles |
| | | – | 1 | 0 | 00b | 1 wait | 2 BCLK cycles |
| | Multiplexed bus ⁽²⁾ | – | – | 0 | 00b | 1 wait | 3 BCLK cycles |
| | | – | – | 0 | 01b | 2 waits | 3 BCLK cycles |
| | | – | – | 0 | 10b | 3 waits | 4 BCLK cycles |
| | | – | 1 | 0 | 00b | 1 wait | 3 BCLK cycles |

NOTES:

1. To use the $\overline{\text{RDY}}$ signal, set this bit to 0.
2. To access in multiplexed bus mode, set the corresponding bit of bits CS0W to CS3W to 0 (with wait state).
3. After reset, the PM17 bit is set to 0 (without wait state), all of bits CS0W to CS3W are set to 0 (with wait state), and the CSE register is set to 00h (one wait state for CS0 to CS3). Therefore, the internal RAM and internal ROM are accessed with no wait state, and all external areas are accessed with one wait state.
4. When the selected CPU clock source is the PLL clock, the number of wait cycles can be altered by the PM20 bit in the PM2 register. When using PLL clock over 16 MHz, be sure to set the PM20 bit to 0 (2 wait cycles).
5. When the PM17 bit is set to 1 and access an external area, set the CSiW bits (i = 0 to 3) to 0 (with wait state).

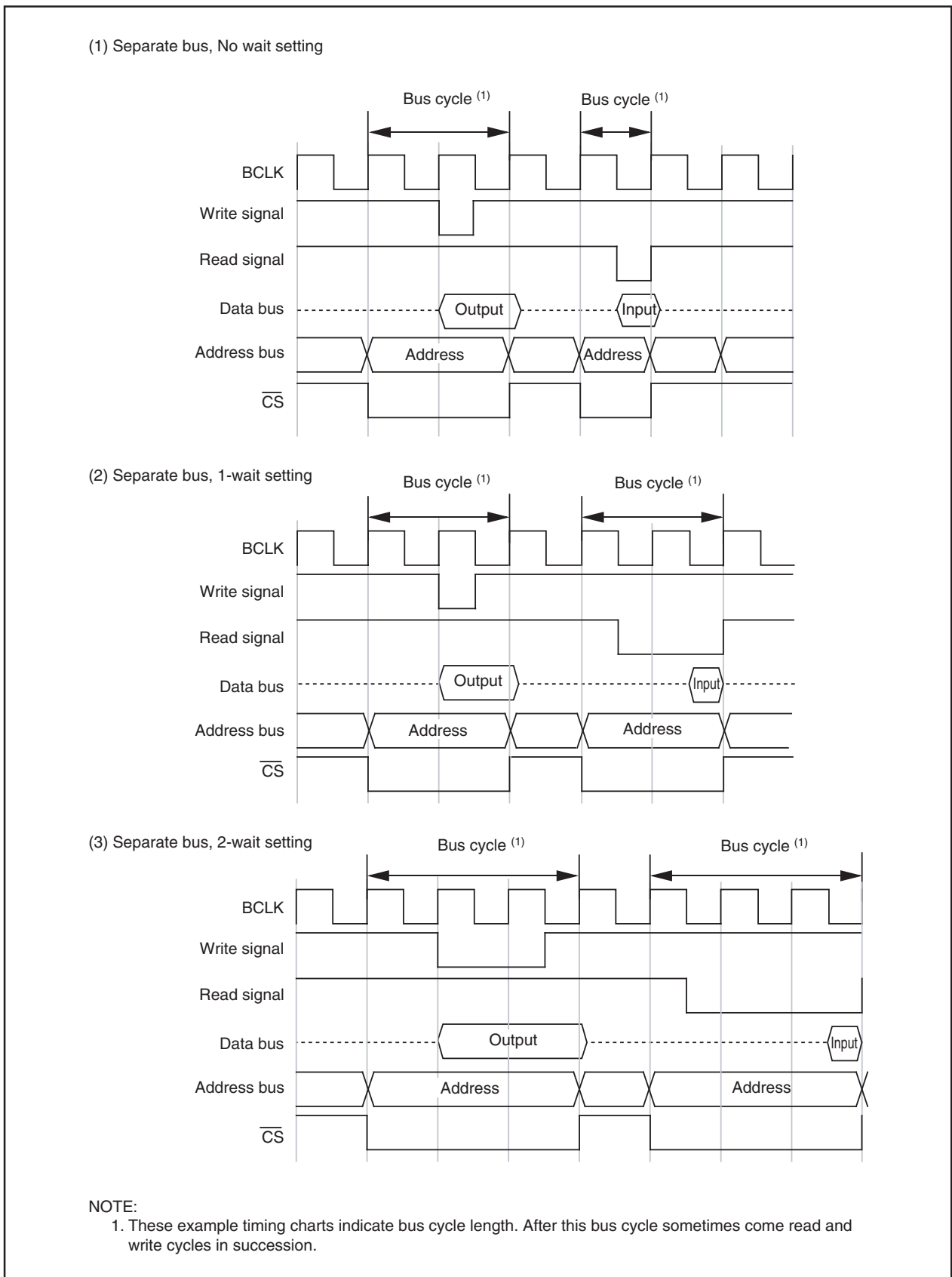


Figure 7.7 Typical Bus Timings Using Software Wait (1)

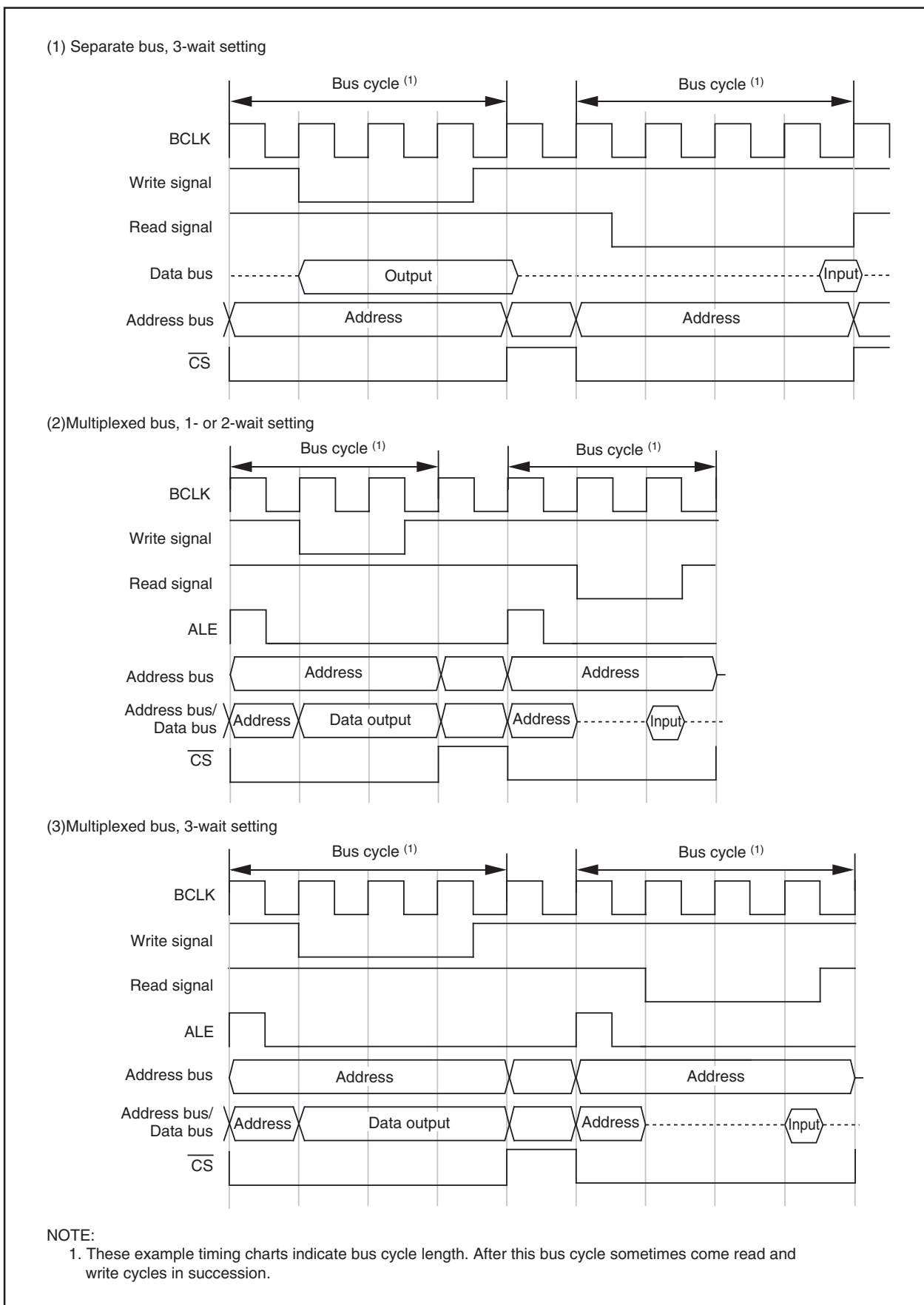


Figure 7.8 Typical Bus Timings Using Software Wait (2)

8. Clock Generation Circuit

8.1 Types of Clock Generation Circuit

Four circuits are incorporated to generate the system clock signal:

- Main clock oscillation circuit
- Sub clock oscillation circuit
- On-chip oscillator
- PLL frequency synthesizer

Table 8.1 lists the Clock Generation Circuit Specifications. Figure 8.1 shows the Clock Generation Circuit. Figures 8.2 to 8.8 show the clock-related registers.

Table 8.1 Clock Generation Circuit Specifications

| Item | Main Clock Oscillation Circuit | Sub Clock Oscillation Circuit | On-chip Oscillator | PLL Frequency Synthesizer |
|--|--|--|---|--|
| Use of clock | <ul style="list-style-type: none"> • CPU clock source • Peripheral function clock source | <ul style="list-style-type: none"> • CPU clock source • Clock source of timer A, B | <ul style="list-style-type: none"> • CPU clock source • Peripheral function clock source • CPU and peripheral function clock sources when the main clock stops oscillating | <ul style="list-style-type: none"> • CPU clock source • Peripheral function clock source |
| Clock frequency | 0 to 16 MHz | 32.768 kHz | About 1 MHz | 16 MHz, 20 MHz, 24 MHz ⁽¹⁾ |
| Usable oscillator | <ul style="list-style-type: none"> •Ceramic oscillator •Crystal oscillator | •Crystal oscillator | - | - |
| Pins to connect oscillator | XIN, XOUT | XCIN, XCOU | - | - |
| Oscillation stop and re-oscillation detection function | Available | Available | Available | Available |
| Oscillation status after reset | Oscillating | Stopped | Stopped | Stopped |
| Other | Externally derived clock can be input | | - | - |

NOTE:

- 1. 24 MHz is available Normal-ver. only.

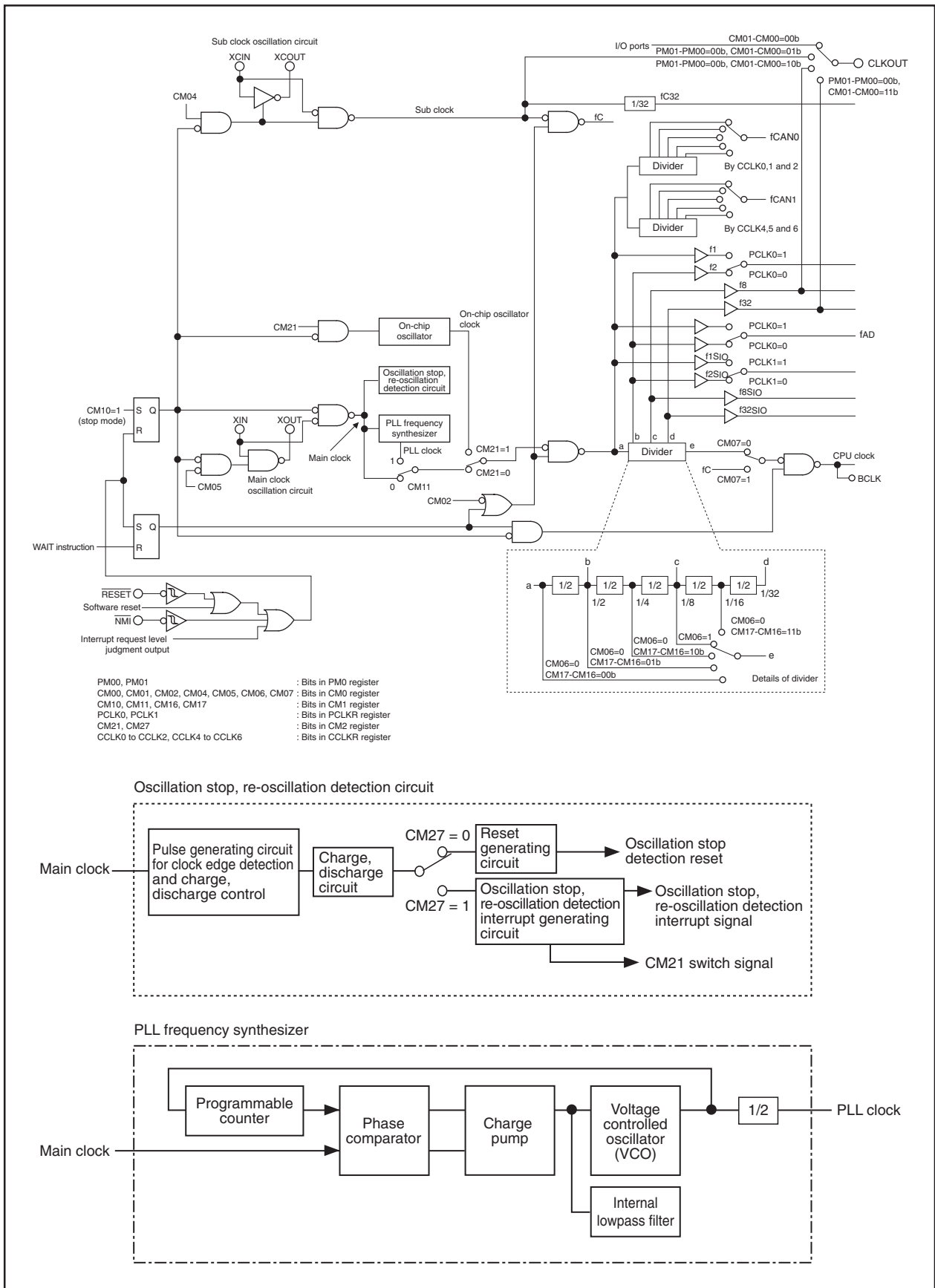


Figure 8.1 Clock Generation Circuit

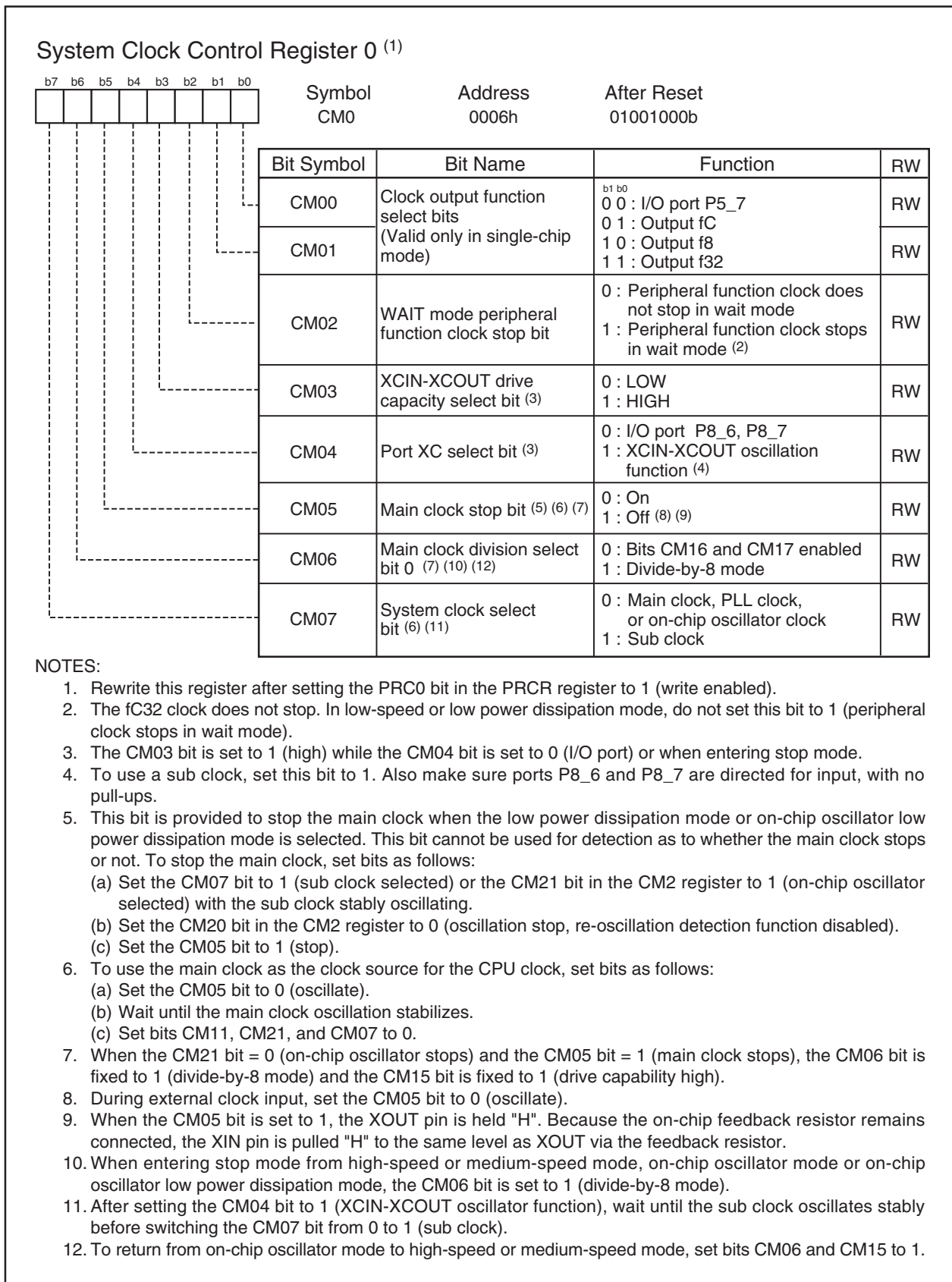


Figure 8.2 CM0 Register

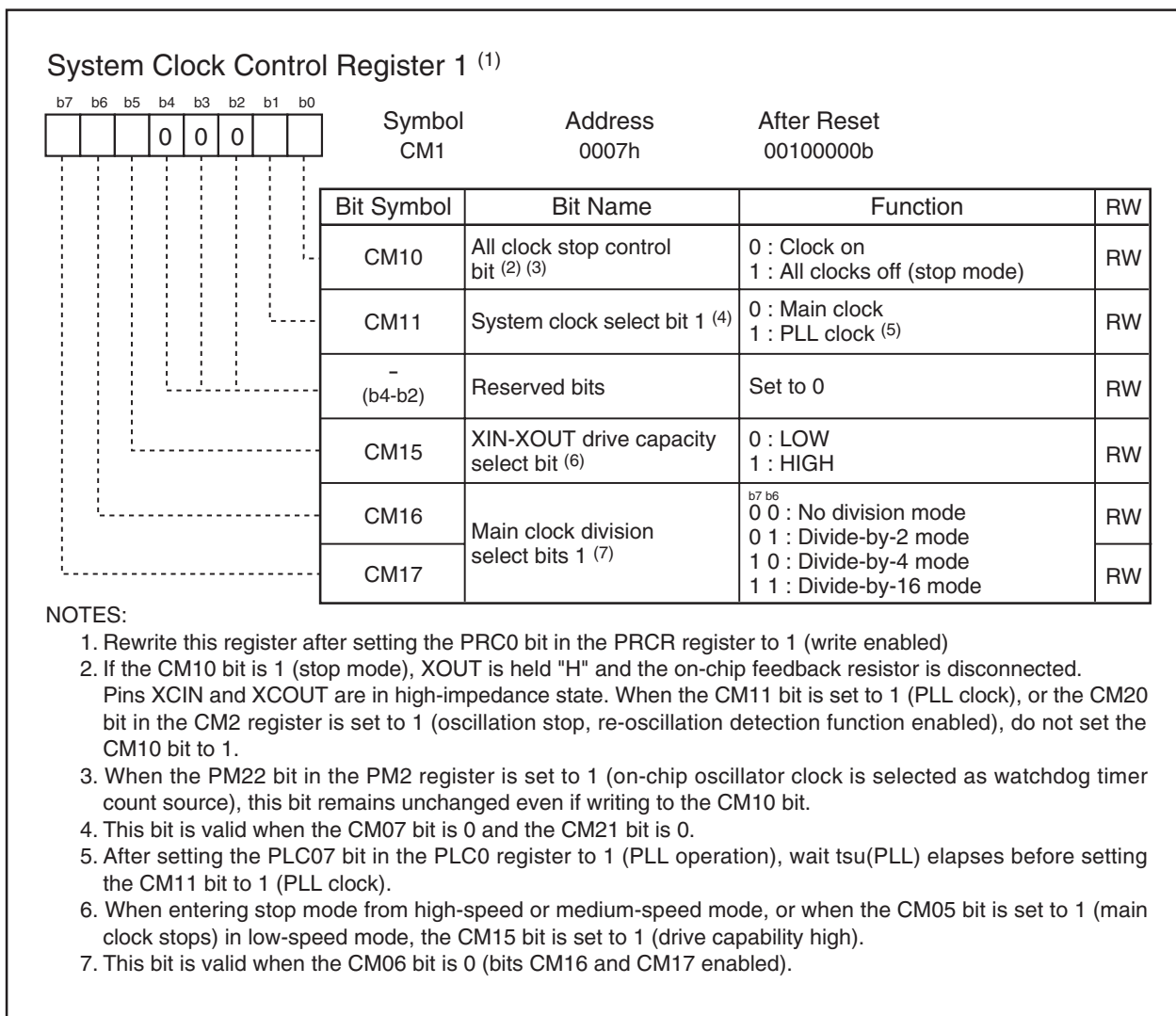


Figure 8.3 CM1 Register

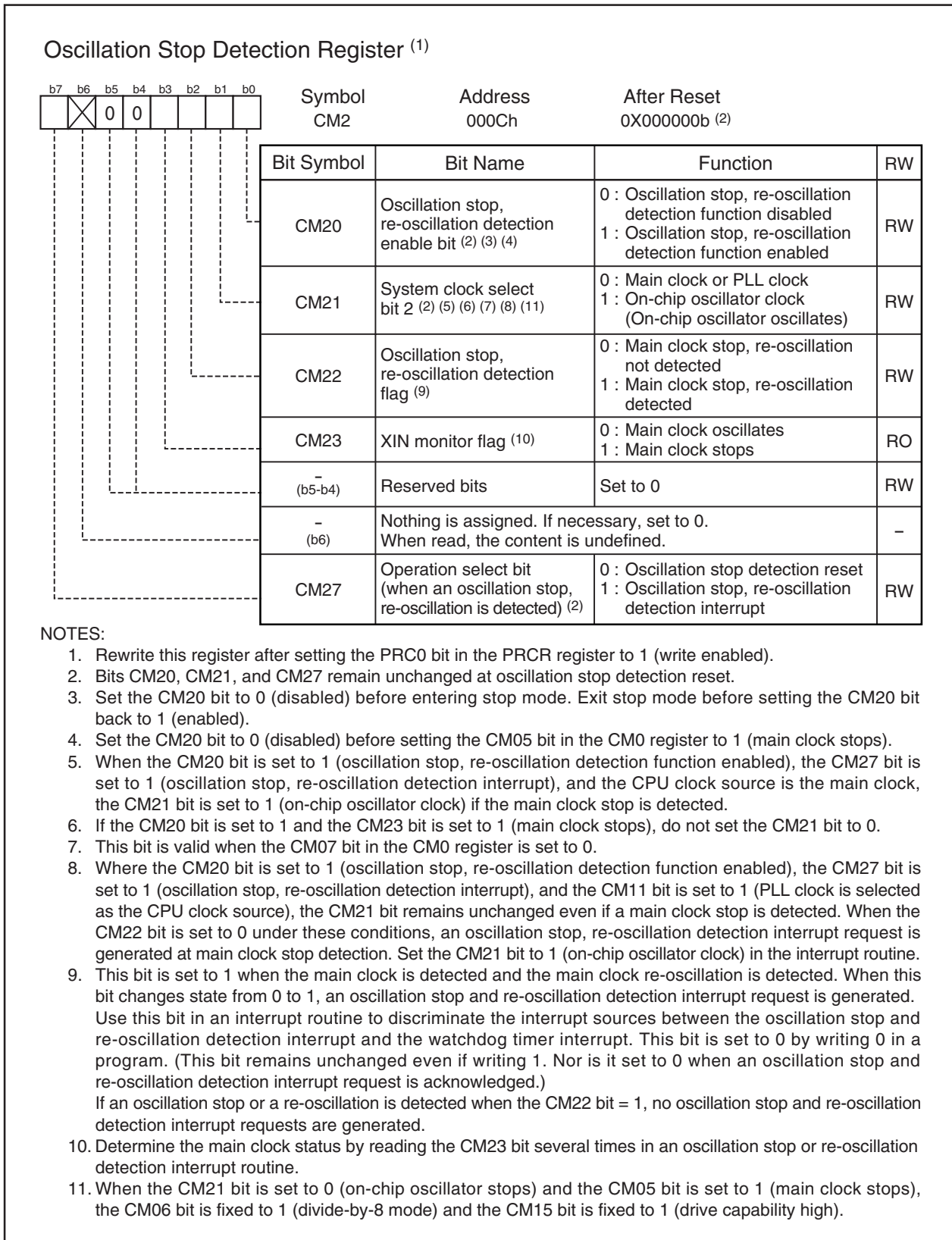


Figure 8.4 CM2 Register

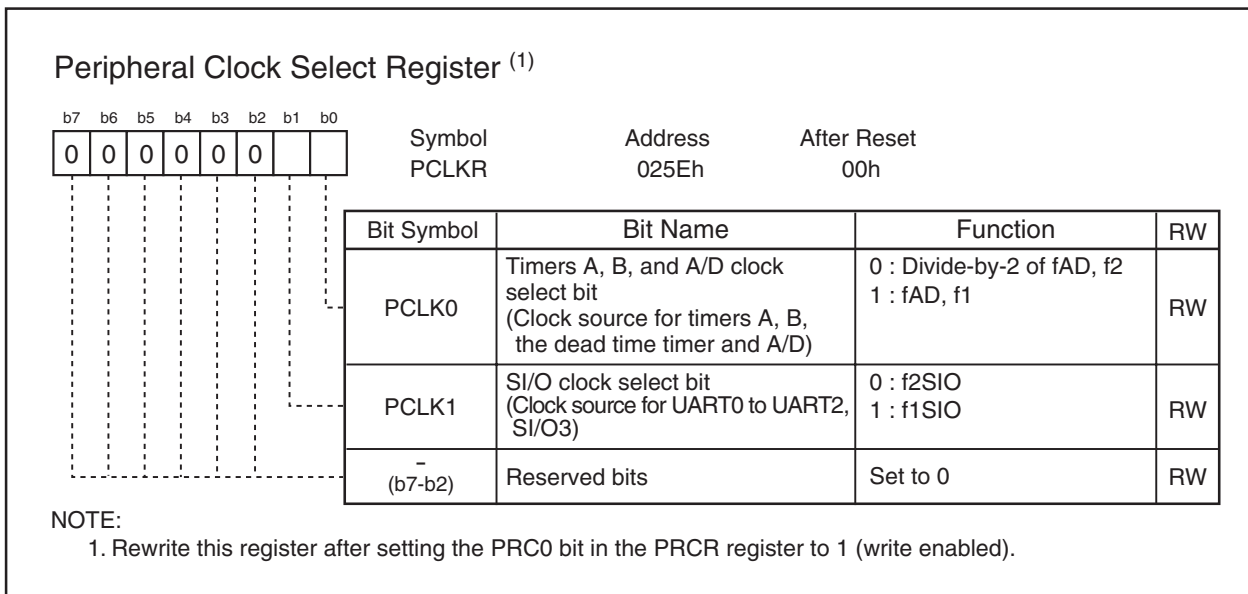


Figure 8.5 PCLKR Register

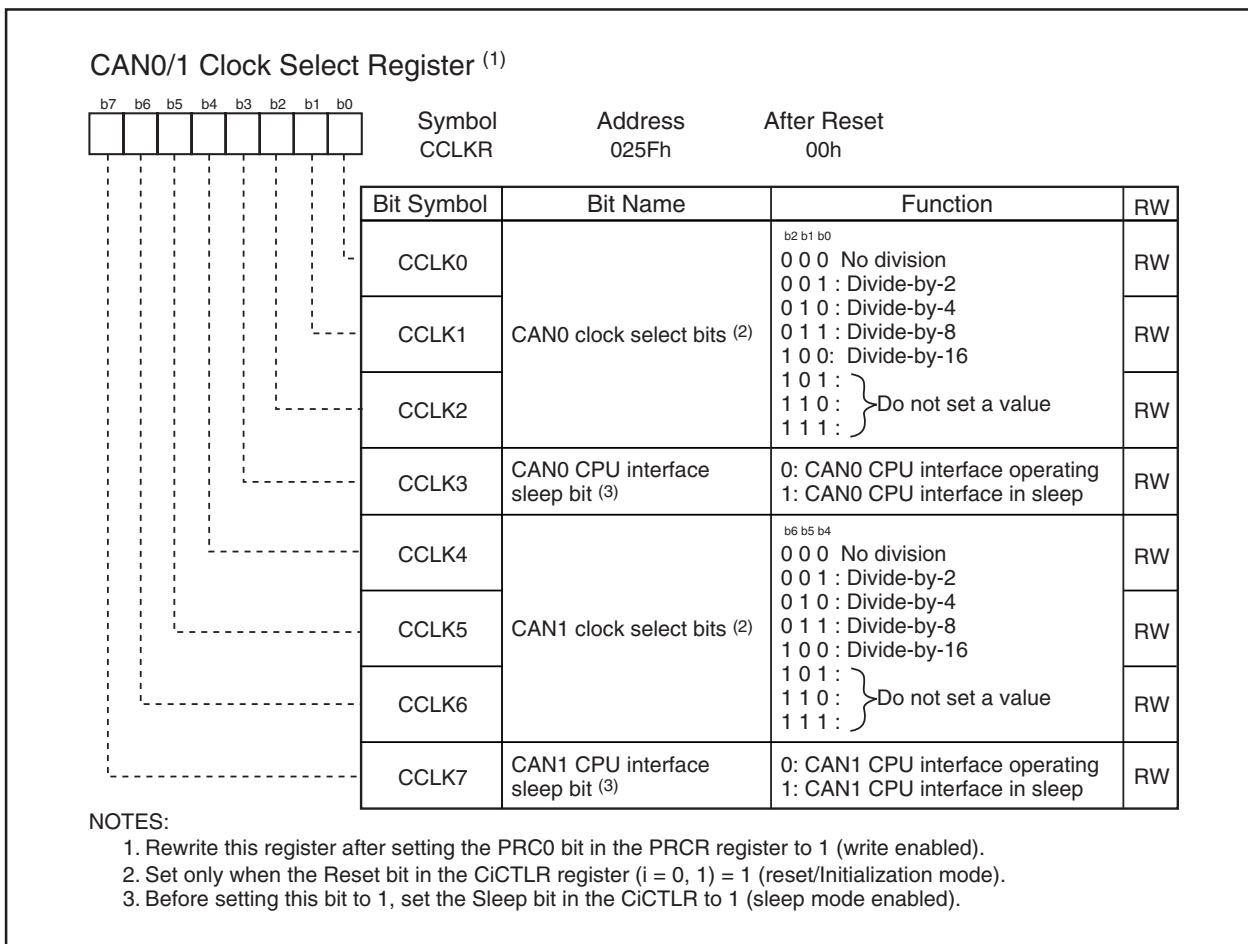


Figure 8.6 CCLKR Register

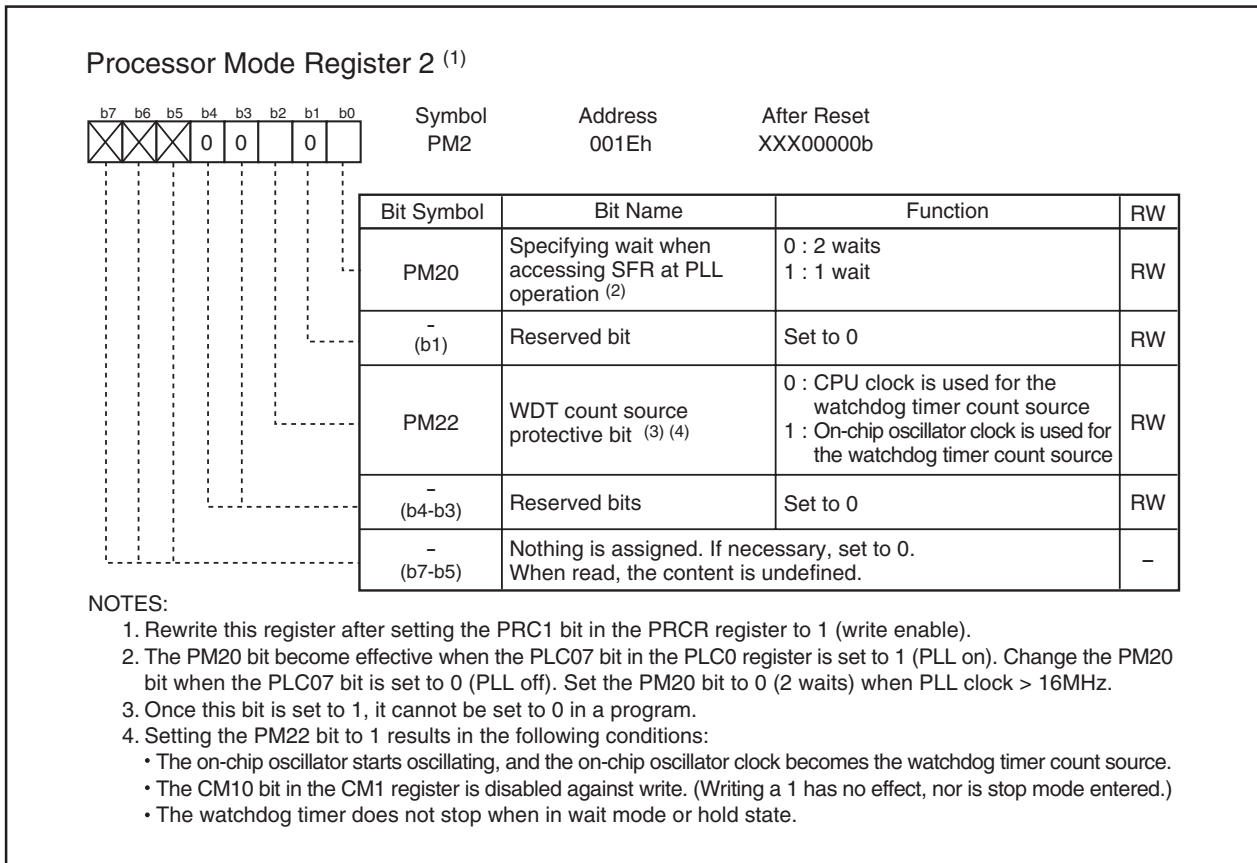


Figure 8.7 PM2 Register

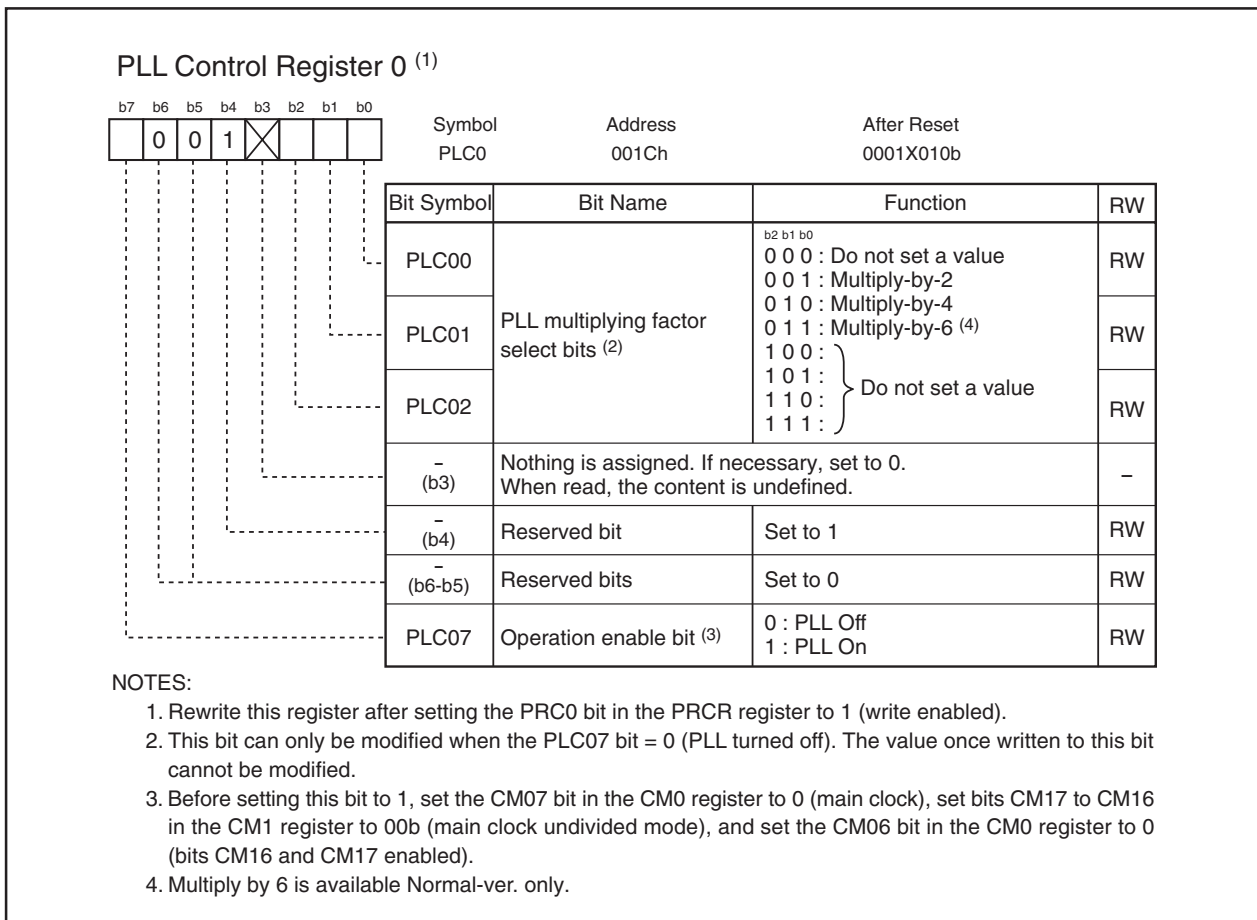


Figure 8.8 PLC0 Register

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

8.1.1 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 oscillation circuit is configured by connecting a resonator between pins XIN and XOUT. The main clock oscillation circuit has an on-chip feedback resistor, which is disconnected from the oscillation circuit during stop mode in order to reduce the amount of power consumed in the chip. The main clock oscillation circuit may also be configured by feeding an externally generated clock to the XIN pin. Figure 8.9 shows an Examples of Main Clock Connection Circuit.

After reset, the main clock divided by 8 is selected for the CPU clock.

The power consumption in the chip can be reduced by setting the CM05 bit in the CM0 register to 1 (main clock oscillation 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 an on-chip feedback resistor remains on, XIN is pulled "H" to XOUT via the feedback resistor. Note, that if an externally generated clock is fed into the XIN pin, the main clock cannot be turned off by setting the CM05 bit to 1, unless the sub clock is selected as a CPU clock. If necessary, use an external circuit to turn off the clock.

During stop mode, all clocks including the main clock are turned off. Refer to **8.4 Power Control**.

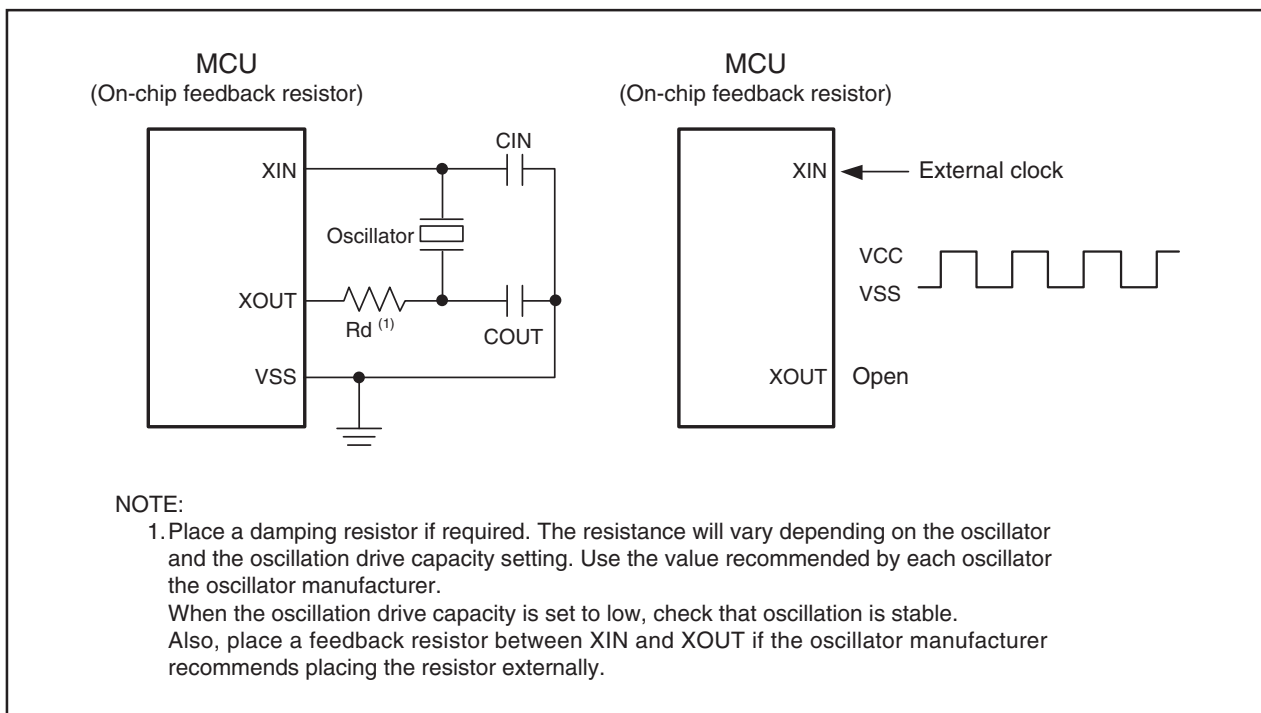


Figure 8.9 Examples of Main Clock Connection Circuit

8.1.2 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. In addition, an fC clock with the same frequency as that of the sub clock can be output from the CLKOUT pin.

The sub clock oscillation circuit is configured by connecting a crystal resonator between pins XCIN and XCOUT. The sub clock oscillation circuit has an on-chip feedback resistor, which is disconnected from the oscillation circuit during stop mode in order to reduce the amount of power consumed in the chip. The sub clock oscillation circuit may also be configured by feeding an externally generated clock to the XCIN pin. Figure 8.10 shows an Examples of Sub Clock Connection Circuit.

After reset, the sub clock is turned off. At this time, the feedback resistor is disconnected from the oscillation 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 **8.4 Power Control**.

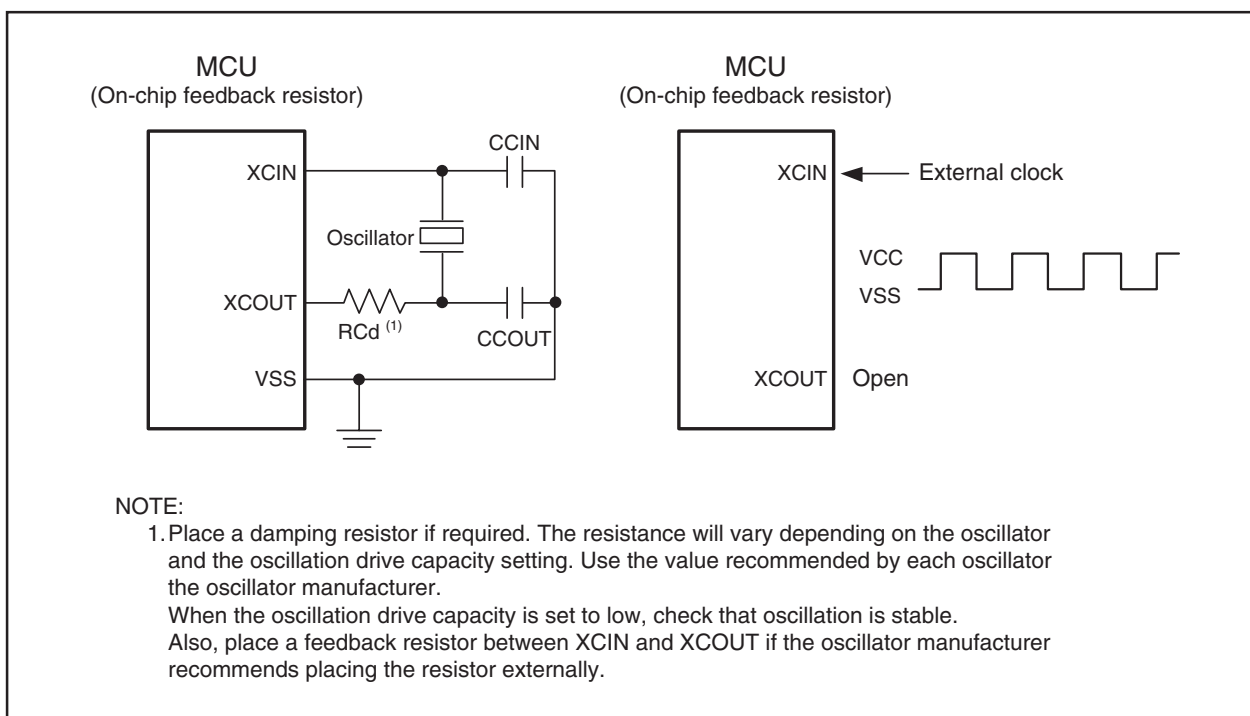


Figure 8.10 Examples of Sub Clock Connection Circuit

8.1.3 On-chip Oscillator Clock

This clock, approximately 1 MHz, is supplied by a 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 **11.1 Count Source Protective Mode**).

After reset, the on-chip oscillator is turned off. It is 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, in place of the main clock. 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.

8.1.4 PLL Clock

The PLL clock is generated 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 8.11 shows the Procedure to Use PLL Clock as CPU Clock Source.

The PLL clock frequency is determined by the equation below. When the PLL clock frequency is 16 MHz or more, set the PM20 bit in the PM2 register to 0 (2 waits).

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

(However, PLL clock frequency = 16 MHz, 20 MHz or 24 MHz ⁽¹⁾)

NOTE:

1. 24 MHz is available Normal-ver. only.

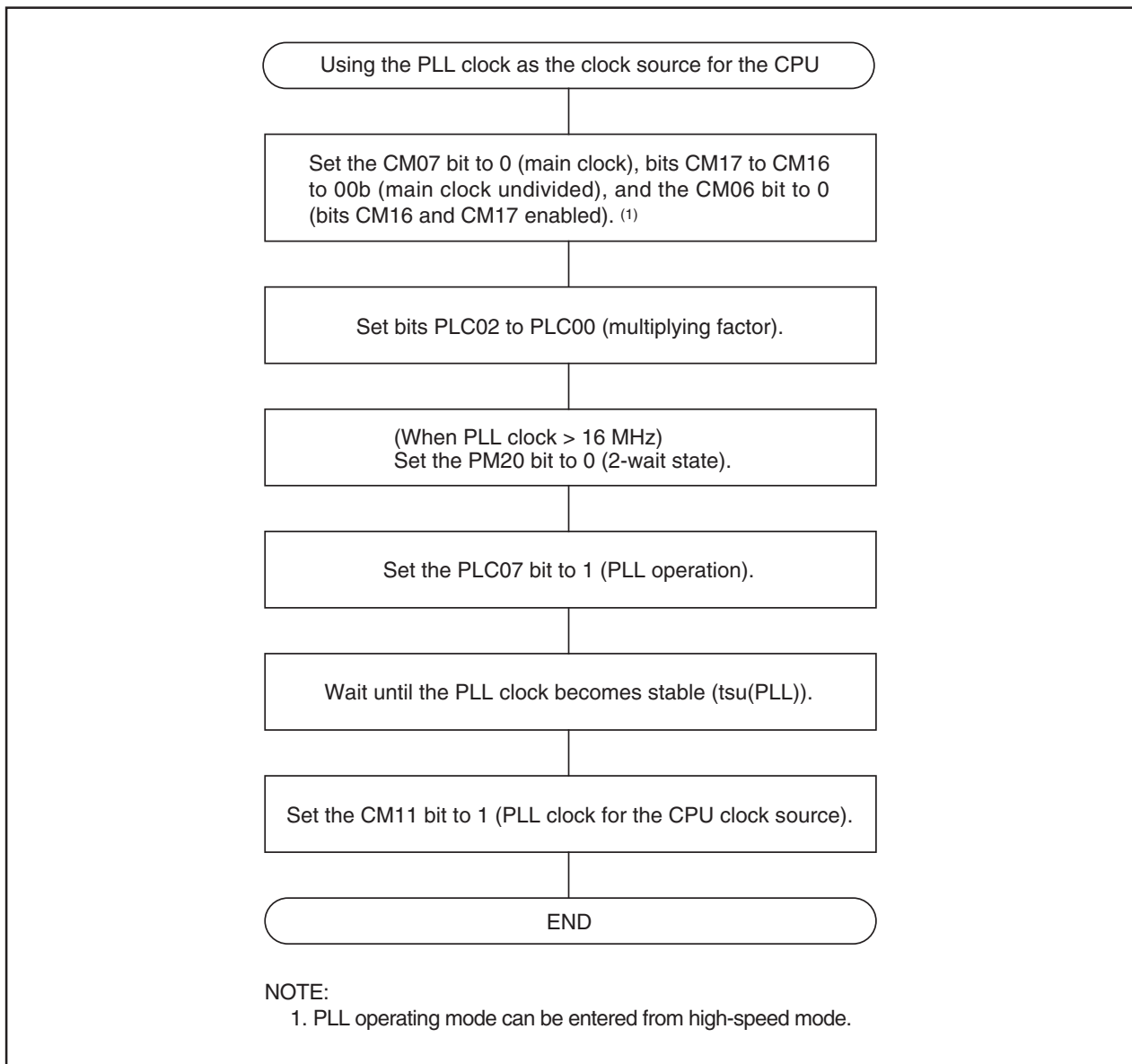
Bits PLC02 to PLC00 can be set only once after reset. Table 8.2 shows an Example for Setting PLL Clock Frequencies.

Table 8.2 Example for Setting PLL Clock Frequencies

| XIN (MHz) | PLC02 | PLC01 | PLC00 | Multiply Factor | PLL Clock (MHz) ⁽¹⁾ |
|-----------|-------|-------|-------|------------------|--------------------------------|
| 8 | 0 | 0 | 1 | 2 | 16 |
| 4 | 0 | 1 | 0 | 4 | |
| 10 | 0 | 0 | 1 | 2 | 20 |
| 5 | 0 | 1 | 0 | 4 | |
| 12 | 0 | 0 | 1 | 2 | 24 ⁽²⁾ |
| 6 | 0 | 1 | 0 | 4 | |
| 4 | 0 | 1 | 1 | 6 ⁽³⁾ | |

NOTES:

1. PLL clock frequency = 16 MHz , 20 MHz or 24 MHz
2. 24 MHz is available Normal-ver. only.
3. Multiply by 6 is available Normal-ver. only.

**Figure 8.11 Procedure to Use PLL Clock as CPU Clock Source**

8.2 CPU Clock and Peripheral Function Clock

Two type clocks: CPU clock to operate the CPU and peripheral function clocks to operate the peripheral functions.

8.2.1 CPU Clock and BCLK

These are operating clocks 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 the CM0 register and bits CM17 to CM16 in the 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 to CM16 to 00b (undivided).

After reset, the main clock divided by 8 provides the CPU clock.

During memory expansion or microprocessor mode, a BCLK signal with the same frequency as the CPU clock can be output from the BCLK pin by setting the PM07 bit of PM0 register to 0 (output enabled).

Note that when entering stop mode from high-speed or medium-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).

8.2.2 Peripheral Function Clock (f1, f2, f8, f32, f1SIO, f2SIO, f8SIO, f32SIO, fAD, fCAN0, fCAN1, fC32)

These are operating clocks for the peripheral functions.

Two of these, f_i ($i = 1, 2, 8, 32$) and f_{iSIO} are derived from the main clock, PLL clock or on-chip oscillator clock by dividing them by i . The clock f_i is used for timers A and B, and f_{iSIO} is used for serial interface. The f_8 and f_{32} clocks can be output from the CLKOUT pin.

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_{CANi} ($i = 0, 1$) clock is derived from the main clock, PLL clock or on-chip oscillator clock by dividing them 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_{iSIO} , f_{AD} , f_{CAN0} , and f_{CAN1} clocks are turned off ⁽¹⁾.

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

NOTE:

1. f_{CAN0} and f_{CAN1} clocks stop at "H" in CAN0, 1 sleep mode.

8.3 Clock Output Function

During single-chip mode, the f_8 , f_{32} , or f_C clock can be output from the CLKOUT pin. Use bits CM01 to CM00 in the CM0 register to select.

8.4 Power Control

Normal operating mode, wait mode and stop mode are provided as the power consumption control. All mode states, except wait mode and stop mode, are called normal operating mode in this document.

8.4.1 Normal Operating Mode

Normal operating mode is further classified into seven sub modes.

In normal operating 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 to which switched must be oscillating stably. 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 operating modes cannot be changed directly from low speed or low power dissipation mode to on-chip oscillator or on-chip oscillator low power dissipation mode. Nor can operating modes be changed directly from on-chip oscillator or on-chip oscillator low power dissipation mode to low-speed or low power dissipation mode. Where the CPU clock source is changed from the on-chip oscillator to the main clock, change the operating mode to the medium-speed mode (divide-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.

8.4.1.1 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.

8.4.1.2 PLL Operating Mode

The main clock multiplied by 2, 4, or 6⁽¹⁾ 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 operating mode can be entered from high speed mode. If PLL operating mode is to be changed to wait or stop mode, first go to high speed mode before changing.

NOTE:

1. The main clock multiplied by 6 is available Normal-ver. only.

8.4.1.3 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.

8.4.1.4 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 in the CM2 register 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.

8.4.1.5 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.

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

8.4.1.6 On-chip Oscillator Mode

The 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. When the operating mode is returned to the high-speed and medium-speed modes, set the CM06 bit in the CM0 register to 1 (divide-by-8 mode).

8.4.1.7 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 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 8.3 lists the Setting Clock Related Bit and Modes.

Table 8.3 Setting Clock Related Bit and Modes

| Modes | | CM2 Register | CM1 Register | | CM0 Register | | | |
|---|--------------|--------------|--------------|------------|--------------|------------------|------------------|------|
| | | CM21 | CM11 | CM17, CM16 | CM07 | CM06 | CM05 | CM04 |
| PLL operating mode | | 0 | 1 | 00b | 0 | 0 | 0 | - |
| High-speed mode | | 0 | 0 | 00b | 0 | 0 | 0 | - |
| Medium-speed mode | Divide-by-2 | 0 | 0 | 01b | 0 | 0 | 0 | - |
| | Divide-by-4 | 0 | 0 | 10b | 0 | 0 | 0 | - |
| | Divide-by-8 | 0 | 0 | - | 0 | 1 | 0 | - |
| | Divide-by-16 | 0 | 0 | 11b | 0 | 0 | 0 | - |
| Low-speed mode | | - | 0 | - | 1 | - | 0 | 1 |
| Low power dissipation mode | | 0 | 0 | - | 1 | 1 ⁽¹⁾ | 1 ⁽¹⁾ | 1 |
| On-chip oscillator mode | No division | 1 | 0 | 00b | 0 | 0 | 0 | - |
| | Divide-by-2 | 1 | 0 | 01b | 0 | 0 | 0 | - |
| | Divide-by-4 | 1 | 0 | 10b | 0 | 0 | 0 | - |
| | Divide-by-8 | 1 | 0 | - | 0 | 1 | 0 | - |
| | Divide-by-16 | 1 | 0 | 11b | 0 | 0 | 0 | - |
| On-chip oscillator low power dissipation mode | | 1 | 0 | (NOTE 2) | 0 | (NOTE 2) | 1 | - |

-: 0 or 1

NOTES:

1. 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 the CM06 bit is set to 1 (divide-by-8 mode) simultaneously.
2. The divide-by-n value can be selected the same way as in on-chip oscillator mode.

8.4.2 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 and on-chip oscillator clock all are on, the peripheral functions using these clocks keep operating.

8.4.2.1 Peripheral Function Clock Stop Function

If the CM02 bit in the CM0 register is 1 (peripheral function clocks turned off during wait mode), the f1, f2, f8, f32, f1SIO, f8SIO, f32SIO, fAD, fCAN0, and fCAN1 clocks are turned off when in wait mode, with the power consumption reduced that much. However, fC32 remains on.

8.4.2.2 Entering Wait Mode

The MCU is placed into wait mode by executing the WAIT instruction.

When the CM11 bit = 1 (CPU clock source is the PLL clock), be sure to set the CM11 bit in the CM1 register to 0 (CPU clock source is the main clock) before going to wait mode. The power consumption of the chip can be reduced by setting the PLC07 bit in the PLC0 register to 0 (PLL stops).

8.4.2.3 Pin Status During Wait Mode

Table 8.4 lists the Pin Status During Wait Mode.

Table 8.4 Pin Status During Wait Mode

| Pin | Memory Expansion Mode Microprocessor Mode | Single-chip Mode | |
|--|--|-----------------------------------|--|
| A0 to A19, D0 to D15, CS0 to CS3, BHE | Retains status before wait mode | Does not become a bus control pin | |
| RD, WR, WRL, WRH | “H” | | |
| HLDA, BCLK | “H” | | |
| ALE | “L” | | |
| I/O ports | Retains status before wait mode | Retains status before wait mode | |
| CLKOUT | When fC selected | Does not become a CLKOUT pin | Does not stop |
| | When f8, f32 selected | | <ul style="list-style-type: none"> •CM02 bit = 0: Does not stop •CM02 bit = 1: Retains status before wait mode |

8.4.2.4 Exiting Wait Mode

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

If the MCU exits wait mode by a hardware reset or $\overline{\text{NMI}}$ interrupt, set the peripheral function interrupt priority bits ILVL2 to ILVL0 to 000b (interrupt disabled) before executing the WAIT instruction.

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

Table 8.5 lists the Interrupts to Exit Wait Mode and Use Conditions.

Table 8.5 Interrupts to Exit Wait Mode and Use Conditions

| Interrupt | CM02 Bit = 0 | CM02 Bit = 1 |
|--|--|--|
| NMI interrupt | Can be used | Can be used |
| Serial interface interrupt | Can be used when operating with internal or external clock | Can be used when operating with external clock |
| Key input interrupt | Can be used | Can be used |
| A/D conversion interrupt | Can be used in one-shot mode or single sweep mode | - (Do not use) |
| Timer A interrupt Timer B interrupt | Can be used in all modes | Can be used in event counter mode or when the count source is fC32 |
| INT interrupt | Can be used | Can be used |
| CAN0/1 wake-up interrupt | Can be used in CAN sleep mode | Can be used in CAN sleep mode |

If the MCU exits wait mode by a peripheral function interrupt, set up the following before executing the WAIT instruction.

- (1) Set bits ILVL2 to ILVL0 in the interrupt control register, for peripheral function interrupts used to exit wait mode.
Bits ILVL2 to ILVL0 in all other interrupt control registers, for peripheral function interrupts not used to exit wait mode, are set to 000b (interrupt disabled).
- (2) Set the I flag to 1.
- (3) Start operating the peripheral functions used to exit wait mode.
When the peripheral function interrupt is used, an interrupt routine is performed as soon as an interrupt request is acknowledged and the CPU clock is supplied again.

When the MCU exits wait mode by the peripheral function interrupt, the CPU clock is the same clock as the CPU clock executing the WAIT instruction.

8.4.3 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.

However, the peripheral functions clocked by external signals keep operating.

Table 8.6 lists the Interrupts to Stop Mode and Use Conditions.

Table 8.6 Interrupts to Stop Mode and Use Conditions

| Interrupt | Condition |
|--|--|
| NMI interrupt | Can be used |
| Key input interrupt | Can be used |
| INT interrupt | Can be used |
| Timer A interrupt Timer B interrupt | Can be used (when counting external pulses in event counter mode) |
| Serial interface interrupt | Can be used (when external clock is selected) |
| CAN0/1 wake-up interrupt | Can be used (when CAN sleep mode is selected) |

8.4.3.1 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 CM1 register is set to 1 (main clock oscillator circuit drive capability high).

Before entering stop mode, set the CM20 bit in the CM2 register to 0 (oscillation stop, re-oscillation detection function disabled).

Also, if the CM11 bit in the CM1 register 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 in the PLC0 register to 0 (PLL turned off) before entering stop mode.

8.4.3.2 Pin Status in Stop Mode

Table 8.7 lists the Pin Status in Stop Mode.

Table 8.7 Pin Status in Stop Mode

| Pin | Memory Expansion Mode Microprocessor Mode | Single-chip Mode | |
|--|--|-----------------------------------|---------------------------------|
| A0 to A19, D0 to D15, CS0 to CS3, BHE | Retains status before stop mode | Does not become a bus control pin | |
| RD, WR, WRL, WRH | "H" | | |
| HLDA, BCLK | "H" | | |
| ALE | undefined | | |
| I/O ports | Retains status before stop mode | Retains status before stop mode | |
| CLKOUT | When fC selected | Does not become a CLKOUT pin | "H" |
| | When f8, f32 selected | | Retains status before stop mode |

8.4.3.3 Exiting Stop Mode

Stop mode is exited by a hardware reset, $\overline{\text{NMI}}$ interrupt or peripheral function interrupt.

When the hardware reset or $\overline{\text{NMI}}$ interrupt is used to exit stop mode, set all ILVL2 to ILVL0 bits in the interrupt control registers for the peripheral function interrupt to 000b (interrupt disabled) before setting the CM10 bit in the CM1 register to 1.

When the peripheral function interrupt is used to exit stop mode, set the CM10 bit to 1 after the following settings are completed.

(1) Set bits ILVL2 to ILVL0 in the interrupt control registers to decide the peripheral priority level of the peripheral function interrupt.

Set the interrupt priority levels of the interrupts, not being used to exit stop mode, to 0 by setting the all ILVL2 to ILVL0 bits to 000b (interrupt disabled).

(2) Set the I flag to 1.

(3) Start operation of peripheral function being used to exit wait mode.

When exiting stop mode by the peripheral function interrupt, the interrupt routine is performed when an interrupt request is generated and the CPU clock is supplied again.

When stop mode is exited by the peripheral function interrupt or $\overline{\text{NMI}}$ interrupt, the CPU clock source is as follows, in accordance with the CPU clock source setting before the MCU had entered stop mode.

- When the sub clock is the CPU clock before entering stop mode: Sub clock
- When the main clock is the CPU clock source before entering stop mode:

Main clock divided by 8

- When the on-chip oscillator clock is the CPU clock source before entering stop mode:

On-chip oscillator clock divided by 8

Figure 8.12 shows the State Transition to Stop Mode and Wait Mode. Figure 8.13 shows the State Transition in Normal Operating Mode.

Table 8.8 shows a state transition matrix describing allowed transition and setting. The vertical line shows current state and horizontal line show state after transition.

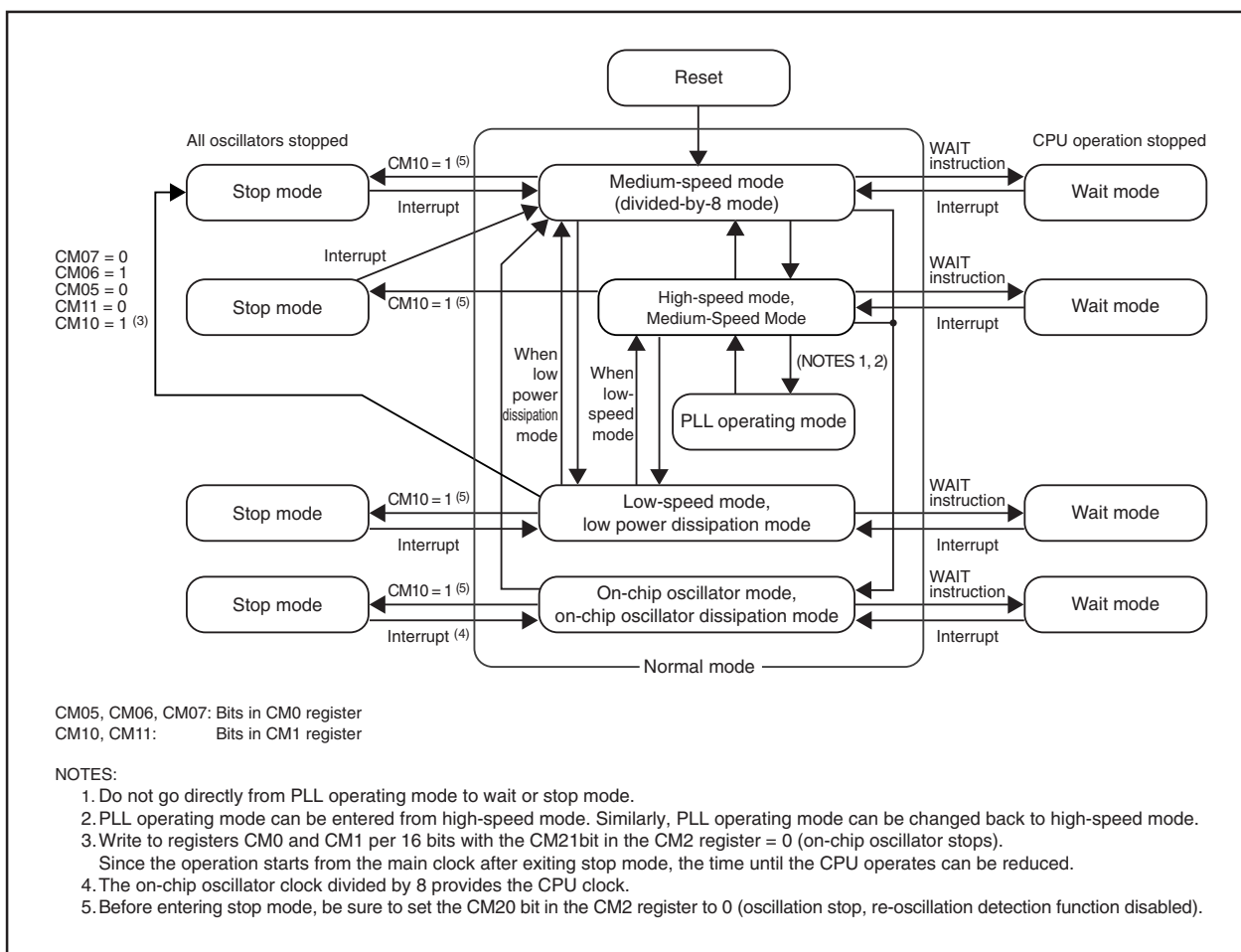


Figure 8.12 State Transition to Stop Mode and Wait Mode

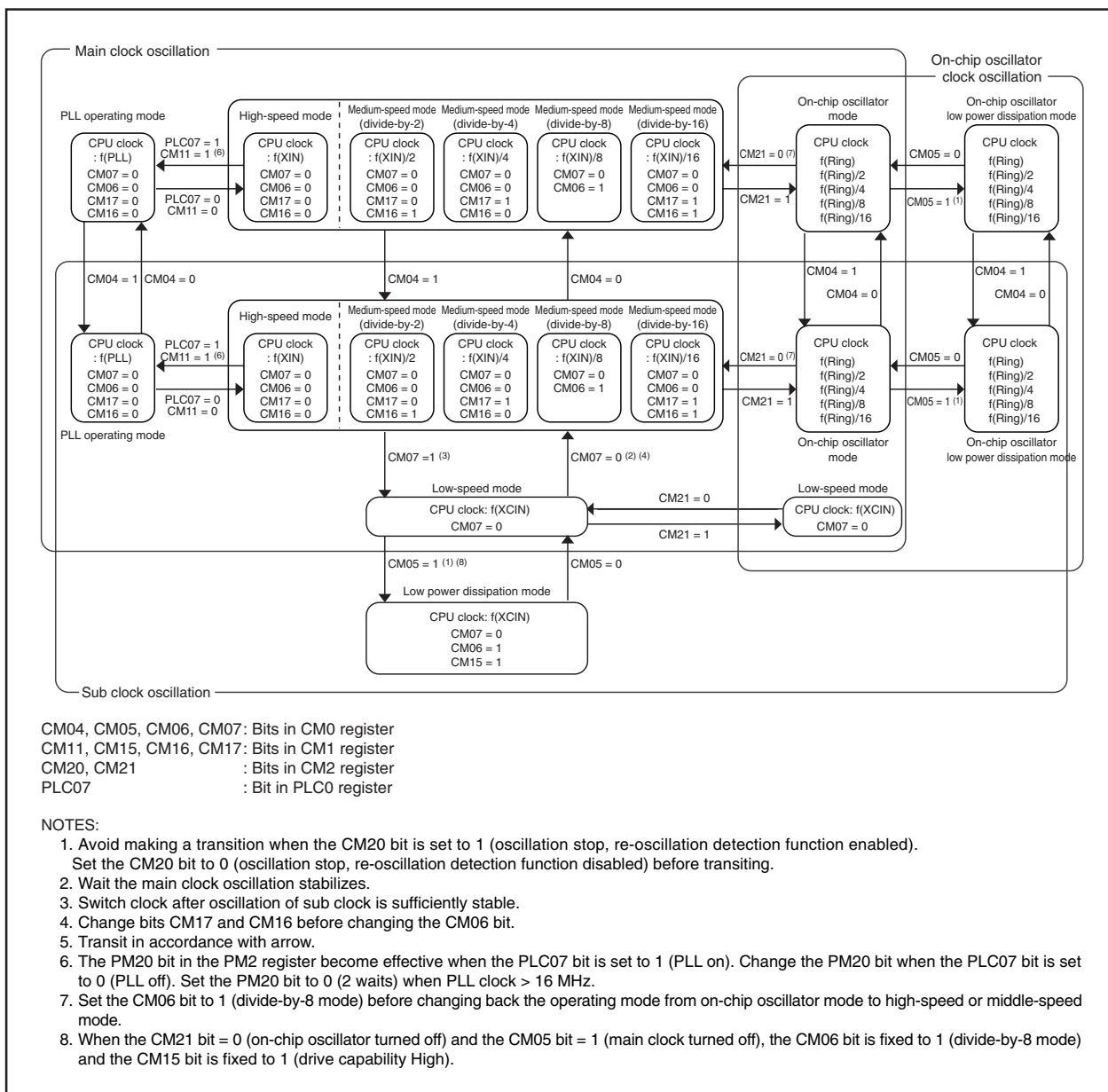


Figure 8.13 State Transition in Normal Operating Mode

Table 8.8 Allowed Transition and Setting ⁽⁹⁾

| | | State after Transition | | | | | | | |
|---------------|---|------------------------------------|-------------------------------|----------------------------|------------------------|-------------------------|---|---------------------|-----------|
| | | High-Speed Mode, Medium-Speed Mode | Low-Speed Mode ⁽²⁾ | Low Power Dissipation Mode | PLL Operating Mode (2) | On-chip Oscillator Mode | On-chip Oscillator Low Power Dissipation Mode | Stop Mode | Wait Mode |
| Current State | High-speed mode, medium-speed mode | (NOTE 8) | (9) ⁽⁷⁾ | - | (13) ⁽³⁾ | (15) | - | (16) ⁽¹⁾ | (17) |
| | Low-speed mode ⁽²⁾ | (8) | / | (11) ^{(1) (6)} | - | - | - | (16) ⁽¹⁾ | (17) |
| | Low power dissipation mode | - | (10) | / | - | - | - | (16) ⁽¹⁾ | (17) |
| | PLL operating mode ⁽²⁾ | (12) ⁽³⁾ | - | - | / | - | - | - | - |
| | On-chip oscillator mode | (14) ⁽⁴⁾ | - | - | - | (NOTE 8) | (11) ⁽¹⁾ | (16) ⁽¹⁾ | (17) |
| | On-chip oscillator low power dissipation mode | - | - | - | - | (10) | (NOTE 8) | (16) ⁽¹⁾ | (17) |
| | Stop mode | (18) ⁽⁵⁾ | (18) | (18) | - | (18) ⁽⁵⁾ | (18) ⁽⁵⁾ | / | - |
| | Wait mode | (18) | (18) | (18) | - | (18) | (18) | - | / |

-: Cannot transit

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 transiting.
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 operating mode. In this mode, sub clock can be used as peripheral function clock.
3. PLL operating mode can only be entered from and changed to high-speed mode.
4. Set the CM06 bit to 1 (divide-by-8 mode) before transiting from on-chip oscillator mode to high-speed or medium-speed mode.
5. When exiting stop mode, the CM06 bit is set to 1 (divide-by-8 mode).
6. If the CM05 bit is set to 1 (main clock stop), then the CM06 bit is set to 1 (divide-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 sub clock oscillation turned on or off) are shown in the table below.

| | | Sub Clock Oscillating | | | | | Sub Clock Turned Off | | | | |
|-----------------------|--------------|-----------------------|-------------|-------------|-------------|--------------|----------------------|-------------|-------------|-------------|--------------|
| | | No Division | Divide-by-2 | Divide-by-4 | Divide-by-8 | Divide-by-16 | No Division | Divide-by-2 | Divide-by-4 | Divide-by-8 | Divide-by-16 |
| Sub Clock Oscillating | No division | / | (4) | (5) | (7) | (6) | (1) | - | - | - | - |
| | Divide-by-2 | (3) | / | (5) | (7) | (6) | - | (1) | - | - | - |
| | Divide-by-4 | (3) | (4) | / | (7) | (6) | - | - | (1) | - | - |
| | Divide-by-8 | (3) | (4) | (5) | / | (6) | - | - | - | (1) | - |
| | Divide-by-16 | (3) | (4) | (5) | (7) | / | - | - | - | - | (1) |
| Sub Clock Turned Off | No division | (2) | - | - | - | - | (4) | (5) | (7) | (6) | / |
| | Divide-by-2 | - | (2) | - | - | - | (3) | (5) | (7) | (6) | / |
| | Divide-by-4 | - | - | (2) | - | - | (3) | (4) | (7) | (6) | / |
| | Divide-by-8 | - | - | - | (2) | - | (3) | (4) | (5) | (6) | / |
| | Divide-by-16 | - | - | - | - | (2) | (3) | (4) | (5) | (7) | / |

9. () :setting method. See right table.

| 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 divide-by-2 mode |
| (5) CM06=0 CM17=1 CM16=0 | CPU clock divide-by-4 mode |
| (6) CM06=0 CM17=1 CM16=1 | CPU clock divide-by-16 mode |
| (7) CM06=1 | CPU clock divide-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 CM0 register
 CM10, CM11, CM16, CM17: Bits in CM1 register
 CM20, CM21 : Bits in CM2 register
 PLC07 : Bit in PLC0 register

8.5 Oscillation Stop and Re-oscillation Detection Function

The oscillation stop and re-oscillation detection function is such that main clock oscillation circuit stop and re-oscillation are detected. At oscillation stop, re-oscillation detection, reset or oscillation stop, re-oscillation detection interrupt request are generated. Which is to be generated can be selected using the CM27 bit in the CM2 register.

The oscillation stop and re-oscillation detection function can be enabled and disabled using the CM20 bit in the CM2 register.

Table 8.9 lists a Specification Overview of Oscillation Stop and Re-oscillation Detection Function.

Table 8.9 Specification Overview of Oscillation Stop and Re-oscillation Detection Function

| Item | Specification |
|---|--|
| Oscillation stop detectable clock and frequency bandwidth | $f(XIN) \geq 2 \text{ MHz}$ |
| Enabling condition for oscillation stop and re-oscillation detection function | Set CM20 bit to 1 (enabled) |
| 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 is generated (when CM27 bit =1) |

8.5.1 Operation when CM27 Bit = 0 (Oscillation Stop Detection Reset)

Where 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 **4. Special Function Registers (SFRs)**, **5. Resets**).

This status is reset with hardware reset. 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).

8.5.2 Operation when CM27 Bit = 1 (Oscillation Stop, Re-oscillation Detection Interrupt)

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

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

Where 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, re-oscillation detection interrupt request is generated.
- CM22 bit = 1 (main clock stop detected)
- CM23 bit = 1 (main clock stopped)
- CM21 bit remains unchanged

Where 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, re-oscillation detection interrupt request is generated.
- CM22 bit = 1 (main clock re-oscillation detected)
- CM23 bit = 0 (main clock oscillation)
- CM21 bit remains unchanged

8.5.3 How to Use Oscillation Stop and Re-oscillation Detection Function

- The oscillation stop, re-oscillation detection 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, the clock source for the CPU clock and peripheral function must be switched to the main clock in the program. Figure 8.14 shows the Procedure to Switch Clock Source from On-chip Oscillator to Main Clock.
- Simultaneously with oscillation stop, re-oscillation detection interrupt request 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 in the 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 and re-oscillation detection function, set the CM02 bit to 0 (peripheral function clocks not turned off during wait mode).
- Since the oscillation stop and re-oscillation detection function is provided in preparation for main clock stop due to external sources, set the CM20 bit to 0 (oscillation stop, re-oscillation detection function disabled) where the main clock is stopped or oscillated in the 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.

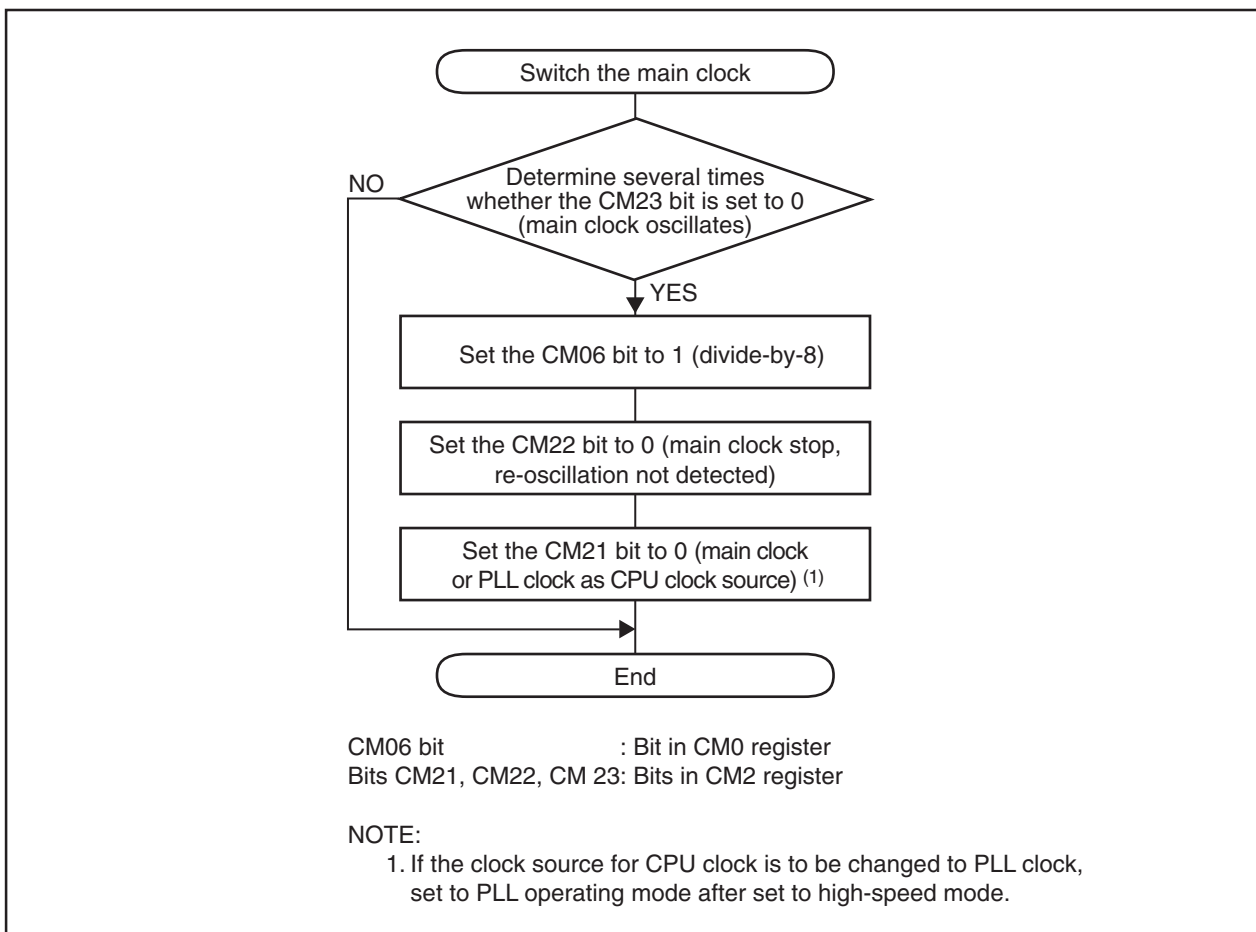


Figure 8.14 Procedure to Switch Clock Source from On-chip Oscillator to Main Clock

9. 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 9.1 shows the PRCR Register. The registers protected by the PRCR register are listed below.

- Registers protected by the PRC0 bit: Registers CM0, CM1, CM2, PLC0, PCLKR, and CCLKR
- Registers protected by the PRC1 bit: Registers PM0, PM1, PM2, TB2SC, INVC0, and INVC1
- Registers protected by the PRC2 bit: Registers PD7, PD9, and S3C

Set the PRC2 bit to 1 (write enabled) and then write to given address, and the PRC2 bit will be set 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. Bits PRC0 and PRC1 are not automatically set to 0 by writing to given address. They can only be set to 0 in a program.

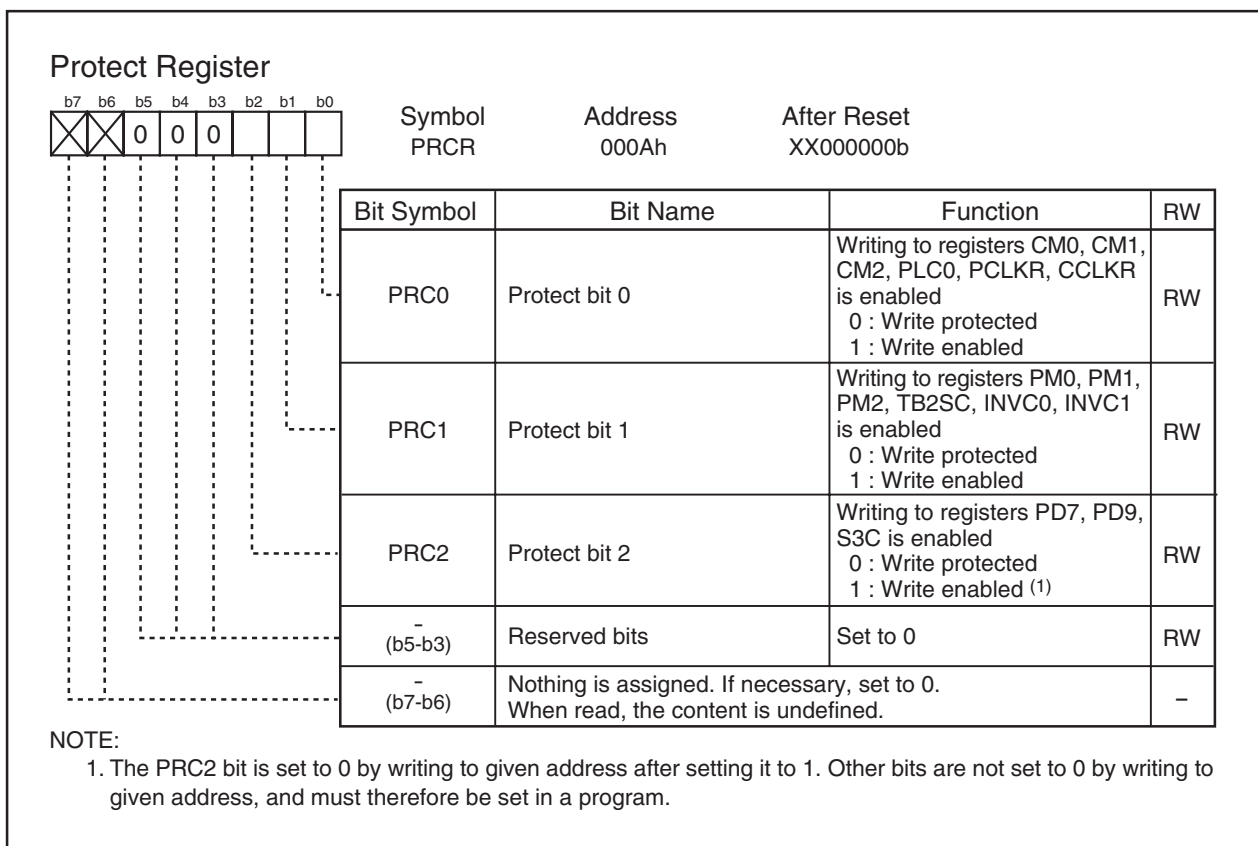


Figure 9.1 PRCR Register

10. Interrupts

10.1 Type of Interrupts

Figure 10.1 shows the Types of Interrupts.

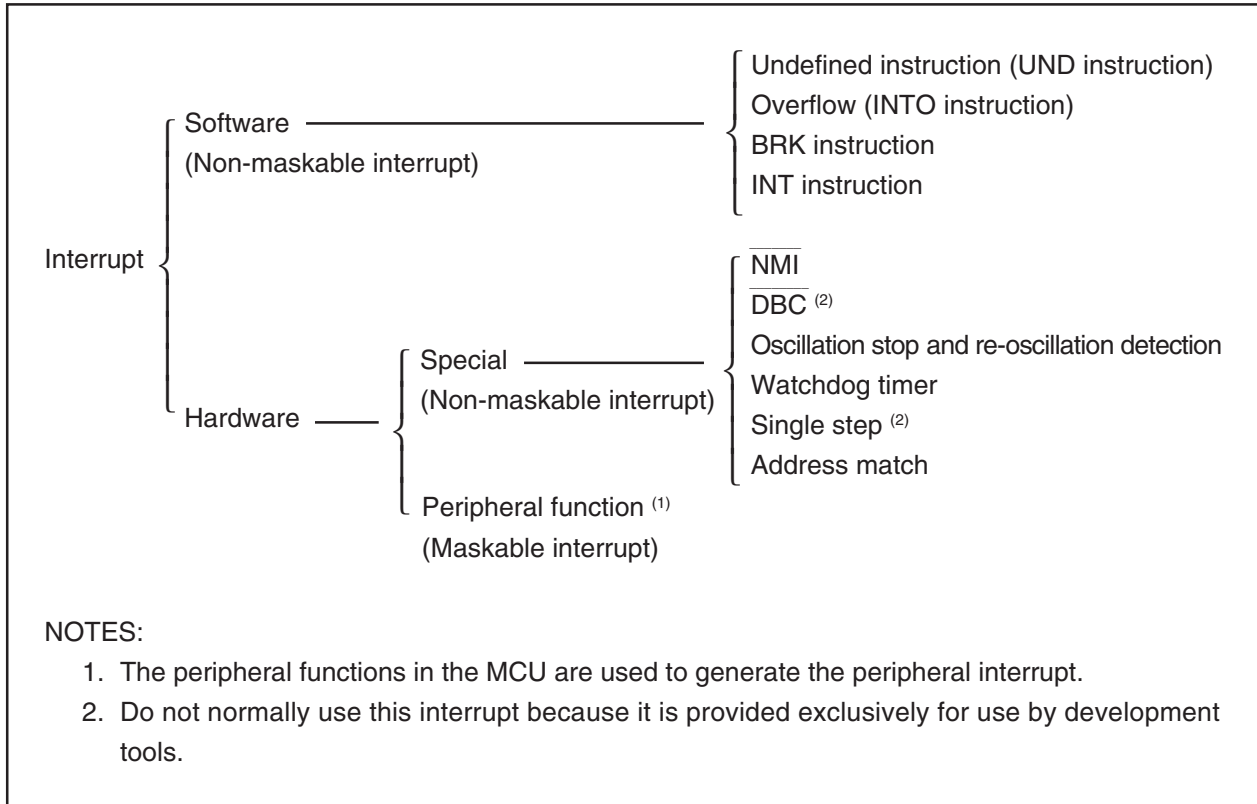


Figure 10.1 Types of Interrupts

- 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.

10.2 Software Interrupts

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

10.2.1 Undefined Instruction Interrupt

An undefined instruction interrupt is generated when executing the UND instruction.

10.2.2 Overflow Interrupt

An overflow interrupt is generated when executing the INTO instruction with the O flag in the FLG register 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

10.2.3 BRK Interrupt

A BRK interrupt is generated when executing the BRK instruction.

10.2.4 INT Instruction Interrupt

An INT instruction interrupt is generated 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 set 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.

10.3 Hardware Interrupts

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

10.3.1 Special Interrupts

Special interrupts are non-maskable interrupts.

10.3.1.1 $\overline{\text{NMI}}$ Interrupt

An $\overline{\text{NMI}}$ interrupt is generated when input on the $\overline{\text{NMI}}$ pin changes state from high to low. For details, refer to **10.7 $\overline{\text{NMI}}$ Interrupt**.

10.3.1.2 $\overline{\text{DBC}}$ Interrupt

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

10.3.1.3 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 **11. Watchdog Timer**.

10.3.1.4 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 **8. Clock Generation Circuit**.

10.3.1.5 Single-Step Interrupt

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

10.3.1.6 Address Match Interrupt

An address match interrupt is generated immediately before executing the instruction at the address indicated by registers RMAD0 to RMAD3 that corresponds to one of the AIER0 or AIER1 bit in the AIER register or the AIER20 or AIER21 bit in the AIER2 register which is 1 (address match interrupt enabled). For details, refer to **10.10 Address Match Interrupt**.

10.3.2 Peripheral Function Interrupts

The peripheral function interrupt is generated when a request from the peripheral functions in the MCU is acknowledged. The peripheral function interrupt is a maskable interrupt. See **Table 10.2 Relocatable Vector Tables** about how the peripheral function interrupt occurs. Refer to the descriptions of each function for details.

10.4 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 10.2 shows the Interrupt Vector.

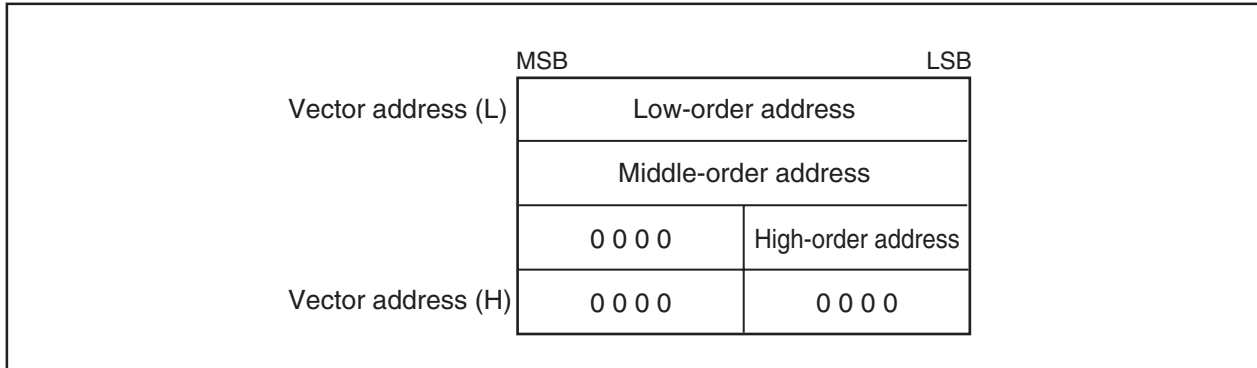


Figure 10.2 Interrupt Vector

10.4.1 Fixed Vector Tables

The fixed vector tables are allocated to the addresses from FFFDCh to FFFFFh. Table 10.1 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 **21.2 Functions to Prevent Flash Memory from Rewriting**.

Table 10.1 Fixed Vector Tables

| Interrupt Source | Vector table Addresses Address (L) to Address (H) | Reference |
|--|--|---|
| Undefined instruction (UND instruction) | FFFDCh to FFFDFh | M16C/60, M16C/20, M16C/Tiny |
| Overflow (INTO instruction) | FFFE0h to FFFE3h | Series Software Manual |
| BRK instruction ⁽²⁾ | FFFE4h to FFFE7h | |
| Address match | FFFE8h to FFEFBh | 10.10 Address Match Interrupt |
| Single step ⁽¹⁾ | FFFECh to FFEFh | - |
| Oscillation stop and re-oscillation detection, Watchdog timer | FFFF0h to FFFF3h | 8. Clock Generation Circuit 11. Watchdog Timer |
| DBC ⁽¹⁾ | FFFF4h to FFFF7h | - |
| NMI | FFFF8h to FFFFh | 10.7 NMI Interrupt |
| Reset | FFFFCh to FFFFh | 5. Resets |

NOTES:

- Do not normally use this interrupt because it is provided exclusively for use by development tools.
- If the contents of address FFFE7h is FFh, program execution starts from the address shown by the vector in the relocatable vector table.

10.4.2 Relocatable Vector Tables

The 256 bytes beginning with the start address set in the INTB register comprise a relocatable vector table area. Table 10.2 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 10.2 Relocatable Vector Tables

| Interrupt Source | Vector Address ⁽¹⁾ Address (L) to Address (H) | Software Interrupt Number | Reference |
|--|--|------------------------------|--|
| BRK Instruction ⁽²⁾ | +0 to +3 (0000h to 0003h) | 0 | M16C/60, M16C/20, M16C/Tiny Series Software Manual |
| CAN0/1 wake-up ⁽¹⁰⁾ | +4 to +7 (0004h to 0007h) | 1 | 19. CAN Module |
| CAN0 successful reception | +8 to +11 (0008h to 000Bh) | 2 | |
| CAN0 successful transmission | +12 to +15 (000Ch to 000Fh) | 3 | |
| INT3 | +16 to +19 (0010h to 0013h) | 4 | 10.6 INT Interrupt |
| Timer B5 | +20 to +23 (0014h to 0017h) | 5 | 13. Timers |
| Timer B4, UART1 bus collision detection ^{(3) (9)} | +24 to +27 (0018h to 001Bh) | 6 | 13. Timers |
| Timer B3, UART0 bus collision detection ^{(4) (9)} | +28 to +31 (001Ch to 001Fh) | 7 | 15. Serial Interface |
| CAN1 successful reception, INT5 ⁽⁵⁾ | +32 to +35 (0020h to 0023h) | 8 | 19. CAN Module, 10.6 INT Interrupt |
| SI/O3, CAN1 successful transmission, INT4 ⁽⁶⁾ | +36 to +39 (0024h to 0027h) | 9 | 15. Serial Interface, 19. CAN Module, 10.6 INT Interrupt |
| UART2 bus collision detection ⁽⁹⁾ | +40 to +43 (0028h to 002Bh) | 10 | 15. Serial Interface |
| DMA0 | +44 to +47 (002Ch to 002Fh) | 11 | 12. DMAC |
| DMA1 | +48 to +51 (0030h to 0033h) | 12 | |
| CAN0/1 error ⁽¹¹⁾ | +52 to +55 (0034h to 0037h) | 13 | 19. CAN Module |
| A/D, key input ⁽⁷⁾ | +56 to +59 (0038h to 003Bh) | 14 | 16. A/D Converter, 10.8 Key Input Interrupt |
| UART2 transmission, NACK2 ⁽⁸⁾ | +60 to +63 (003Ch to 003Fh) | 15 | 15. Serial Interface |
| UART2 reception, ACK2 ⁽⁸⁾ | +64 to +67 (0040h to 0043h) | 16 | |
| UART0 transmission, NACK0 ⁽⁸⁾ | +68 to +71 (0044h to 0047h) | 17 | |
| UART0 reception, ACK0 ⁽⁸⁾ | +72 to +75 (0048h to 004Bh) | 18 | |
| UART1 transmission, NACK1 ⁽⁸⁾ | +76 to +79 (004Ch to 004Fh) | 19 | |
| UART1 reception, ACK1 ⁽⁸⁾ | +80 to +83 (0050h to 0053h) | 20 | |
| Timer A0 | +84 to +87 (0054h to 0057h) | 21 | |
| Timer A1 | +88 to +91 (0058h to 005Bh) | 22 | |
| Timer A2 | +92 to +95 (005Ch to 005Fh) | 23 | |
| Timer A3 | +96 to +99 (0060h to 0063h) | 24 | |
| Timer A4 | +100 to +103 (0064h to 0067h) | 25 | |
| Timer B0 | +104 to +107 (0068h to 006Bh) | 26 | |
| Timer B1 | +108 to +111 (006Ch to 006Fh) | 27 | |
| Timer B2 | +112 to +115 (0070h to 0073h) | 28 | |
| INT0 | +116 to +119 (0074h to 0077h) | 29 | 10.6 INT Interrupt |
| INT1 | +120 to +123 (0078h to 007Bh) | 30 | |
| INT2 | +124 to +127 (007Ch to 007Fh) | 31 | |
| INT instruction interrupt ⁽²⁾ | +128 to +131 (0080h to 0083h) to +252 to +255 (00FCh to 00FFh) | 32 to 63 | M16C/60, M16C/20, M16C/Tiny Series Software Manual |

NOTES:

- Address relative to address in INTB.
- These interrupts cannot be disabled using the I flag.
- Use the IFSR07 bit in the IFSR0 register to select.
- Use the IFSR06 bit in the IFSR0 register to select.
- Use the IFSR17 bit in the IFSR1 register to select.
- Use the IFSR16 bit in the IFSR1 register to select.
Furthermore, use the IFSR00 bit in the IFSR0 register to select, when selecting SI/O3 or CAN1 successful transmission.
- Use the IFSR01 bit in the IFSR0 register to select.
- During I²C mode, NACK and ACK interrupts comprise the interrupt source.
- Bus collision detection: During IE mode, this bus collision detection constitutes the interrupt source.
During I²C mode, a start condition or a stop condition detection constitutes the interrupt source.
- Use the IFSR02 bit in the IFSR0 register to select. When the IFSR02 bit = 0, CAN0/1 wake-up is selected. When the IFSR02 bit = 1, CAN0 wake-up/error is selected.
- Use the IFSR02 bit in the IFSR0 register to select. When the IFSR02 bit = 0, CAN0/1 error is selected. When the IFSR02 bit = 1, CAN1 wake-up/error is selected.

10.5 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 non-maskable interrupts.

Use the I flag in 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 the each interrupt control register.

Figures 10.3 and 10.4 show the Interrupt Control Registers.

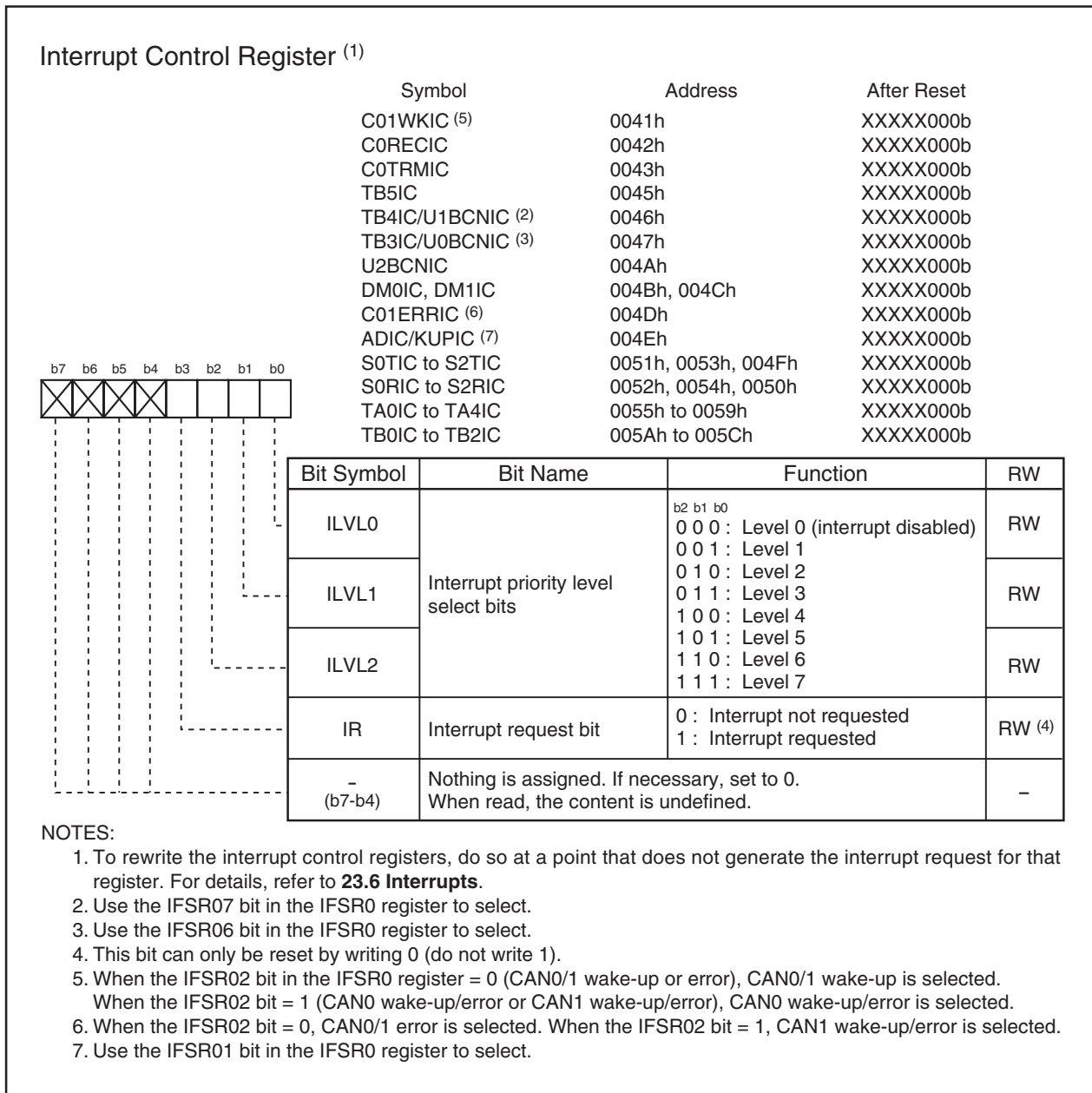


Figure 10.3 Interrupt Control Registers (1)

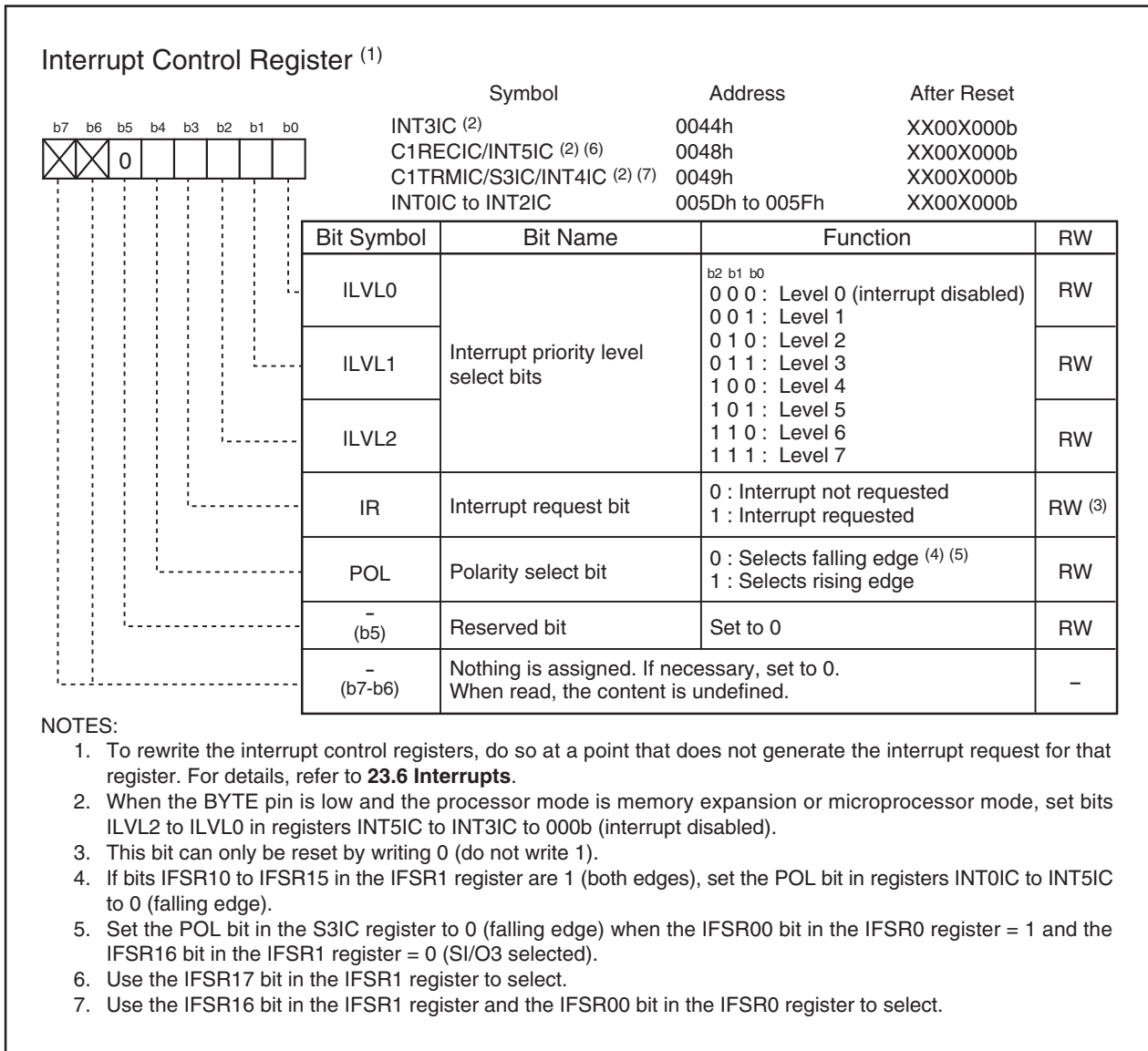


Figure 10.4 Interrupt Control Registers (2)

10.5.1 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.

10.5.2 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 set to 0 (interrupt not requested).

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

10.5.3 Bits ILVL2 to ILVL0 and IPL

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

Table 10.3 shows the settings of interrupt priority levels and Table 10.4 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 10.3 Settings of Interrupt Priority Levels


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

Table 10.4 Interrupt Priority Levels Enabled by IPL

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

10.5.4 Interrupt Sequence

An interrupt sequence — what are performed over a period from the instant an interrupt is accepted to the instant the interrupt routine is executed — is described here.

If an interrupt request is generated while an instruction is being executing, the CPU determines its priority when the execution of the instruction is completed, and transfers control to the interrupt sequence from the next cycle. However, for the SMOVB, SMOVF, SSTR or RMPA instruction, if an interrupt request is generated while the instruction is being executing, the MCU temporarily suspends the instruction being executed, and transfers control to the interrupt sequence.

The CPU behavior during the interrupt sequence is described below.

Figure 10.5 shows the Time Required for Executing Interrupt Sequence.

- (1) The CPU obtains interrupt information (interrupt number and interrupt request level) by reading address 000000h. Then, the IR bit applicable to the interrupt information is set to 0 (interrupt requested).
- (2) The FLG register, prior to an interrupt sequence, is saved to a temporary register ⁽¹⁾ within the CPU.
- (3) Flags I, D, and U in the FLG register become as follows:
 - The I flag is set to 0 (interrupt disabled)
 - The D flag is set to 0 (single-step interrupt disabled)
 - The U flag is set to 0 (ISP selected)
 However, the U flag does not change state if an INT instruction for software interrupt Nos. 32 to 63 is executed.
- (4) The temporary register ⁽¹⁾ within the CPU is saved to the stack.
- (5) The PC is saved to the stack.
- (6) The interrupt priority level of the acknowledged interrupt in IPL is set.
- (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, an instruction is executed from the starting address of the interrupt routine.

NOTE:

1. This register cannot be accessed by user.

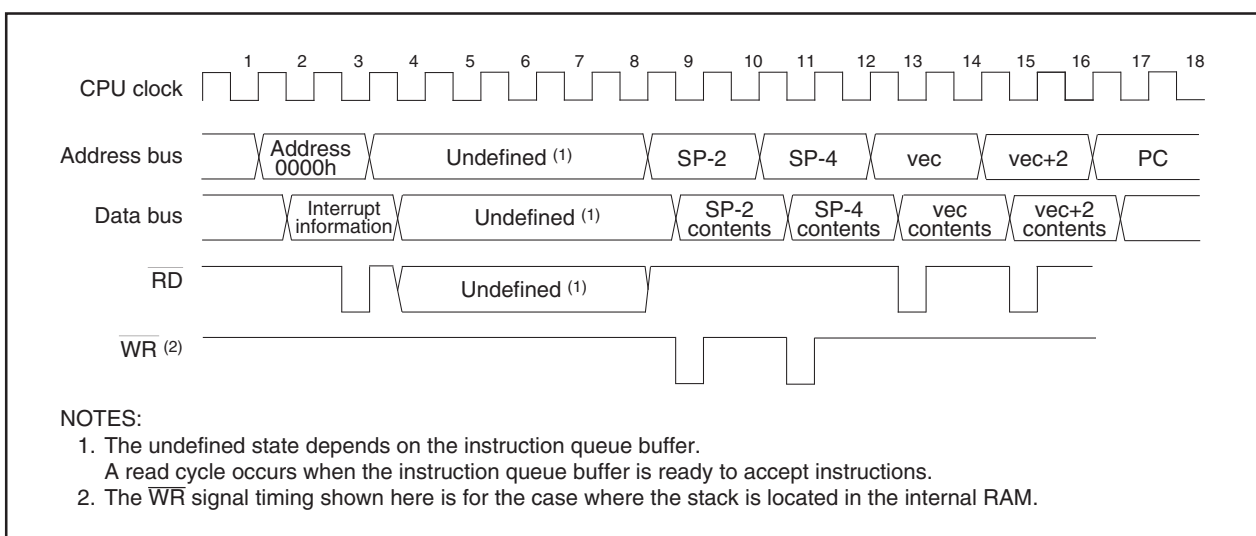


Figure 10.5 Time Required for Executing Interrupt Sequence

10.5.5 Interrupt Response Time

Figure 10.6 shows the Interrupt Response Time. The interrupt response or interrupt acknowledge time denotes a time from when an interrupt request is generated till when the first instruction in the interrupt routine is executed. Specifically, it consists of a time from when an interrupt request is generated till when the instruction then executing is completed ((a) on Figure 10.6) and a time during which the interrupt sequence is executed ((b) on Figure 10.6).

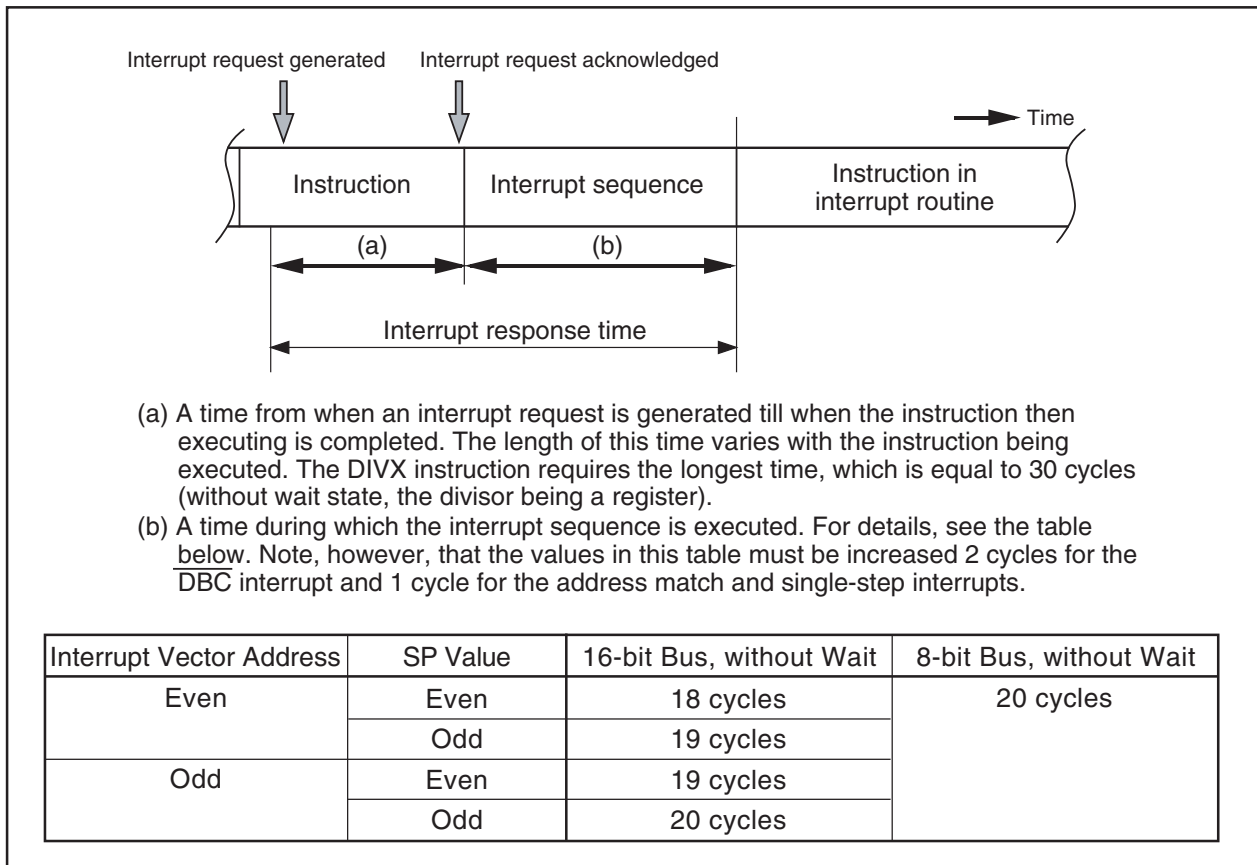


Figure 10.6 Interrupt Response Time

10.5.6 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 10.5 is set in the IPL. Table 10.5 shows the IPL Level that is Set to IPL when Software or Special Interrupts is Accepted.

Table 10.5 IPL Level that is Set to IPL when Software or Special Interrupt is Accepted

| Interrupt Sources | Value that is Set to IPL |
|--|--------------------------|
| Oscillation stop and re-oscillation detection, Watchdog timer, NMI | 7 |
| Software, Address match, DBC, Single-step | Not changed |

10.5.7 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 in 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 10.7 shows the Stack Status Before and After Acceptance of Interrupt Request.

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.

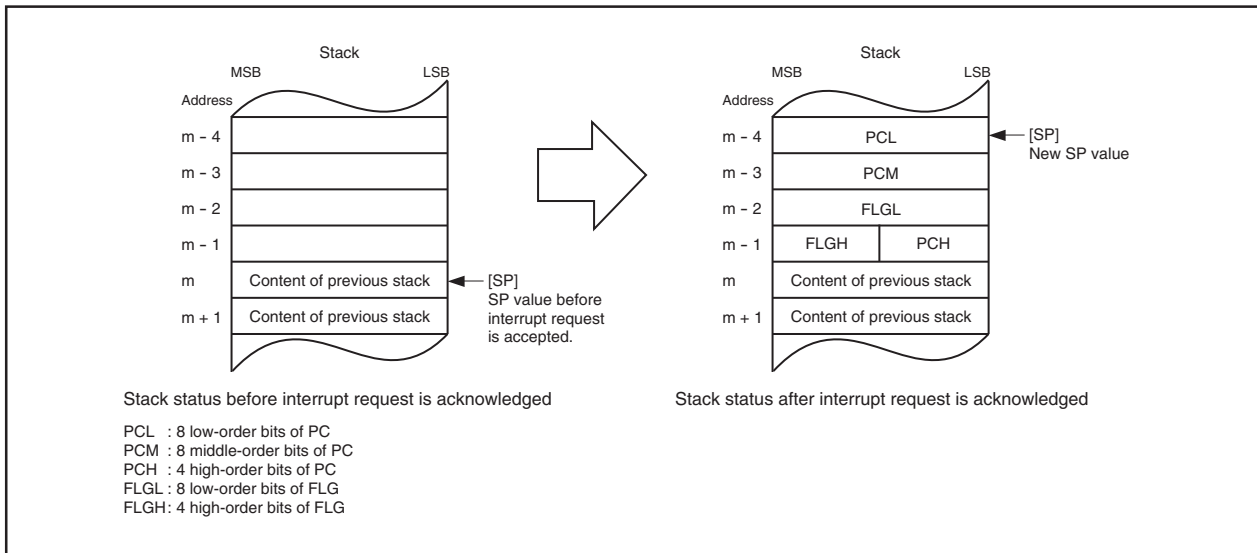


Figure 10.7 Stack Status Before and After Acceptance of Interrupt Request

The register saving operation 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 SP⁽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 10.8 shows the Register Saving Operation.

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.

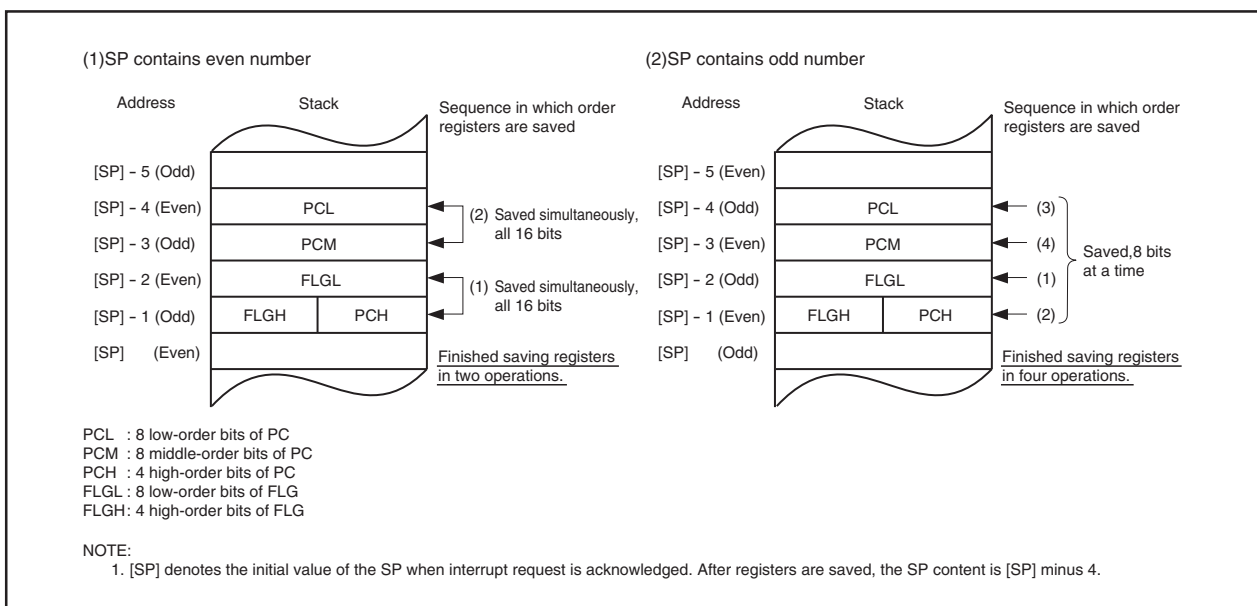


Figure 10.8 Register Saving Operation

10.5.8 Returning from 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.

Register bank is switched back to the bank used prior to the interrupt sequence by the REIT instruction.

10.5.9 Interrupt Priority

If two or more interrupt requests are sampled at the same sampling points (a timing to detect whether an interrupt request is generated or not), the interrupt request with the highest priority is acknowledged.

For maskable interrupts (peripheral functions interrupt), 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 10.9 shows the Hardware Interrupts Priority.

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

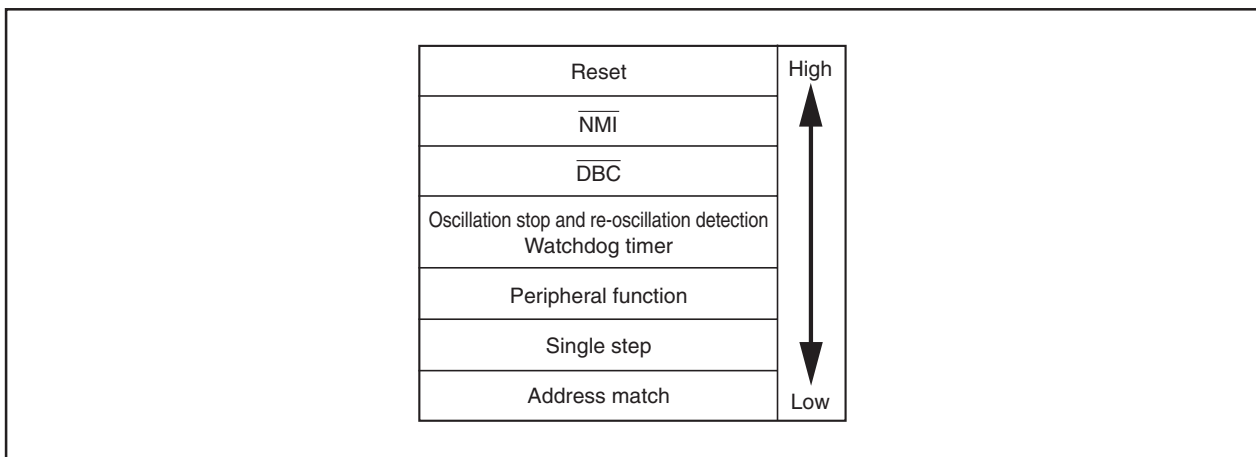


Figure 10.9 Hardware Interrupt Priority

10.5.10 Interrupt Priority Level Select Circuit

The interrupt priority level select circuit selects the highest priority interrupt when two or more interrupt requests are sampled at the same sampling point.

Figure 10.10 shows the Interrupts Priority Select Circuit.

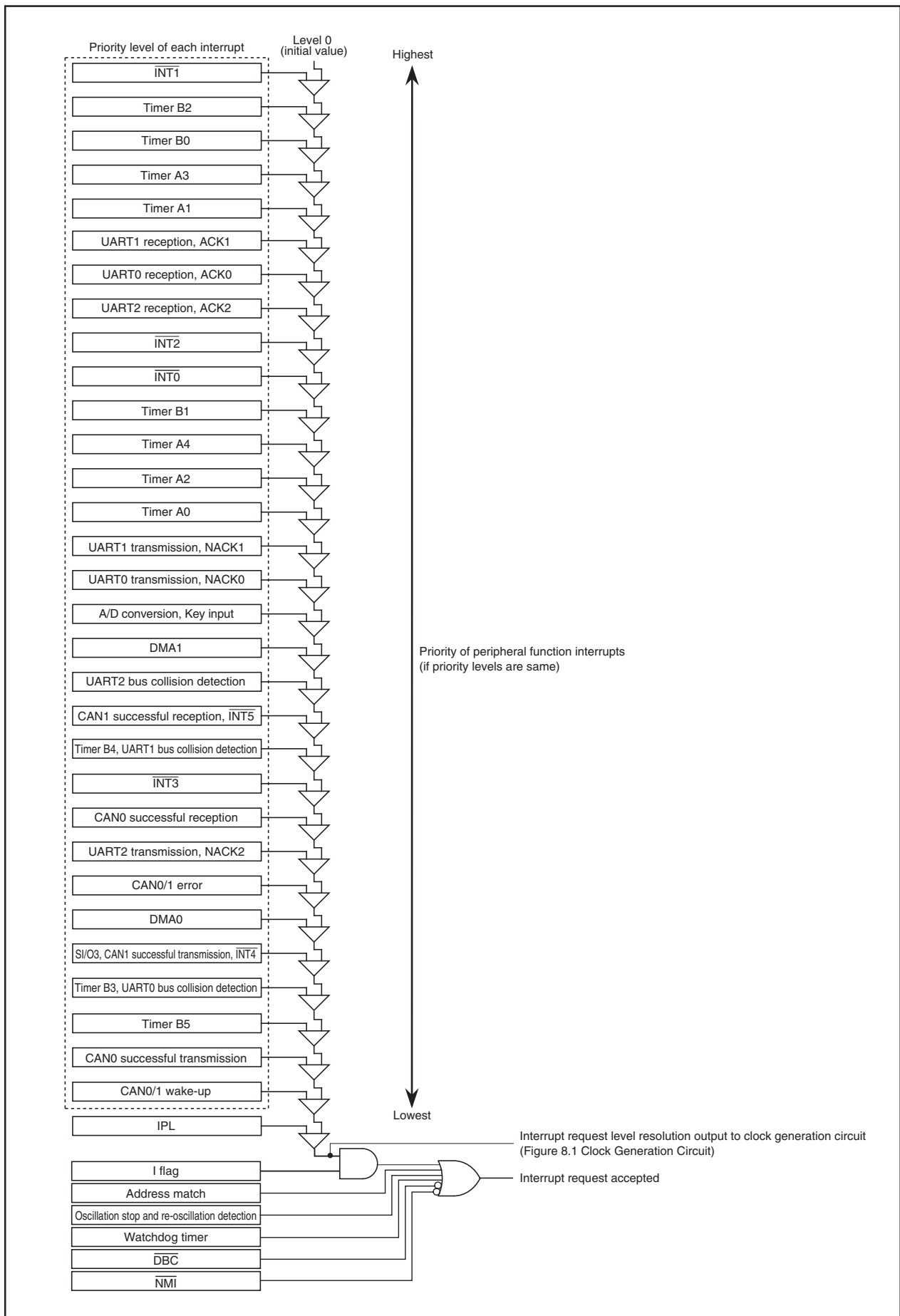


Figure 10.10 Interrupts Priority Select Circuit

10.6 $\overline{\text{INT}}$ Interrupt

$\overline{\text{INT}}_i$ interrupt ($i = 0$ to 5) is triggered by the edges of external inputs. The edge polarity is selected using bits IFSR10 to IFSR15 in the IFSR1 register.

$\overline{\text{INT}}_4$ share the interrupt vector and interrupt control register with CAN1 successful transmission and SI/O3.

$\overline{\text{INT}}_5$ share the interrupt vector and interrupt control register with CAN1 successful reception. To use the $\overline{\text{INT}}_4$ interrupt, set the IFSR16 bit in the IFSR1 register to 1 ($\overline{\text{INT}}_4$). To use the $\overline{\text{INT}}_5$ interrupt, set the IFSR17 bit in the IFSR1 register to 1 ($\overline{\text{INT}}_5$).

After modifying the IFSR16 or IFSR17 bit, set the corresponding IR bit to 0 (interrupt not requested) before enabling the interrupt.

Figure 10.11 shows Registers IFSR0 and IFSR1.

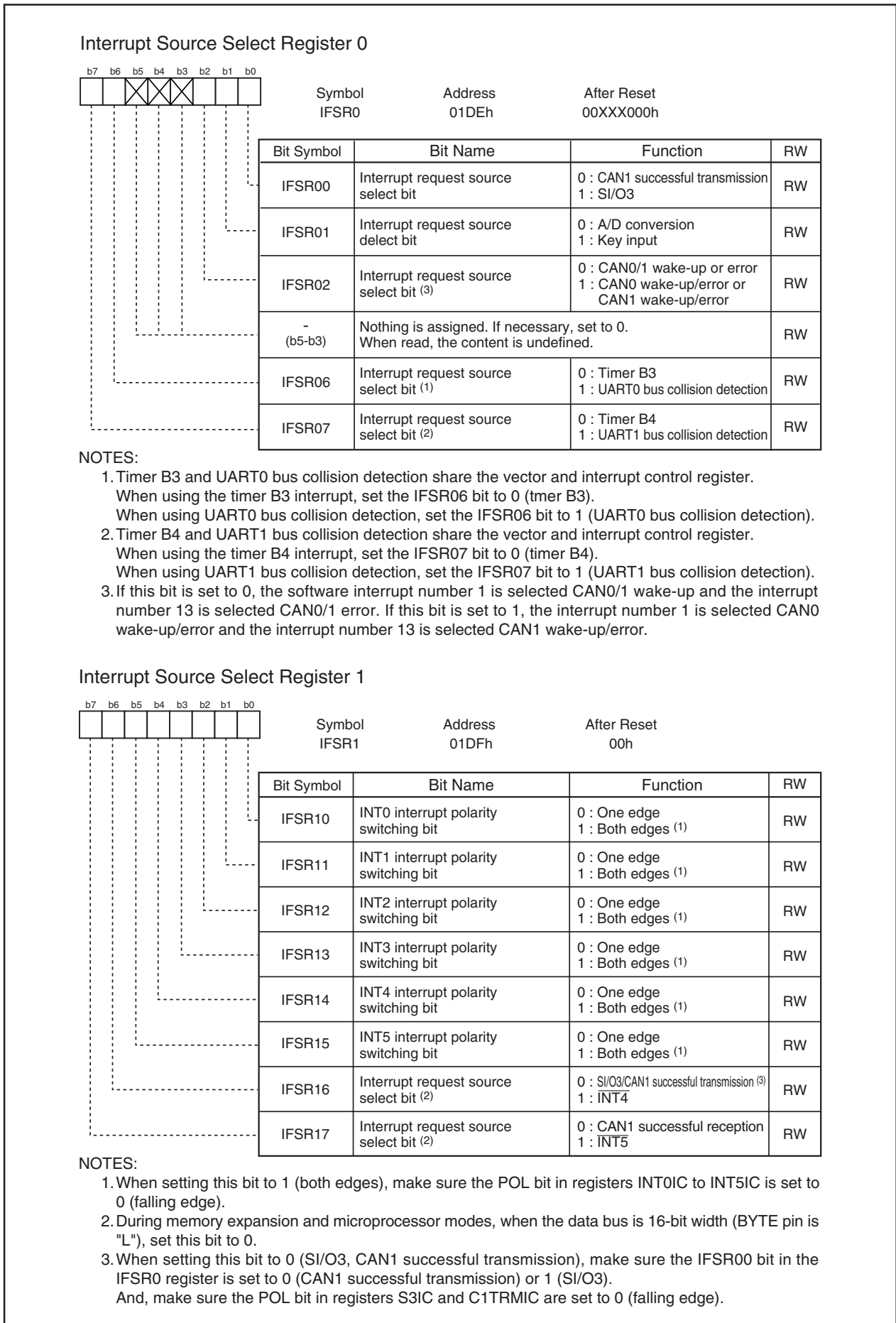


Figure 10.11 Registers IFSR0 and IFSR1

10.7 $\overline{\text{NMI}}$ Interrupt

An $\overline{\text{NMI}}$ interrupt request is generated when input on the $\overline{\text{NMI}}$ pin changes state from high to low. The $\overline{\text{NMI}}$ interrupt is a non-maskable interrupt.

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

This pin cannot be used as an input port.

10.8 Key Input Interrupt

Of P10_4 to P10_7, a key input interrupt request is generated when input on any of pins P10_4 to P10_7 which has had bits PD10_4 to PD10_7 in the PD10 register set to 0 (input) goes low. Key input interrupts can be used as a key-on wake up function, the function which gets the MCU out of wait or stop mode. However, if you intend to use the key input interrupt, do not use P10_4 to P10_7 as analog input ports. Figure 10.12 shows the Key Input Interrupt Block Diagram. Note, however, that while input on any pin which has had bits PD10_4 to PD10_7 set to 0 (input mode) is pulled low, inputs on all other pins of the port are not detected as interrupts.

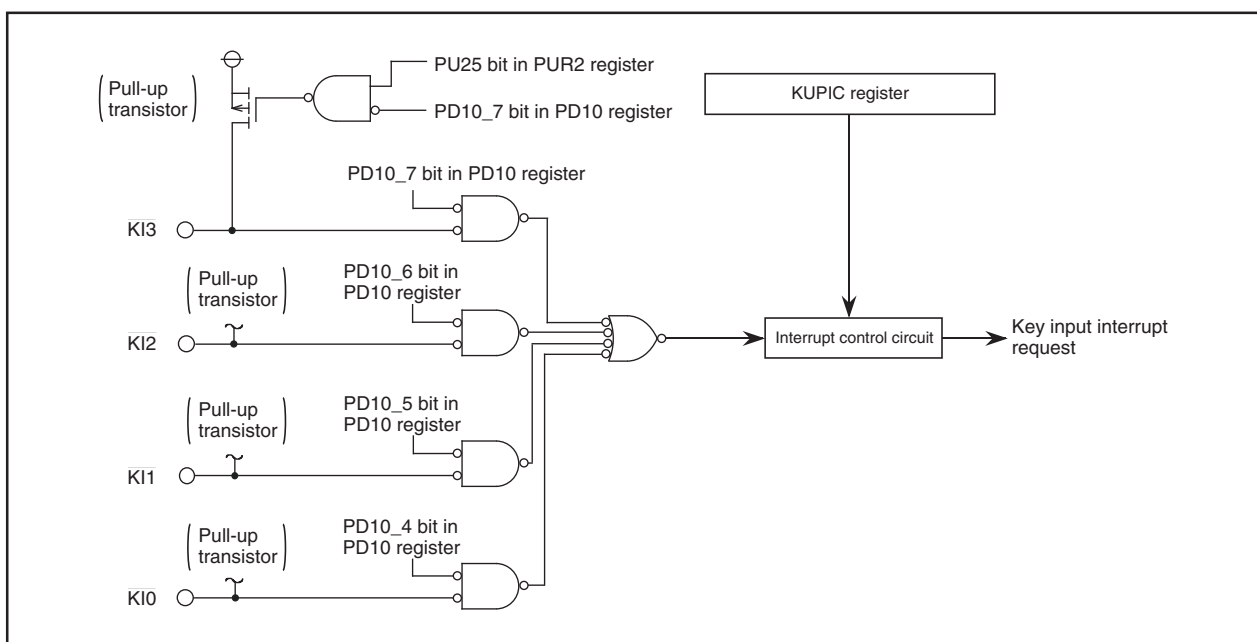


Figure 10.12 Key Input Interrupt Block Diagram

10.9 CAN0/1 Wake-up Interrupt

CAN0/1 wake-up interrupt request is generated when a falling edge is input to CRX0 or CRX1. One interrupt is allocated to CAN0/1. The CAN0/1 wake-up interrupt is enabled only when the PortEn bit = 1 (CTX/CRX function) and Sleep bit = 1 (sleep mode enabled) in the CiCTLR register (i = 0, 1). Figure 10.13 shows the CAN0/1 Wake-up Interrupt Block Diagram. Please note that the wake-up message will be lost.

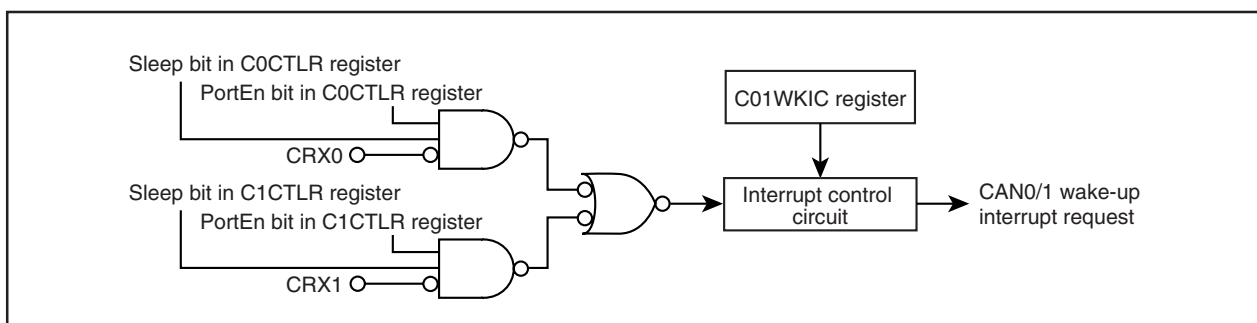


Figure 10.13 CAN0/1 Wake-up Interrupt Block Diagram

10.10 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 3). Set the start address of any instruction in the RMAD_i register. Use bits AIER0 and AIER1 in the AIER register and bits AIER20 and AIER21 in the AIER2 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 **10.5.7 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 10.6 shows the Value of PC that is Saved to Stack Area when Address Match Interrupt Request is Accepted. Table 10.7 shows the Relationship between Address Match Interrupt Sources and Associated Registers.

Note that when using the external bus in 8-bit width, no address match interrupts can be used for external areas.

Figure 10.14 shows Registers AIER, AIER2, and RMAD0 to RMAD3.

Table 10.6 Value of PC that is Saved to Stack Area when Address Match Interrupt Request is Accepted

| Instruction at Address Indicated by RMAD _i Register | Value of PC that is Saved to Stack Area |
|---|---|
| <ul style="list-style-type: none"> • 16-bit operation code instruction • Instruction shown below among 8-bit operation code instructions <pre> ADD.B:S #IMM8,dest SUB.B:S #IMM8,dest AND.B:S #IMM8,dest OR.B:S #IMM8,dest MOV.B:S #IMM8,dest STZ.B:S #IMM8,dest STNZ.B:S #IMM8,dest STZX.B:S #IMM81,#IMM82,dest CMP.B:S #IMM8,dest PUSHM src POPM dest JMPS #IMM8 JSRS #IMM8 MOV.B:S #IMM,dest (However, dest = A0 or A1) </pre> | Address indicated by RMAD _i register + 2 |
| Instructions other than the above | Address indicated by RMAD _i register + 1 |

Value of PC that is saved to stack area: Refer to **10.5.7 Saving Registers**.

Table 10.7 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 |
| Address match interrupt 2 | AIER20 | RMAD2 |
| Address match interrupt 3 | AIER21 | RMAD3 |

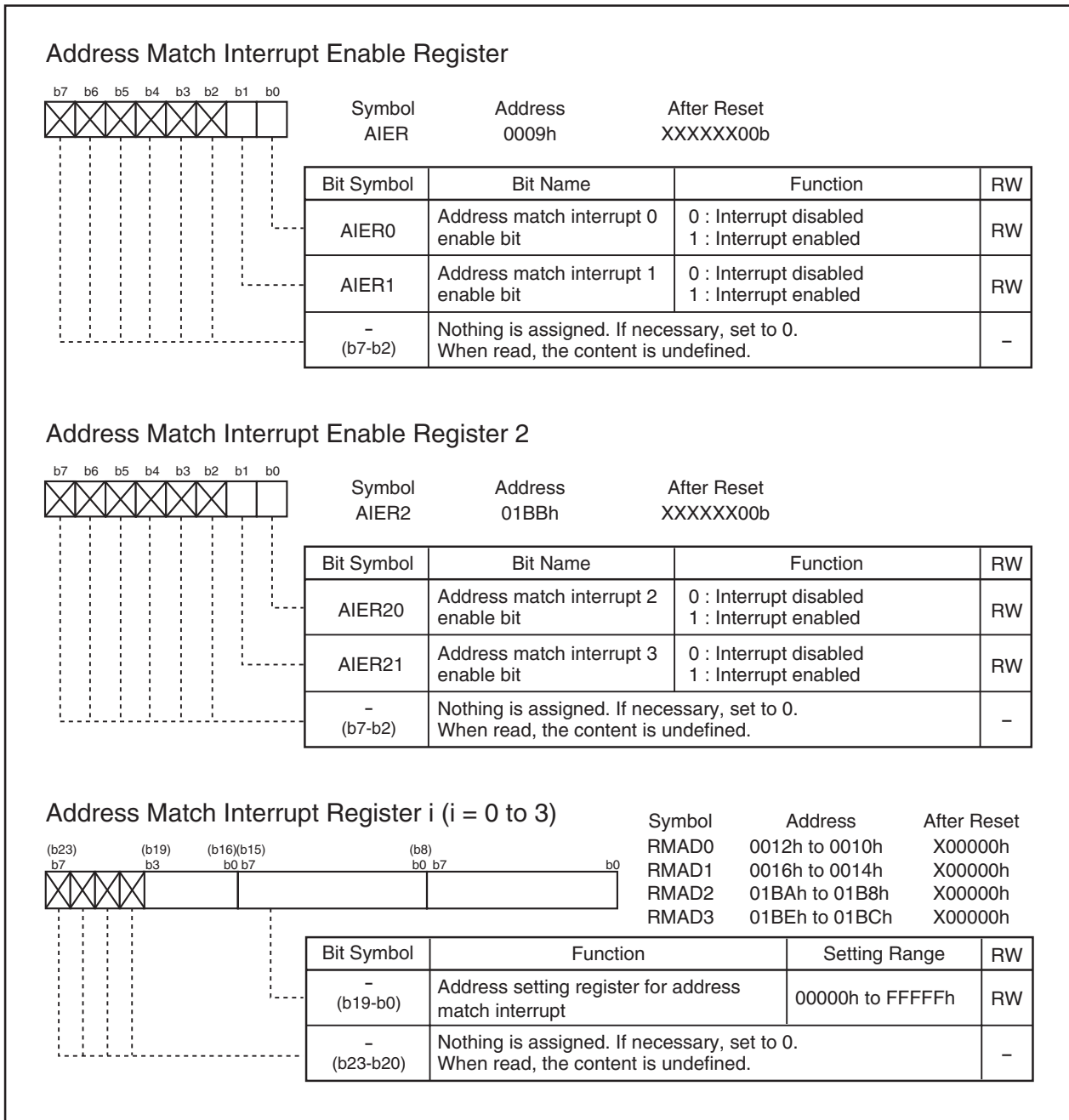


Figure 10.14 Registers AIER, AIER2, and RMAD0 to RMAD3

11. 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 (watchdog timer 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 details about watchdog timer reset.

When the main clock, on-chip oscillator clock or PLL clock is selected for CPU clock, the divide-by-n value for the prescaler can be selected to be 16 or 128. If a sub clock is selected for CPU clock, the divide-by-n value for 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, on-chip oscillator clock or PLL clock selected for CPU clock

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

With sub clock selected for CPU clock

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

For example, when CPU clock = 16 MHz and the divide-by-n value for the prescaler = 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.

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

Figure 11.1 shows the Watchdog Timer Block Diagram. Figure 11.2 shows Registers WDC and WDTS.

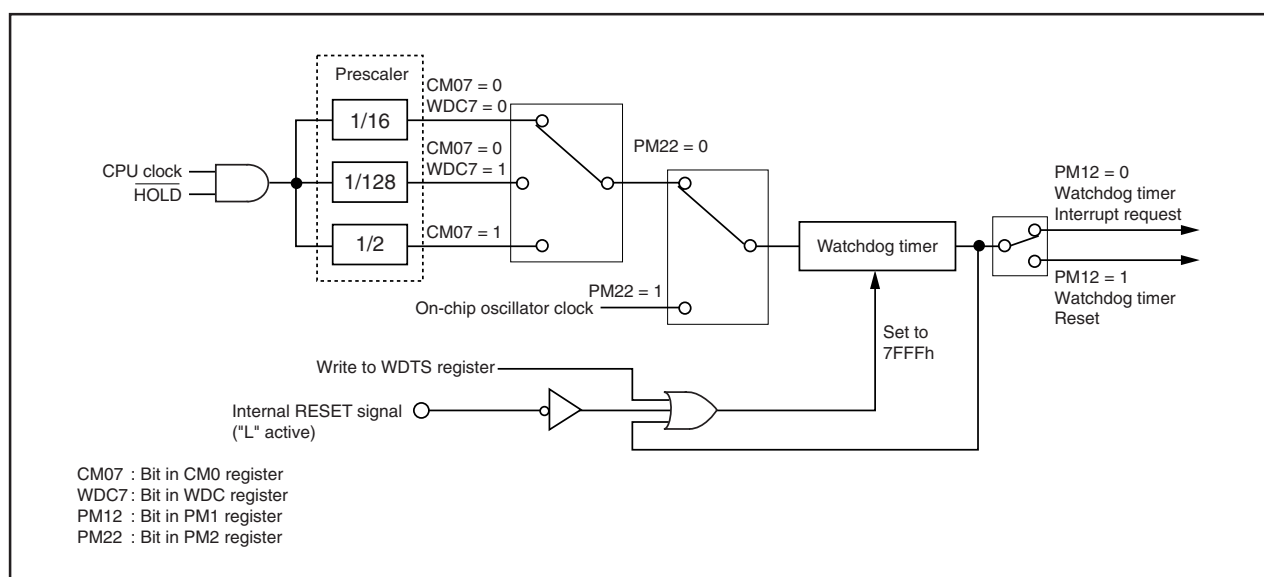


Figure 11.1 Watchdog Timer Block Diagram

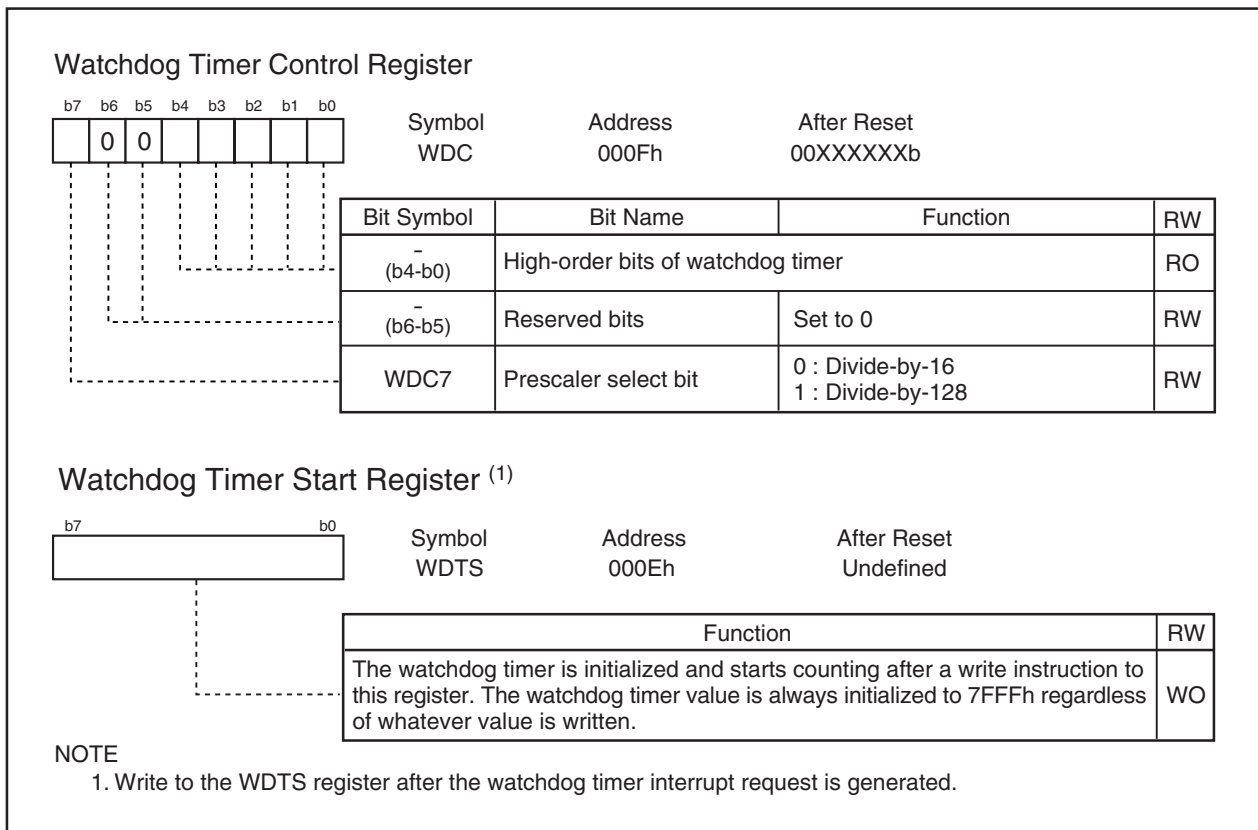


Figure 11.2 Registers WDC and WDTS

11.1 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 runaway.

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

- (1) Set the PRC1 bit in the PRCR register to 1 (write to registers PM1 and PM2 enabled).
- (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 (write to registers PM1 and PM2 disabled).
- (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 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 or hold state.

12. DMAC

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 12.1 shows the DMAC Block Diagram. Table 12.1 lists the DMAC Specifications. Figures 12.2 to 12.4 show the DMAC related-registers.

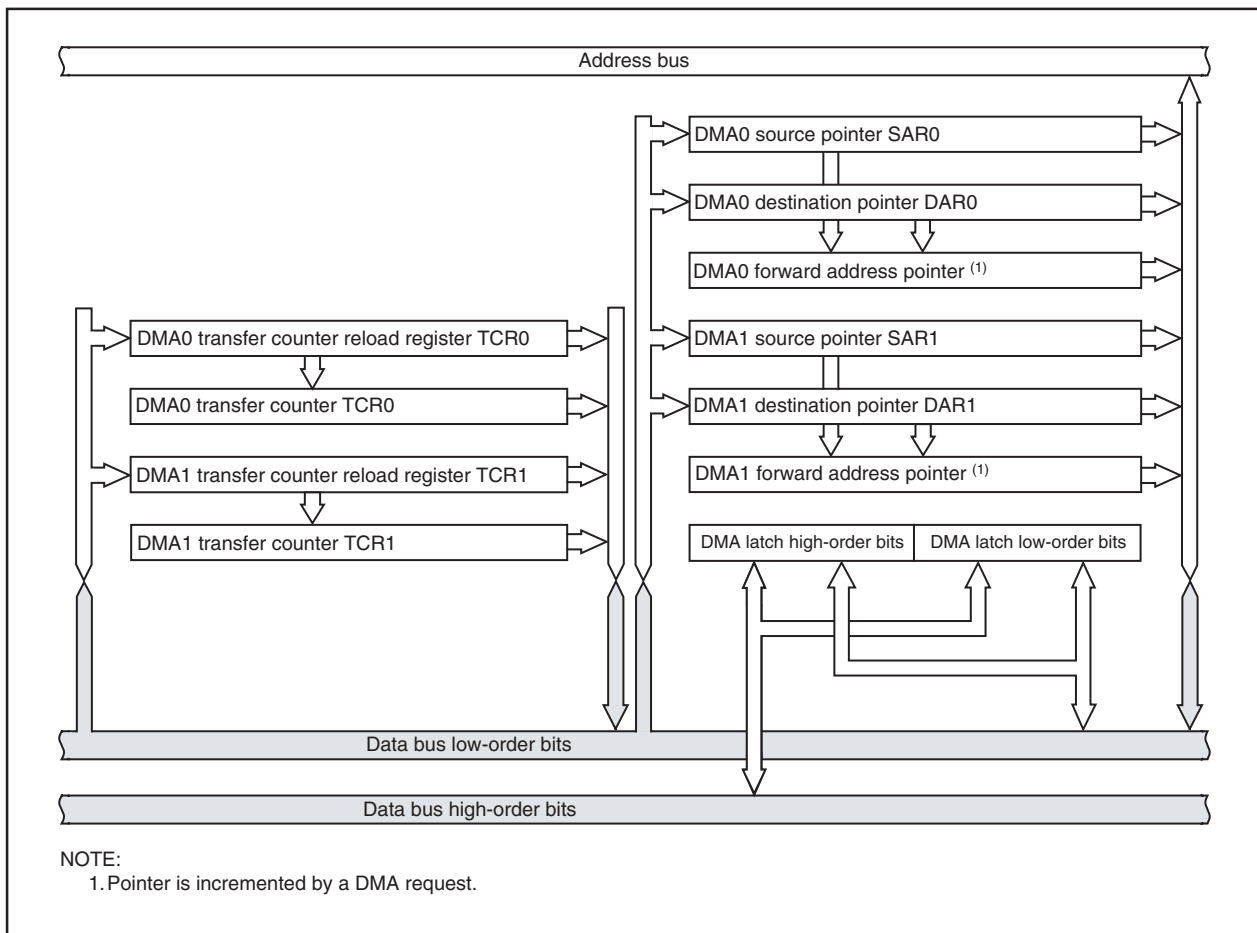


Figure 12.1 DMAC Block Diagram

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 bits DMS, and 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 = 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 **12.4 DMA Request**.

Table 12.1 DMAC Specifications

| Item | | Specification |
|--|-----------------|--|
| No. of channels | | 2 (cycle steal method) |
| Transfer memory space | | <ul style="list-style-type: none"> • From given address in the 1-Mbyte space to a fixed address • From a fixed address to given address in the 1-Mbyte space • From a fixed address to a fixed address |
| Maximum no. of bytes transferred | | 128 Kbytes (with 16-bit transfer) or 64 Kbytes (with 8-bit transfer) |
| DMA request sources ^{(1) (2)} | | Falling edge of INT0 or INT1 Both edge of $\overline{\text{INT0}}$ or $\overline{\text{INT1}}$ Timers A0 to A4 interrupt requests Timers B0 to B5 interrupt requests UART0 transmit, UART0 receive interrupt requests UART1 transmit, UART1 receive interrupt requests UART2 transmit, UART2 receive interrupt requests SI/O3 interrupt request A/D conversion interrupt 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 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 start up | | Data transfer is initiated each time a DMA request is generated when the 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. |
| DMA transfer cycles | | Minimum 3 cycles between SFR and internal RAM |

i = 0, 1

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 DMA request sources differ with each channel.
3. Make sure that no DMAC-related registers (addresses 0020h to 003Fh) are accessed by the DMAC.

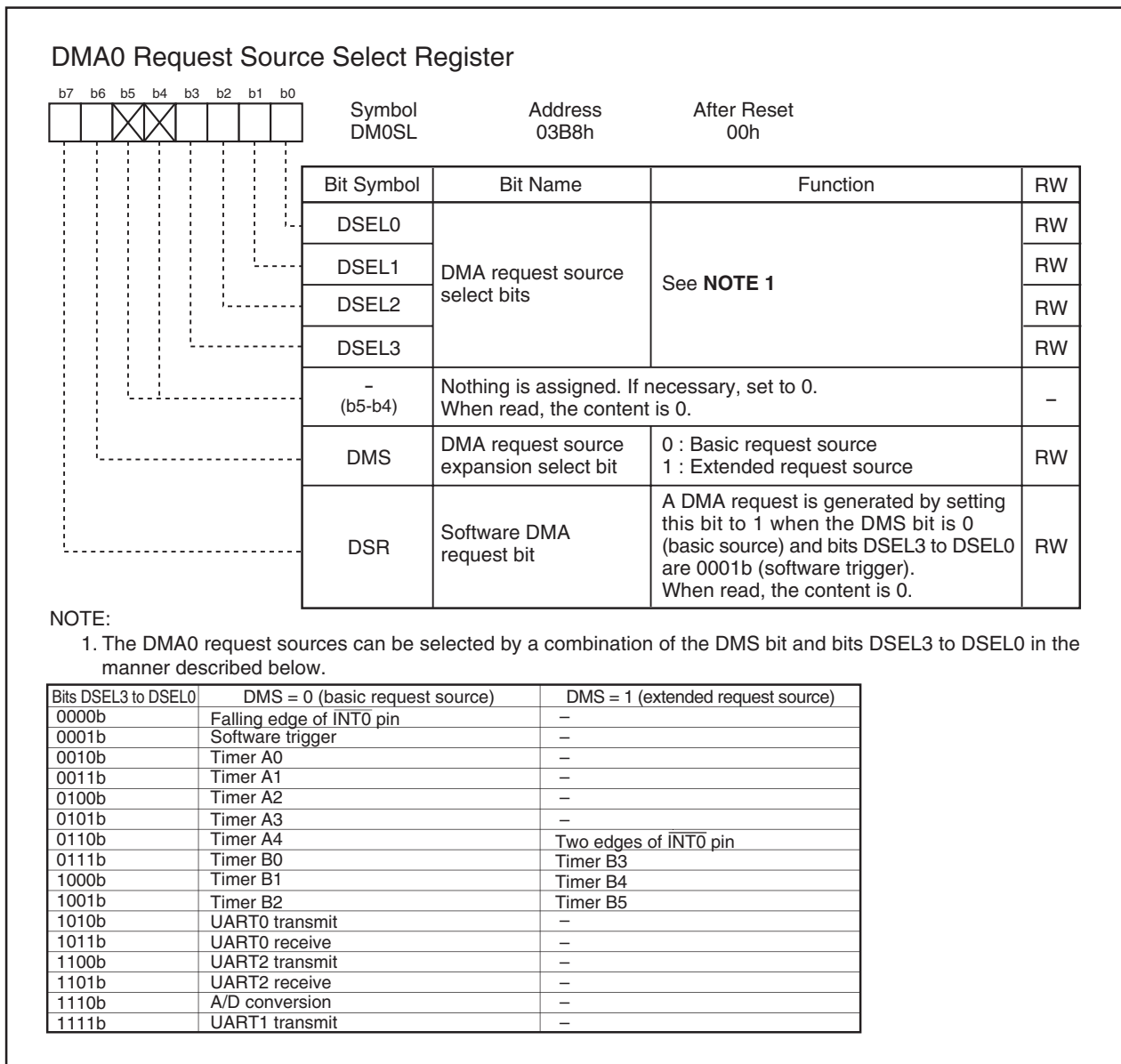


Figure 12.2 DM0SL Register

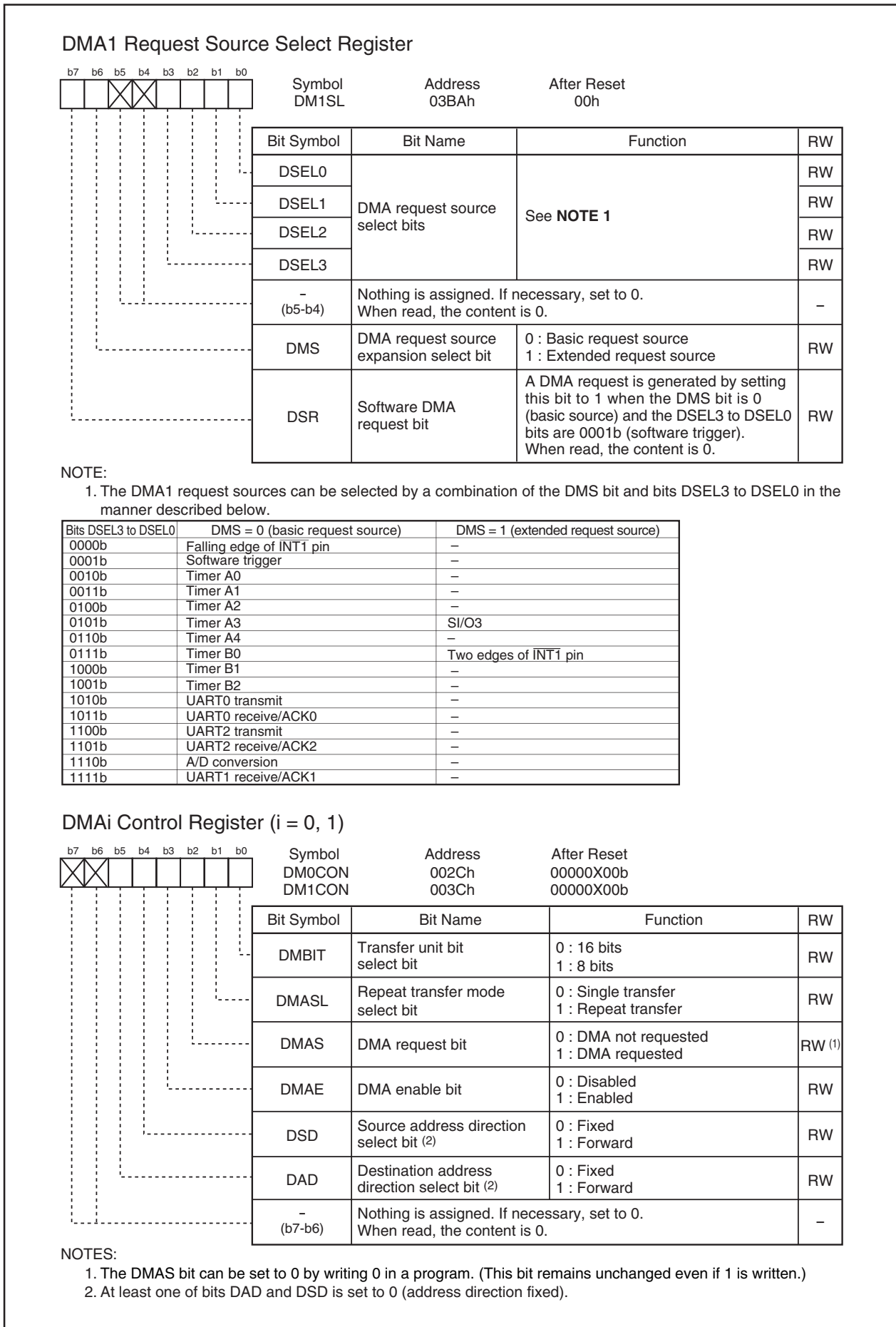


Figure 12.3 Registers DM1SL, DM0CON, and DM1CON

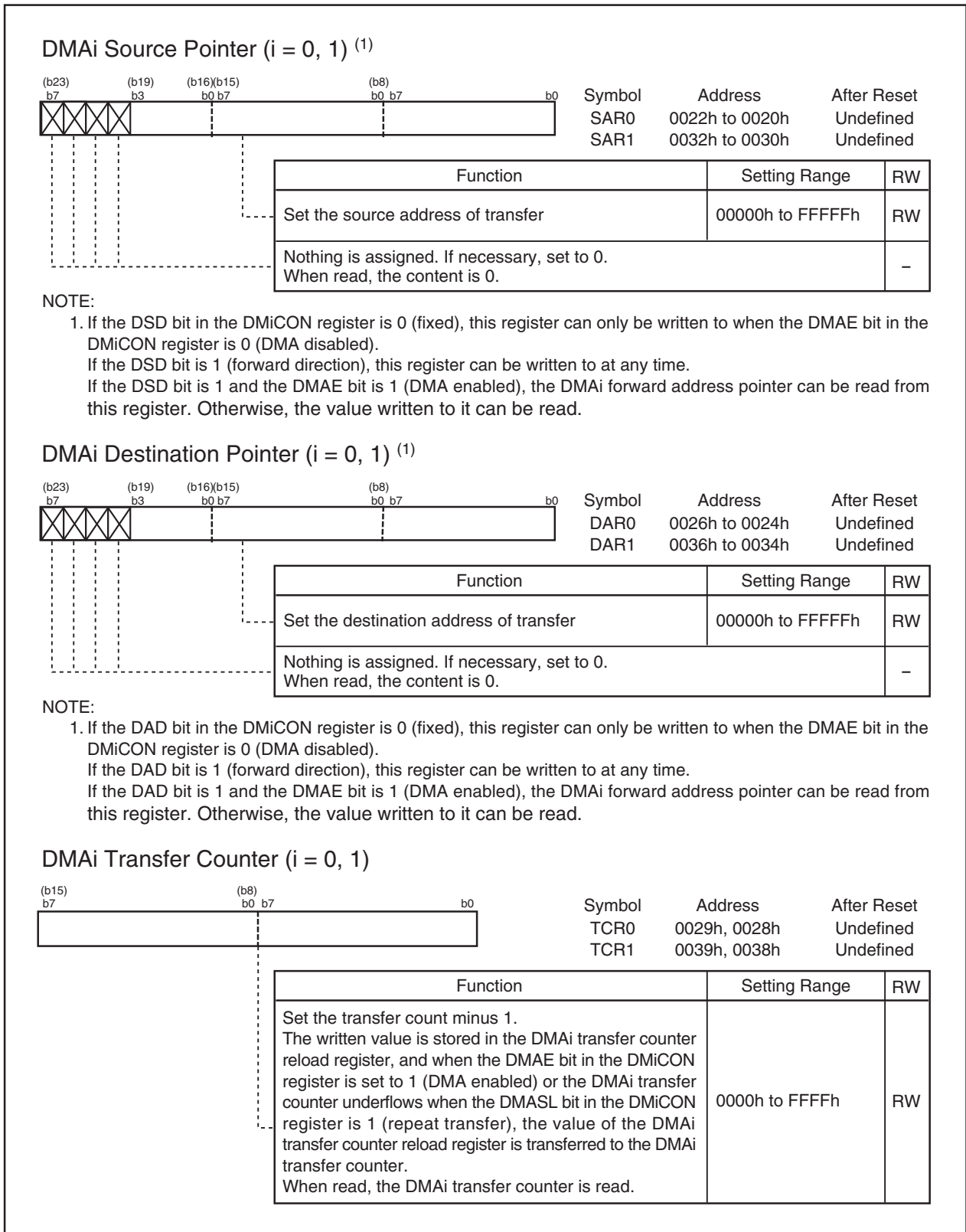


Figure 12.4 Registers SAR0, SAR1, DAR0, DAR1, TCR0, and TCR1

12.1 Transfer Cycle

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. During memory expansion and microprocessor modes, it is also affected by the BYTE pin level. Furthermore, the bus cycle itself is extended by a software wait or RDY signal.

12.1.1 Effect of Source and Destination Addresses

If the transfer unit and data bus both are 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 and data bus both are 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.

12.1.2 Effect of BYTE Pin Level

During memory expansion and microprocessor modes, if 16 bits of data are to be transferred on an 8-bit data bus (input on the BYTE pin = high), the operation is accomplished by transferring 8 bits of data twice. Therefore, this operation requires two bus cycles to read data and two bus cycles to write data.

Furthermore, if the DMAC is to access the internal area (internal ROM, internal RAM, or SFR), unlike in the case of the CPU, the DMAC does it through the data bus width selected by the BYTE pin.

12.1.3 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.

12.1.4 Effect of $\overline{\text{RDY}}$ Signal

During memory expansion and microprocessor modes, DMA transfers to and from an external area are affected by the $\overline{\text{RDY}}$ signal. Refer to **7.2.6 $\overline{\text{RDY}}$ Signal**.

Figure 12.5 shows the Transfer Cycles for 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 unit using an 8-bit bus ((2) on Figure 12.5), two source read bus cycles and two destination write bus cycles are required.

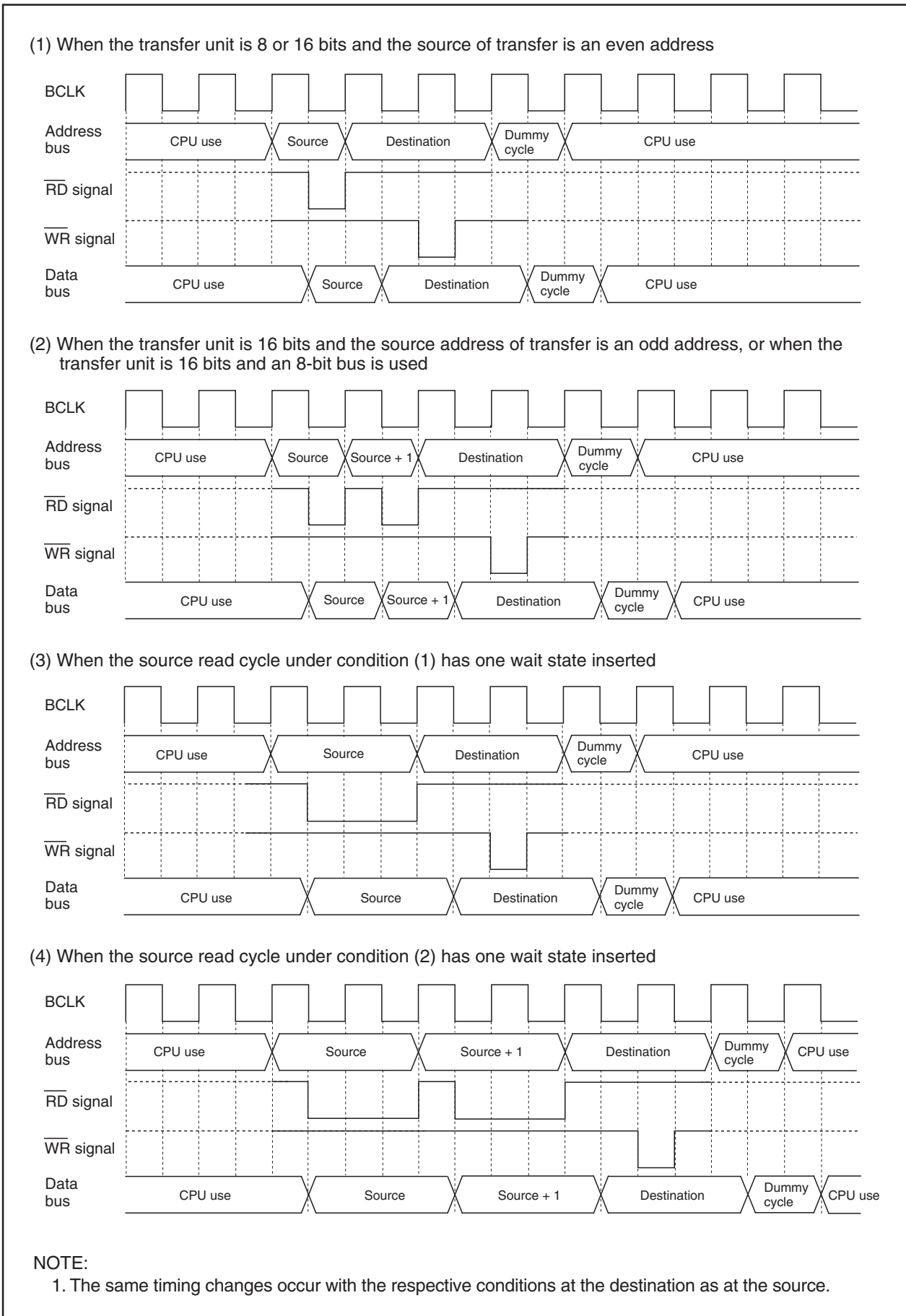


Figure 12.5 Transfer Cycles for Source Read

12.2 DMA Transfer Cycles

Any combination of even or odd transfer read and write addresses is possible.

Table 12.2 lists the DMA Transfer Cycles. Table 12.3 lists 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 12.2 DMA Transfer Cycles

| Transfer Unit | Bus Width | Access Address | Single-chip Mode | | Memory Expansion Mode Microprocessor Mode | |
|--------------------------------|-----------------------|----------------|--------------------|---------------------|--|---------------------|
| | | | No. of Read Cycles | No. of Write Cycles | No. of Read Cycles | No. of Write Cycles |
| 8-bit transfer (DMBIT = 1) | 16 bits (BYTE = L) | Even | 1 | 1 | 1 | 1 |
| | | Odd | 1 | 1 | 1 | 1 |
| | 8 bits (BYTE = H) | Even | - | - | 1 | 1 |
| | | Odd | - | - | 1 | 1 |
| 16-bit transfer (DMBIT = 0) | 16 bits (BYTE = L) | Even | 1 | 1 | 1 | 1 |
| | | Odd | 2 | 2 | 2 | 2 |
| | 8 bits (BYTE = H) | Even | - | - | 2 | 2 |
| | | Odd | - | - | 2 | 2 |

-: This condition does not exist.

Table 12.3 Coefficient j, k

| | Internal Area | | | | External Area | | | | | | |
|---|-------------------|-----------|-----------------------|------------------------|---------------|--------------------------|---------|-----------------|--------------------------|---------|---------|
| | Internal ROM, RAM | | SFR | | Separate Bus | | | Multiplexed Bus | | | |
| | No Wait | With Wait | 1 Wait ⁽¹⁾ | 2 Waits ⁽¹⁾ | No Wait | With Wait ⁽²⁾ | | | With Wait ⁽²⁾ | | |
| | | | | | | 1 Wait | 2 Waits | 3 Waits | 1 Wait | 2 Waits | 3 Waits |
| j | 1 | 2 | 2 | 3 | 1 | 2 | 3 | 4 | 3 | 3 | 4 |
| k | 1 | 2 | 2 | 3 | 2 | 2 | 3 | 4 | 3 | 3 | 4 |

NOTES:

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

12.3 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:

- (1) Reload the forward address pointer with the SAR_i register value when the DSD bit in the DMiCON register is 1 (forward) or the DAR_i register value when the DAD bit in the DMiCON register is 1 (forward).
- (2) 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.

Step 1: Write 1 to the DMAE bit and DMAS bit in the DMiCON register simultaneously.

Step 2: Make sure that the DMA_i is in an initial state as described above (1) and (2) in a program.

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

12.4 DMA Request

The DMAC can generate a DMA request as triggered by the request source that is selected with bits DMS, and DSEL3 to DSEL0 in the DMiSL register ($i = 0, 1$) on either channel.

Table 12.4 lists the Timing at which 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 in a program (it can only be set to 0).

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

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

Table 12.4 Timing at which DMAS Bit Changes State

| DMA Source | DMAS Bit in 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 in a program |
| Peripheral function | When the interrupt control register for the peripheral function that is selected by bits DSEL3 to DSEL0, and DMS in the DMiSL register has its IR bit set to 1. | |

$i = 0, 1$

12.5 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 BCLK), 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 12.6 shows an example of DMA Transfer by External Sources.

In Figure 12.6, 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 12.6, occurs more than one time, the DMAS bit is set to 0 as soon as getting the bus arbitration.

Refer to 7.2.7 **HOLD Signal** for details about bus arbitration between the CPU and DMA.

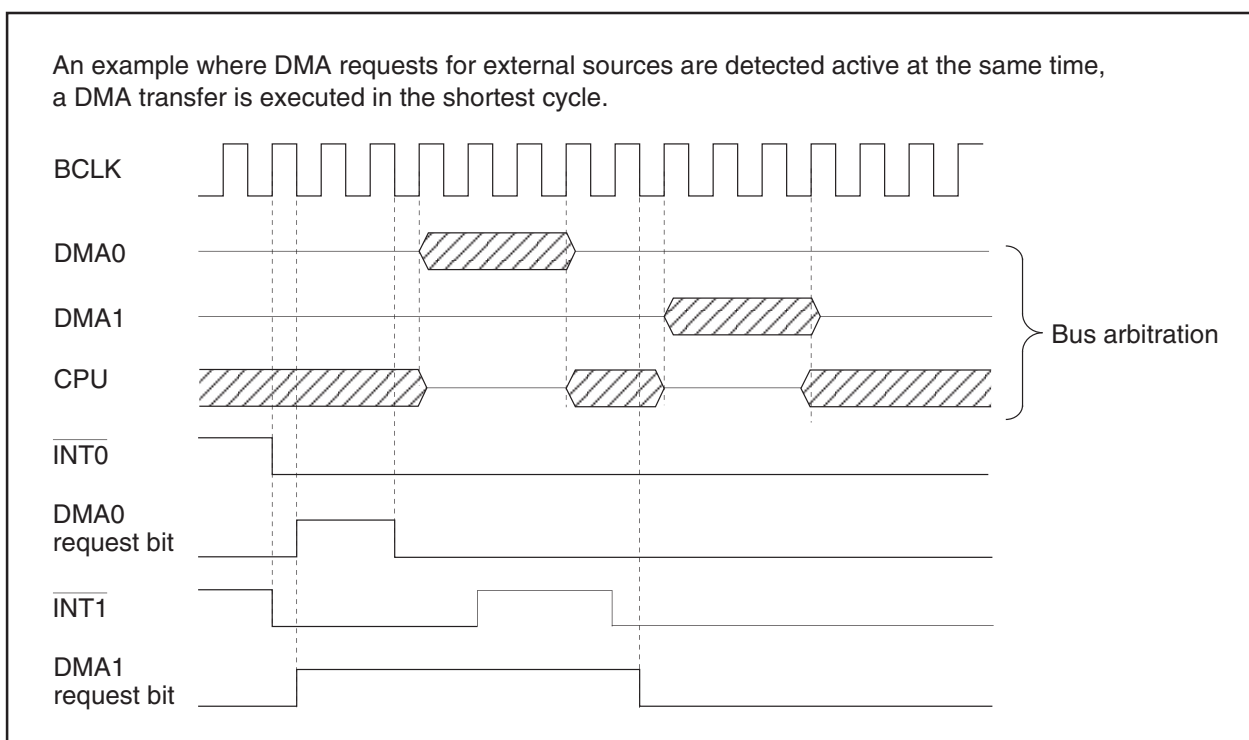


Figure 12.6 DMA Transfer by External Sources

13. Timers

Eleven 16-bit timers, each capable of operating independently of the others, can be classified by function as either timer A (five) and timer B (six). The count source for each timer acts as a clock, to control such timer operations as counting, reloading, etc.

Figures 13.1 and 13.2 show the Timer A and Timer B Configurations.

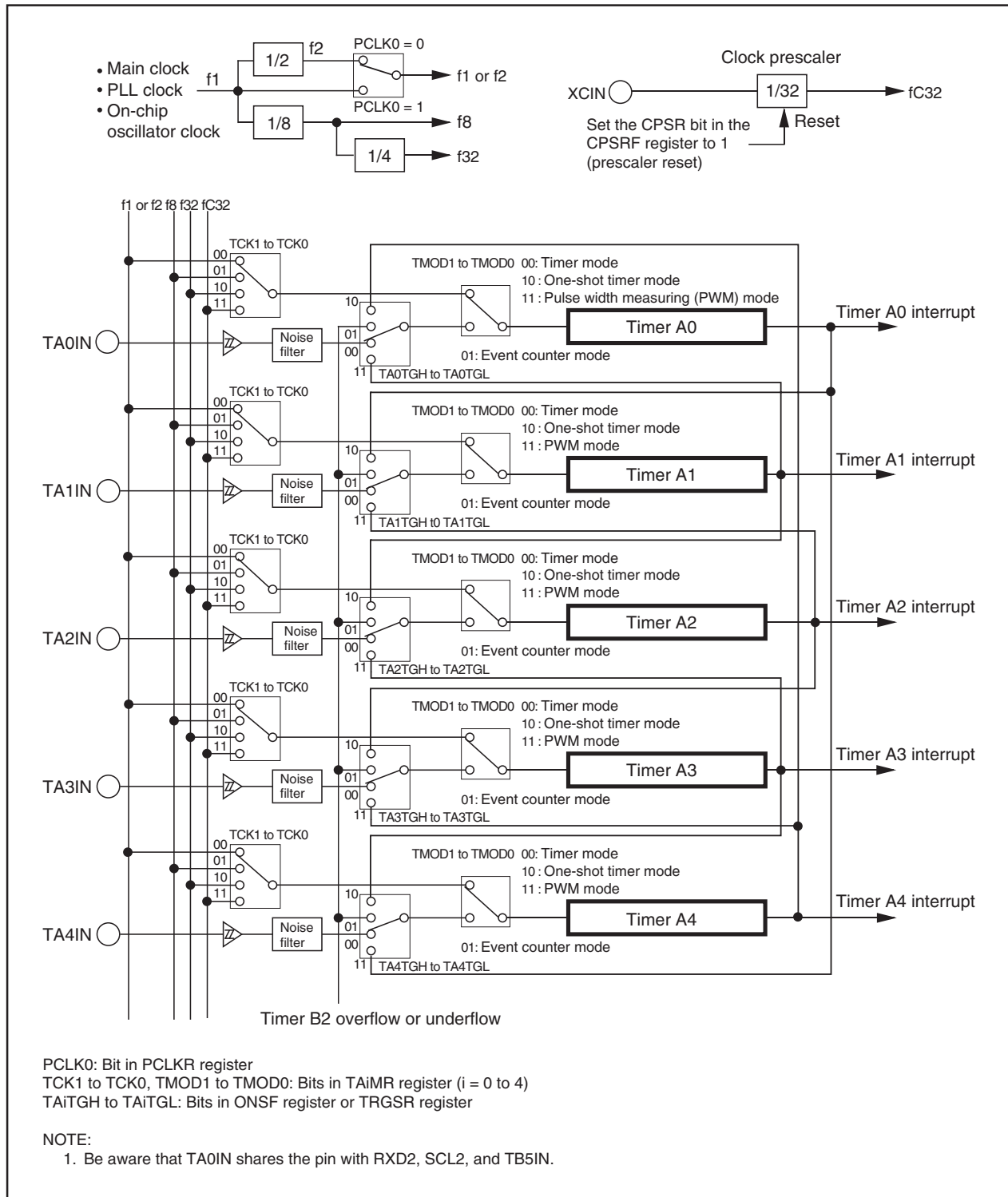


Figure 13.1 Timer A Configuration

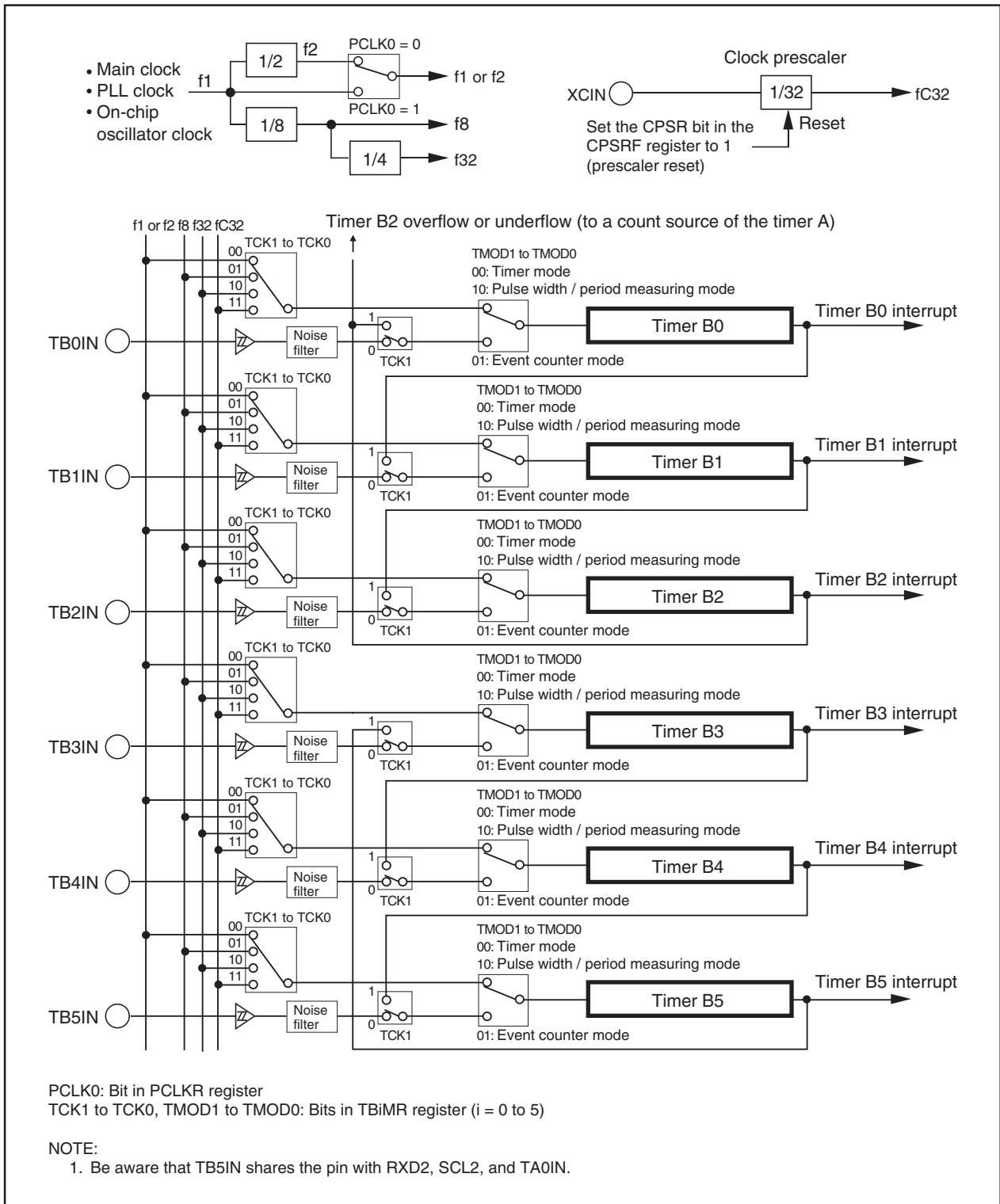


Figure 13.2 Timer B Configuration

13.1 Timer A

Figure 13.3 shows the Timer A Block Diagram. Figures 13.4 to 13.6 show the timer A-related registers. 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_iMR 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 0000h.
- Pulse width modulation (PWM) mode: The timer outputs pulses in a given width successively.

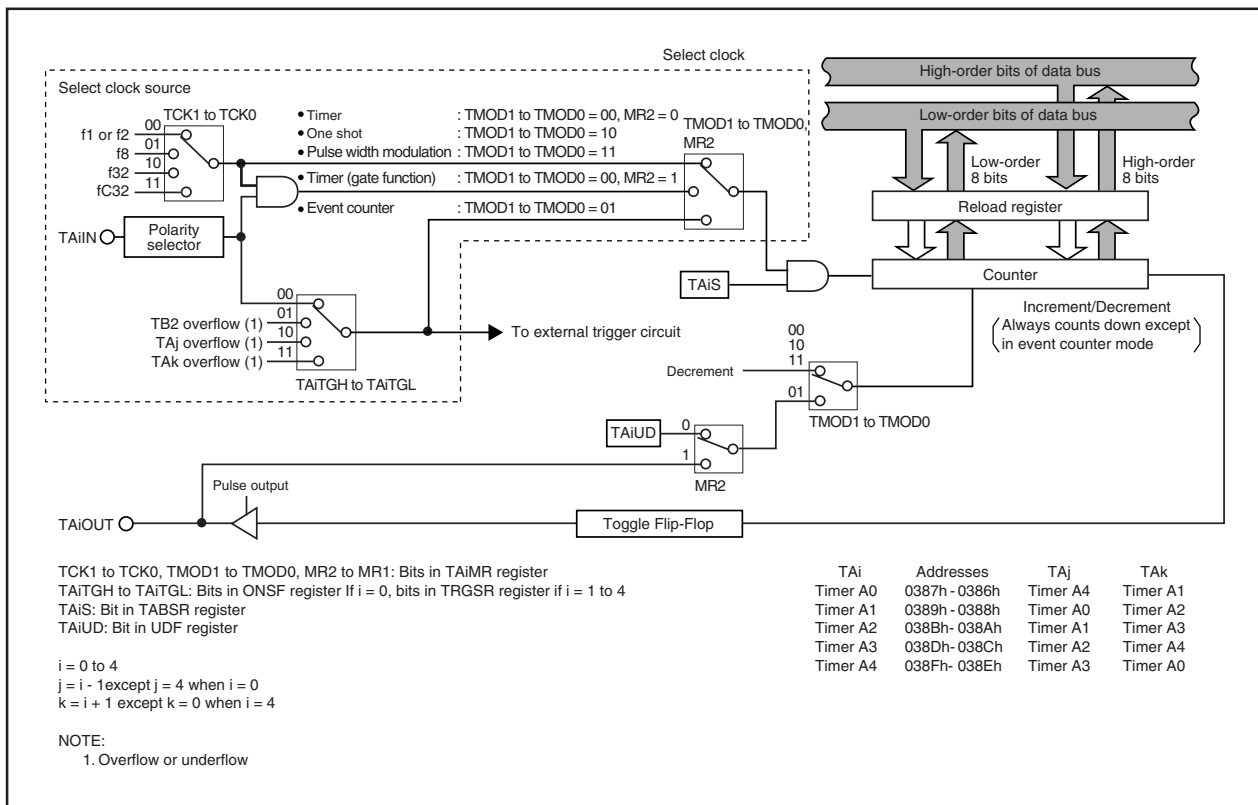


Figure 13.3 Timer A Block Diagram

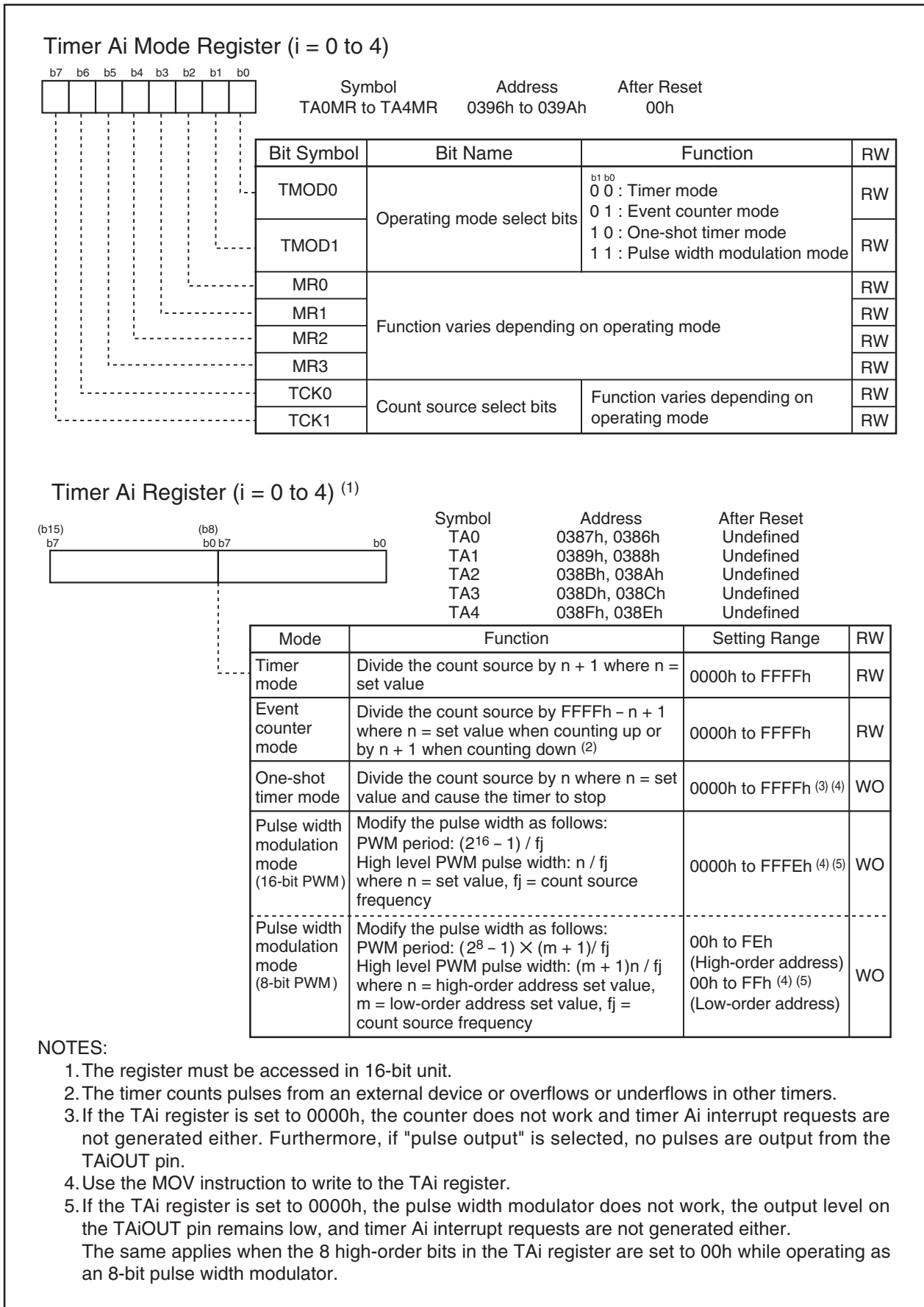


Figure 13.4 Registers TA0MR to TA4MR, and TA0 to TA4

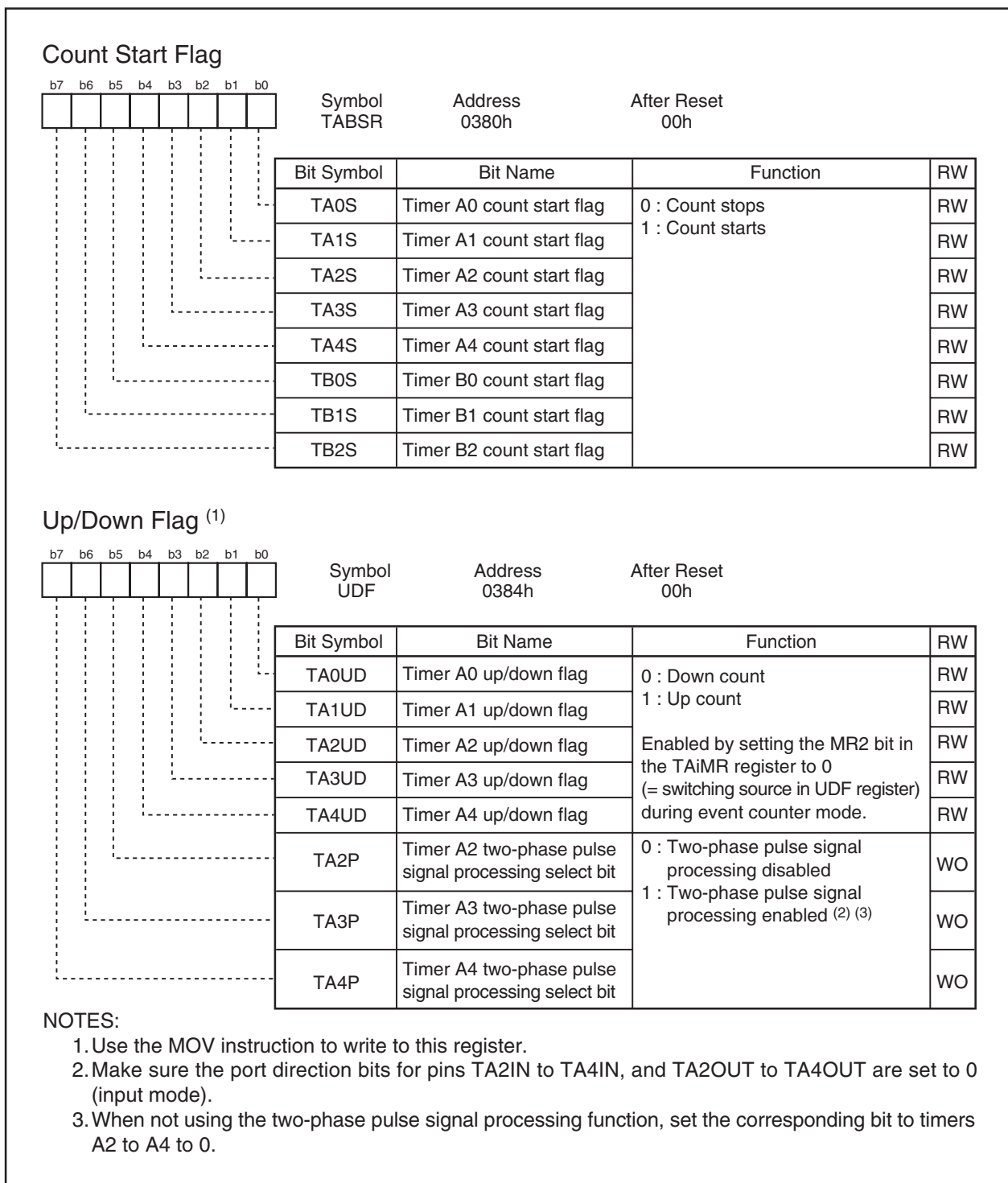


Figure 13.5 Registers TABSR and UDF

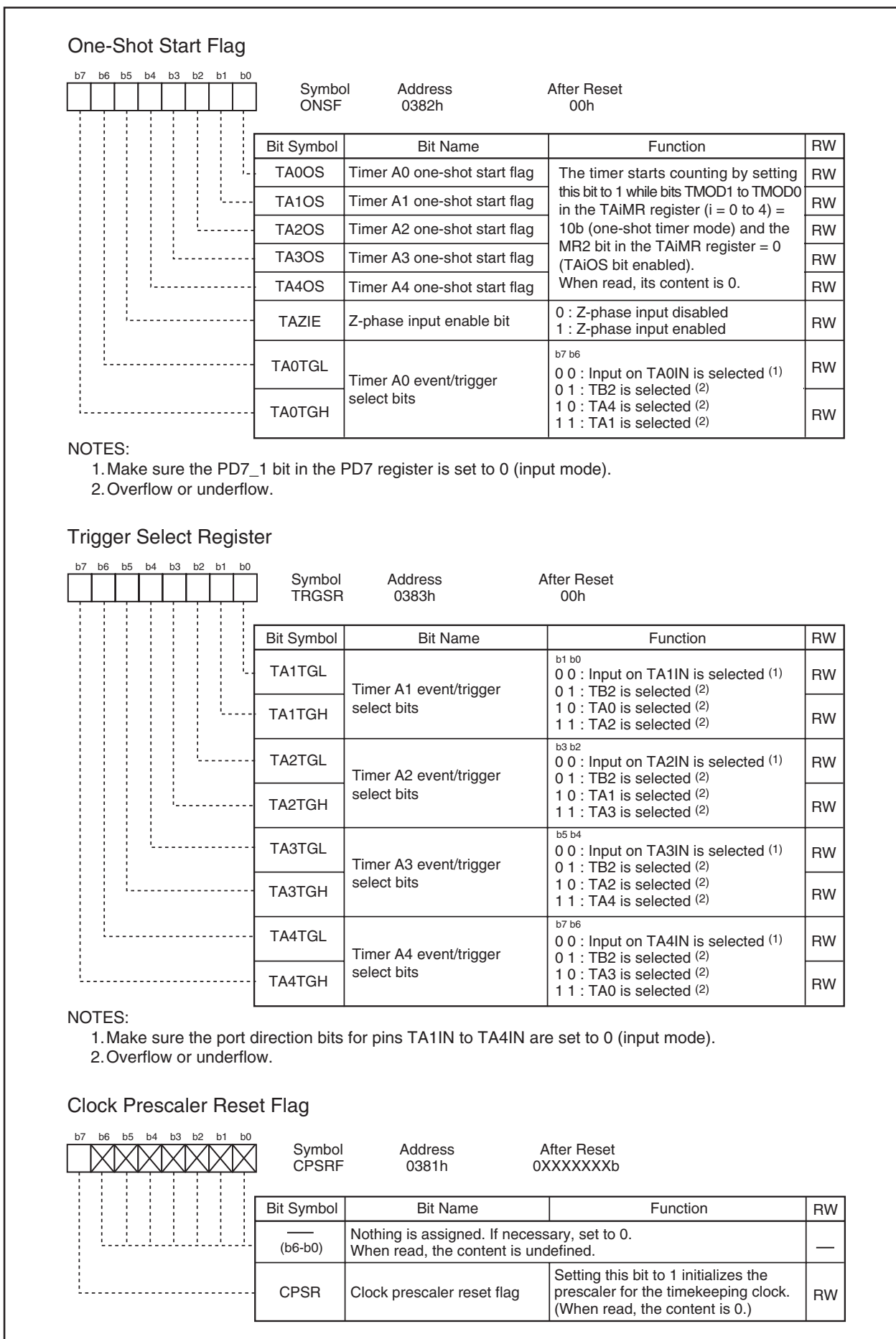


Figure 13.6 Registers ONSF, TRGSR, and CPSRF

13.1.1 Timer Mode

In timer mode, the timer counts a count source generated internally.

Table 13.1 lists the Timer Mode Specifications. Figure 13.7 shows Registers TA0MR to TA4MR in Timer Mode.

Table 13.1 Timer Mode Specifications

| Item | Specification |
|-------------------------------------|--|
| Count source | f1, f2, f8, f32, fC32 |
| Count operation | <ul style="list-style-type: none"> Down-count When the timer underflows, it reloads the reload register contents and continues counting |
| Divide ratio | 1/(n+1) n: set value of the TAI register 0000h to FFFFh |
| Count start condition | Set the TAI _S bit in the TABSR register to 1 (count starts) |
| Count stop condition | Set the TAI _S bit to 0 (count stops) |
| Interrupt request generation timing | Timer underflow |
| TAiIN pin function | I/O port or gate input |
| TAiOUT pin function | I/O port or pulse output |
| Read from timer | Count value can be read by reading the 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 the TAI register is written to both reload register and counter When counting (after 1st count source input) Value written to the 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 TAIIN pin Pulse output function Whenever the timer underflows, the output polarity of TAIOUT pin is inverted. When TAI_S bit is set to 0 (count stops), the pin outputs a low. |

i = 0 to 4

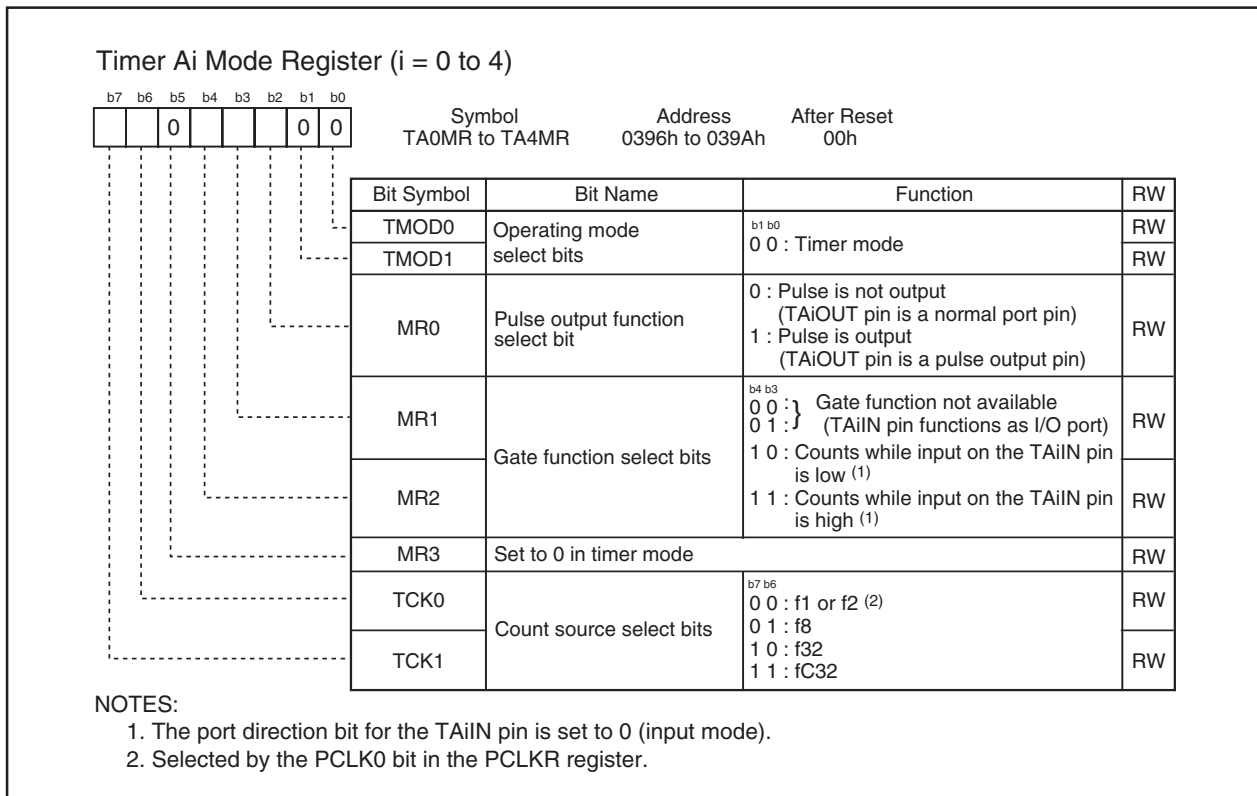


Figure 13.7 Registers TA0MR to TA4MR in Timer Mode

13.1.2 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 13.2 lists the Event Counter Mode Specifications (when not using two-phase pulse signal processing). Figure 13.8 shows TAIiMR Register in Event Counter Mode (when not using two-phase pulse signal processing). Table 13.3 lists the Event Counter Mode Specifications (when using two-phase pulse signal processing with timers A2, A3, and A4). Figure 13.9 shows Registers TA2MR to TA4MR in Event Counter Mode (when using two-phase pulse signal processing with timers A2, A3, and A4).

Table 13.2 Event Counter Mode Specifications (when not using two-phase pulse signal processing)

| Item | Specification |
|-------------------------------------|--|
| Count source | <ul style="list-style-type: none"> External signals input to TAIiN pin (effective edge can be selected in program) Timer B2 overflows or underflows, Timer Aj overflows or underflows, Timer Ak overflows or underflows |
| Count operation | <ul style="list-style-type: none"> Up-count or down-count 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/ (FFFFh - n + 1) for up-count 1/ (n + 1) for down-count n : set value of the TAI register 0000h to FFFFh |
| Count start condition | Set the TAIiS bit in the TABSR register to 1 (count starts) |
| Count stop condition | Set the TAIiS bit to 0 (count stops) |
| Interrupt request generation timing | Timer overflow or underflow |
| TAiIN pin function | I/O port or count source input |
| TAiOUT pin function | I/O port, pulse output, or up/down-count select input |
| Read from timer | Count value can be read by reading the 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 the TAI register is written to both reload register and counter When counting (after 1st count source input) Value written to the 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 TAIOUT pin is inverted. When TAIiS bit is set to 0 (count stops), the pin outputs a low. |

i = 0 to 4

j = i - 1, except j = 4 if i = 0

k = i + 1, except k = 0 if i = 4

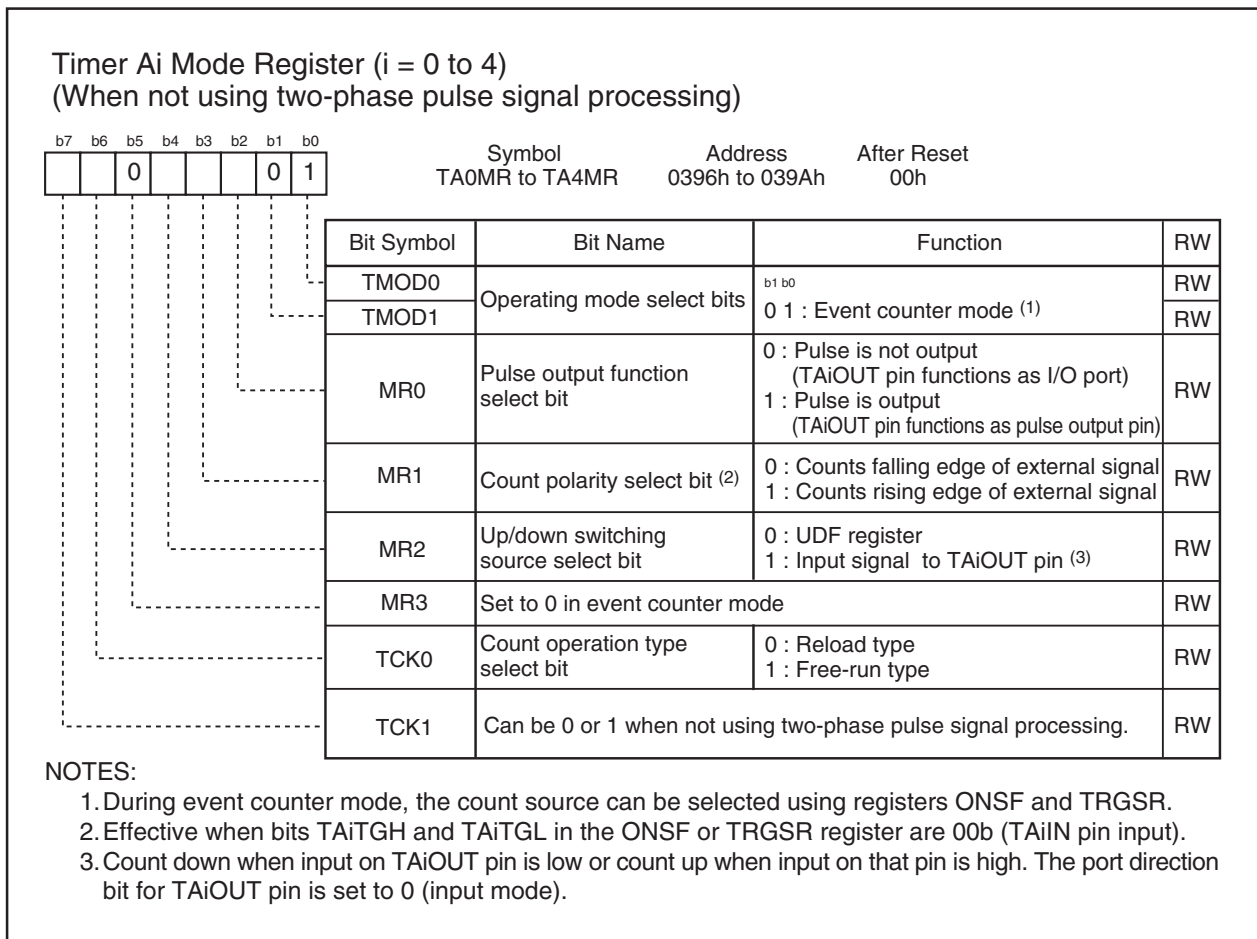


Figure 13.8 Registers TA0MR to TA4MR in Event Counter Mode (when not using two-phase pulse signal processing)

Table 13.3 Event Counter Mode Specifications (when using two-phase pulse signal processing with timers A2, A3, and A4)

| Item | Specification |
|-------------------------------------|--|
| Count source | • Two-phase pulse signals input to TAIiN or TAIiOUT pins |
| Count operation | • Up-count 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/ (FFFFh - n + 1) for up-count 1/ (n + 1) for down-count n : set value of the TAI register 0000h to FFFFh |
| Count start condition | Set the TAIiS bit in the TABSR register to 1 (count starts) |
| Count stop condition | Set the TAIiS bit to 0 (count stops) |
| 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 the TAI register |
| Write to timer | • 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 ⁽¹⁾ | <ul style="list-style-type: none"> • Normal processing operation (timers A2 and A3) The timer counts up rising edges or counts down falling edges on TAJiN pin when input signals on TAJiOUT pin is "H". <ul style="list-style-type: none"> • Multiply-by-4 processing operation (timers A3 and A4) If the phase relationship is such that TAKiN pin goes "H" when the input signal on TAKiOUT pin is "H", the timer counts up rising and falling edges on pins TAKiOUT and TAKiN. If the phase relationship is such that TAKiN pin goes "L" when the input signal on TAKiOUT pin is "H", the timer counts down rising and falling edges on pins TAKiOUT and TAKiN. <ul style="list-style-type: none"> • Counter initialization by Z-phase input (timer A3) The timer count value is initialized to 0 by Z-phase input. |

i = 2 to 4

j = 2, 3

k = 3, 4

NOTE:

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

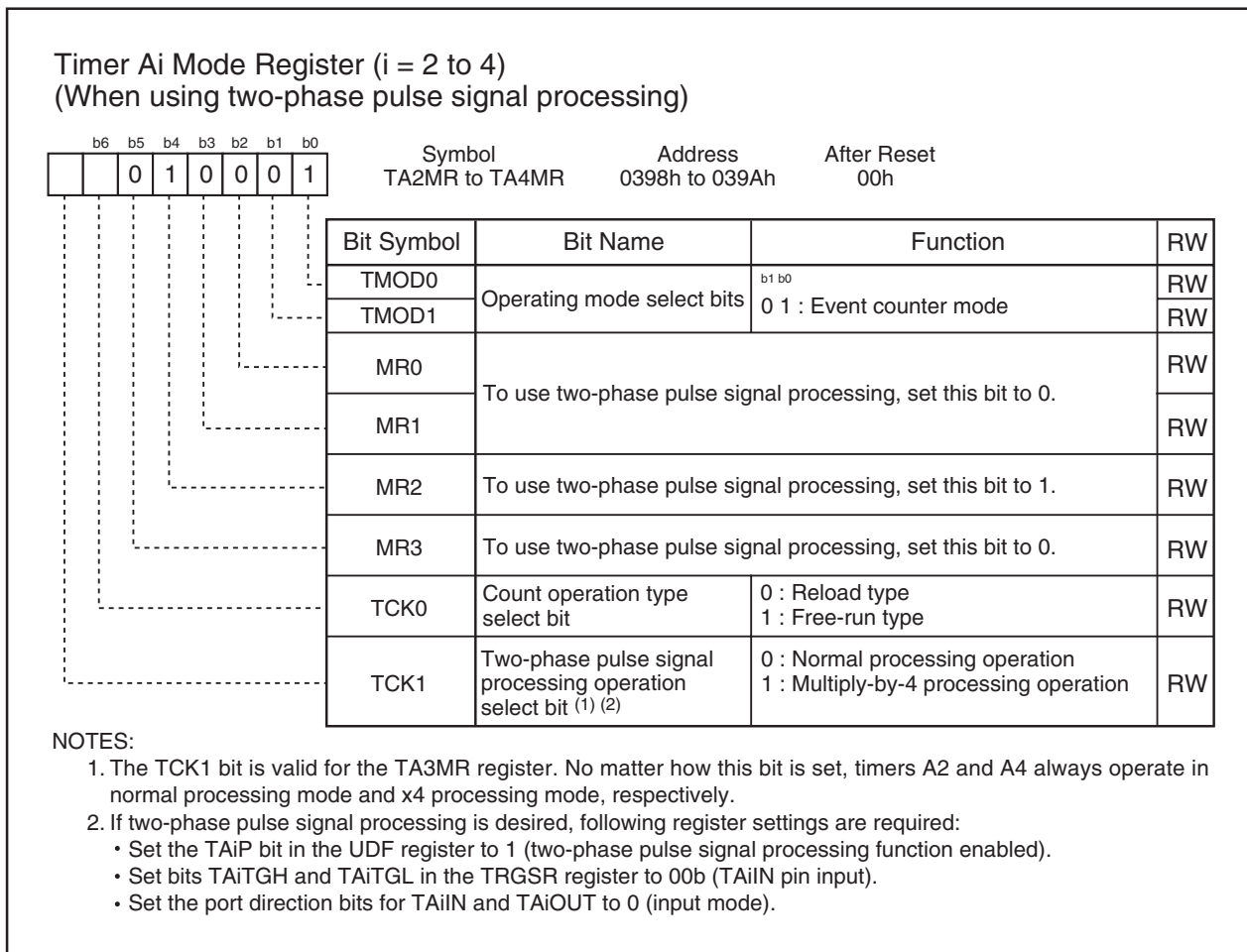


Figure 13.9 Registers TA2MR to TA4MR in Event Counter Mode (when using two-phase pulse signal processing with timers A2, A3, and A4)

13.1.2.1 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 ZP pin.

Counter initialization by Z-phase input is enabled by writing 0000h to the TA3 register and setting the TAZIE bit in the ONSF register to 1 (Z-phase input enabled).

Counter initialization is accomplished by detecting Z-phase input edge. The active edge can be selected 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 13.10 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.

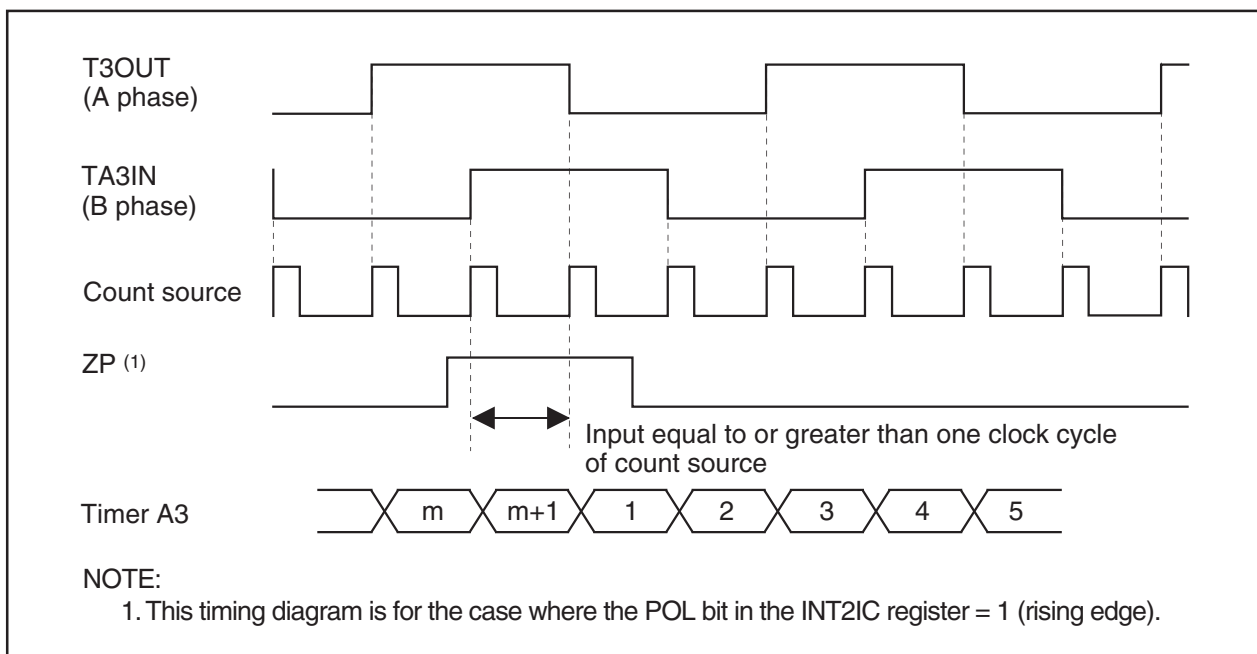


Figure 13.10 Two-phase Pulse (A Phase and B Phase) and Z Phase

13.1.3 One-shot Timer Mode

In one-shot timer mode, the timer is activated only once by one trigger. When the trigger occurs, the timer starts up and continues operating for a given period. Table 13.4 lists the One-shot Timer Mode Specifications. Figure 13.11 shows Registers TA0MR to TA4MR in One-shot Timer Mode.

Table 13.4 One-shot Timer Mode Specifications

| Item | Specification |
|-------------------------------------|---|
| Count source | f1, f2, f8, f32, fC32 |
| Count operation | <ul style="list-style-type: none"> • Down-count • When the counter reaches 0000h, 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 the TAI register 0000h to FFFFh However, the counter does not work if the divide-by-n value is set to 0000h. |
| Count start condition | <p>The TAI_iS bit in the TABSR register = 1 (count starts) and one of the following triggers occurs.</p> <ul style="list-style-type: none"> • External trigger input from the TAI_iIN pin • Timer B2 overflow or underflow, Timer A_j overflow or underflow, Timer A_k overflow or underflow • The TAI_iOS 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 0000h • TAI_iS bit is set to 0 (count stops) |
| Interrupt request generation timing | When the counter reaches 0000h |
| TAI _i IN pin function | I/O port or trigger input |
| TAI _i OUT pin function | I/O port or pulse output |
| Read from timer | An undefined value is read by reading the 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 the TAI register is written to both reload register and counter • When counting (after 1st count source input) Value written to the 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 <p>The timer outputs a low when not counting and a high when counting.</p> |

i = 0 to 4

j = i - 1, except j = 4 if i = 0

k = i + 1, except k = 0 if i = 4

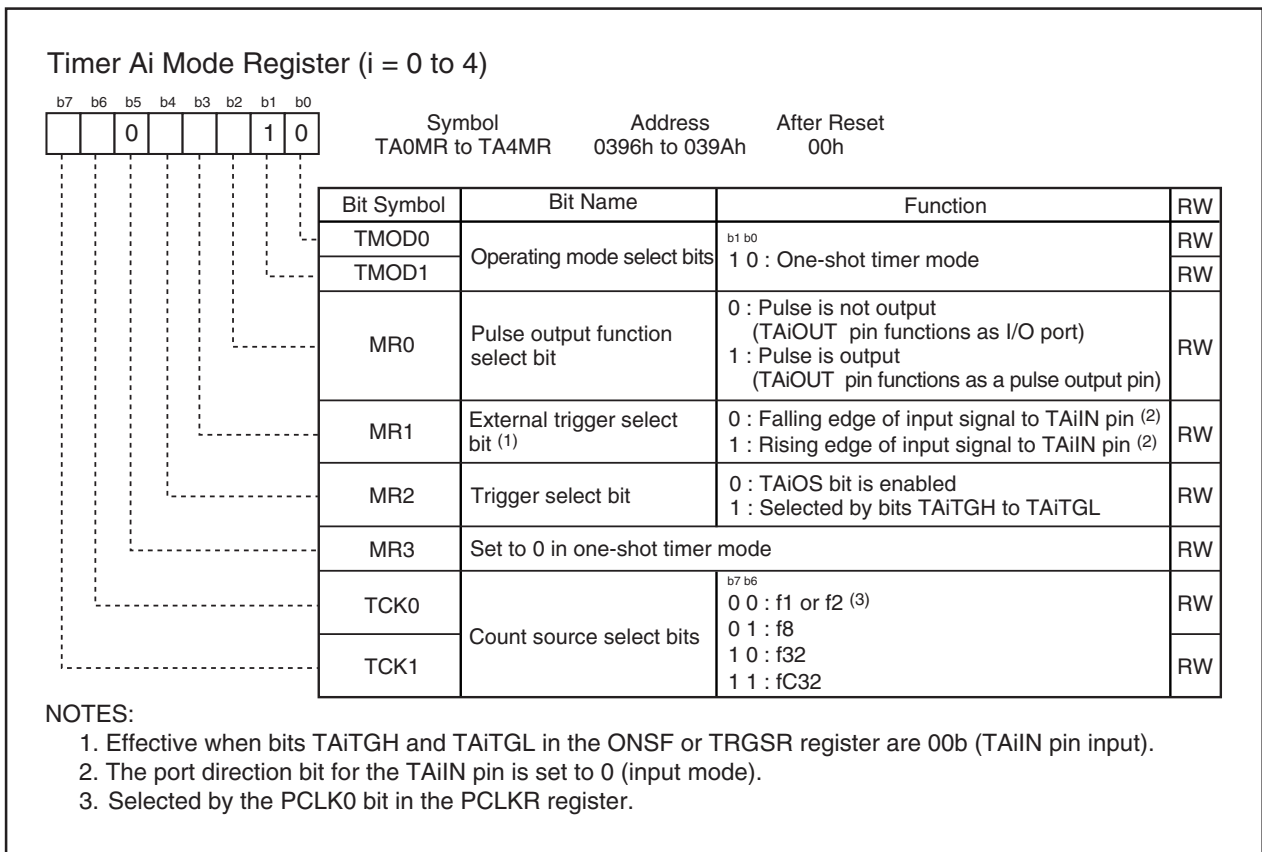


Figure 13.11 Registers TA0MR to TA4MR in One-shot Timer Mode

13.1.4 Pulse Width Modulation (PWM) Mode

In PWM mode, the timer outputs pulses of a given width in succession. The counter functions as either 16-bit pulse width modulator or 8-bit pulse width modulator.

Table 13.5 lists the PWM Mode Specifications. Figure 13.12 shows Registers TA0MR to TA4MR in PWM Mode. Figures 13.13 and 13.14 show an Example of 16-bit Pulse Width Modulator Operation and 8-bit Pulse Width Modulator Operation.

Table 13.5 PWM Mode Specifications

| Item | Specification |
|-------------------------------------|--|
| Count source | f1, f2, f8, f32, fC32 |
| Count operation | <ul style="list-style-type: none"> • Down-count (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 the TAI register • 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 the TAI register high-order address • Cycle time $(2^8-1) \times (m+1) / f_j$ m : set value of the TAI register low-order address |
| Count start condition | <ul style="list-style-type: none"> • The TAI_S bit in the TABSR register is set to 1 (count starts) • 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 overflow or underflow, Timer A_k overflow or underflow |
| Count stop condition | The TAI _S bit is set to 0 (count stops) |
| Interrupt request generation timing | On the falling edge of the PWM pulse |
| 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 the 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 the TAI register is written to both reload register and counter • When counting (after 1st count source input) Value written to the TAI register is written to only reload register (Transferred to counter when reloaded next) |

$i = 0$ to 4

$j = i - 1$, except $j = 4$ if $i = 0$

$k = i + 1$, except $k = 0$ if $i = 4$

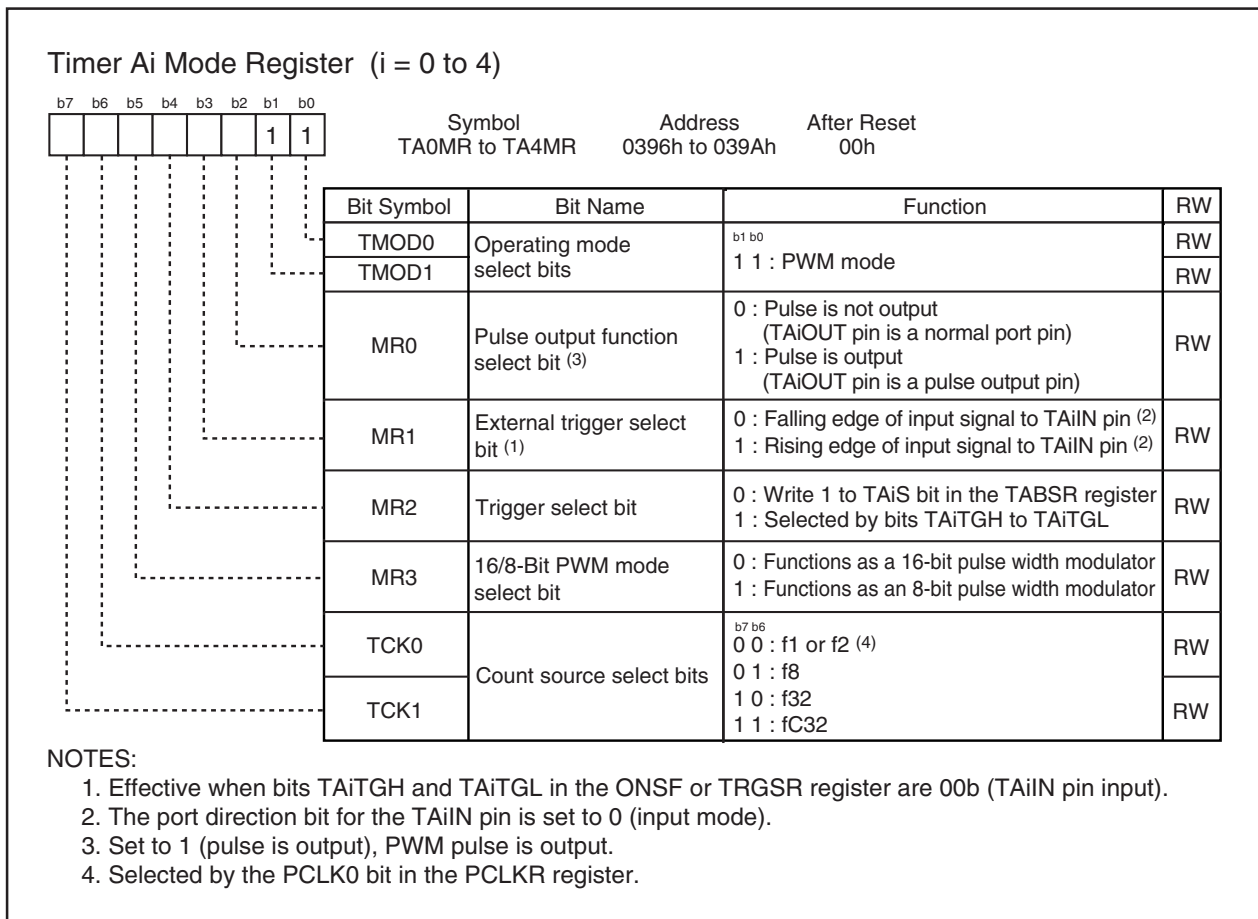


Figure 13.12 Registers TA0MR to TA4MR in PWM Mode

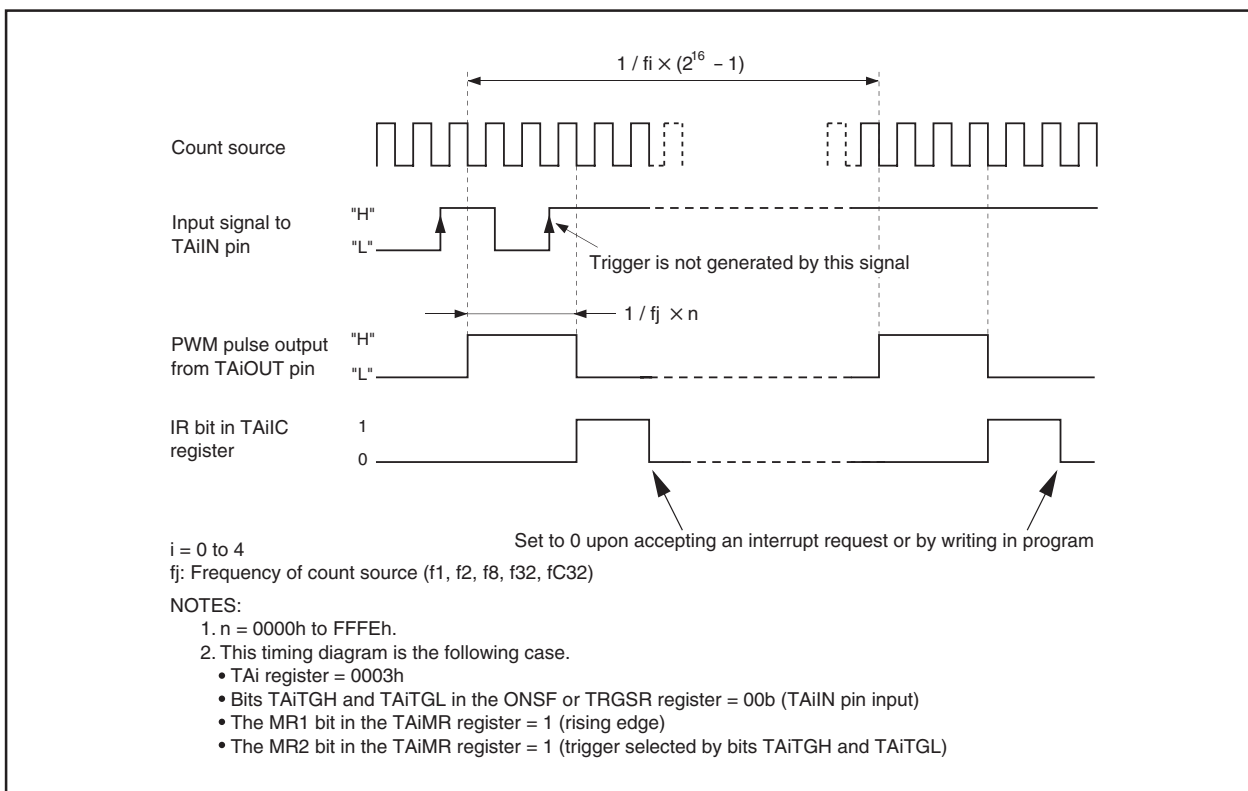


Figure 13.13 Example of 16-bit Pulse Width Modulator Operation

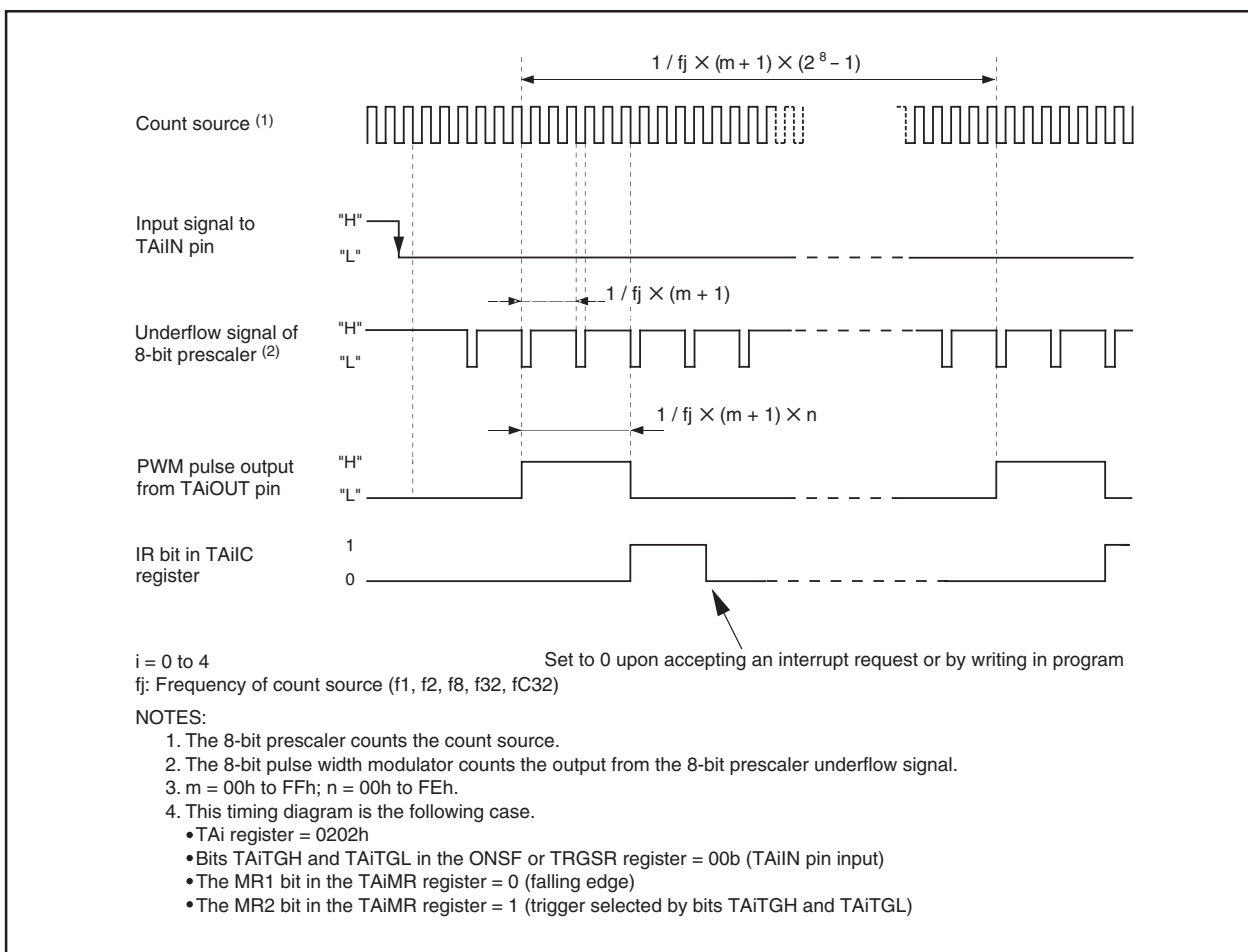


Figure 13.14 Example of 8-bit Pulse Width Modulator Operation

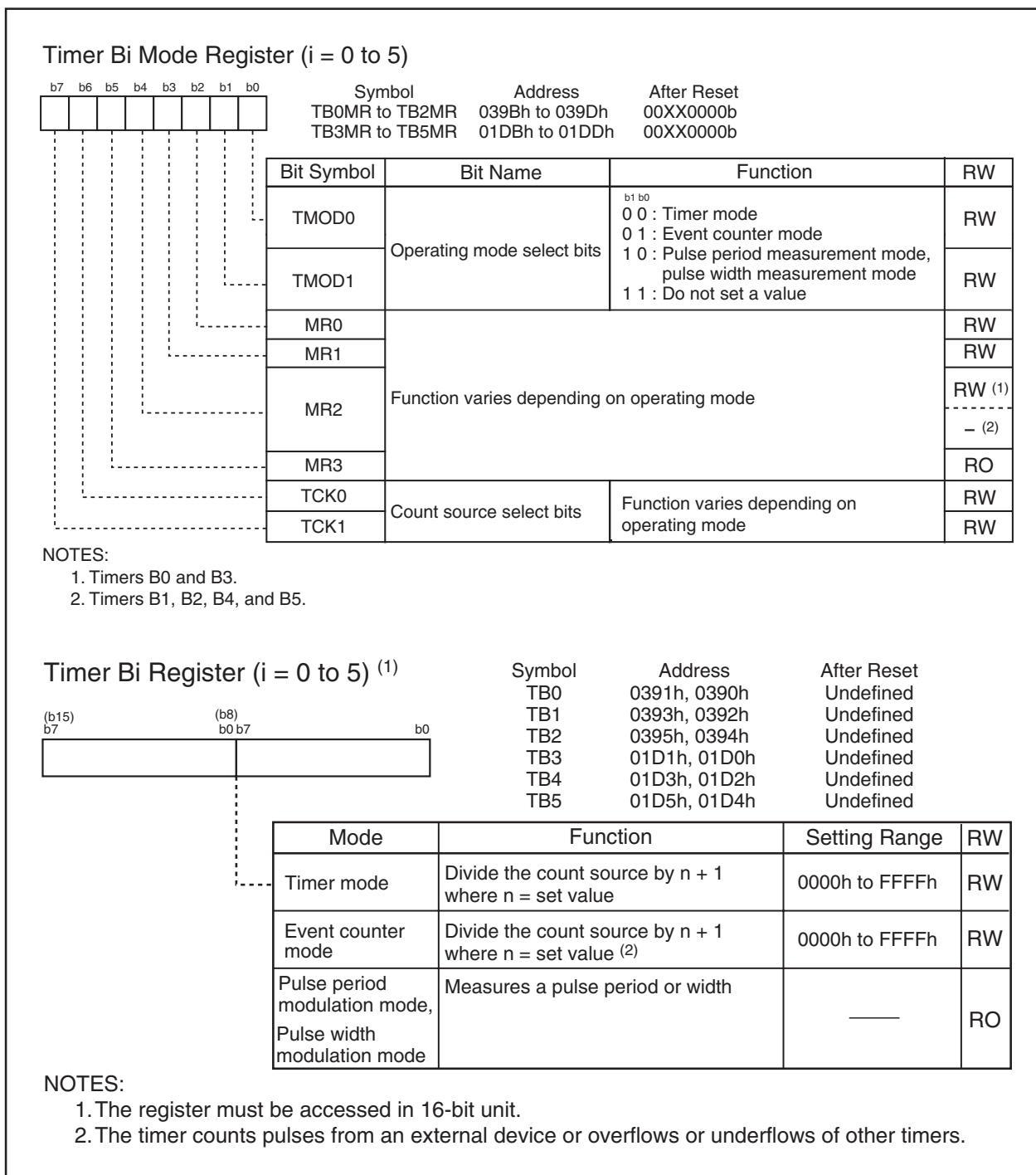


Figure 13.16 Registers TB0MR to TB5MR, and TB0 to TB5

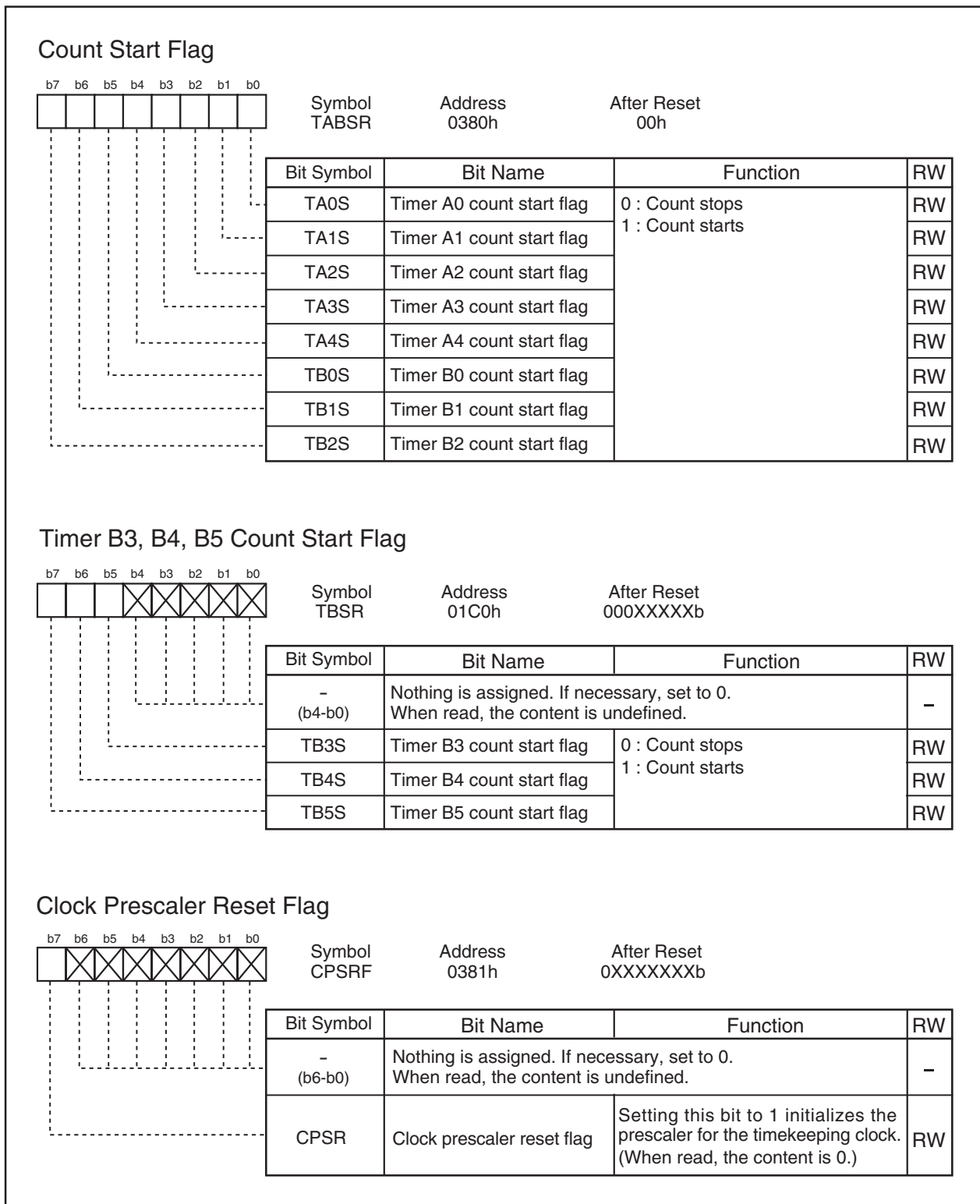


Figure 13.17 Registers TABSR, TBSR, and CPSRF

13.2.1 Timer Mode

In timer mode, the timer counts a count source generated internally.

Table 13.6 lists the Timer Mode Specifications. Figure 13.18 shows Registers TB0MR to TB5MR in Timer Mode.

Table 13.6 Timer Mode Specifications

| Item | Specification |
|-------------------------------------|--|
| Count source | f1, f2, f8, f32, fC32 |
| Count operation | <ul style="list-style-type: none"> Down-count When the timer underflows, it reloads the reload register contents and continues counting |
| Divide ratio | 1/(n+1) n: set value of the TBi register 0000h to FFFFh |
| Count start condition | Set the TBiS bit ⁽¹⁾ to 1 (count starts) |
| Count stop condition | Set the TBiS bit to 0 (count stops) |
| Interrupt request generation timing | Timer underflow |
| TBiIN pin function | I/O port |
| Read from timer | Count value can be read by reading the 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 the TBi register is written to both reload register and counter When counting (after 1st count source input) Value written to the TBi register is written to only reload register (Transferred to counter when reloaded next) |

i = 0 to 5

NOTE:

- Bits TB0S to TB2S are assigned to bits 5 to 7 in the TABSR register, and bits TB3S to TB5S are assigned to bits 5 to 7 in the TBSR register.

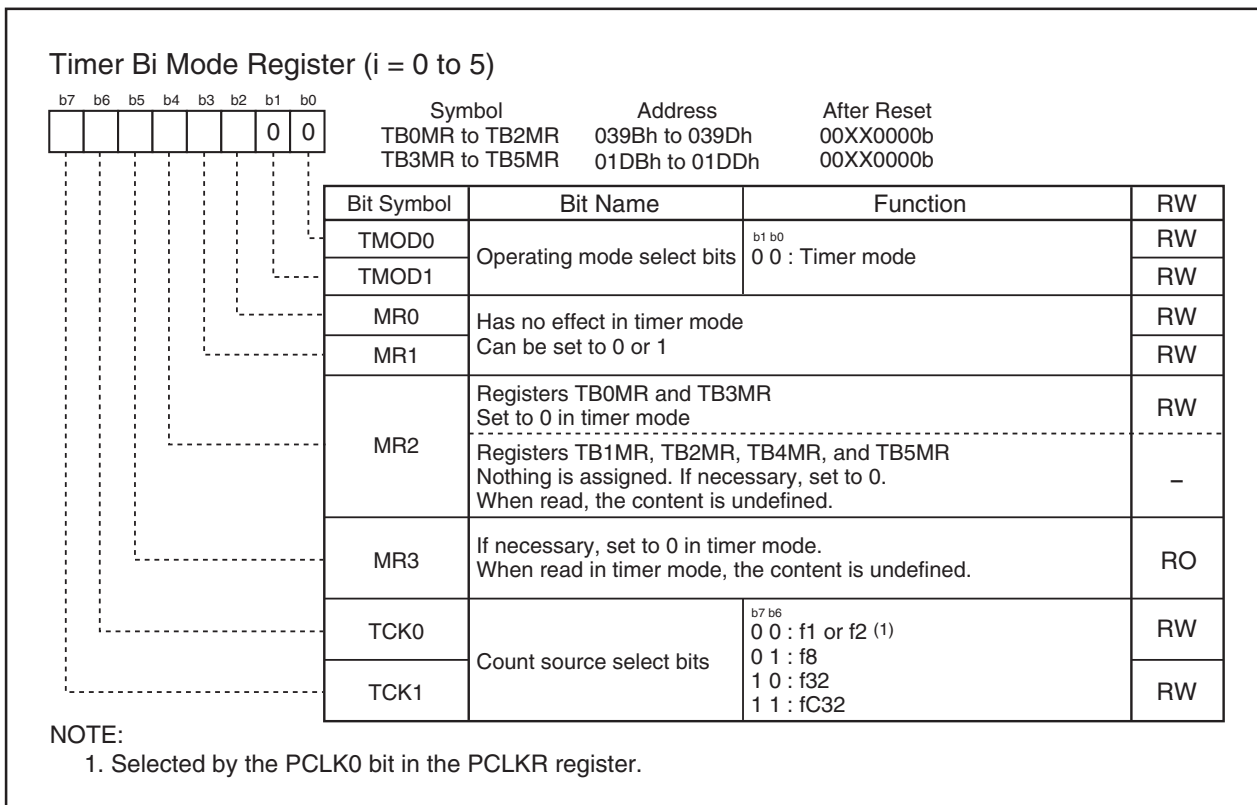


Figure 13.18 Registers TB0MR to TB5MR in Timer Mode

13.2.2 Event Counter Mode

In event counter mode, the timer counts pulses from an external device or overflows and underflows of other timers. Table 13.7 lists the Event Counter Mode Specifications. Figure 13.19 shows Registers TB0MR to TB5MR in Event Counter Mode.

Table 13.7 Event Counter Mode Specifications

| Item | Specification |
|-------------------------------------|--|
| Count source | <ul style="list-style-type: none"> External signals input to TBIiN pin (effective edge can be selected in program) Timer Bj overflow or underflow |
| Count operation | <ul style="list-style-type: none"> Down-count When the timer underflows, it reloads the reload register contents and continues counting |
| Divide ratio | 1/(n+1) n: set value of the TBI register 0000h to FFFFh |
| Count start condition | Set TBI <i>S</i> bit ⁽¹⁾ to 1 (count starts) |
| Count stop condition | Set TBI <i>S</i> bit to 0 (count stops) |
| Interrupt request generation timing | Timer underflow |
| TBIiN pin function | Count source input |
| Read from timer | Count value can be read by reading the 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 the TBI register is written to both reload register and counter When counting (after 1st count source input) Value written to the TBI register is written to only reload register (Transferred to counter when reloaded next) |

i = 0 to 5

j = i - 1, except j = 2 if i = 0, j = 5 if i = 3

NOTE:

- Bits TB0S to TB2S are assigned to bits 5 to 7 in the TABSR register, and bits TB3S to TB5S are assigned to bits 5 to 7 in the TBSR register.

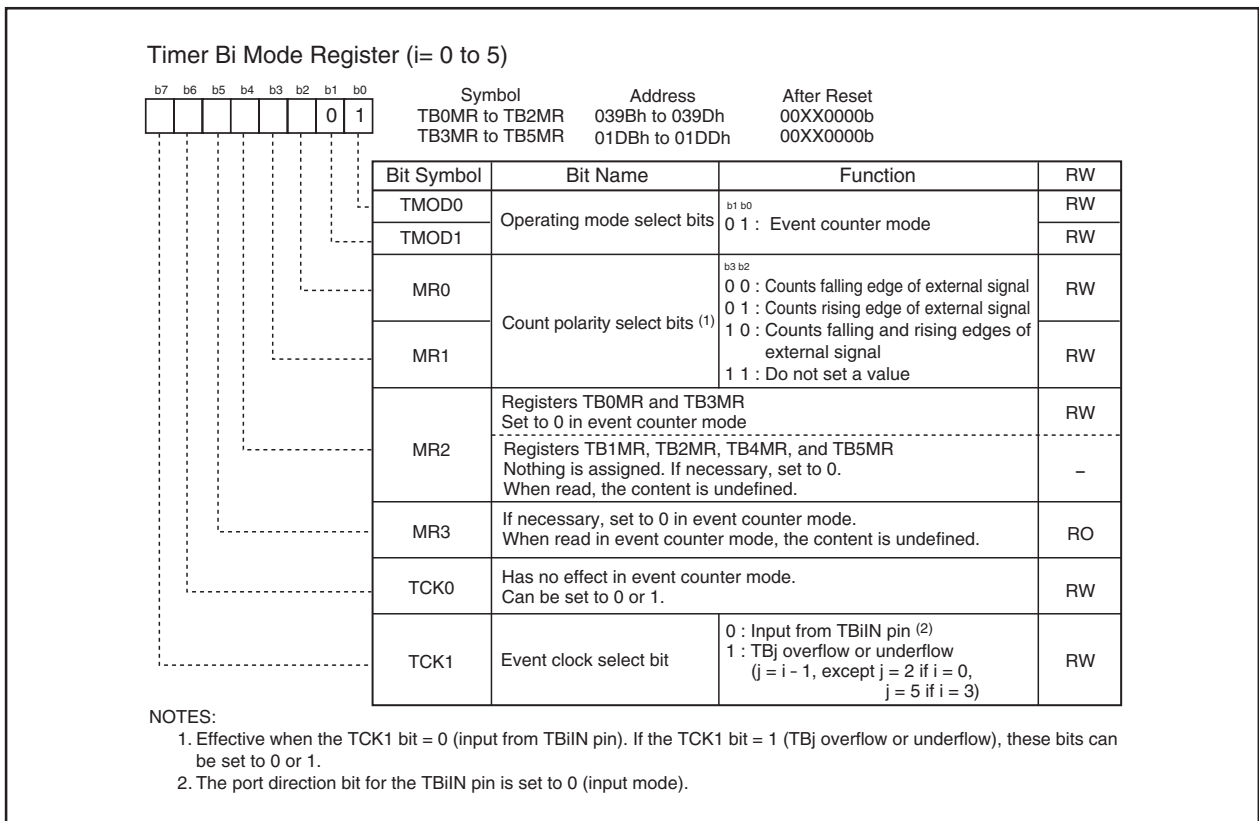


Figure 13.19 Registers TB0MR to TB5MR in Event Counter Mode

13.2.3 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. Table 13.8 lists the Pulse Period and Pulse Width Measurement Mode Specifications. Figure 13.20 shows Registers TB0MR to TB5MR in Pulse Period and Pulse Width Measurement mode. Figure 13.21 shows the Operation Timing when Measuring Pulse Period. Figure 13.22 shows the Operation Timing when Measuring Pulse Width.

Table 13.8 Pulse Period and Pulse Width Measurement Mode Specifications

| Item | Specification |
|-------------------------------------|--|
| Count source | f1, f2, f8, f32, fC32 |
| Count operation | <ul style="list-style-type: none"> • Up-count • Counter value is transferred to reload register at an effective edge of measurement pulse. The counter value is set to 0000h to continue counting. |
| Count start condition | Set the TBiS bit ⁽¹⁾ to 1 (count starts) |
| Count stop condition | Set the TBiS bit to 0 (count stops) |
| Interrupt request generation timing | <ul style="list-style-type: none"> • When an effective edge of measurement pulse is input ⁽²⁾ • Timer overflow. If an overflow occurs, the MR3 bit in the TBiMR register is set to 1 (overflow) simultaneously. The MR3 bit is set to 0 (no overflow) by writing to the TBiMR register at the next count timing or later after the MR3 bit was set to 1. At this time, make sure the TBiS bit is set to 1 (count starts). |
| TBiIN pin function | Measurement pulse input |
| Read from timer | Contents of the reload register (measurement result) can be read by reading TBi register ⁽³⁾ |
| Write to timer | Value written to the TBi register is written to neither reload register nor counter |

i = 0 to 5

NOTES:

1. Bits TB0S to TB2S are assigned to bits 5 to 7 in the TABSR register, and bits TB3S to TB5S are assigned to bits 5 to 7 in the TBSR register.
2. Interrupt request is not generated when the first effective edge is input after the timer started counting.
3. Value read from the TBi register is undefined until the second valid edge is input after the timer starts counting.

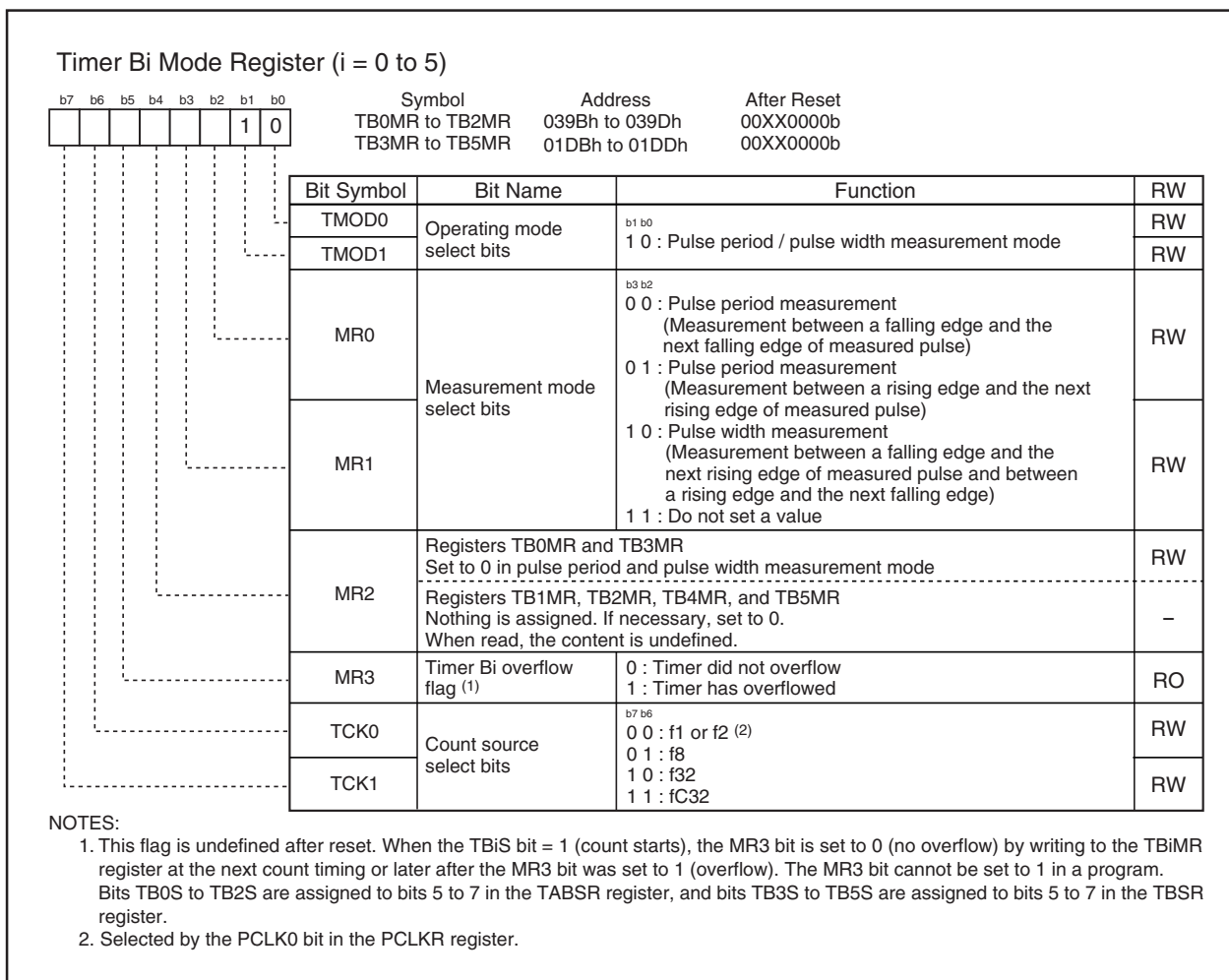


Figure 13.20 Registers TB0MR to TB5MR in Pulse Period and Pulse Width Measurement Mode

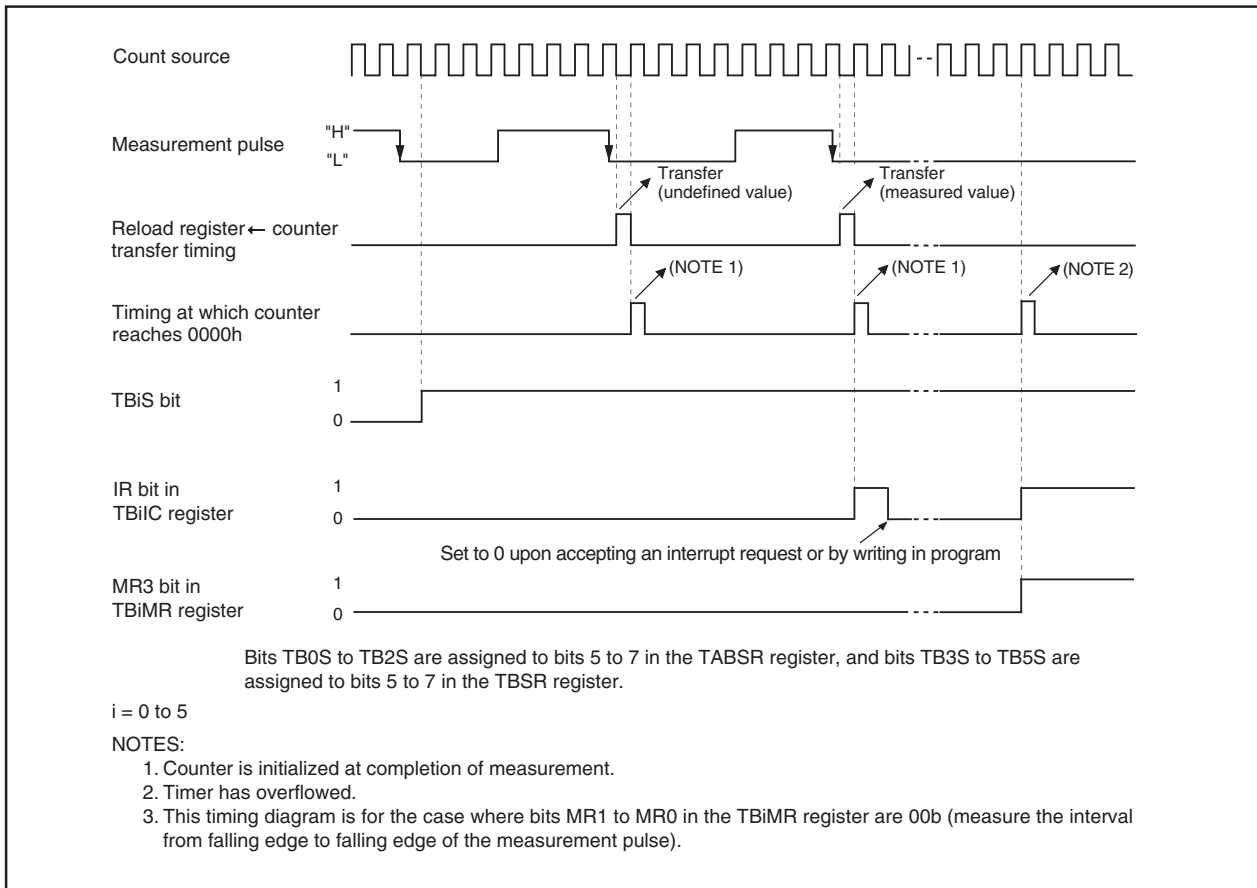


Figure 13.21 Operation Timing When Measuring Pulse Period

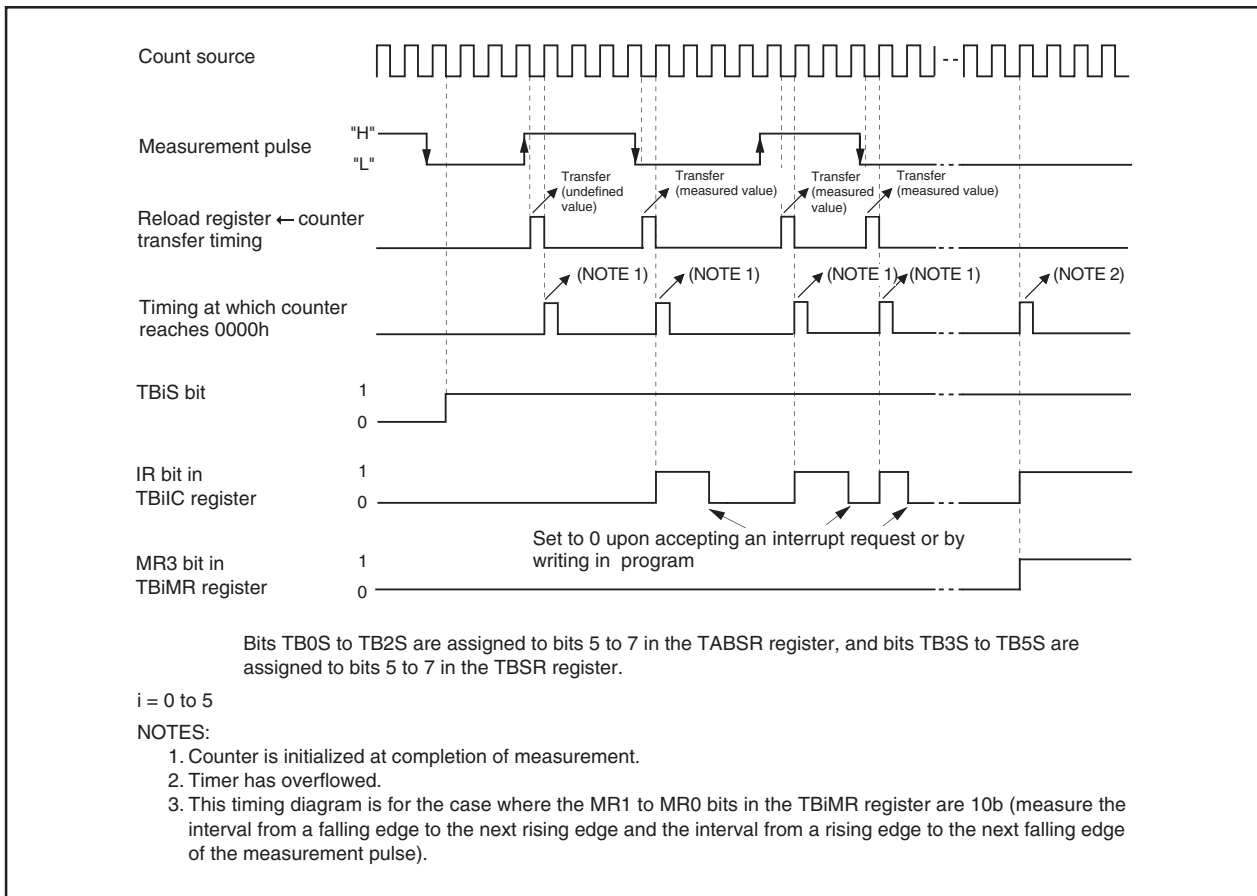


Figure 13.22 Operation Timing When Measuring Pulse Width

14. Three-Phase Motor Control Timer Function

Timers A1, A2, A4, and B2 can be used to output three-phase motor drive waveforms. Table 14.1 lists the Three-phase Motor Control Timer Function Specifications. Figure 14.1 shows the Three-phase Motor Control Timer Function Block Diagram. Figures 14.2 to 14.8 shows the Three-phase Motor Control Timer Function related registers.

Table 14.1 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 NMI pin |
| Used timers | Timer A4, A1, A2 (used in the one-shot timer mode) <ul style="list-style-type: none"> • 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) <ul style="list-style-type: none"> • Carrier wave cycle control Dead time timer (3 eight-bit timer and shared reload register) <ul style="list-style-type: none"> • Dead time control |
| Output waveform | Triangular wave modulation, Sawtooth wave modification <ul style="list-style-type: none"> • 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 $\times (m+1) \times 2$ Sawtooth wave modulation: count source $\times (m+1)$ m: Setting value of the TB2 register, 0000h to FFFFh Count source: f1, f2, f8, f32, fC32 |
| Three-Phase PWM output width | Triangular wave modulation: count source $\times n \times 2$ Sawtooth wave modulation: count source $\times n$ n: Setting value of registers TA4, TA1, and TA2 (of registers TA4, TA41, TA1, TA11, TA2, and TA21 when setting the INV11 bit to 1), 0001h to FFFFh Count source: f1, f2, f8, f32, fC32 |
| Dead time | Count source $\times p$, or no dead time p: Setting value of the DTT register, 01h to FFh Count source: f1, f2, f1 divided by 2, f2 divided by 2 |
| Active level | Enable to select "H" or "L" |
| Positive and negative-phase concurrent active disable function | 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:

1. Forced cutoff with \bar{NMI} input is effective when the IVPCR1 bit in the TB2SC register is set to 1 (three-phase output forcible cutoff by \bar{NMI} input enabled). If an "L" signal is applied to the \bar{NMI} pin when the IVPCR1 bit is 1, the related pins go to a high-impedance state regardless of which functions of those pins are being used.

Related pins: • P7_2/CLK2/TA1OUT/V
 • P7_3/CTS2/RTS2/TA1IN/ \bar{V}
 • P7_4/TA2OUT/W
 • P7_5/TA2IN/ \bar{W}
 • P8_0/TA4OUT/U
 • P8_1/TA4IN/ \bar{U}

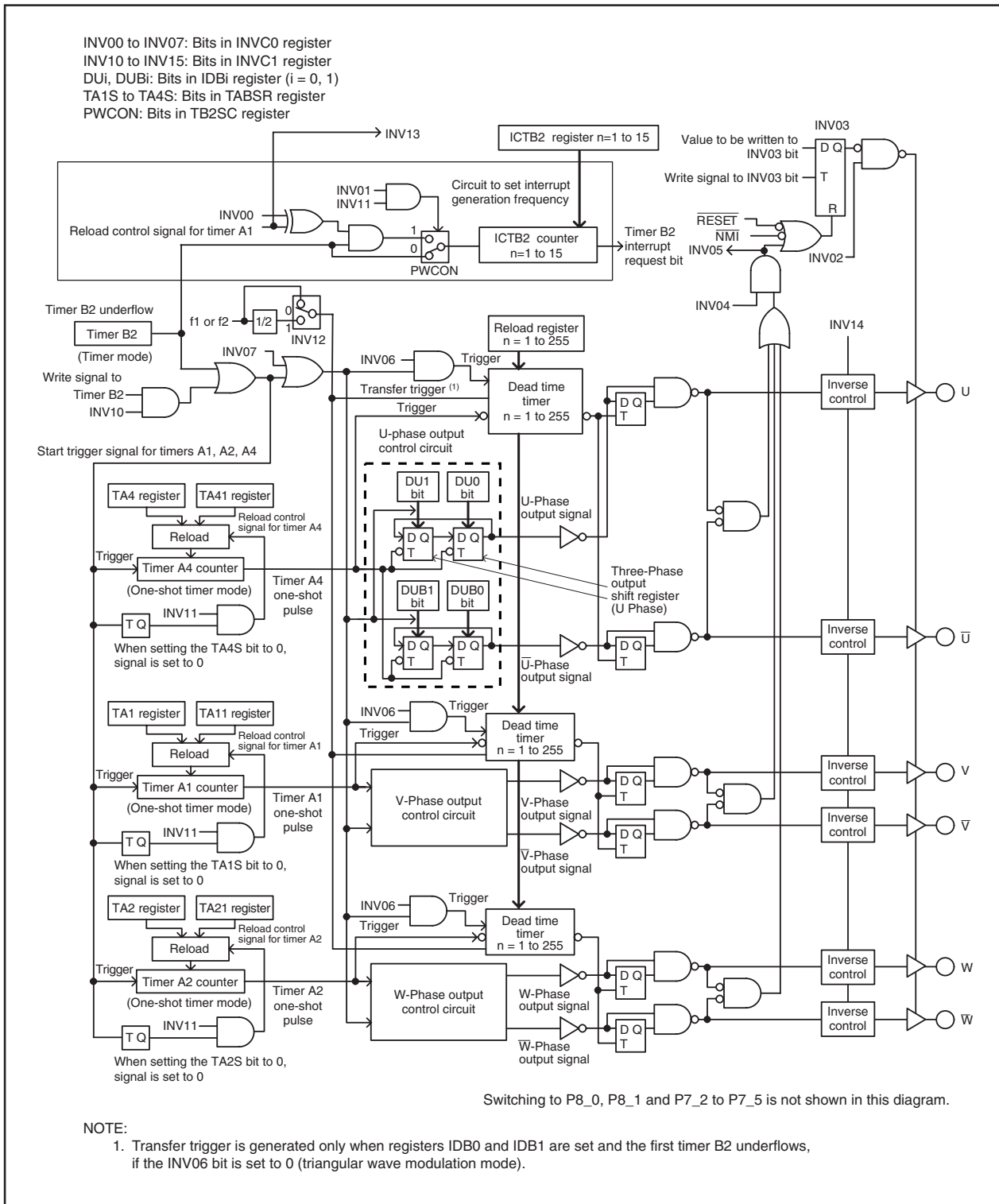


Figure 14.1 Three-Phase Motor Control Timer Function Block Diagram

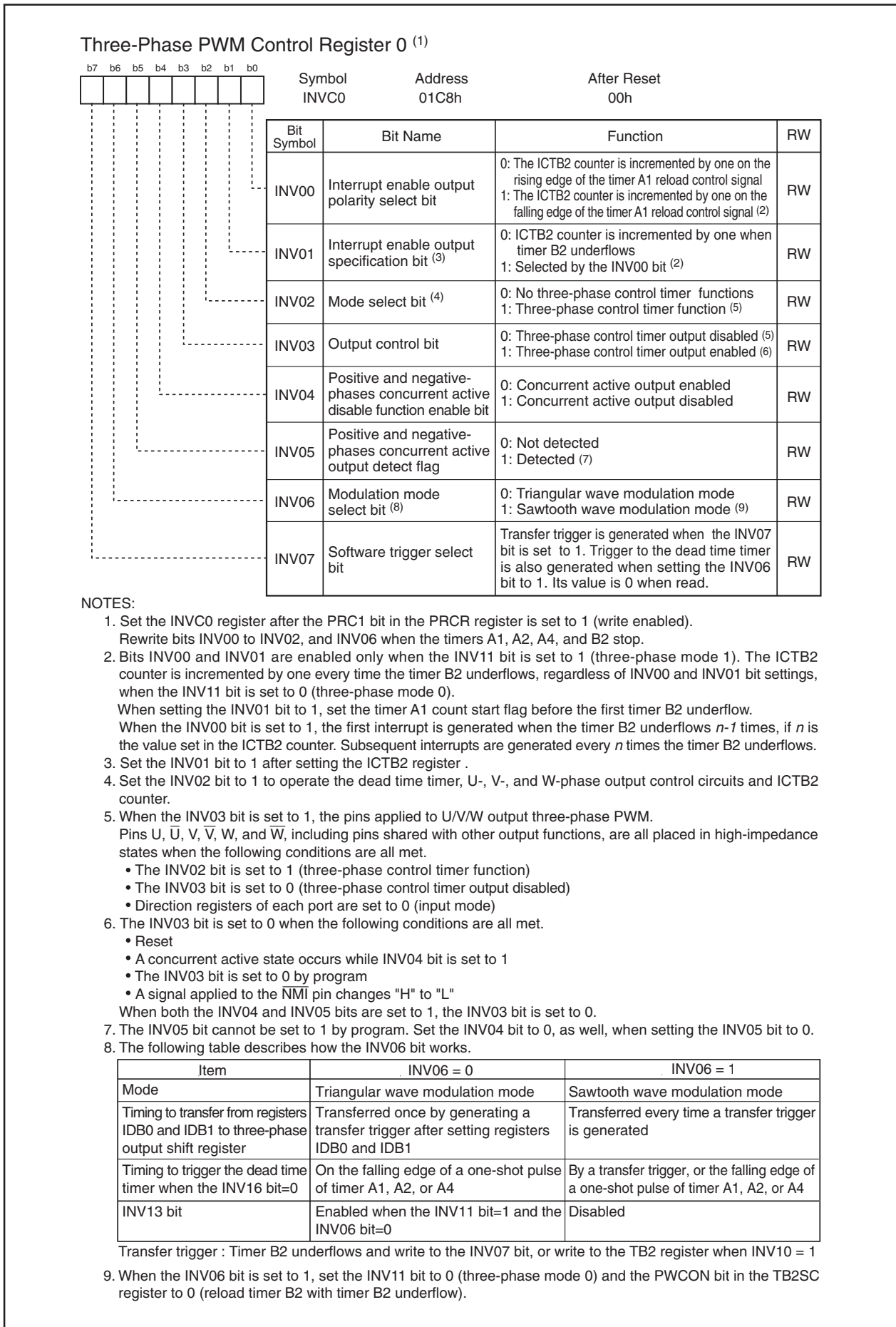


Figure 14.2 INVC0 Register

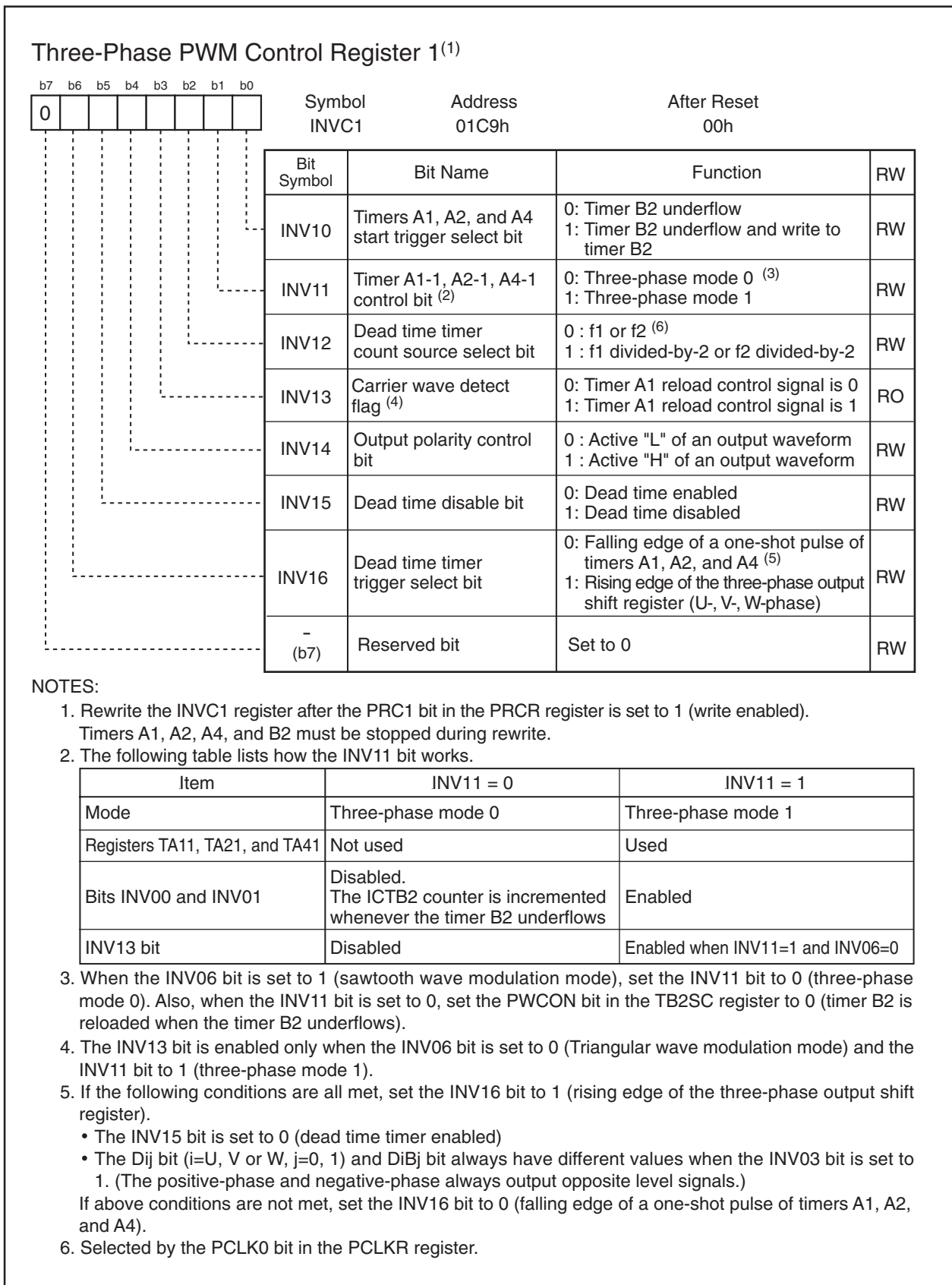


Figure 14.3 INVC1 Register

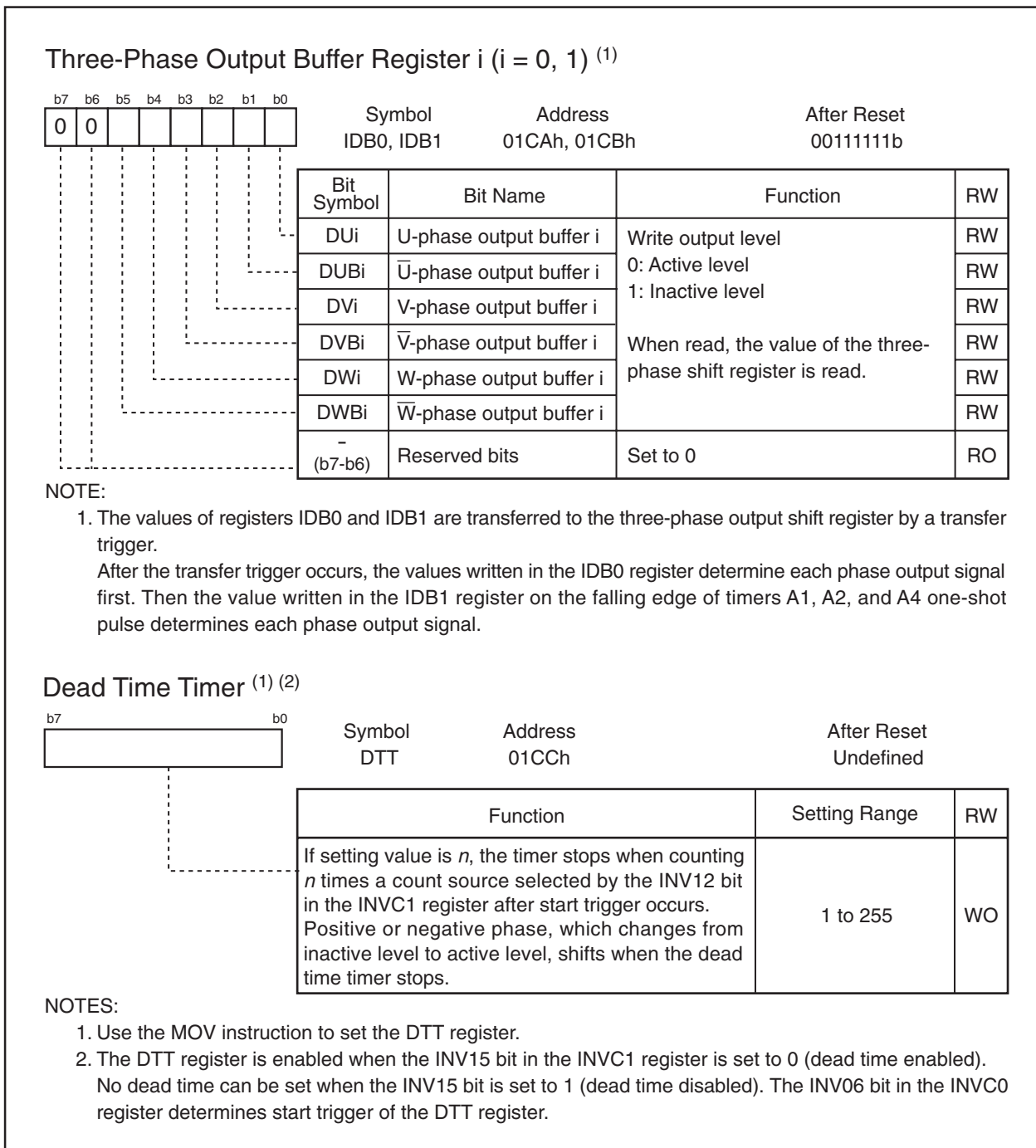


Figure 14.4 Registers IDB0, IDB1, and DTT

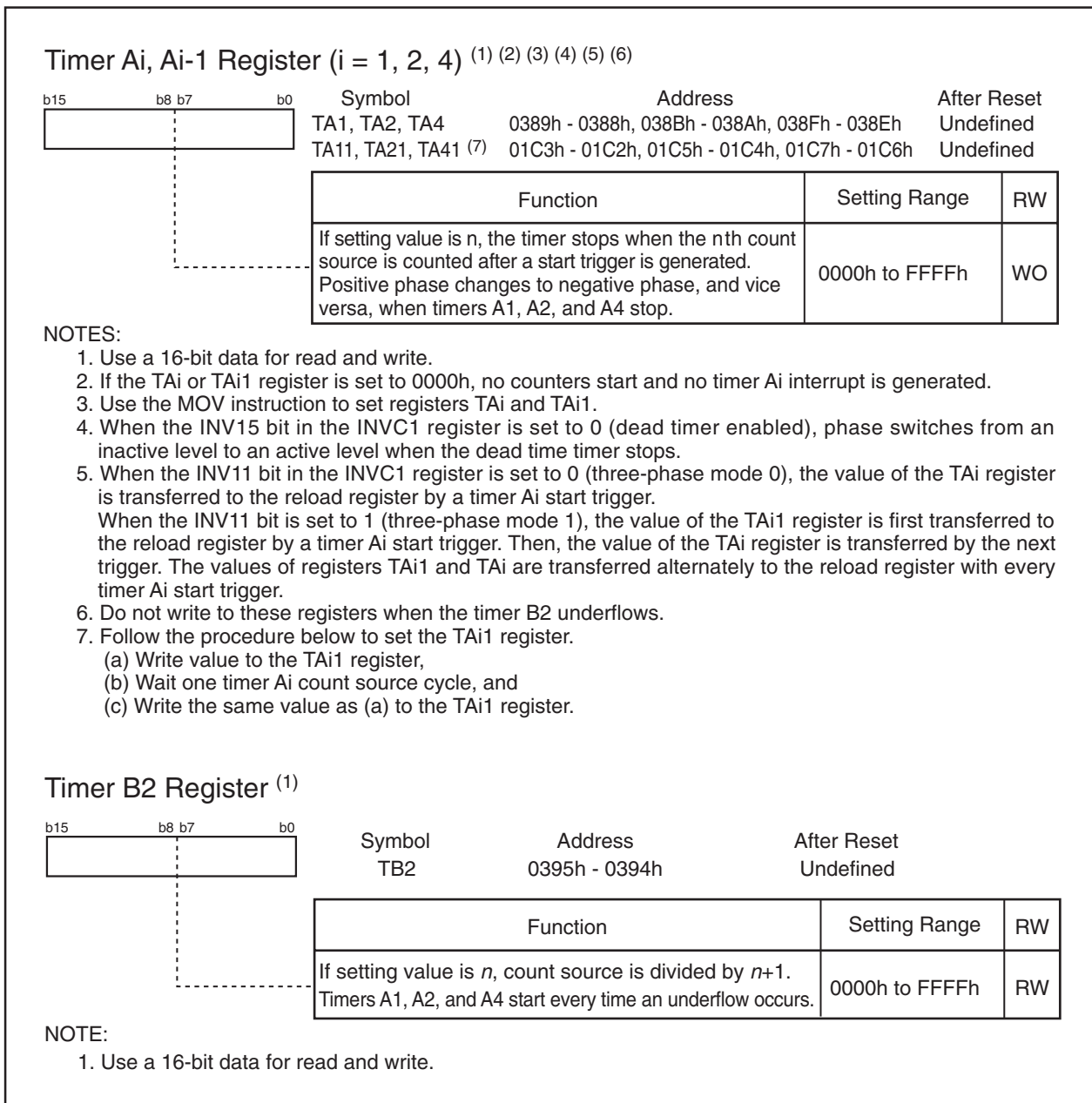


Figure 14.5 Registers TA1, TA2, TA4, TA11, TA21, TA41, and TB2

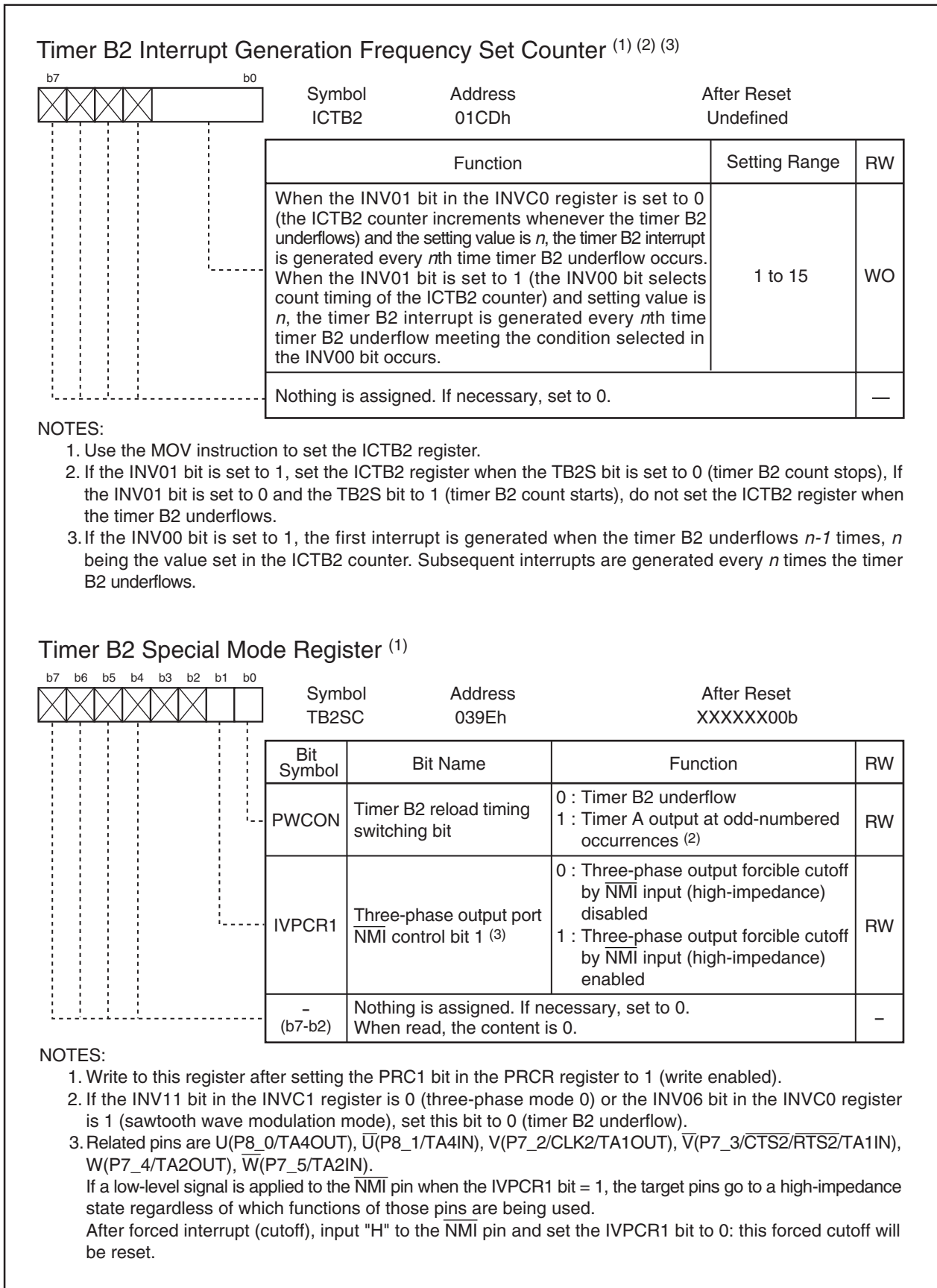


Figure 14.6 Registers ICTB2 and TB2SC

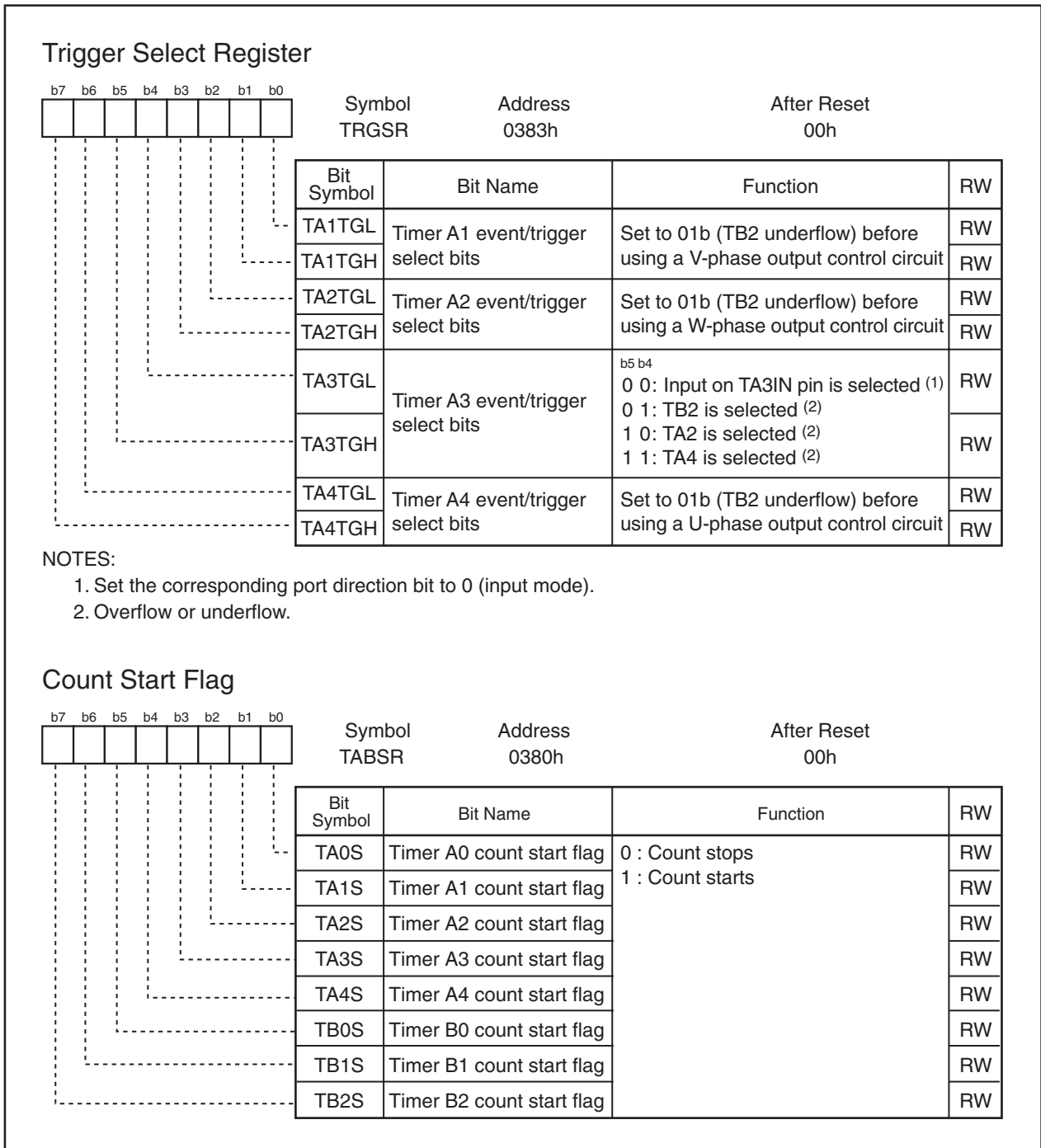


Figure 14.7 Registers TRGSR and TRBSR

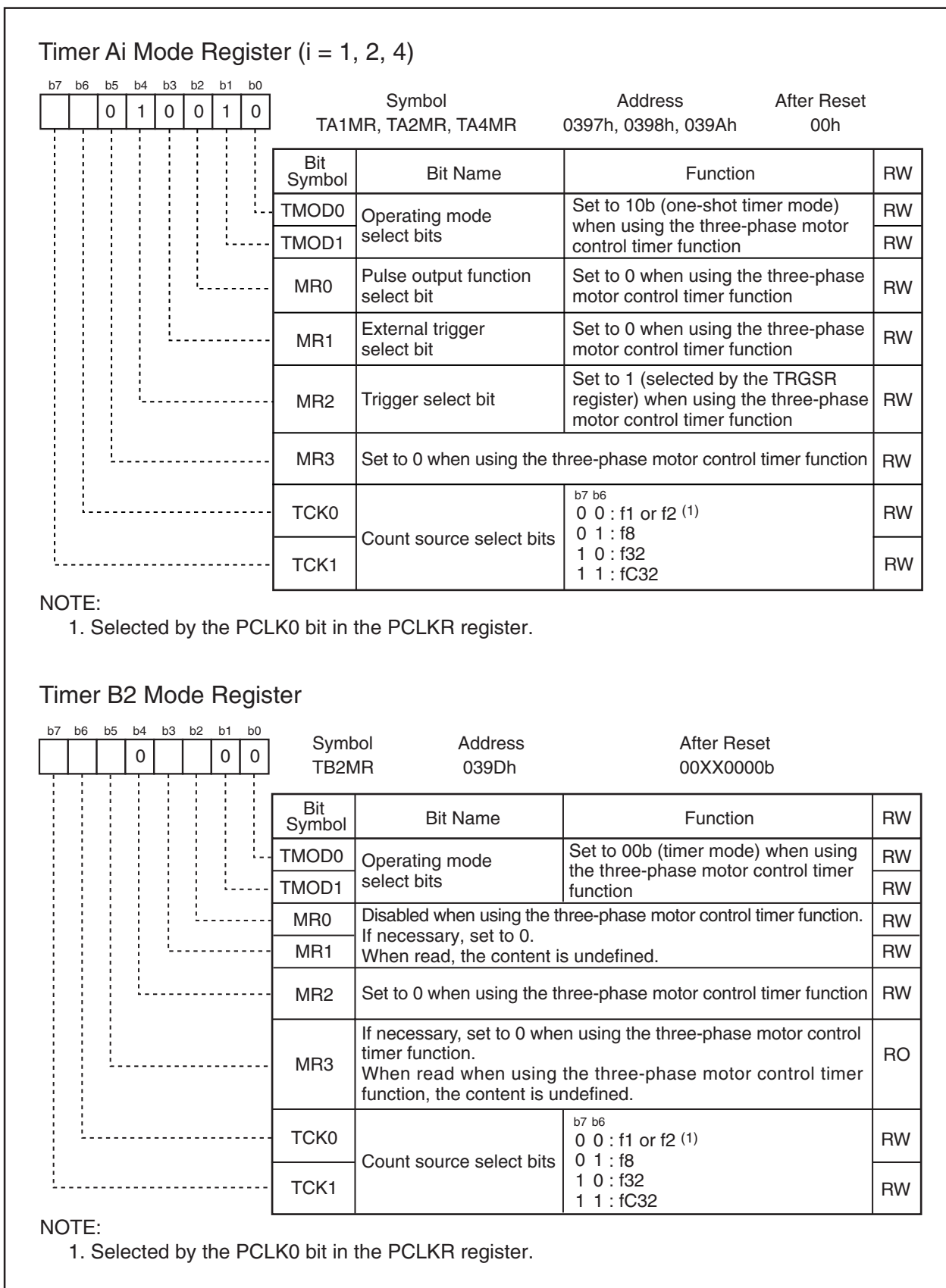


Figure 14.8 Registers TA1MR, TA2MR, TA4MR, and TB2MR

The three-phase motor control timer function is enabled by setting the INV02 bit in the INVC0 register to 1. When this function is selected, 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 14.9 shows an Example of Triangular Wave Modulation Operation and Figure 14.10 shows an Example of Sawtooth Wave Modulation Operation.

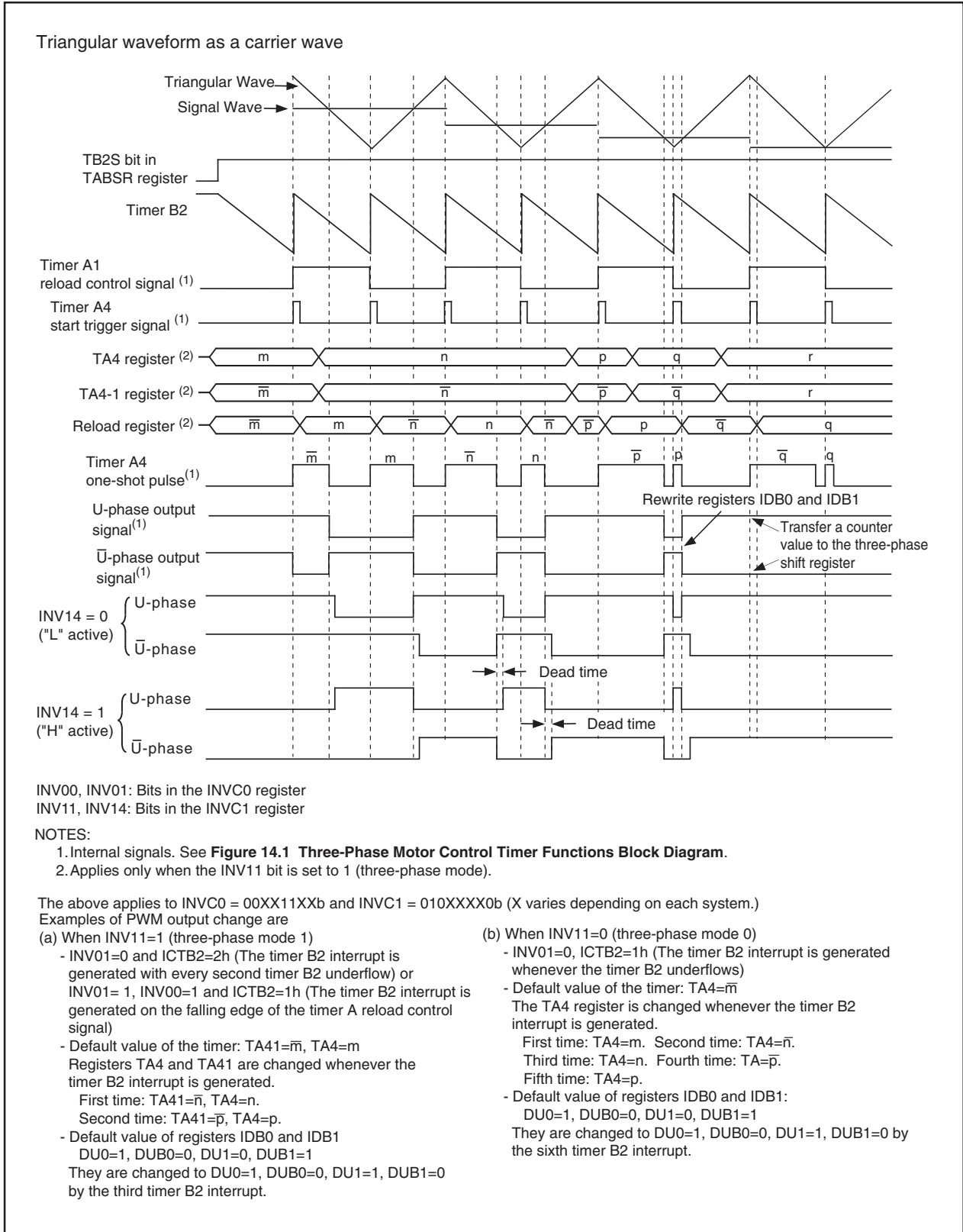


Figure 14.9 Triangular Wave Modulation Operation

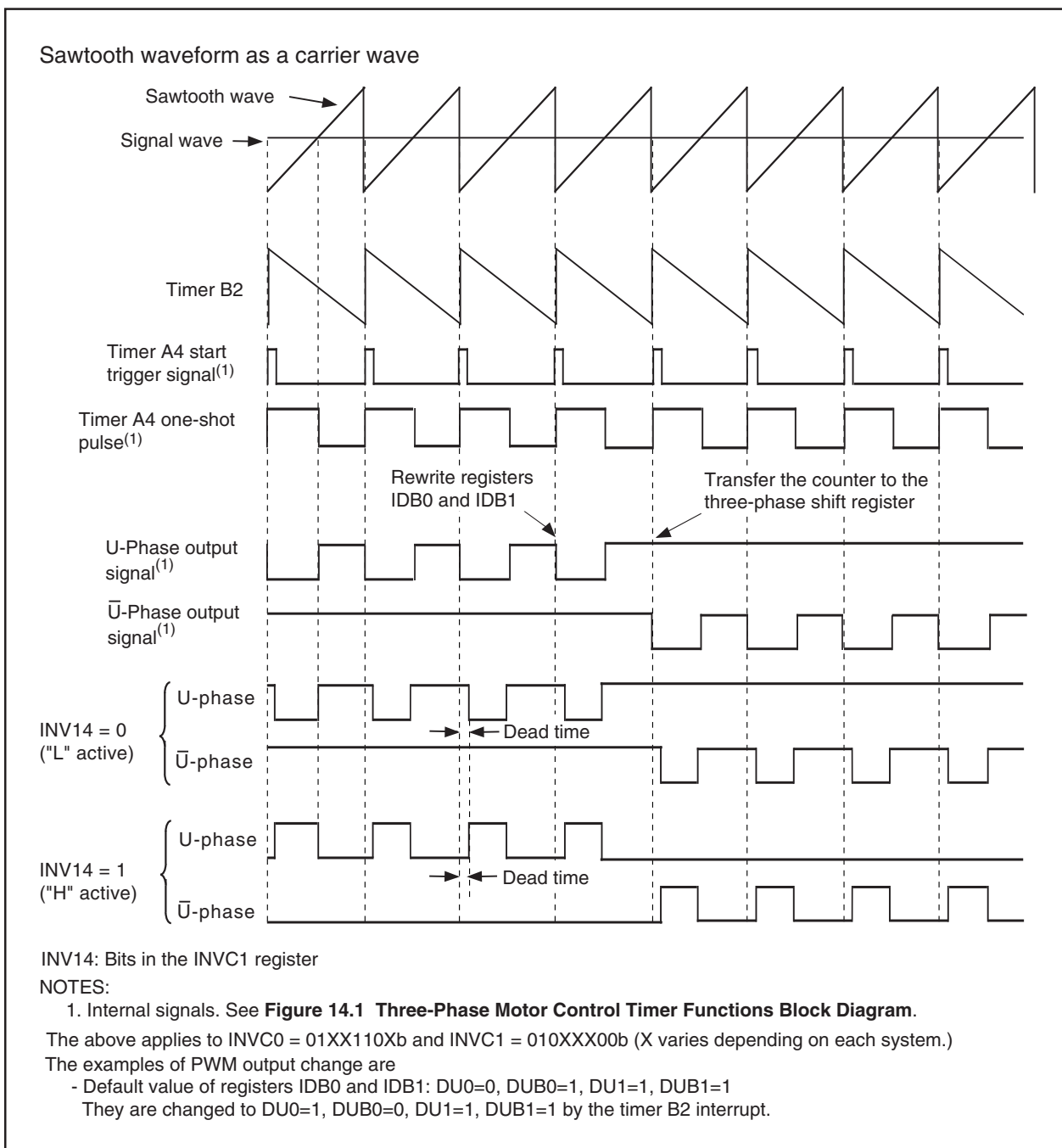


Figure 14.10 Sawtooth Wave Modulation Operation

15. Serial Interface

Serial interface is configured with 4 channels: UART0 to UART2 and SI/O3.

15.1 UARTi (i = 0 to 2)

UARTi each have an exclusive timer to generate a transfer clock, so they operate independently of each other. Figures 15.1 to 15.3 show the UARTi Block Diagram. Figure 15.4 shows the UARTi Transmit/Receive Unit.

UARTi has the following modes:

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

Figures 15.5 to 15.10 show the UARTi-related registers.

Refer to tables listing each mode for register setting.

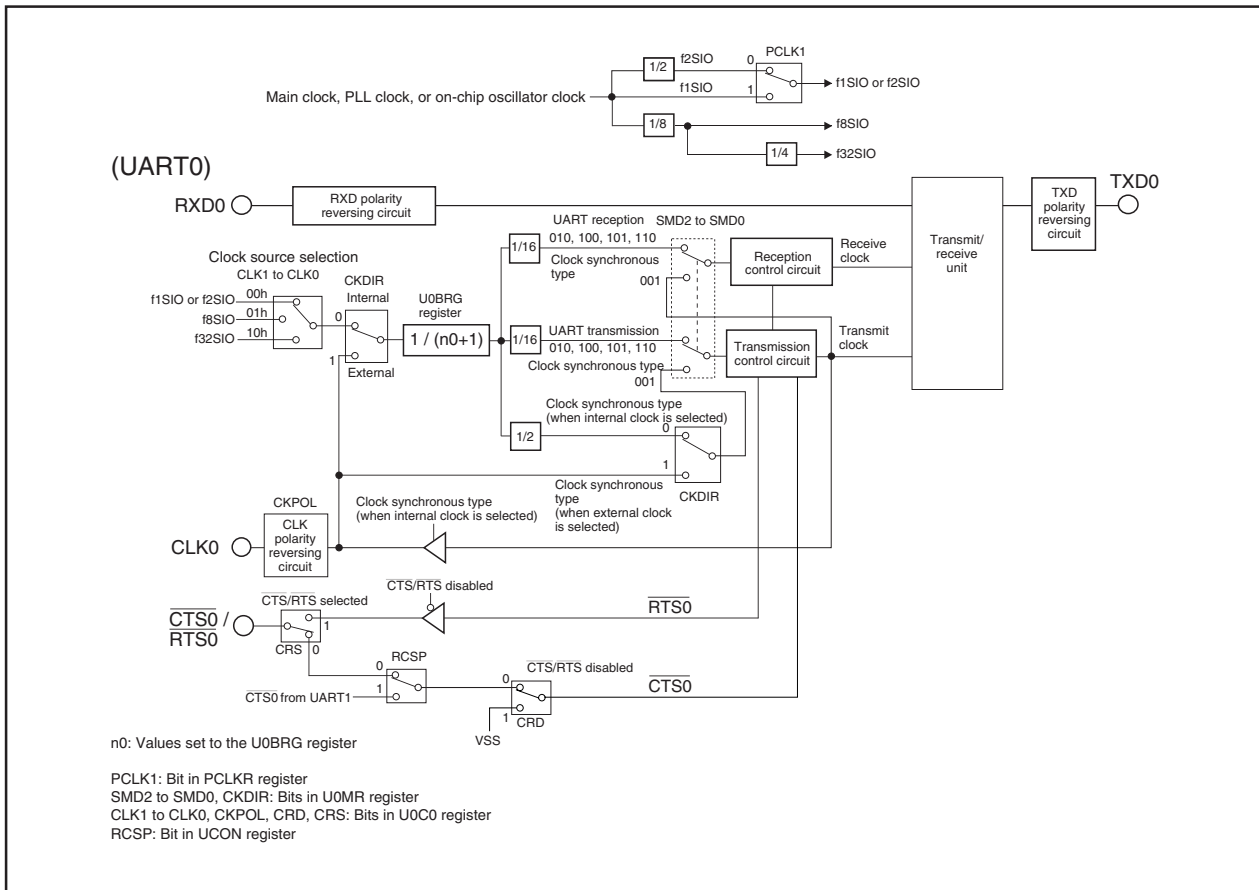


Figure 15.1 UART0 Block Diagram

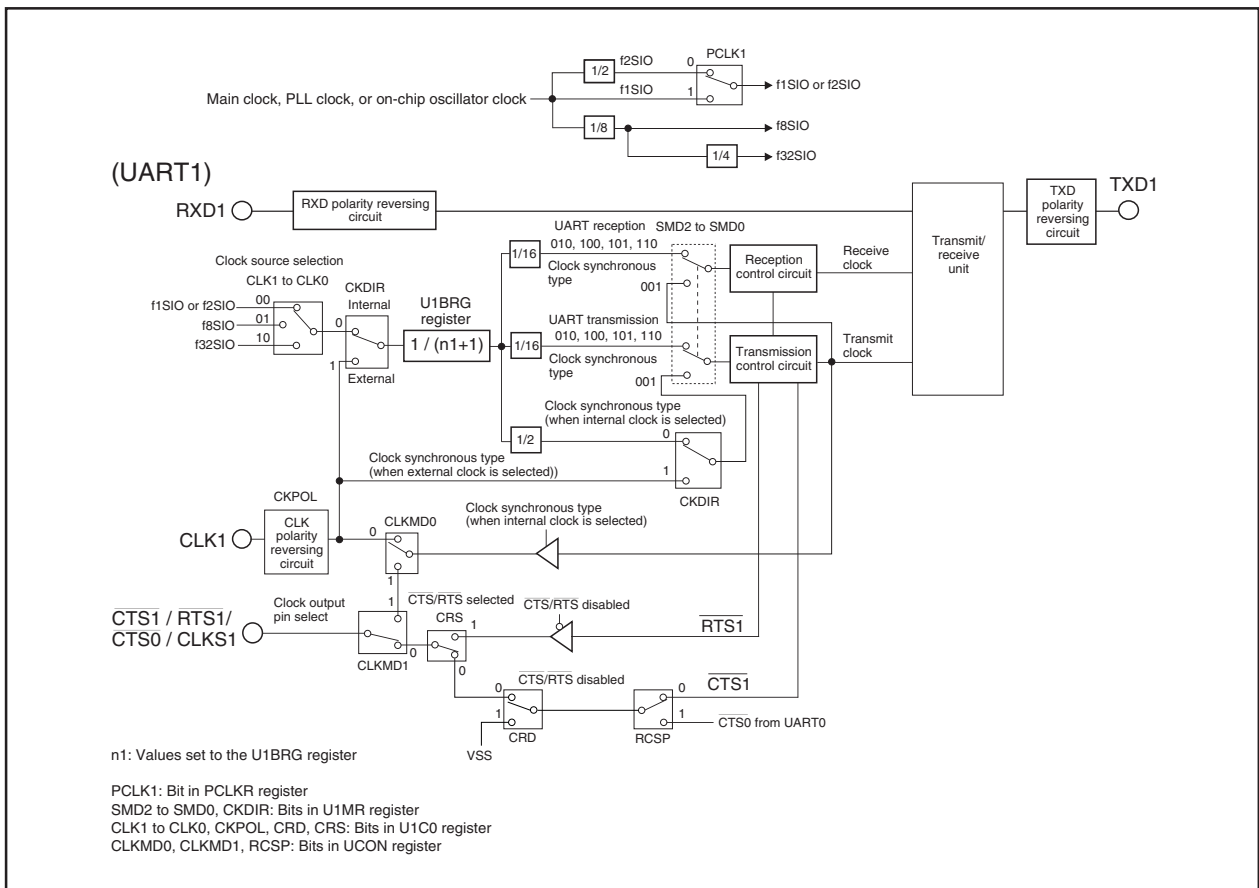


Figure 15.2 UART1 Block Diagram

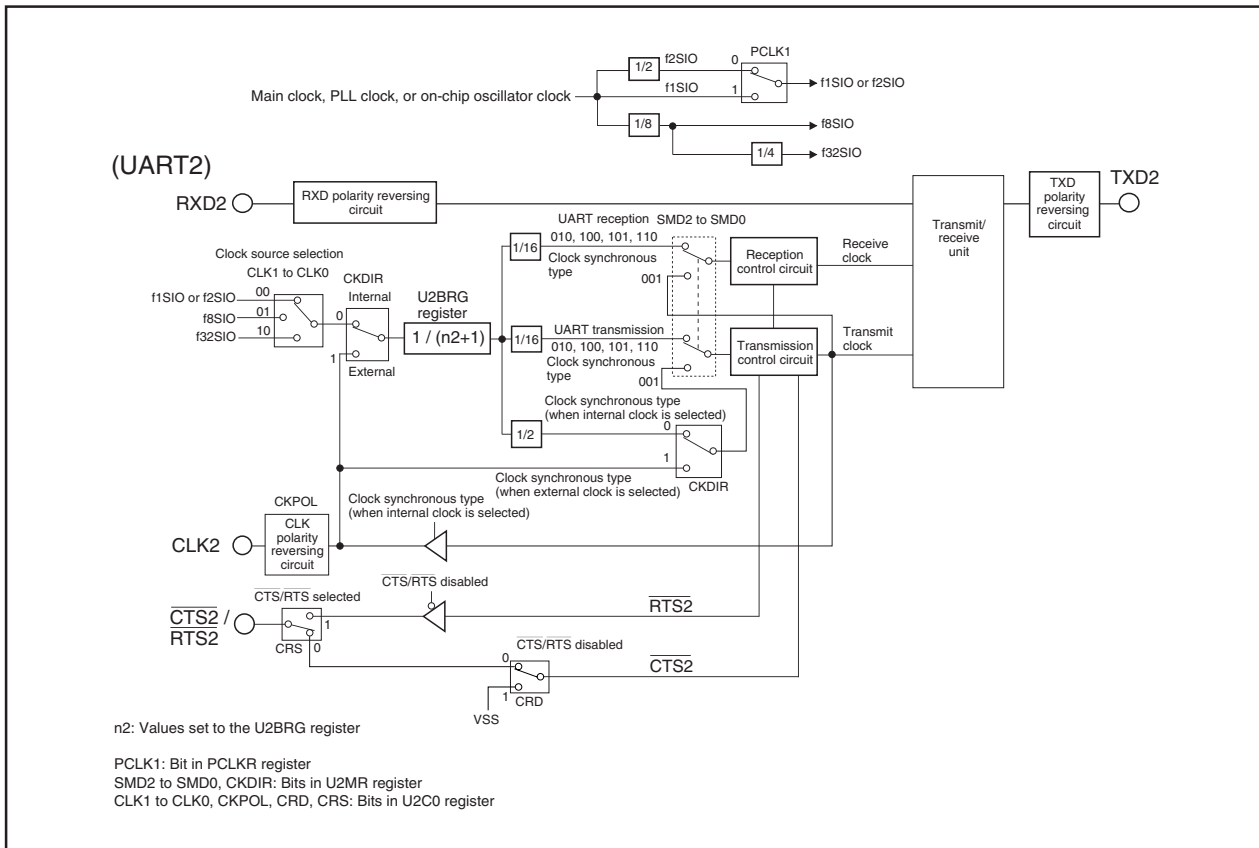


Figure 15.3 UART2 Block Diagram

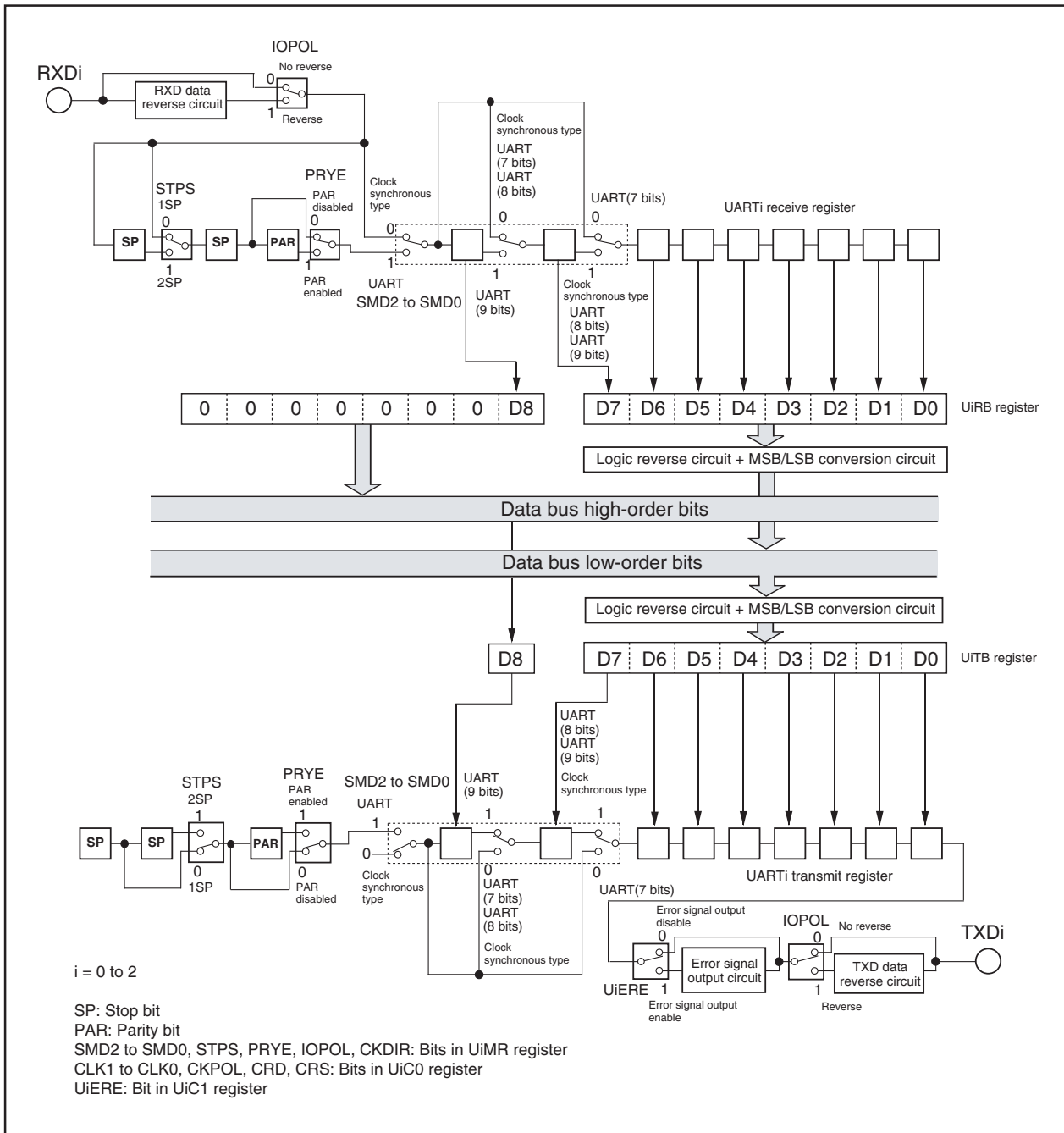


Figure 15.4 UARTi Transmit/Receive Unit

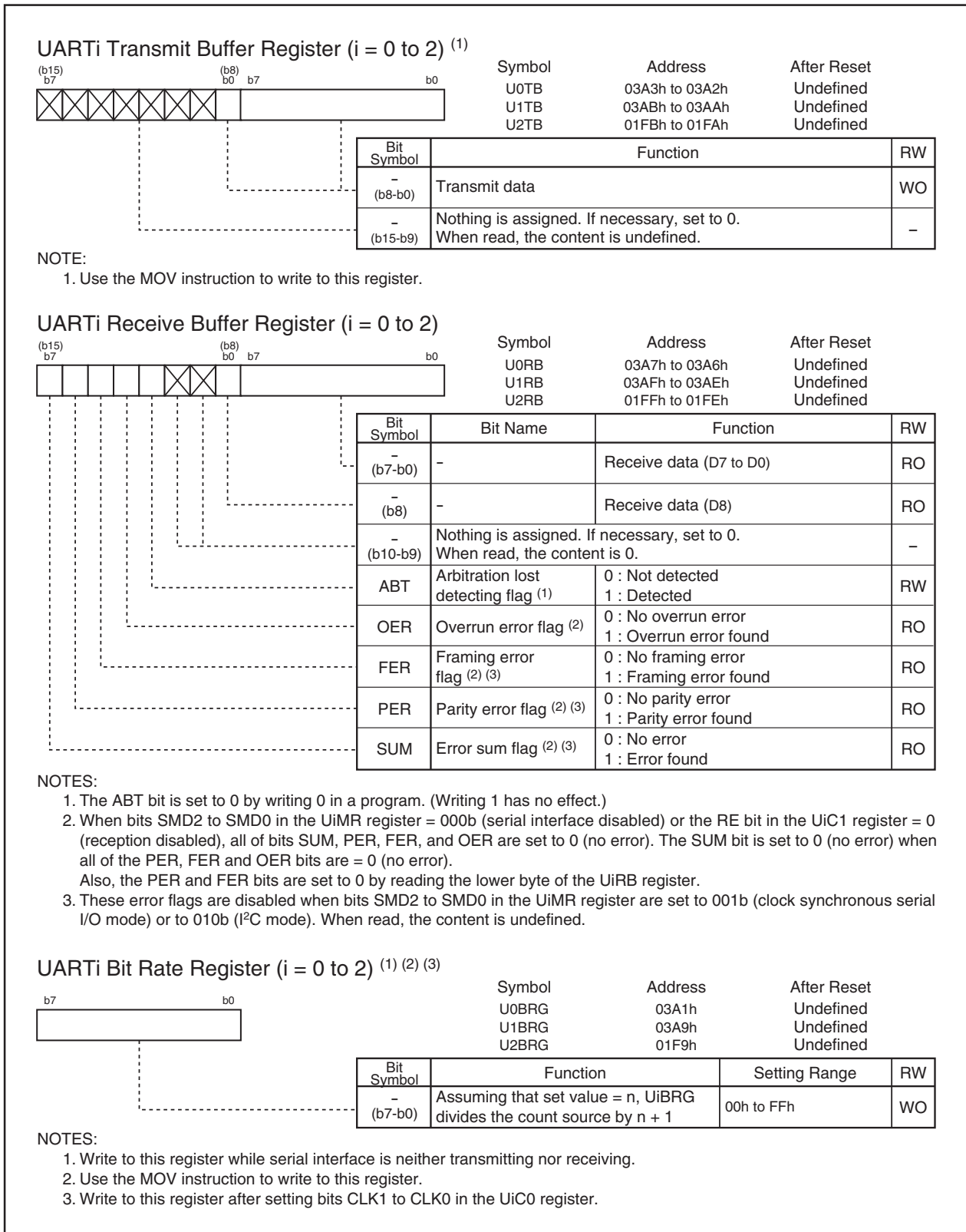


Figure 15.5 Registers U0TB to U2TB, U0RB to U2RB, and U0BRG to U2BRG

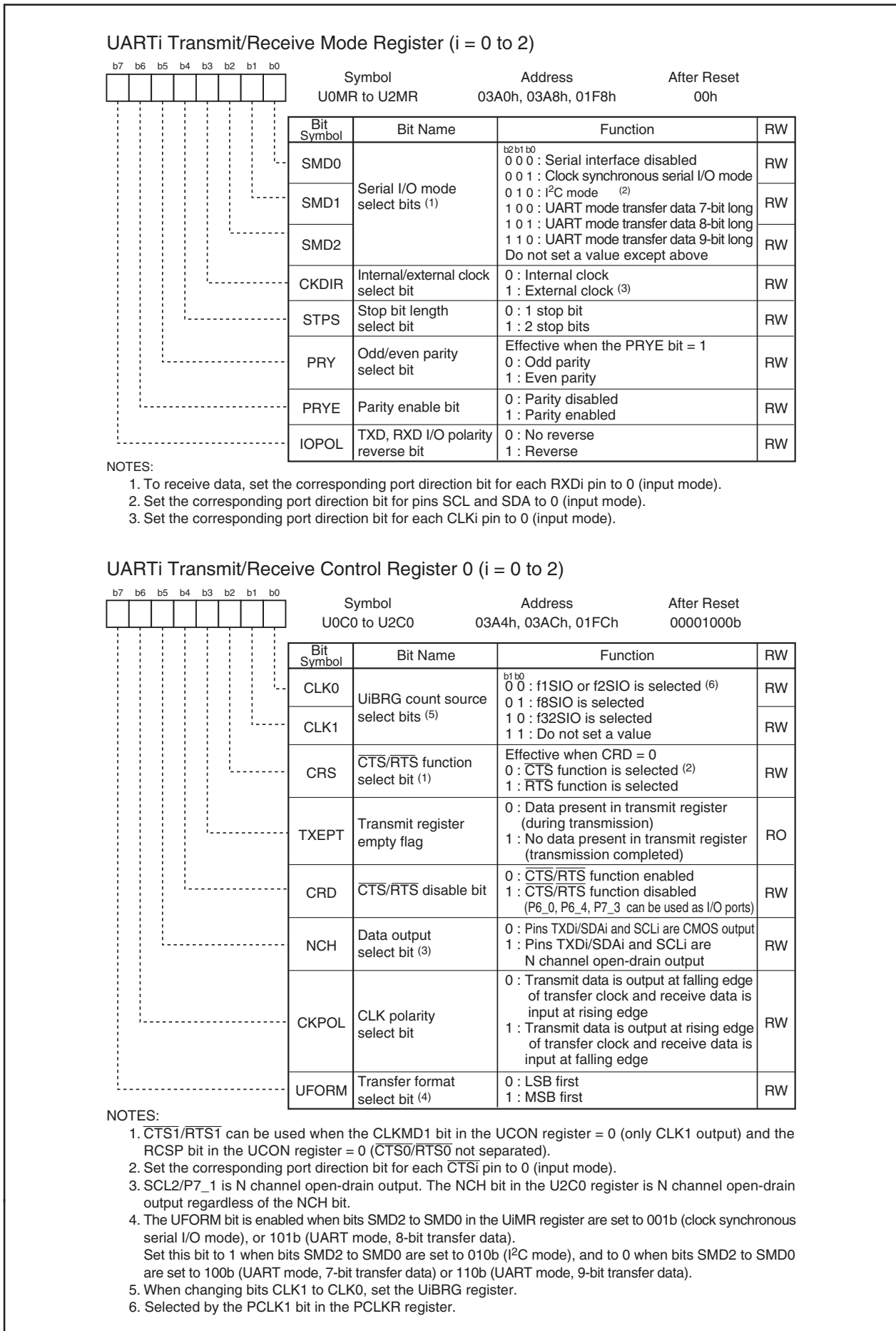


Figure 15.6 Registers U0MR to U2MR and U0C0 to U2C0

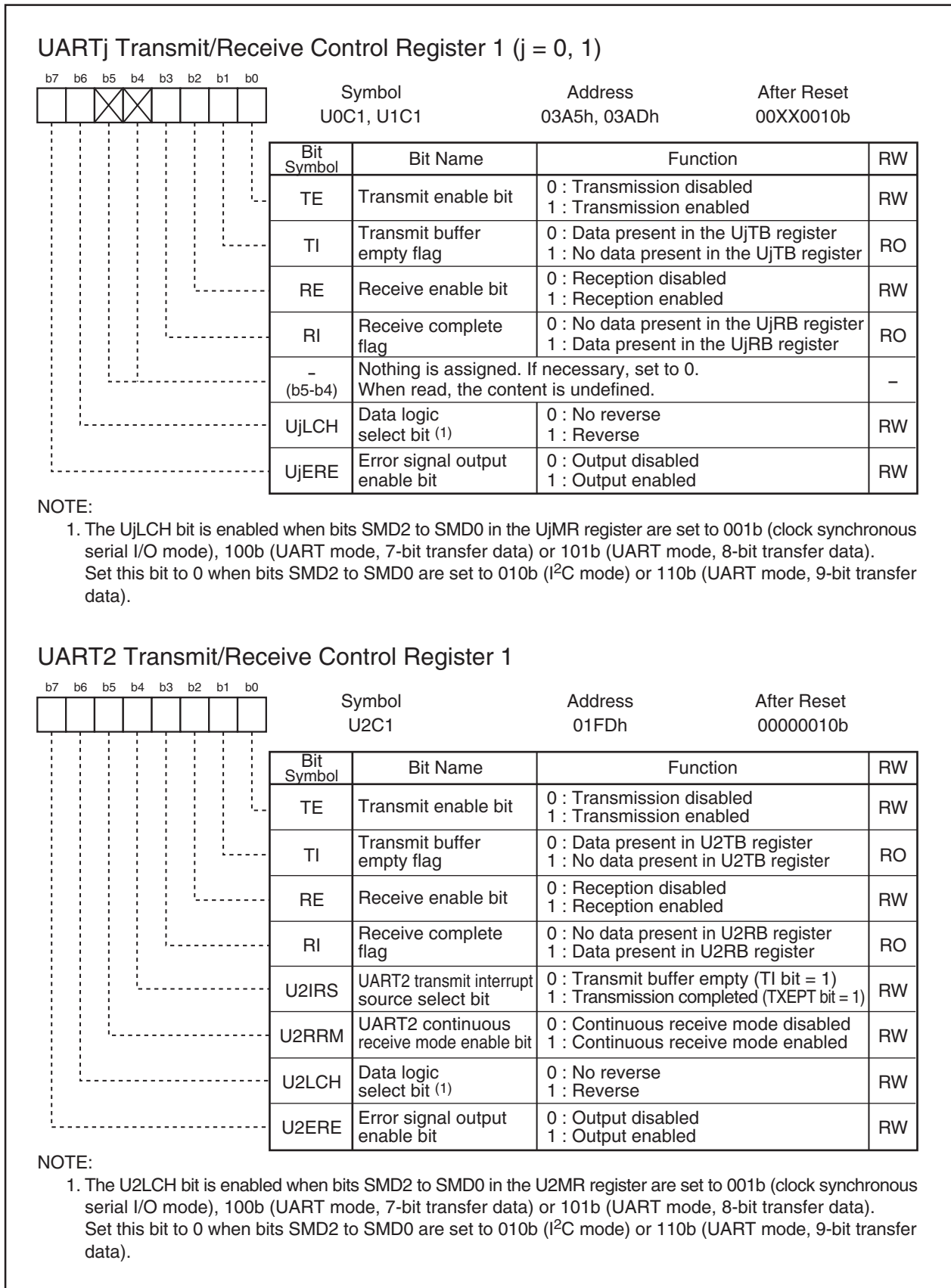


Figure 15.7 Registers U0C1, U1C1, and U2C1

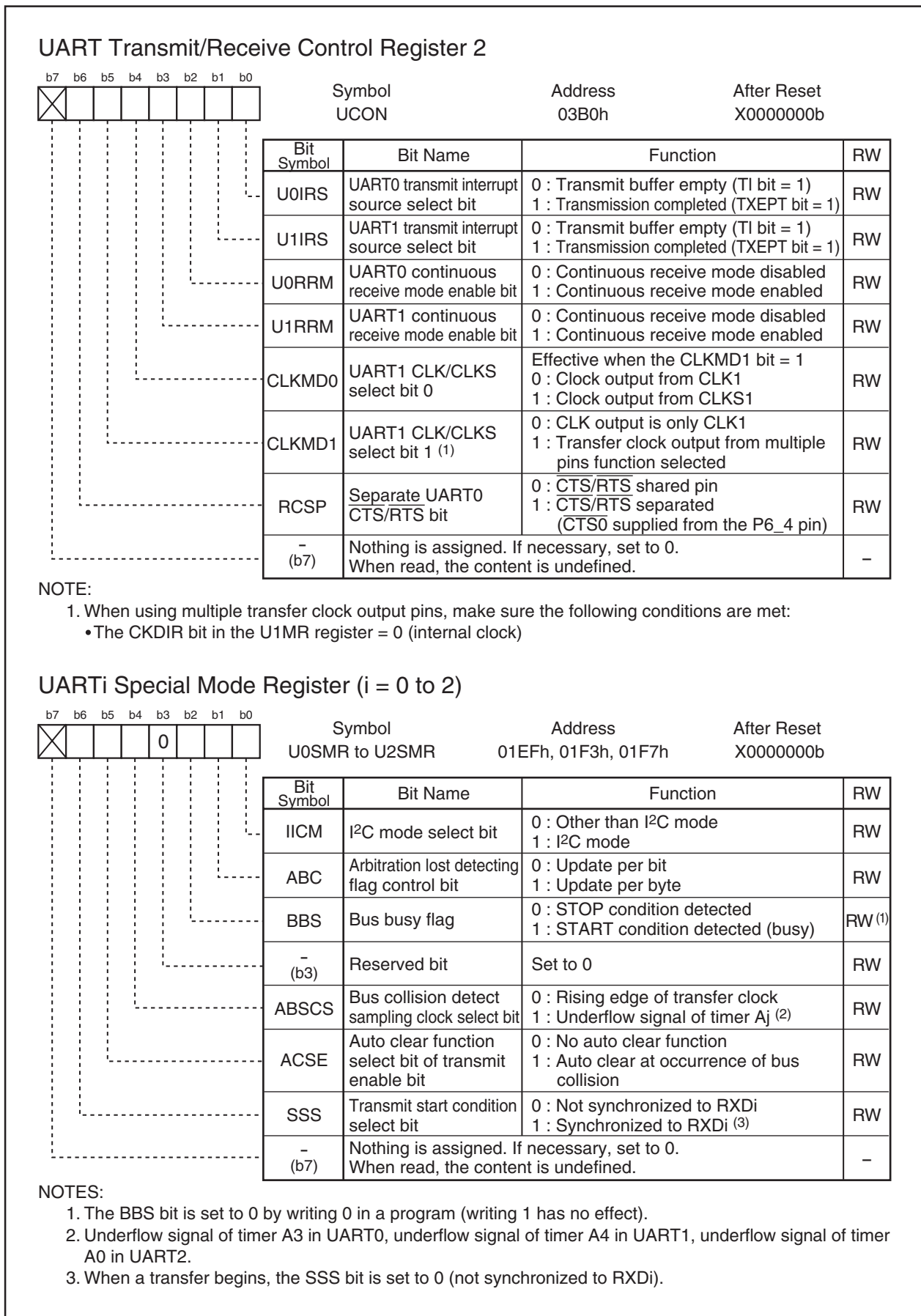


Figure 15.8 Registers UCON, and U0SMR to U2SMR

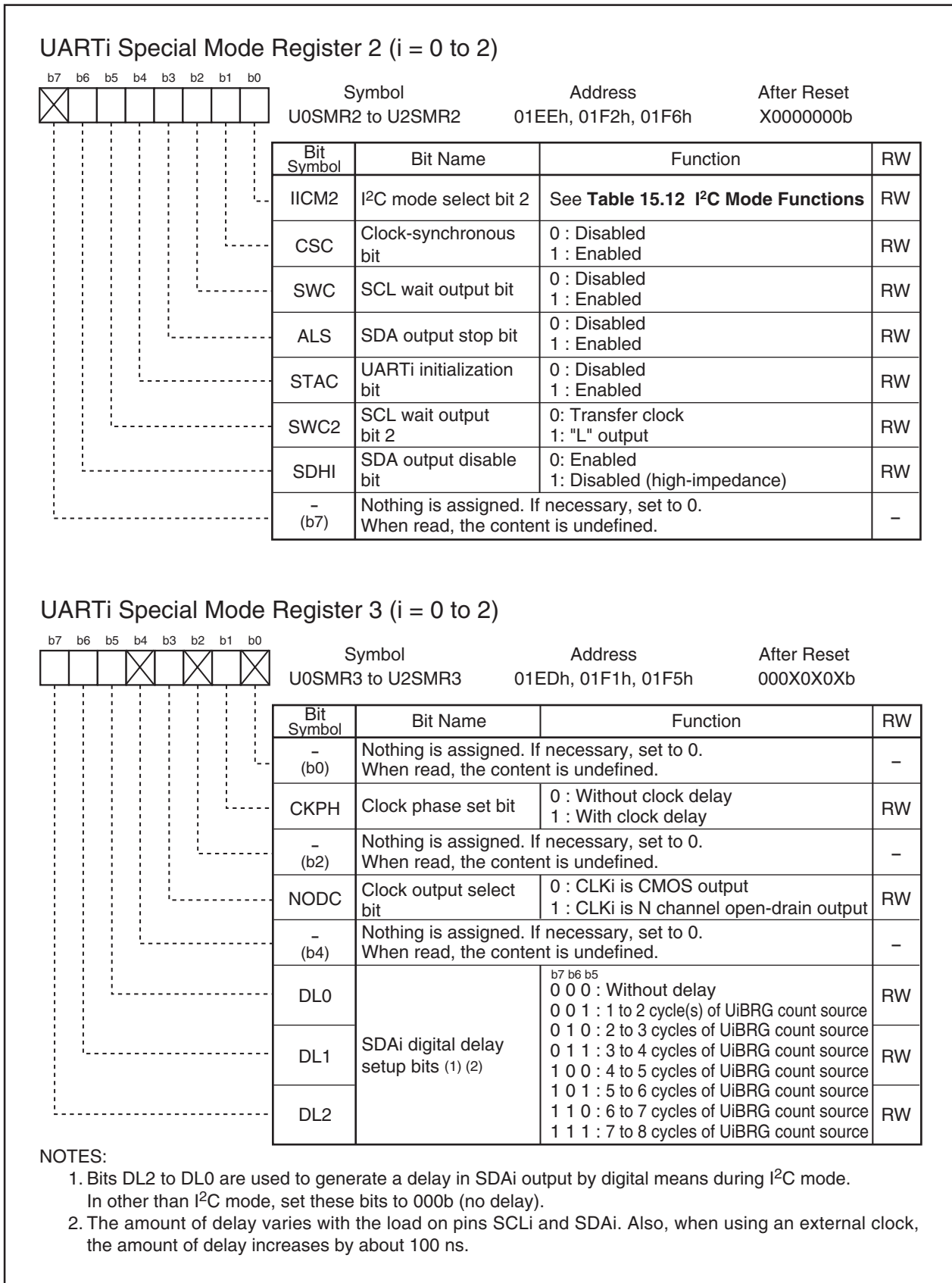


Figure 15.9 Registers U0SMR2 to U2SMR2 and U0SMR3 to U2SMR3

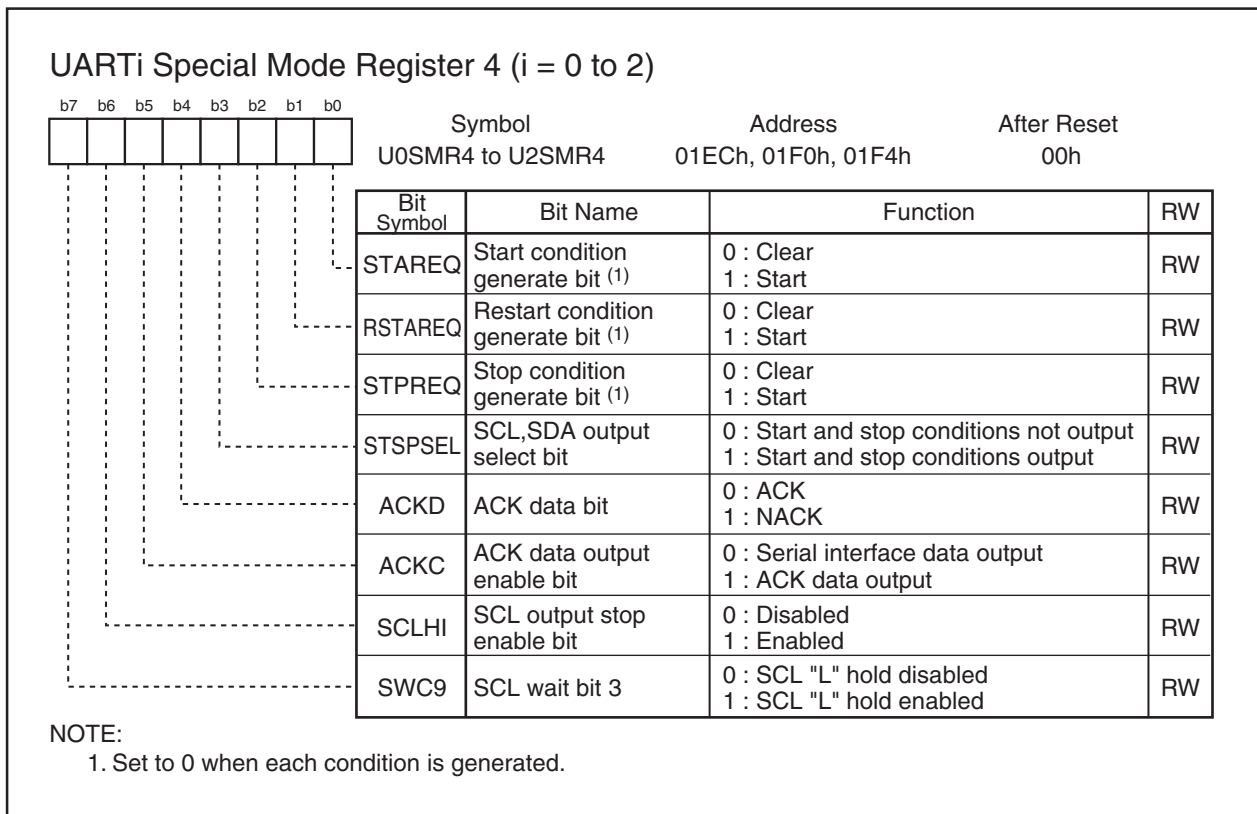


Figure 15.10 Registers U0SMR4 to U2SMR4

15.1.1 Clock Synchronous Serial I/O Mode

The clock synchronous serial I/O mode uses a transfer clock to transmit and receive data.

Table 15.1 lists the Clock Synchronous Serial I/O Mode Specifications. Table 15.2 lists the Registers to be Used in and Setting in Clock Synchronous Serial I/O Mode.

Table 15.1 Clock Synchronous Serial I/O Mode Specifications

| Item | Specification |
|-------------------------------------|---|
| Transfer data format | Transfer data length: 8 bits |
| Transfer clock | The CKDIR bit in the UiMR register = 0 (internal clock) : $f_j/(2(n+1))$ <ul style="list-style-type: none"> $f_j = f1SIO, f2SIO, f8SIO, f32SIO$. n: Setting value of the UiBRG register 00h to FFh The CKDIR bit = 1 (external clock) : Input from CLKi pin |
| Transmit/receive control | Selectable from \overline{CTS} function, \overline{RTS} function or $\overline{CTS}/\overline{RTS}$ function disabled |
| Transmit start condition | Before transmission can start, meet the following requirements ⁽¹⁾ <ul style="list-style-type: none"> The TE bit in the UiC1 register = 1 (transmission enabled) The TI bit in the UiC1 register = 0 (data present in the UiTB register) If CTS function is selected, input on the \overline{CTS}i pin = L |
| Receive start condition | Before reception can start, meet the following requirements ⁽¹⁾ <ul style="list-style-type: none"> The RE bit in the UiC1 register = 1 (reception enabled) The TE bit in the UiC1 register = 1 (transmission enabled) The TI bit in the UiC1 register = 0 (data present in the UiTB register) |
| Interrupt request generation timing | For transmission, one of the following conditions can be selected <ul style="list-style-type: none"> The UiIRS bit ⁽²⁾ = 0 (transmit buffer empty): when transferring data from the UiTB register to the UARTi transmit register (at start of transmission) The UiIRS bit = 1 (transmission completed): when the serial interface finished transmitting 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 | Overrun error ⁽³⁾ This error occurs if the serial interface started receiving the next data before reading the UiRB register and received the 7th bit of the next data |
| Select function | <ul style="list-style-type: none"> CLK polarity selection Transfer data input/output can be selected to occur synchronously with the rising or the falling edge of the transfer clock LSB first, MSB first selection Whether to start transmitting or receiving data begins with bit 0 or begins with bit 7 can be selected Continuous receive mode selection Reception is enabled immediately by reading the UiRB register Switching serial data logic 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) $\overline{CTS0}$ and $\overline{RTS0}$ are input/output from separate pins |

i = 0 to 2

NOTES:

- When an external clock is selected, the conditions must be met while if the CKPOL bit in the UiC0 register = 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 = 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.
- Bits U0IRS and U1IRS are bits 0 and 1 in the UCON register; the U2IRS bit is bit 4 in the U2C1 register.
- If an overrun error occurs, the receive data of UiRB register will be undefined. The IR bit in the SiRIC register remains unchanged.

Table 15.2 Registers to be Used and Settings in Clock Synchronous Serial I/O Mode

| Register | Bit | Function |
|---------------------|----------------------|---|
| UiTB ⁽¹⁾ | 0 to 7 | Set transmit data |
| UiRB ⁽¹⁾ | 0 to 7 | Receive data can be read |
| | OER | Overflow error flag |
| UiBRG | 0 to 7 | Set a bit rate |
| UiMR ⁽¹⁾ | SMD2 to SMD0 | Set to 001b |
| | CKDIR | Select the internal clock or external clock |
| | IOPOL | Set to 0 |
| UiC0 | CLK1 to CLK0 | Select the count source for the UiBRG register |
| | CRS | Select CTS or RTS to use |
| | TXEPT | Transmit register empty flag |
| | CRD | Select CTS/RTS function enabled or disabled |
| | 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 |
| | TI | Transmit buffer empty flag |
| | RE | Set this bit to 1 to enable reception |
| | RI | Reception complete flag |
| | U2IRS ⁽²⁾ | Select the UART2 transmit interrupt source |
| | U2RRM ⁽²⁾ | Set this bit to 1 to use continuous receive mode |
| | UiLCH | Set this bit to 1 to use inverted data logic |
| | UiERE | Set to 0 |
| UiSMR | 0 to 7 | Set to 0 |
| UiSMR2 | 0 to 7 | Set to 0 |
| UiSMR3 | 0 to 2 | Set to 0 |
| | NODC | Select clock output mode |
| | 4 to 7 | Set to 0 |
| UiSMR4 | 0 to 7 | Set to 0 |
| UCON | U0IRS, U1IRS | Select the UART0/UART1 transmit interrupt source |
| | U0RRM, U1RRM | Set this bit to 1 to use continuous receive mode |
| | CLKMD0 | Select the transfer clock output pin when the CLKMD1 bit = 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 $\overline{\text{CTS0}}$ signal of the UART0 from the P6_4 pin |
| | 7 | Set to 0 |

i = 0 to 2

NOTES:

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

Table 15.3 lists the I/O Pin Functions (when not select multiple transfer clock output pin select function) in clock synchronous serial I/O mode. Table 15.4 lists the P6_4 Pin Functions in clock synchronous serial I/O mode.

Note that for a period from when the UART_i operating mode is selected to when transfer starts, the TXD_i pin outputs an “H”.

Figure 15.11 shows the Transmit/Receive Operation during clock synchronous serial I/O mode.

Table 15.3 I/O Pin Functions (when not select multiple transfer clock output pin select function)

| Pin Name | Function | Method of Selection |
|--|-----------------------|---|
| TXD _i (P6_3, P6_7, P7_0) | Serial data output | (Outputs dummy data when performing reception only) |
| RXD _i (P6_2, P6_6, P7_1) | Serial data input | Bits PD6_2 and PD6_6 in PD6 register = 0 PD7_1 bit in PD7 register = 0 (Can be used as an input port when performing transmission only) |
| CLK _i (P6_1, P6_5, P7_2) | Transfer clock output | CKDIR bit in UiMR register = 0 |
| | Transfer clock input | CKDIR bit = 1 Bits PD6_1 and PD6_5 in PD6 register = 0 PD7_2 bit in PD7 register = 0 |
| CTS _i /RTS _i (P6_0, P6_4, P7_3) | CTS input | CRD bit in UiC0 register = 0 CRS bit in UiC0 register = 0 Bits PD6_0 and PD6_4 in PD6 register = 0 PD7_3 bit in PD7 register = 0 |
| | RTS output | CRD bit = 0 CRS bit = 1 |
| | I/O port | CRD bit = 1 |

i = 0 to 2

Table 15.4 P6_4 Pin Functions

| Pin Function | Bit set Value | | | | | |
|---------------------|---------------|---------|---------------|------------------|------------|---------------------|
| | U1C0 Register | | UCON Register | | | PD6 Register |
| | CRD bit | CRS bit | RCSP bit | CLKMD1 bit | CLKMD0 bit | PD6_4 bit |
| P6_4 | 1 | - | 0 | 0 | - | Input: 0, Output: 1 |
| CTS1 | 0 | 0 | 0 | 0 | - | 0 |
| RTS1 | 0 | 1 | 0 | 0 | - | - |
| CTS0 ⁽¹⁾ | 0 | 0 | 1 | 0 | - | 0 |
| CLKS1 | - | - | - | 1 ⁽²⁾ | 1 | - |

∴: 0 or 1

NOTES:

- In addition to this, set the CRD bit in the U0C0 register to 0 (CTS₀/RTS₀ enabled) and the CRS bit in the U0C0 register to 1 (RTS₀ selected).
- When the CLKMD1 bit = 1 and the CLKMD0 bit = 0, the following logic levels are output:
 - High if the CLKPOL bit in the U1C0 register = 0
 - Low if the CLKPOL bit = 1

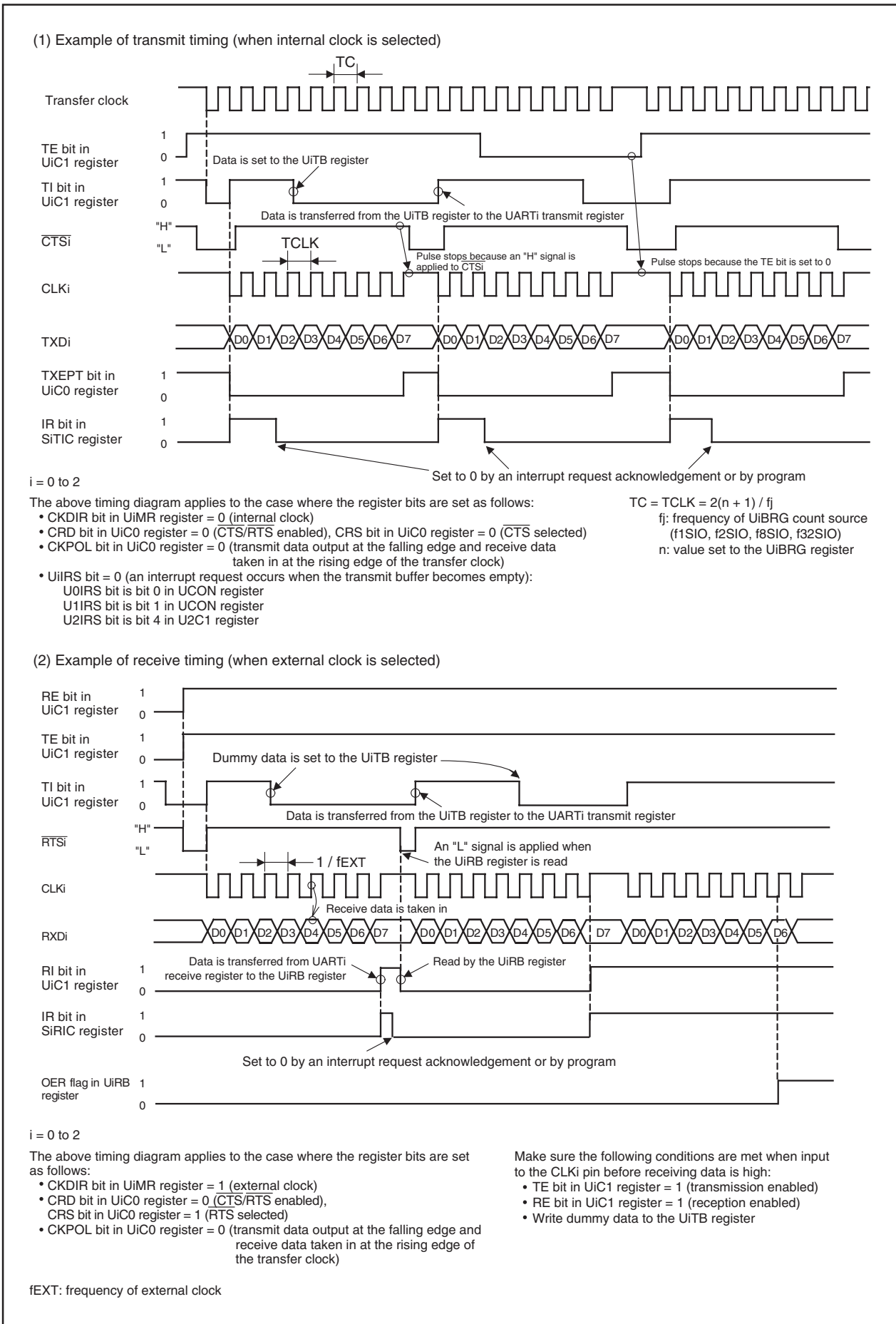


Figure 15.11 Transmit and Receive Operation

15.1.1.1 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 000b (serial interface disabled)
 - (3) Set bits SMD2 to SMD0 in the UiMR register to 001b (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 000b (serial interface disabled)
 - (2) Set bits SMD2 to SMD0 in the UiMR register to 001b (clock synchronous serial I/O mode)
 - (3) 1 (transmission enabled) is written to the TE bit in the UiC1 register, regardless of the TE bit

15.1.1.2 CLK Polarity Select Function

Use the CKPOL bit in the UiC0 register ($i = 0$ to 2) to select the transfer clock polarity. Figure 15.12 shows the Transfer Clock Polarity.

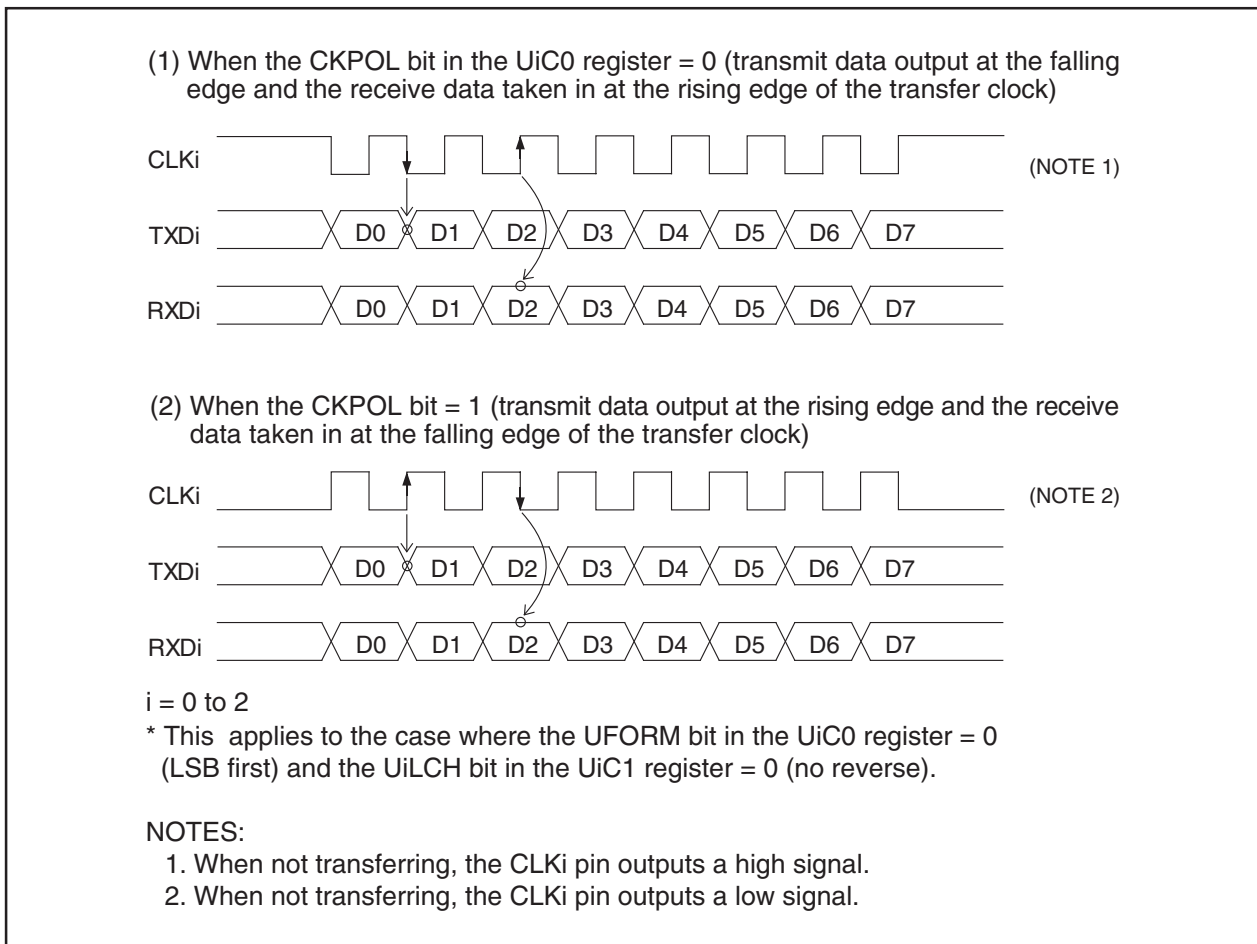


Figure 15.12 Transfer Clock Polarity

15.1.1.3 LSB First/MSB First Select Function

Use the UFORM bit in the UiC0 register ($i = 0$ to 2) to select the transfer format.

Figure 15.13 shows the Transfer Format.

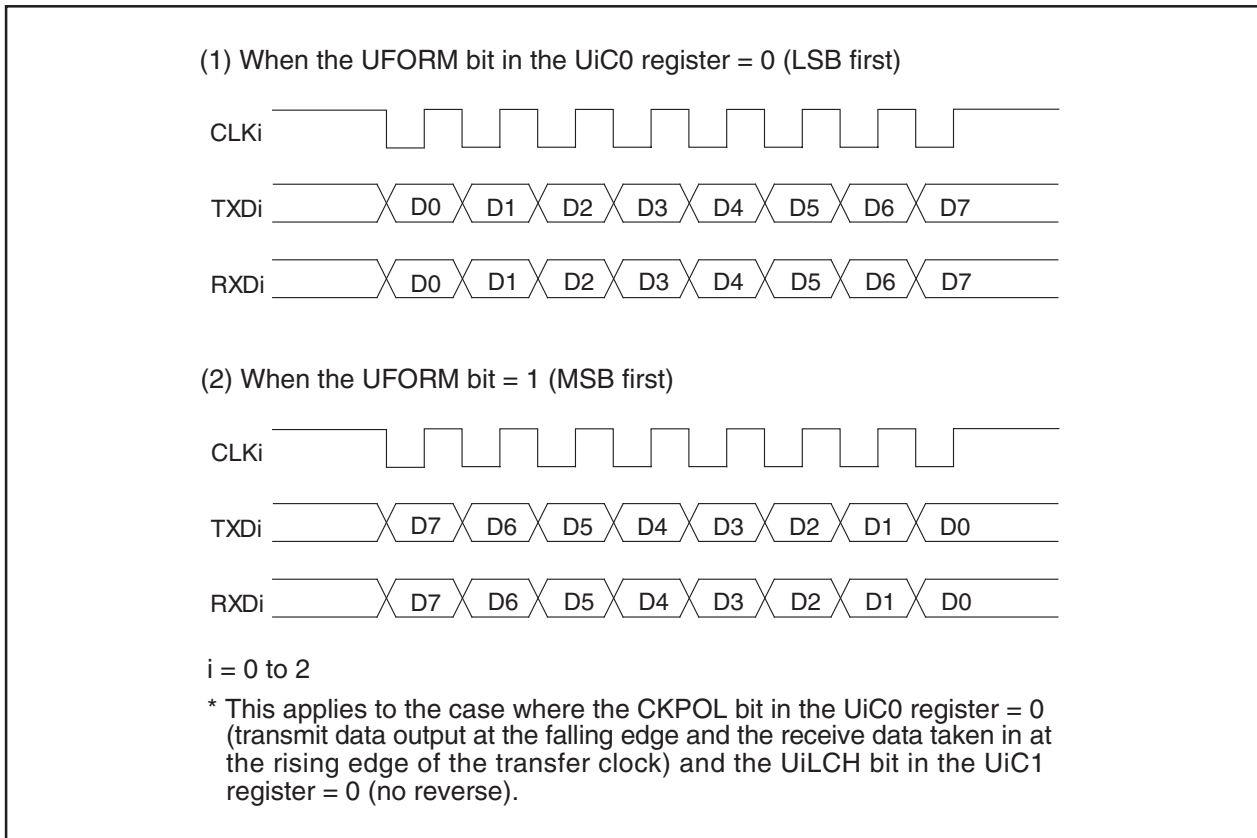


Figure 15.13 Transfer Format

15.1.1.4 Continuous Receive Mode

In continuous receive mode, receive operation becomes enable when the receive buffer register is read. It is not necessary to write dummy data into the transmit buffer register to enable receive operation in this mode. However, a dummy read of the receive buffer register is required when starting the operating mode.

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

15.1.1.5 Serial Data Logic Switching Function

When the UiLCH bit in the UiC1 register ($i = 0$ to 2) = 1 (reverse), the data written to the UiTB register has its logic reversed before being transmitted. Similarly, the receive data has its logic reversed when read from the UiRB register. Figure 15.14 shows the Serial Data Logic Switching.

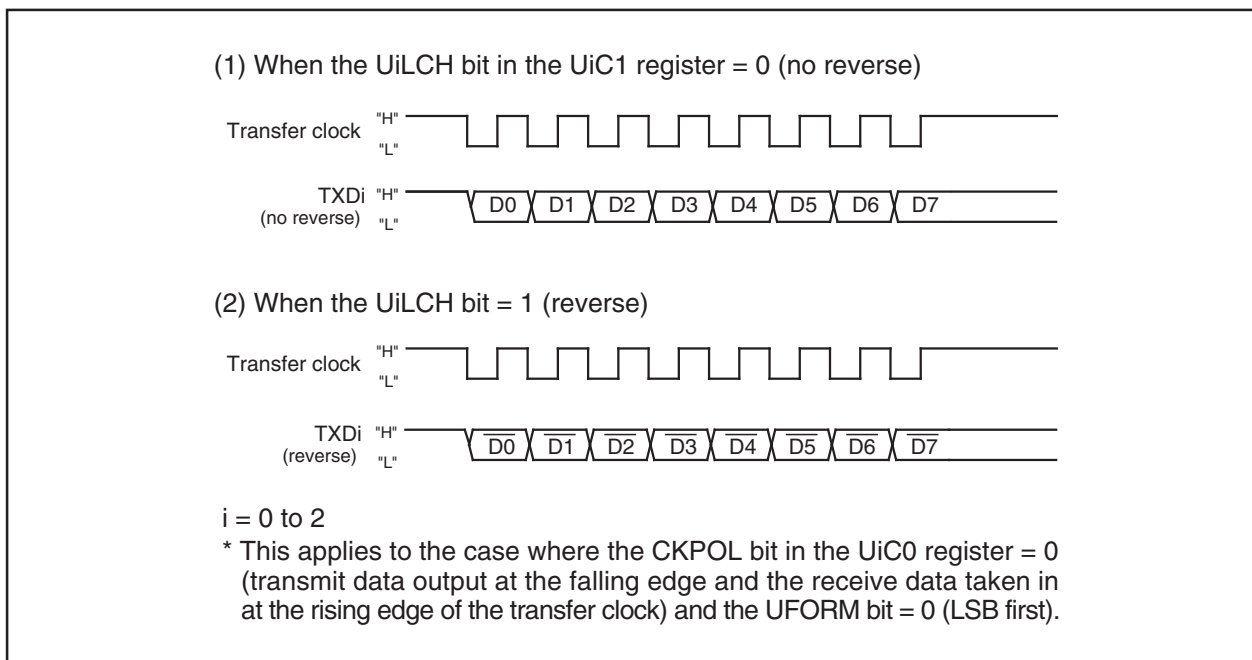


Figure 15.14 Serial Data Logic Switching

15.1.1.6 Transfer Clock Output From Multiple Pins (UART1)

Use bits CLKMD1 to CLKMD0 in the UCON register to select one of the two transfer clock output pins. Figure 15.15 shows the Transfer Clock Output from Multiple Pins. This function can be used when the selected transfer clock for UART1 is an internal clock.

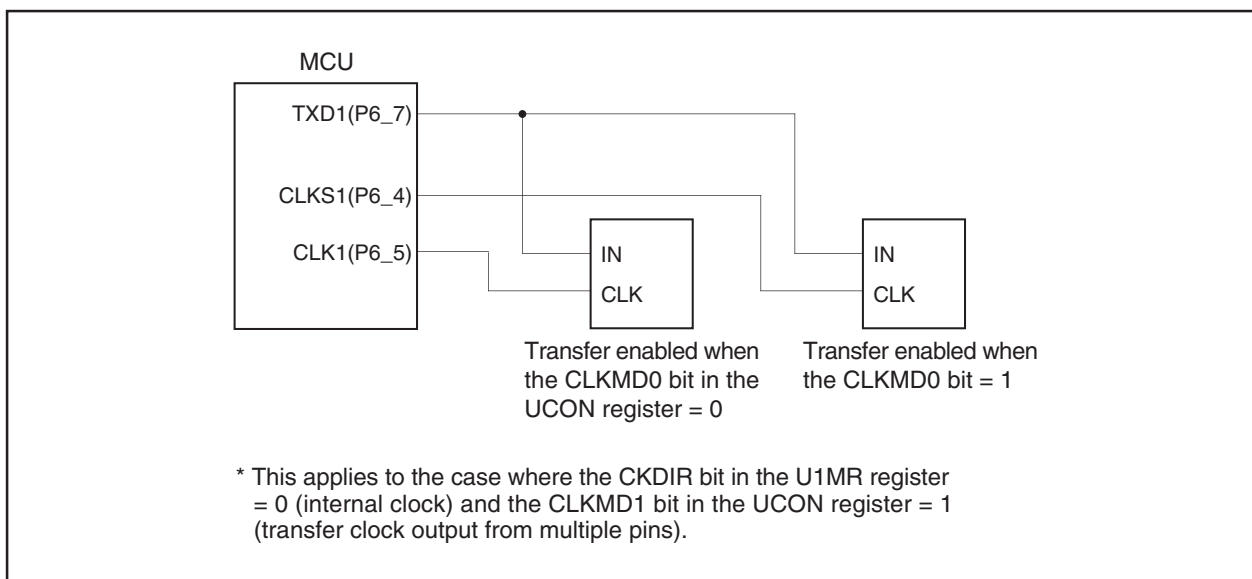


Figure 15.15 Transfer Clock Output from Multiple Pins

15.1.1.7 $\overline{\text{CTS}}/\overline{\text{RTS}}$ Function

When the $\overline{\text{CTS}}$ function is used transmit and receive operation start when “L” is applied to the $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$ ($i = 0$ to 2) pin. Transmit and receive operation begins when the $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$ pin is held “L”. If the “L” signal is switched to “H” during a transmit or receive operation, the operation stops before the next data.

When the $\overline{\text{RTS}}$ function is used, the $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$ pin outputs on “L” signal when the MCU is ready to receive. The output level becomes “H” on the first falling edge of the CLK_i pin.

- CRD bit in UiC0 register = 1 ($\overline{\text{CTS}}/\overline{\text{RTS}}$ function disabled) $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$ pin is programmable I/O function
- CRD bit = 0, CRS bit in UiC0 register = 0 ($\overline{\text{CTS}}$ function is selected) $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$ pin is $\overline{\text{CTS}}$ function
- CRD bit = 0, CRS bit = 1 ($\overline{\text{RTS}}$ function is selected) $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$ pin is $\overline{\text{RTS}}$ function

15.1.1.8 $\overline{\text{CTS}}/\overline{\text{RTS}}$ Separate Function (UART0)

This function separates $\overline{\text{CTS}}_0/\overline{\text{RTS}}_0$, outputs $\overline{\text{RTS}}_0$ from the P6_0 pin, and accepts as input the $\overline{\text{CTS}}_0$ from the P6_4 pin. To use this function, set the register bits as shown below.

- CRD bit in U0C0 register = 0 ($\overline{\text{CTS}}/\overline{\text{RTS}}$ of UART0 enabled)
- CRS bit in U0C0 register = 1 (output $\overline{\text{RTS}}$ of UART0)
- CRD bit in U1C0 register = 0 ($\overline{\text{CTS}}/\overline{\text{RTS}}$ of UART1 enabled)
- CRS bit in U1C0 register = 0 (input $\overline{\text{CTS}}$ of UART1)
- RCSP bit in UCON register = 1 (input $\overline{\text{CTS}}_0$ from the P6_4 pin)
- CLKMD1 bit in UCON register = 0 (CLKS1 not used)

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

Figure 15.16 shows the $\overline{\text{CTS}}/\overline{\text{RTS}}$ Separate Function.

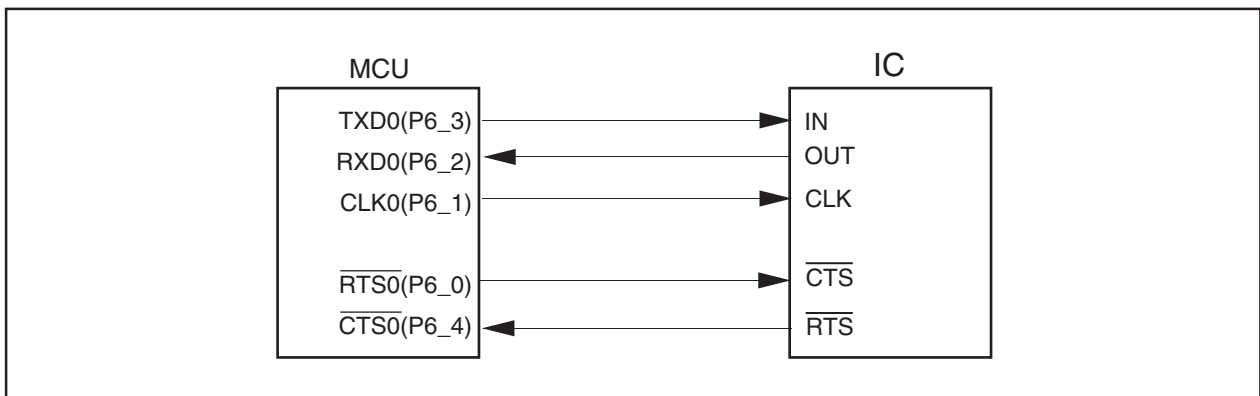


Figure 15.16 $\overline{\text{CTS}}/\overline{\text{RTS}}$ Separate Function

15.1.2 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 15.5 lists the UART Mode Specifications. Table 15.6 lists the Registers to be Used and Setting in UART Mode.

Table 15.5 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"> • CKDIR bit in UiMR register = 0 (internal clock) : $f_j/(16(n+1))$ $f_j = f1SIO, f2SIO, f8SIO, f32SIO$. n: Setting value of the UiBRG register 00h to FFh • The CKDIR bit = 1 (external clock) : $fEXT/(16(n+1))$ $fEXT$: Input from CLKi pin. n: Setting value of the UiBRG register 00h to FFh |
| Transmit/receive control | Selectable from CTS function, RTS function or CTS/RTS function disabled |
| Transmit start condition | <p>Before transmission can start, meet the following requirements</p> <ul style="list-style-type: none"> • The TE bit in the UiC1 register = 1 (transmission enabled) • The TI bit in the UiC1 register = 0 (data present in UiTB register) • If CTS function is selected, input on the CTSi pin = L |
| Receive start condition | <p>Before reception can start, meet the following requirements</p> <ul style="list-style-type: none"> • The RE bit in the UiC1 register = 1 (reception enabled) • Start bit detection |
| Interrupt request generation timing | <p>For transmission, one of the following conditions can be selected</p> <ul style="list-style-type: none"> • The UiIRS bit ⁽¹⁾ = 0 (transmit buffer empty): when transferring data from the UiTB register to the UARTi transmit register (at start of transmission) • The UiIRS bit = 1 (transmission completed): when the serial interface finished transmitting data from the UARTi transmit register <p>For reception</p> <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 interface started receiving the next data before reading the UiRB register and received the bit one before the last stop bit of the next data • Framing error ⁽³⁾ This error occurs when the number of stop bits set is not detected • Parity error ⁽³⁾ This error occurs when if parity is enabled, the number of 1's in parity and character bits does not match the number of 1's set • Error sum flag This flag is set to 1 when any of the overrun, framing, or parity errors occur |
| Select function | <ul style="list-style-type: none"> • LSB first, MSB first selection Whether to start transmitting or receiving data begins with bit 0 or begins with bit 7 can be selected • Serial data logic switch This function reverses the logic of the transmit/receive data. The start and stop bits are not reversed. • TXD, RXD I/O polarity switch 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 |

i = 0 to 2

NOTES:

1. Bits U0IRS and U1IRS are bits 0 and 1 in the UCON register. The U2IRS bit is bit 4 in the U2C1 register.
2. If an overrun error occurs, the receive data of UiRB register will be undefined. The IR bit in the SiRIC register remains unchanged.
3. The timing at which the framing error flag and the parity error flag are set is detected when data is transferred from the UARTi receive register to the UiRB register.

Table 15.6 Registers to Be Used and Settings in UART Mode

| Register | Bit | Function |
|----------|----------------------|---|
| UiTB | 0 to 8 | Set transmit data ⁽¹⁾ |
| UiRB | 0 to 8 | Receive data can be read ⁽¹⁾ |
| | OER,FER,PER,SUM | Error flag |
| UiBRG | 0 to 7 | Set a bit rate |
| UiMR | SMD2 to SMD0 | Set these bits to 100b when transfer data is 7-bit long Set these bits to 101b when transfer data is 8-bit long Set these bits to 110b when transfer data is 9-bit long |
| | CKDIR | Select the internal clock or external clock |
| | STPS | Select the stop bit |
| | PRY, PRYE | Select whether parity is included and whether odd or even |
| | IOPOL | Select the TXD/RXD input/output polarity |
| UiC0 | CLK0 to CLK1 | Select the count source for the UiBRG register |
| | CRS | Select CTS or $\overline{\text{RTS}}$ to use |
| | TXEPT | Transmit register empty flag |
| | CRD | Select CTS/ $\overline{\text{RTS}}$ function enabled or disabled |
| | NCH | Select TXDi pin output mode |
| | CKPOL | Set to 0 |
| | UFORM | LSB first or MSB first can be selected when transfer data is 8-bit long. Set this bit to 0 when transfer data is 7- or 9-bit 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 UART2 transmit interrupt source |
| | U2RRM ⁽²⁾ | Set to 0 |
| | UiLCH | Set this bit to 1 to use inverted data logic |
| | UiERE | 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 UART0/UART1 transmit interrupt source |
| | U0RRM, U1RRM | Set to 0 |
| | CLKMD0 | Invalid because the CLKMD1 bit = 0 |
| | CLKMD1 | Set to 0 |
| | RCSP | Set this bit to 1 to accept as input the $\overline{\text{CTS0}}$ of UART0 signal from the P6_4 pin |
| | 7 | Set to 0 |

i = 0 to 2

NOTES:

- The bits used for transmit/receive data are as follows:
 - Bits 0 to 6 when transfer data is 7-bit long
 - Bits 0 to 7 when transfer data is 8-bit long
 - Bits 0 to 8 when transfer data is 9-bit long.
- Set bits 4 to 5 in registers UOC1 and U1C1 to 0. Bits U0IRS, U1IRS, U0RRM, and U1RRM are included in the UCON register.

Table 15.7 lists the I/O Pins Functions in UART mode. Table 15.8 lists the P6_4 Pin Functions in UART mode. Note that for a period from when the UART_i operating mode is selected to when transfer starts, the TXD_i pin outputs an “H”.

Figure 15.17 shows the Transmit Operation in UART mode. Figure 15.18 shows the Receive Operation in UART mode.

Table 15.7 I/O Pin Functions

| Pin Name | Function | Method of Selection |
|--|----------------------|---|
| TXD _i (P6_3, P6_7, P7_0) | Serial data output | (Outputs “H” when performing reception only) |
| RXD _i (P6_2, P6_6, P7_1) | Serial data input | Bits PD6_2 and PD6_6 in PD6 register = 0 PD7_1 bit in PD7 register = 0 (Can be used as an input port when performing transmission only) |
| CLK _i (P6_1, P6_5, P7_2) | I/O port | CKDIR bit in UiMR register = 0 |
| | Transfer clock input | CKDIR bit in UiMR register = 1 Bits PD6_1 and PD6_5 in PD6 register = 0 PD7_2 bit in PD7 register = 0 |
| CTS _i /RTS _i (P6_0, P6_4, P7_3) | CTS input | CRD bit in UiC0 register = 0 CRS bit in UiC0 register = 0 Bits PD6_0 and PD6_4 in PD6 register = 0 PD7_3 bit in PD7 register = 0 |
| | | RTS output |
| | I/O port | CRD bit = 1 |

i = 0 to 2

Table 15.8 P6_4 Pin Functions

| Pin Function | Bit set Value | | | | |
|---------------------|---------------|---------|---------------|------------|---------------------|
| | U1C0 Register | | UCON Register | | PD6 Register |
| | CRD bit | CRS bit | RCSP bit | CLKMD1 bit | PD6_4 bit |
| P6_4 | 1 | - | 0 | 0 | Input: 0, Output: 1 |
| CTS1 | 0 | 0 | 0 | 0 | 0 |
| RTS1 | 0 | 1 | 0 | 0 | - |
| CTS0 ⁽¹⁾ | 0 | 0 | 1 | 0 | 0 |

-: 0 or 1

NOTE:

- In addition to this, set the CRD bit in the U0C0 register to 0 ($\overline{\text{CTS0}}/\overline{\text{RTS0}}$ enabled) and the CRS bit in the U0C0 register to 1 ($\overline{\text{RTS0}}$ selected).

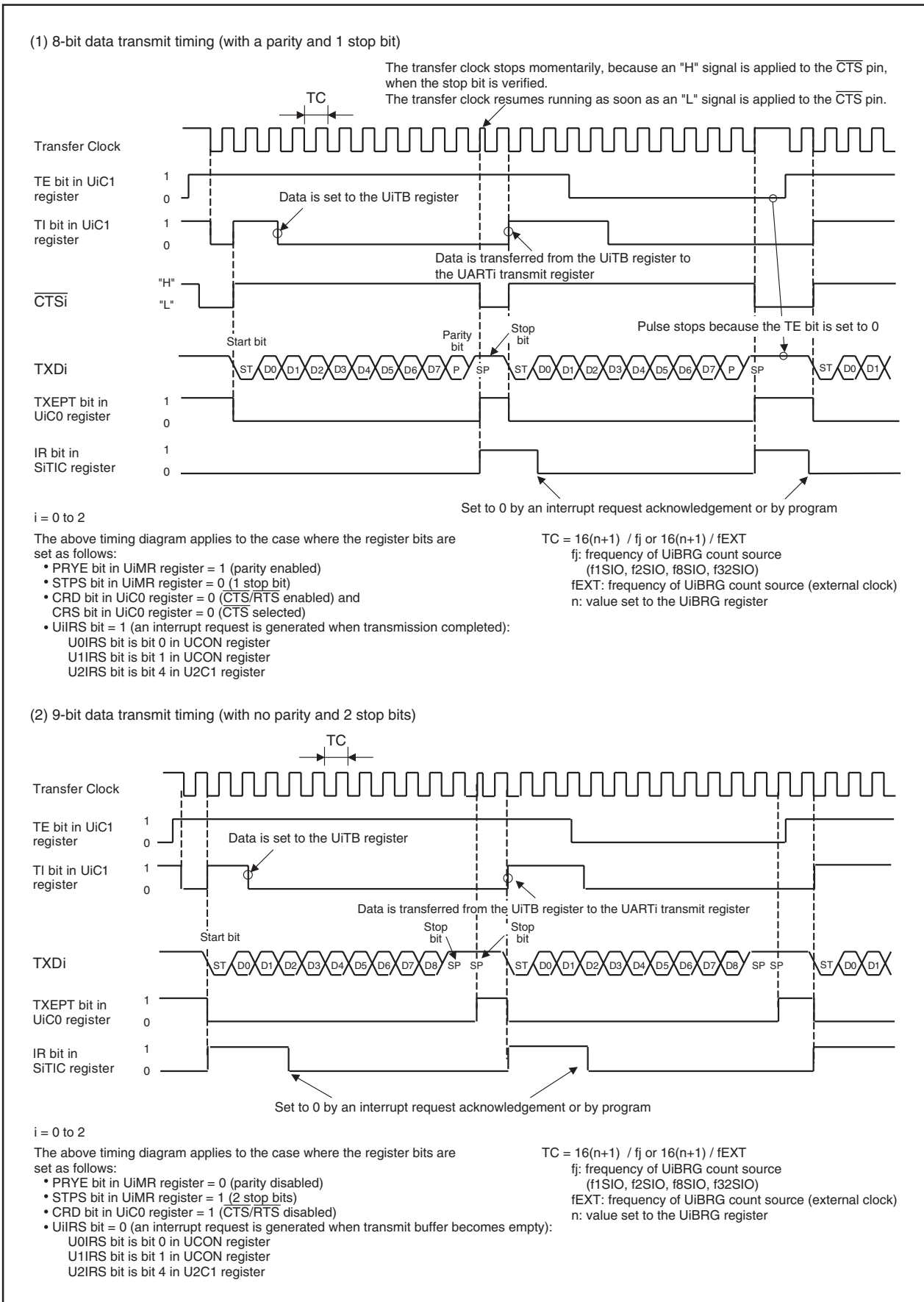


Figure 15.17 Transmit Operation

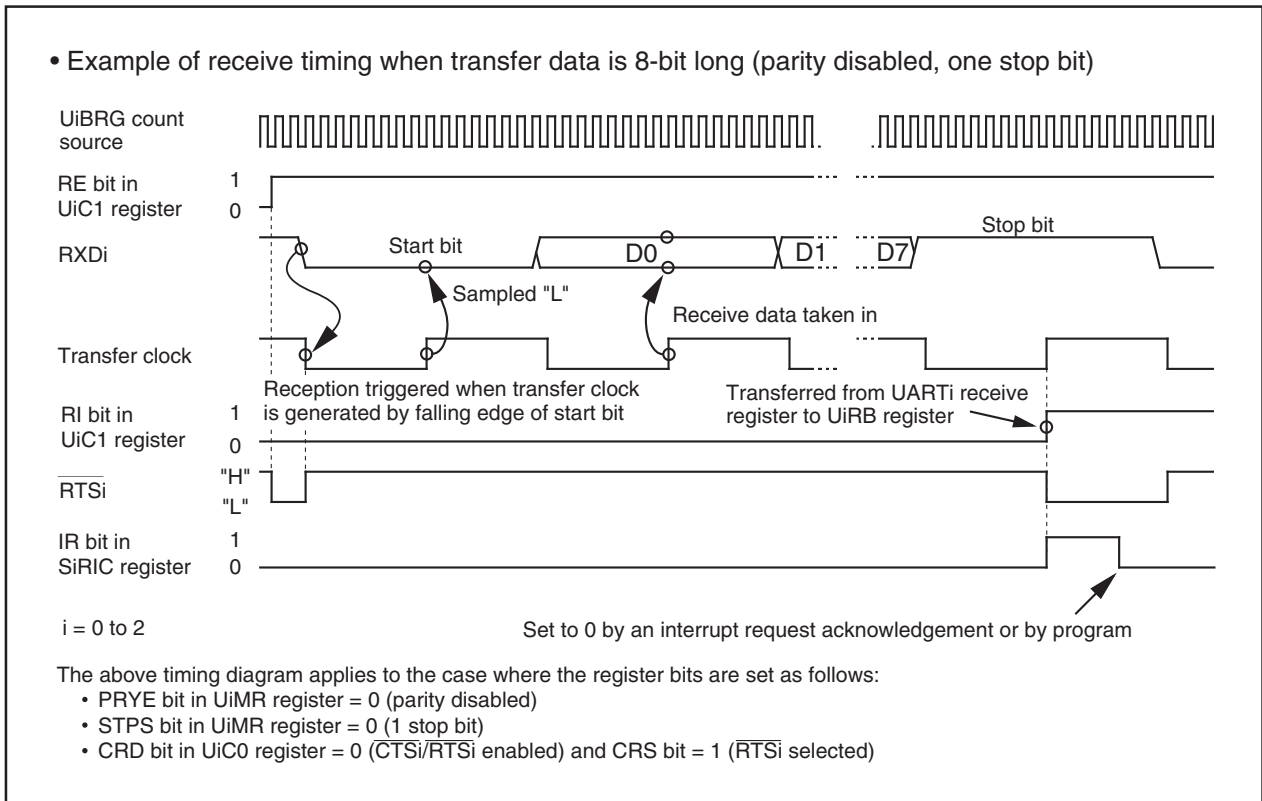


Figure 15.18 Receive Operation

15.1.2.1 Bit Rates

In UART mode, the frequency set by the UiBRG register ($i = 0$ to 2) divided by 16 become the bit rates. Table 15.9 lists an Example of Bit Rates and Settings.

Table 15.9 Example of Bit Rates and Settings

| Bit Rate (bps) | Count Source of UiBRG | Peripheral Function Clock: 16 MHz | | Peripheral Function Clock: 20 MHz | | Peripheral Function Clock: 24 MHz ⁽¹⁾ | |
|----------------|-----------------------|-----------------------------------|----------------|-----------------------------------|----------------|--|----------------|
| | | Set Value of UiBRG: n | Bit Rate (bps) | Set Value of UiBRG: n | Bit Rate (bps) | Set Value of UiBRG: n | Bit Rate (bps) |
| 1200 | f8 | 103 (67h) | 1202 | 129 (81h) | 1202 | 155 (9Bh) | 1202 |
| 2400 | f8 | 51 (33h) | 2404 | 64 (40h) | 2404 | 77 (4Dh) | 2404 |
| 4800 | f8 | 25 (19h) | 4808 | 32 (20h) | 4735 | 38 (26h) | 4808 |
| 9600 | f1 | 103 (67h) | 9615 | 129 (81h) | 9615 | 155 (9Bh) | 9615 |
| 14400 | f1 | 68 (44h) | 14493 | 86 (56h) | 14368 | 103 (67h) | 14423 |
| 19200 | f1 | 51 (33h) | 19231 | 64 (40h) | 19231 | 77 (4Dh) | 19231 |
| 28800 | f1 | 34 (22h) | 28571 | 42 (2Ah) | 29070 | 51 (33h) | 28846 |
| 31250 | f1 | 31 (1Fh) | 31250 | 39 (27h) | 31250 | 47 (2Fh) | 31250 |
| 38400 | f1 | 25 (19h) | 38462 | 32 (20h) | 37879 | 38 (26h) | 38462 |
| 51200 | f1 | 19 (13h) | 50000 | 23 (17h) | 52083 | 28 (1Ch) | 51724 |

$i = 0$ to 2

NOTE:

1. 24 MHz is available Normal-ver. only.

15.1.2.2 Counter Measure for Communication Error Occurs

If a communication error occurs while transmitting or receiving in UART 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 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 000b (serial interface disabled)
 - (2) Set bits SMD2 to SMD0 in the UiMR register to 001b, 101b, 110b
 - (3) 1 (transmission enabled) is written to the TE bit in the UiC1 register, regardless of the TE bit

15.1.2.3 LSB First/MSB First Select Function

As shown in Figure 15.19, use the UFORM bit in the UiC0 register to select the transfer format.

Figure 15.19 shows the Transfer Format. This function is valid when transfer data is 8-bit long.

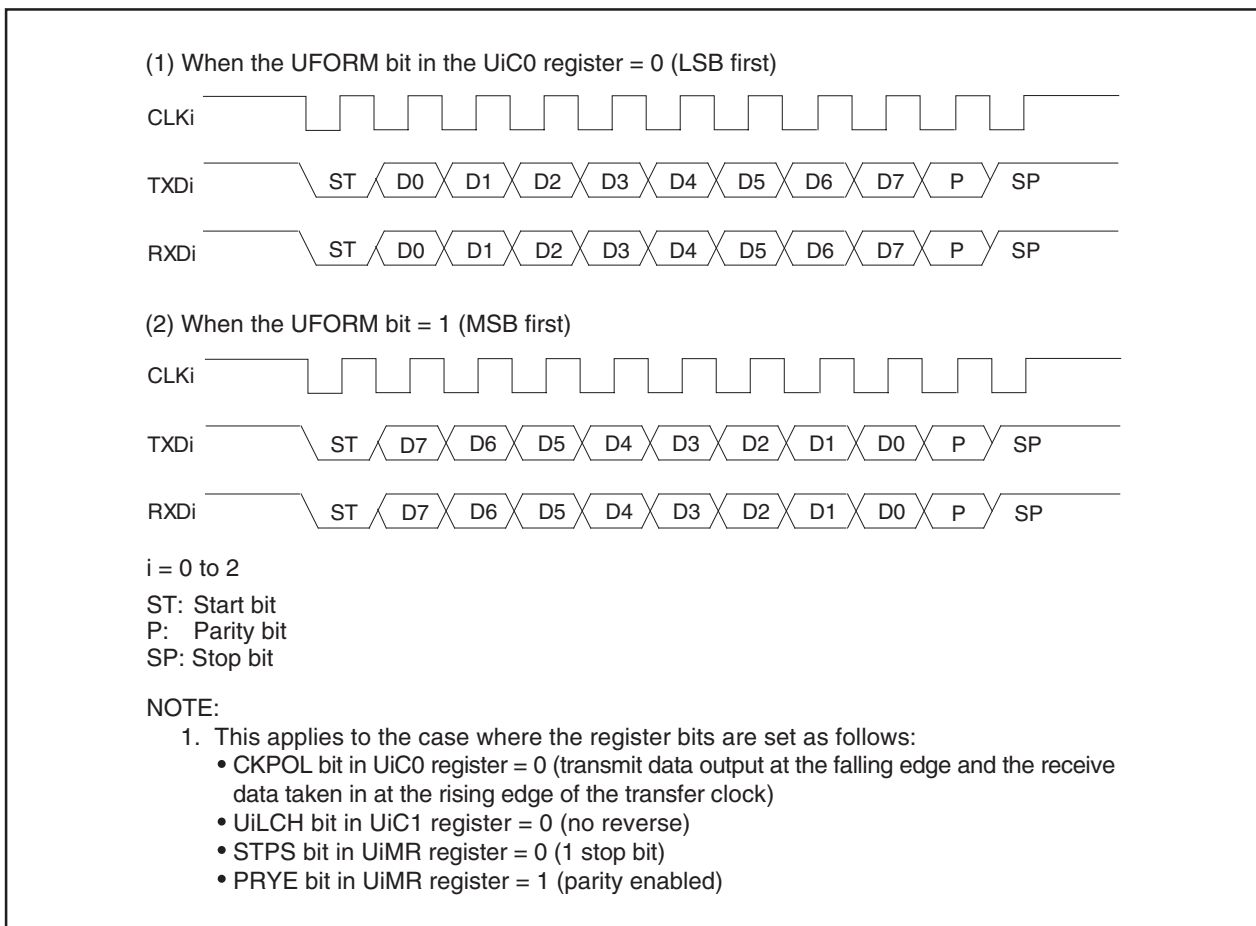


Figure 15.19 Transfer Format

15.1.2.4 Serial Data Logic Switching Function

The data written to the UiTB register has its logic reversed before being transmitted. Similarly, the received data has its logic reversed when read from the UiRB register.

Figure 15.20 shows the Serial Data Logic Switching.

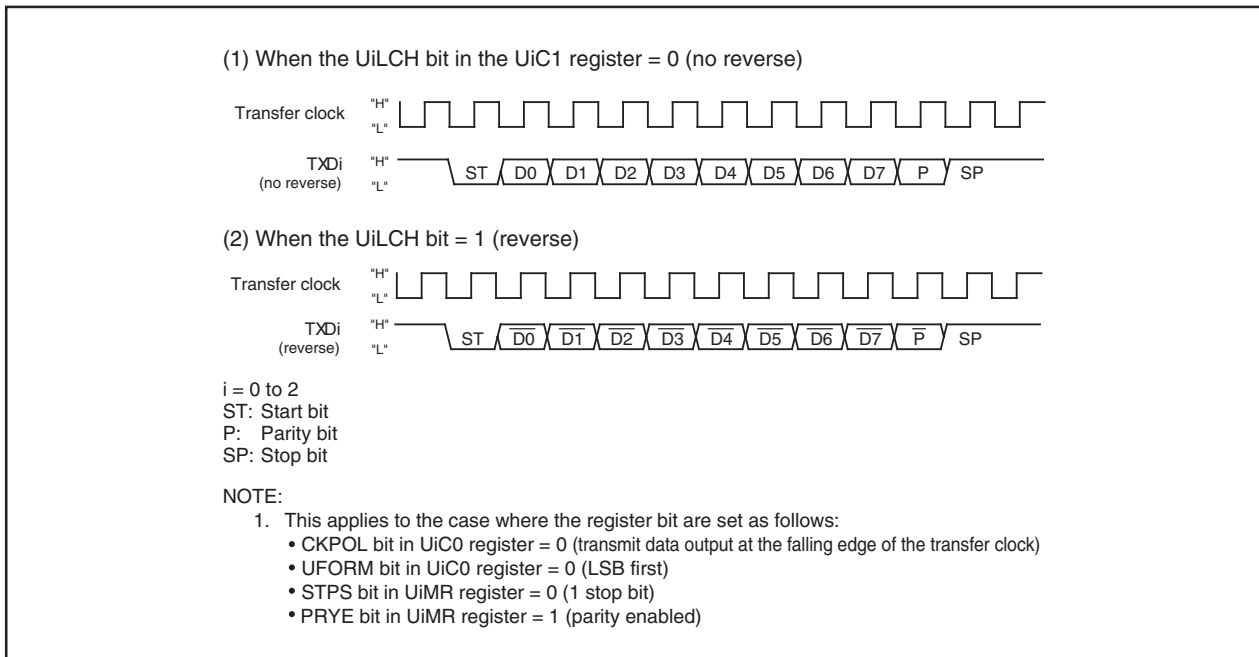


Figure 15.20 Serial Data Logic Switching

15.1.2.5 TXD and RXD I/O Polarity Inverse Function

This function inverses the polarities of the TXDi pin output and RXDi pin input. The logic levels of all input/output data (including the start, stop and parity bits) are inverted.

Figure 15.21 shows the TXD and RXD I/O Polarity Inverse.

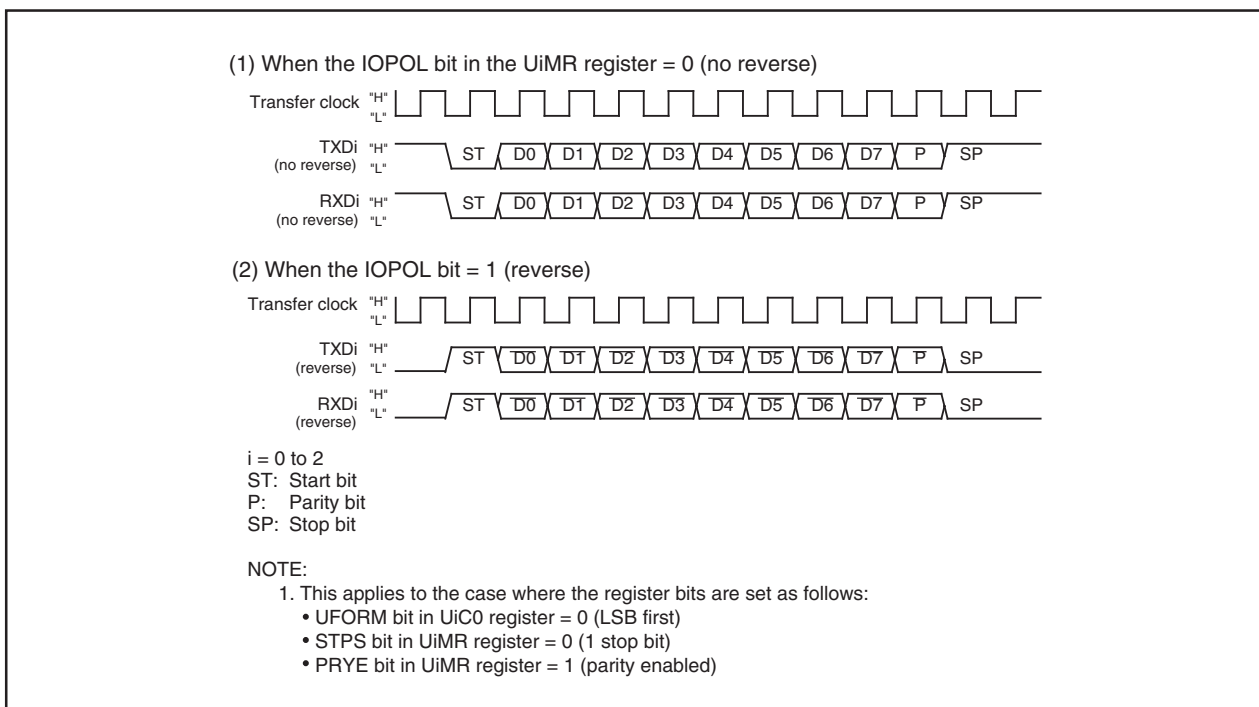


Figure 15.21 TXD and RXD I/O Polarity Inverse

15.1.2.6 $\overline{\text{CTS}}/\overline{\text{RTS}}$ Function

When the $\overline{\text{CTS}}$ function is used transmit operation start when “L” is applied to the $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$ ($i = 0$ to 2) pin. Transmit operation begins when the $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$ pin is held “L”. If the “L” signal is switched to “H” during a transmit operation, the operation stops before the next data.

When the $\overline{\text{RTS}}$ function is used, the $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$ pin outputs on “L” signal when the MCU is ready to receive. The output level becomes “H” on the first falling edge of the CLK_i pin.

- CRD bit in UiC0 register = 1 ($\overline{\text{CTS}}/\overline{\text{RTS}}$ function of UART0 disabled) $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$ pin is programmable I/O function
- CRD bit = 0, CRS bit in UiC0 register = 0 ($\overline{\text{CTS}}$ function is selected) $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$ pin is $\overline{\text{CTS}}$ function
- CRD bit = 0, CRS bit = 1 ($\overline{\text{RTS}}$ function is selected) $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$ pin is $\overline{\text{RTS}}$ function

15.1.2.7 $\overline{\text{CTS}}/\overline{\text{RTS}}$ Separate Function (UART0)

This function separates $\overline{\text{CTS}}_0/\overline{\text{RTS}}_0$, outputs $\overline{\text{RTS}}_0$ from the P6_0 pin, and accepts as input the $\overline{\text{CTS}}_0$ from the P6_4 pin. To use this function, set the register bits as shown below.

- CRD bit in U0C0 register = 0 ($\overline{\text{CTS}}/\overline{\text{RTS}}$ of UART0 enabled)
- CRS bit in U0C0 register = 1 (output $\overline{\text{RTS}}$ of UART0)
- CRD bit in U1C0 register = 0 ($\overline{\text{CTS}}/\overline{\text{RTS}}$ of UART1 enabled)
- CRS bit in U1C0 register = 0 (input $\overline{\text{CTS}}$ of UART1)
- RCSP bit in UCON register = 1 (input $\overline{\text{CTS}}_0$ from the P6_4 pin)
- CLKMD1 bit in UCON register = 0 (CLKS1 not used)

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

Figure 15.22 shows $\overline{\text{CTS}}/\overline{\text{RTS}}$ separate function usage.

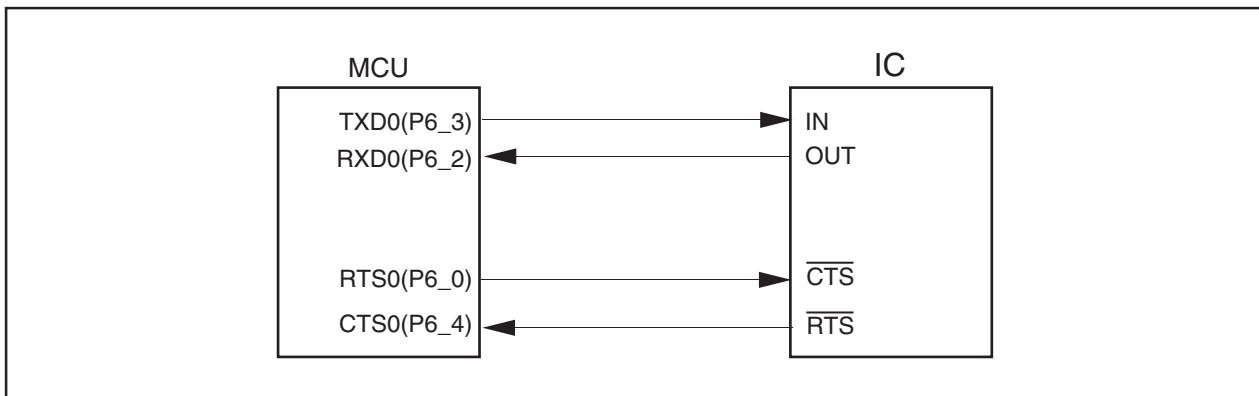


Figure 15.22 $\overline{\text{CTS}}/\overline{\text{RTS}}$ Separate Function

15.1.3 Special Mode 1 (I²C Mode)

I²C mode is provided for use as a simplified I²C interface compatible mode. Table 15.10 lists the I²C Mode Specifications. Figure 15.23 shows the I²C Mode Block Diagram. Table 15.11 lists the Registers to be Used and Setting in I²C Mode. Table 15.12 lists the I²C Mode Functions. Figure 15.24 shows the Transfer to UiRB Register and Interrupt Timing.

As shown in Table 15.12, the MCU is placed in I²C mode by setting bits SMD2 to SMD0 to 010b and the IICM bit to 1. Because SDAi transmit output has a delay circuit attached, SDAi output does not change state until SCLi goes low and remains stably low.

Table 15.10 I²C 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 UiMR register = 0 (internal clock) : $f_j/(2(n+1))$ $f_j = f1SIO, f2SIO, f8SIO, f32SIO$. n: Setting value of the UiBRG register 00h to FFh • During slave The CKDIR bit = 1 (external clock) : Input from SCLi pin |
| Transmit start condition | Before transmission can start, meet the following requirements ⁽¹⁾ <ul style="list-style-type: none"> • The TE bit in the UiC1 register = 1 (transmission enabled) • The TI bit in the UiC1 register = 0 (data present in the UiTB register) |
| Receive start condition | Before reception can start, meet the following requirements ⁽¹⁾ <ul style="list-style-type: none"> • The RE bit in the UiC1 register = 1 (reception enabled) • The TE bit in the UiC1 register = 1 (transmission enabled) • The TI bit in the UiC1 register = 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 | Overrun error ⁽²⁾ This error occurs if the serial I/O started receiving the next data before reading the UiRB register and received the 8th bit of the next data |
| Select function | <ul style="list-style-type: none"> • Arbitration lost Timing at which the ABT bit in the UiRB register is updated can be selected • SDAi digital delay No digital delay or a delay of 2 to 8 UiBRG count source clock cycles selectable • Clock phase setting With or without clock delay selectable |

i = 0 to 2

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, the value of UiRB register will be undefined. The IR bit in the SiRIC register remains unchanged.

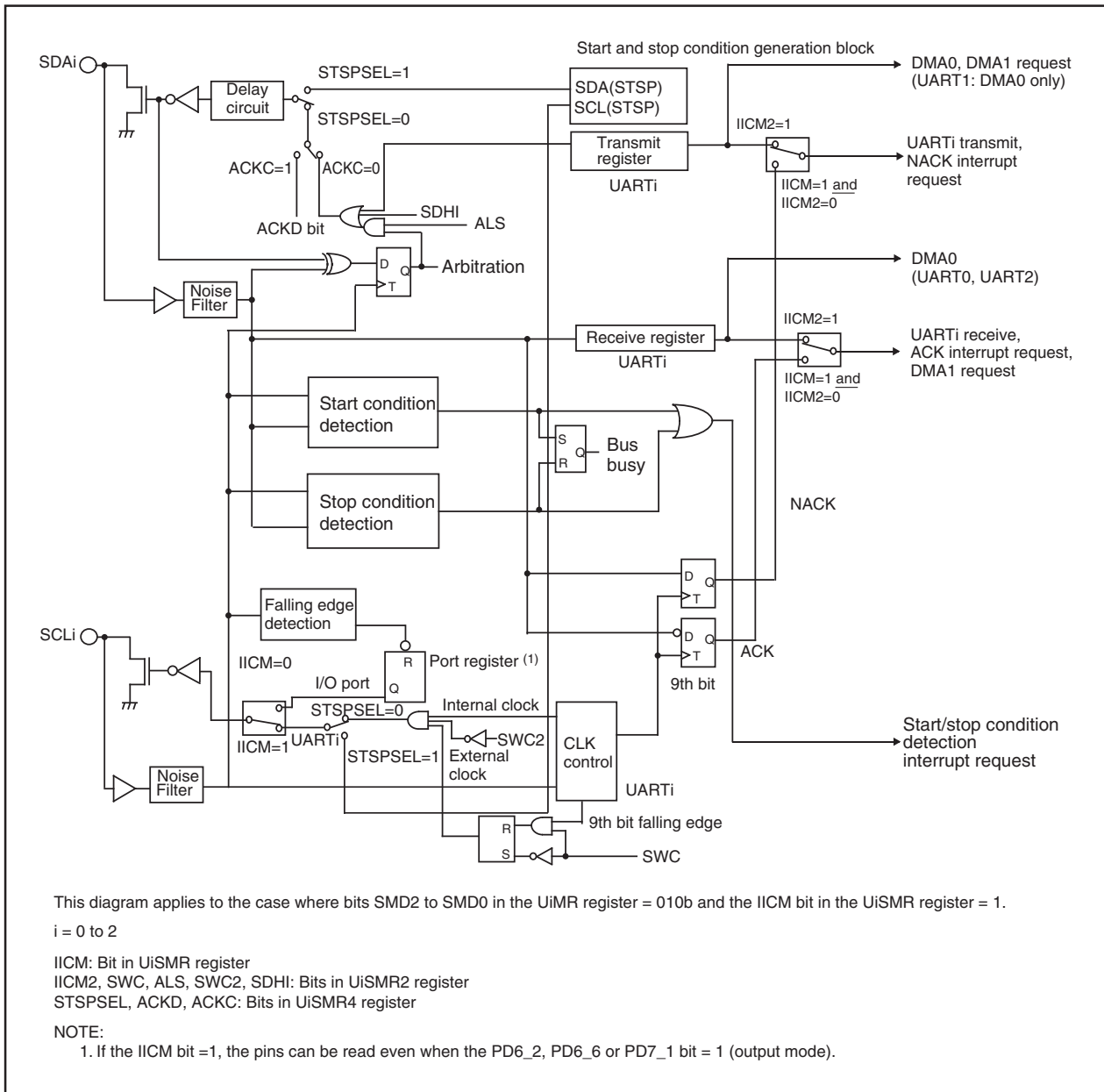


Figure 15.23 I²C Mode Block Diagram

Table 15.11 Registers to Be Used and Settings in I²C Mode

| Register | Bit | Function | |
|----------|----------------------------|--|---|
| | | Master | Slave |
| UiTB (1) | 0 to 7 | Set transmit data | |
| UiRB (1) | 0 to 7 | Receive data can be read | |
| | 8 | ACK or NACK is set in this bit | |
| | ABT | Arbitration lost detection flag | Invalid |
| | OER | Overrun error flag | |
| UiBRG | 0 to 7 | Set a bit rate | Invalid |
| UiMR (1) | SMD2 to SMD0 | Set to 010b | |
| | CKDIR | Set to 0 | Set to 1 |
| | IOPOL | Set to 0 | |
| UiC0 | CLK1 to CLK0 | Select the count source for the UiBRG register | Invalid |
| | CRS | Invalid because the CRD bit = 1 | |
| | TXEPT | Transmit register empty flag | |
| | CRD (3) | Set to 1 | |
| | NCH | Set to 1 | |
| | CKPOL | Set to 0 | |
| | UFORM | Set to 1 | |
| 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) | Invalid | |
| | U2RRM (2), UiLCH, UiERE | Set to 0 | |
| UiSMR | IICM | Set to 1 | |
| | ABC | Select the timing at which arbitration-lost is detected | Invalid |
| | BBS | Bus busy flag | |
| | 3 to 7 | Set to 0 | |
| UiSMR2 | IICM2 | See Table 15.12 I²C Mode Functions | |
| | CSC | Set this bit to 1 to enable clock synchronization | Set to 0 |
| | SWC | Set this bit to 1 to have SCLi output fixed to "L" at the falling edge of the 9th bit of clock | |
| | ALS | Set this bit to 1 to have SDAi output stopped when arbitration-lost is detected | Set to 0 |
| | STAC | Set to 0 | Set this bit to 1 to initialize UARTi at start condition detection |
| | SWC2 | Set this bit to 1 to have SCLi output forcibly pulled low | |
| | SDHI | Set this bit to 1 to disable SDAi output | |
| | 7 | Set to 0 | |
| UiSMR3 | 0, 2, 4, and NODC | Set to 0 | |
| | CKPH | See Table 15.12 I²C Mode Functions | |
| | DL2 to DL0 | Set the amount of SDAi digital delay | |
| UiSMR4 | 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 | |
| | ACKC | Set this bit to 1 to output ACK data | |
| | SCLHI | Set this bit to 1 to have SCLi output stopped when stop condition is detected | Set to 0 |
| | SWC9 | Set to 0 | Set this bit to 1 to set the SCLi to "L" hold at the falling edge of the 9th bit of clock |
| IFSR0 | IFSR06, ISFR07 | Set to 1 | |
| UCON | U0IRS, U1IRS | Invalid | |
| | 2 to 7 | Set to 0 | |

i = 0 to 2

NOTES:

- Not all register bits are described above. Set those bits to 0 when writing to the registers in I²C mode.
- Set bits 4 and 5 in registers U0C1 and U1C1 to 0. Bits U0IRS, U1IRS, U0RRM, and U1RRM are in the UCON register.
- When using UART1 in I²C mode and enabling the $\overline{\text{CTS}}/\text{RTS}$ separate function of UART0, set the CRD bit in the U1C0 register to 0 ($\overline{\text{CTS}}/\text{RTS}$ function enabled) and the CRS bit to 0 (CTS input).

Table 15.12 I²C Mode Functions

| Function | Clock Synchronous Serial I/O Mode (SMD2 to SMD0 = 001b, IICM = 0) | I ² C Mode (SMD2 to SMD0 = 010b, 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) |
| Source of interrupt number 6, 7, and 10 ^{(1) (5) (7)} | - | Start condition detection or stop condition detection (See Table 15.13 STSPSEL Bit Functions) | | | |
| Source of interrupt number 15, 17, and 19 ^{(1) (6)} | UARTi transmission Transmission started or completed (selected by UiIRS) | No acknowledgment detection (NACK) Rising edge of SCLi 9th bit | | UARTi transmission Rising edge of SCLi 9th bit | UARTi transmission Falling edge of SCLi next to the 9th bit |
| Source of interrupt number 16, 18, and 20 ^{(1) (6)} | UARTi reception When 8th bit received CKPOL = 0 (rising edge) CKPOL = 1 (falling edge) | Acknowledgment detection (ACK) Rising edge of SCLi 9th bit | | UARTi reception Falling edge of SCLi 9th bit | |
| Timing for transferring data from UART reception shift register to UiRB register | CKPOL = 0 (rising edge) CKPOL = 1 (falling edge) | Rising edge of SCLi 9th bit | | Falling edge of SCLi 9th bit | Falling and rising edges of SCLi 9th bit |
| UARTi transmission output delay | Not delayed | Delayed | | | |
| Functions of pins P6_3, P6_7, and P7_0 | TXDi output | SDAi input/output | | | |
| Functions of pins P6_2, P6_6, and P7_1 | RXDi input | SCLi input/output | | | |
| Functions of pins P6_1, P6_5, and P7_2 | CLKi input or output selected | - (Cannot be used in I ² C mode) | | | |
| Noise filter width | 15 ns | 200 ns | | | |
| Read RXDi and SCLi pins levels | Possible when the corresponding port direction bit = 0 | Always possible no matter how the corresponding port direction bit is set | | | |
| Initial value of TXDi and SDAi outputs | CKPOL = 0 (H) CKPOL = 1 (L) | The value set in the port register before setting I ² C mode ⁽²⁾ | | | |
| Initial and end value of SCLi | - | H | L | H | L |
| DMA1 source ⁽⁶⁾ | UARTi reception | Acknowledgment detection (ACK) | | UARTi reception Falling edge of SCLi 9th bit | |
| Store received data | 1st to 8th bits of the received data are stored into bits 7 to 0 in the UiRB register | | | 1st to 7th bits of the received data are stored into bits 6 to 0 in the UiRB register, 8th bit is stored into bit 8 in the UiRB register | 1st to 8th bits are stored into bit 7 to bit 0 in UiRB register ⁽³⁾ |
| Read received data | The UiRB register status is read | | | Bit 6 to bit 0 in the UiRB register ⁽⁴⁾ are read as bit 7 to bit 1. Bit 8 in the UiRB register is read as bit 0. | |

i = 0 to 2

NOTES:

- If the interrupt source is changed, the IR bit in the interrupt control register for the changed interrupt may inadvertently be set to 1 (interrupt requested). (Refer to **23.6 Interrupts**.)
If one of the bits shown below is changed, the interrupt source, the interrupt timing, etc. change. Therefore, always be sure to set the IR bit to 0 (interrupt not requested) after changing those bits.
 - Bits SMD2 to SMD0 in UiMR register
 - IICM2 bit in UiSMR2 register
 - IICM bit in UiSMR register
 - CKPH bit in UiSMR3 register
- Set the initial value of SDAi output while bits SMD2 to SMD0 in the UiMR register = 000b (serial interface disabled).
- Second data transfer to the UiRB register (rising edge of SCLi 9th bit)
- First data transfer to the UiRB register (falling edge of SCLi 9th bit)
- See **Figure 15.26 STSPSEL Bit Functions**.
- See **Figure 15.24 Transfer to UiRB Register and Interrupt Timing**.
- When using UART0, be sure to set the IFSR06 bit in the IFSR0 register to 1 (interrupt source: UART0 bus collision detection).
When using UART1, be sure to set the IFSR07 bit in the IFSR0 register to 1 (interrupt source: UART1 bus collision detection).

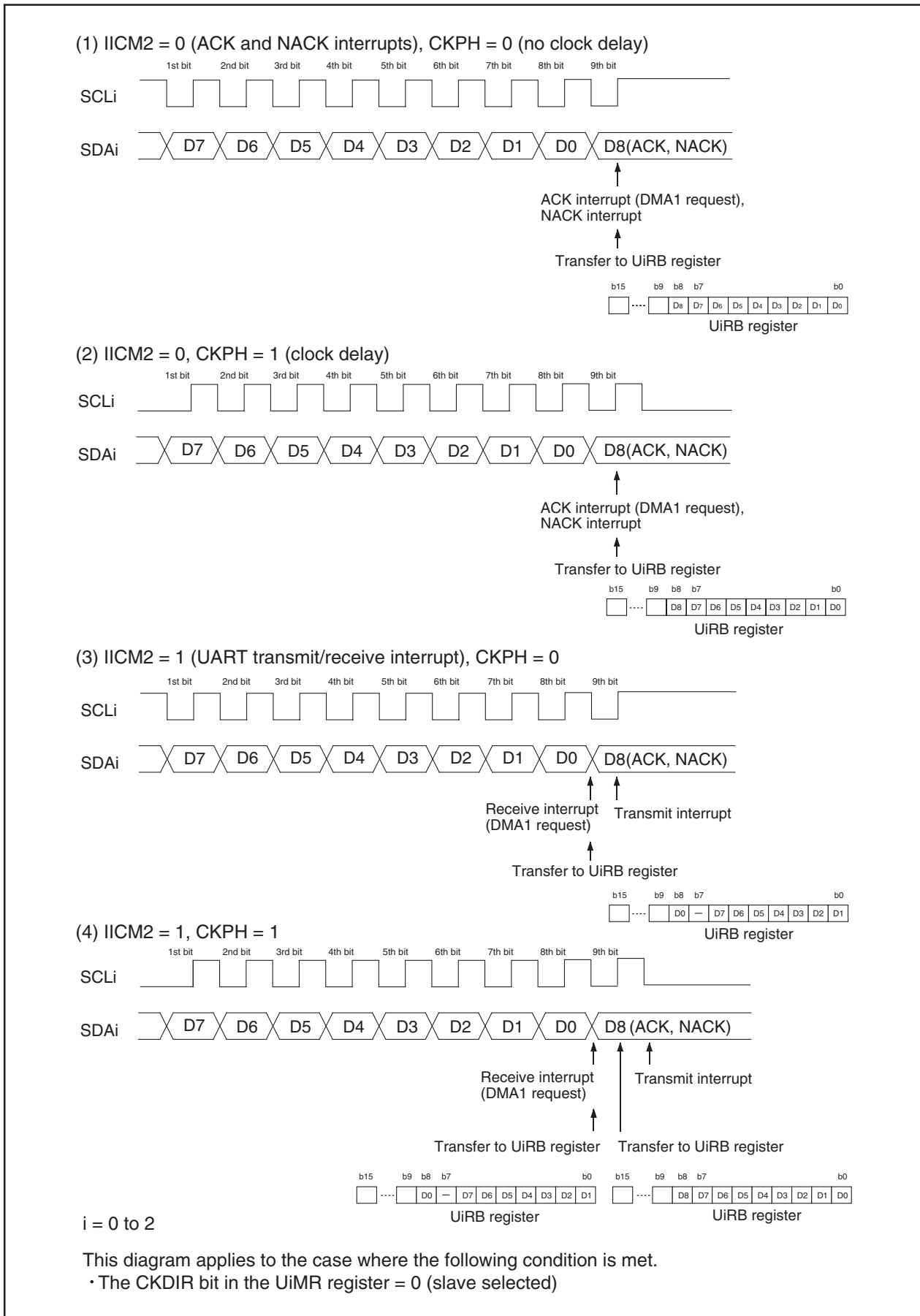


Figure 15.24 Transfer to UiRB Register and Interrupt Timing

15.1.3.1 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 SDAi pin changes state from high to low while the SCLi pin is in the high state. A stop condition-detected interrupt request is generated when the SDAi pin changes state from low to high while the SCLi pin is in the high state.

Figure 15.25 shows the Detection of Start and Stop Condition.

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

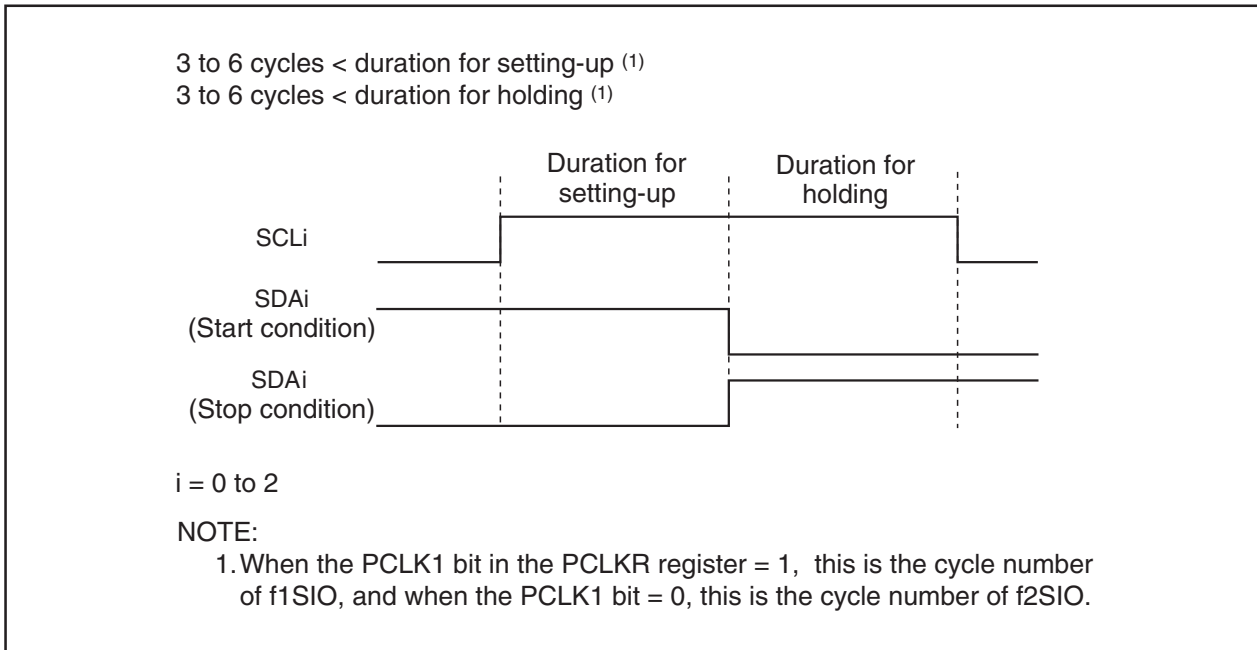


Figure 15.25 Detection of Start and Stop Condition

15.1.3.2 Output of Start and Stop Condition

A start condition is generated by setting the STAREQ bit in the UiSMR4 register (i = 0 to 2) to 1 (start).

A restart condition is generated by setting the RSTAREQ bit in the UiSMR4 register to 1 (start).

A stop condition is generated by setting the STPREQ bit in the UiSMR4 register to 1 (start).

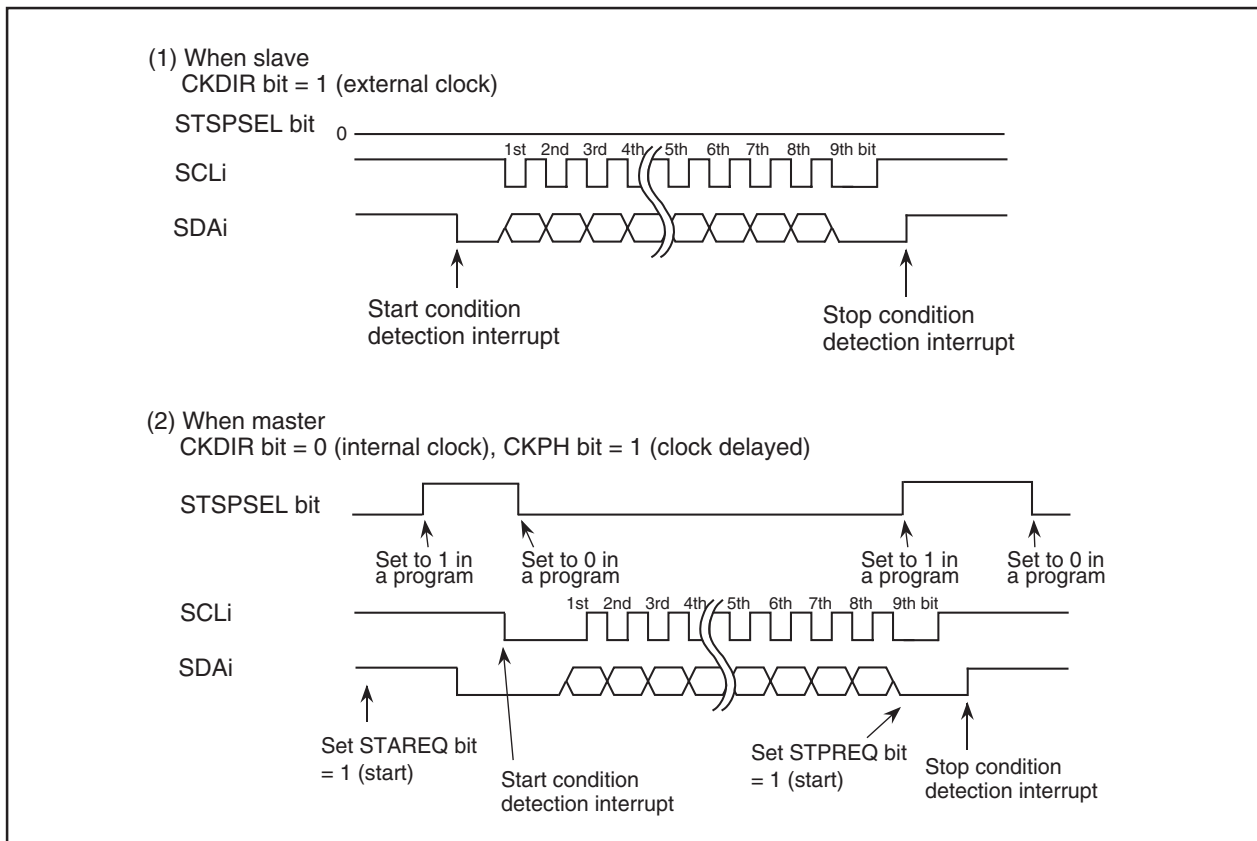
The output procedure is described below.

- (1) Set the STAREQ bit, RSTAREQ bit or STPREQ bit to 1 (start).
- (2) Set the STSPSEL bit in the UiSMR4 register to 1 (output).

Table 15.13 and Figure 15.26 show the STSPSEL Bit Functions.

Table 15.13 STSPSEL Bit Functions

| Function | STSPSEL Bit = 0 | STSPSEL Bit = 1 |
|--|--|--|
| Output of pins SCLi and SDAi | Output of transfer clock and data Output of start/stop condition is accomplished by a program using ports (not automatically generated in hardware) | Output of a start/stop condition depending on bits STAREQ, RSTAREQ, and STPREQ |
| Start/stop condition interrupt request generation timing | Start/stop condition detection | Finish generating start/stop condition |

**Figure 15.26 STSPSEL Bit Functions****15.1.3.3 Arbitration**

Unmatching of the transmit data and SDAi pin input data is checked synchronously with the rising edge of SCLi. Use the ABC bit in the UiSMR register to select the timing at which the ABT bit in the UiRB register is updated. If the ABC bit = 0 (updated per bit), the ABT bit is set to 1 at the same time unmatching is detected during check, and is set 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 per byte, set the ABT bit to 0 (undetected) after detecting acknowledge in the first byte, before transferring the next byte.

Setting the ALS bit in the UiSMR2 register to 1 (SDA output stop enabled) causes arbitration-lost to occur, in which case the SDAi pin is placed in the high-impedance state at the same time the ABT bit is set to 1 (unmatching detected).

15.1.3.4 Transfer Clock

Data is transmitted/received using a transfer clock like the one shown in Figure 15.24 Transfer to UiRB Register and Interrupt Timing.

The CSC bit in the UiSMR2 register is used to synchronize the internally generated clock (internal SCLi) and an external clock supplied to the SCLi pin. In cases when the CSC bit is set to 1 (clock synchronization enabled), if a falling edge on the SCLi pin is detected while the internal SCLi is high, the internal SCLi goes low, at which time the value of the UiBRG register is reloaded with and starts counting in the low-level interval. If the internal SCLi changes state from low to high while the SCLi pin is low, counting stops, and when the SCLi pin goes high, counting restarts.

In this way, the UARTi transfer clock is comprised of the logical product of the internal SCLi and SCLi pin signal. The transfer clock works from a half period before the falling edge of the internal SCLi 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 UiSMR2 register allows to select whether the SCLi 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 UiSMR4 register is set to 1 (enabled), SCLi output is turned off (placed in the high-impedance state) when a stop condition is detected.

Setting the SWC2 bit in the UiSMR2 register = 1 (0 output) makes it possible to forcibly output a low-level signal from the SCLi pin even while sending or receiving data. Setting the SWC2 bit to 0 (transfer clock) allows the transfer clock to be output from or supplied to the SCLi pin, instead of outputting a low-level signal. If the SWC9 bit in the UiSMR4 register is set to 1 (SCL hold low enabled) when the CKPH bit in the UiSMR3 register = 1, the SCLi pin is fixed to low-level output at the falling edge of the clock pulse next to the 9th. Setting the SWC9 bit = 0 (SCL hold low disabled) frees the SCLi pin from low-level output.

15.1.3.5 SDA Output

The data written to bits 7 to 0 (D7 to D0) in the UiTB register is sequentially output beginning with D7. The 9th bit (D8) is ACK or NACK.

The initial value of SDAi transmit output can only be set when IICM = 1 (I²C mode) and bits SMD2 to SMD0 in the UiMR register = 000b (serial interface disabled).

Bits DL2 to DL0 in the UiSMR3 register allow to add no delays or a delay of 2 to 8 UiBRG count source clock cycles to SDAi output.

Setting the SDHI bit in the UiSMR2 register = 1 (SDA output disabled) forcibly places the SDAi pin in the high-impedance state. Do not write to the SDHI bit synchronously with the rising edge of the UARTi transfer clock. This is because the ABT bit may inadvertently be set to 1 (detected).

15.1.3.6 SDA Input

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

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

15.1.3.7 ACK and NACK

If the STSPSEL bit in the UiSMR4 register is set to 0 (start and stop conditions not generated) and the ACKC bit in the UiSMR4 register is set to 1 (ACK data output), the value of the ACKD bit in the UiSMR4 register is output from the SDAi pin.

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

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

15.1.3.8 Initialization of Transmission/Reception

If a start condition is detected while the STAC bit = 1 (UARTi initialization enabled), the serial interface operates as described below.

- The transmit shift register is initialized, and the content of the UiTB register is transferred to the transmit shift register. In this way, the serial interface starts transmitting data synchronously with the next clock pulse applied. However, the UARTi output value does not change state and remains the same as when a start condition was detected until the first bit of data is output synchronously with the input clock.
- The receive shift register is initialized, and the serial interface starts receiving data synchronously with the next clock pulse applied.
- The SWC bit is set to 1 (SCL wait output enabled). Consequently, the SCLi pin is pulled low at the falling edge of the 9th clock pulse.

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

15.1.4 Special Mode 2

Multiple slaves can be serially communicated from one master. Transfer clock polarity and phase are selectable. Table 15.14 lists the Special Mode 2 Specifications. Figure 15.27 shows the Serial Bus Communication Control Example (UART2). Table 15.15 lists the Registers to be Used an Settings in Special Mode 2.

Table 15.14 Special Mode 2 Specifications

| Item | Specification |
|-------------------------------------|--|
| Transfer data format | Transfer data length: 8 bits |
| Transfer clock | <ul style="list-style-type: none"> • Master mode The CKDIR bit in the UiMR register = 0 (internal clock) : $f_j / (2(n+1))$ $f_j = f1SIO, f2SIO, f8SIO, f32SIO$. n: Setting value of the UiBRG register 00h to FFh • Slave mode The CKDIR bit = 1 (external clock selected) : Input from CLKi pin |
| Transmit/receive control | Controlled by input/output ports |
| Transmit start condition | Before transmission can start, meet the following requirements ⁽¹⁾ <ul style="list-style-type: none"> • The TE bit in the UiC1 register = 1 (transmission enabled) • The TI bit in the UiC1 register = 0 (data present in the UiTB register) |
| Receive start condition | Before reception can start, meet the following requirements ⁽¹⁾ <ul style="list-style-type: none"> • The RE bit in the UiC1 register = 1 (reception enabled) • The TE bit in the UiC1 register = 1 (transmission enabled) • The TI bit in the UiC1 register = 0 (data present in the UiTB register) |
| Interrupt request generation timing | For transmission, one of the following conditions can be selected <ul style="list-style-type: none"> • The UiIRS bit ⁽²⁾ = 0 (transmit buffer empty): when transferring data from the UiTB register to the UARTi transmit register (at start of transmission) • The UiIRS bit =1 (transmission completed): when the serial interface finished transmitting 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 | Overrun error ⁽³⁾ This error occurs if the serial interface started receiving the next data before reading the UiRB register and received the 7th bit of the next data |
| Select function | Clock phase setting Selectable from four combinations of transfer clock polarities and phases |

i = 0 to 2

NOTES:

1. When an external clock is selected, the conditions must be met while if the CKPOL bit in the UiC0 register = 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 = 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.
2. Bits U0IRS and U1IRS are bits 0 and 1 in the UCON register ; the U2IRS bit is bit 4 in the U2C1 register.
3. If an overrun error occurs, the value of UiRB register will be undefined. The IR bit in SiRIC register remains unchanged.

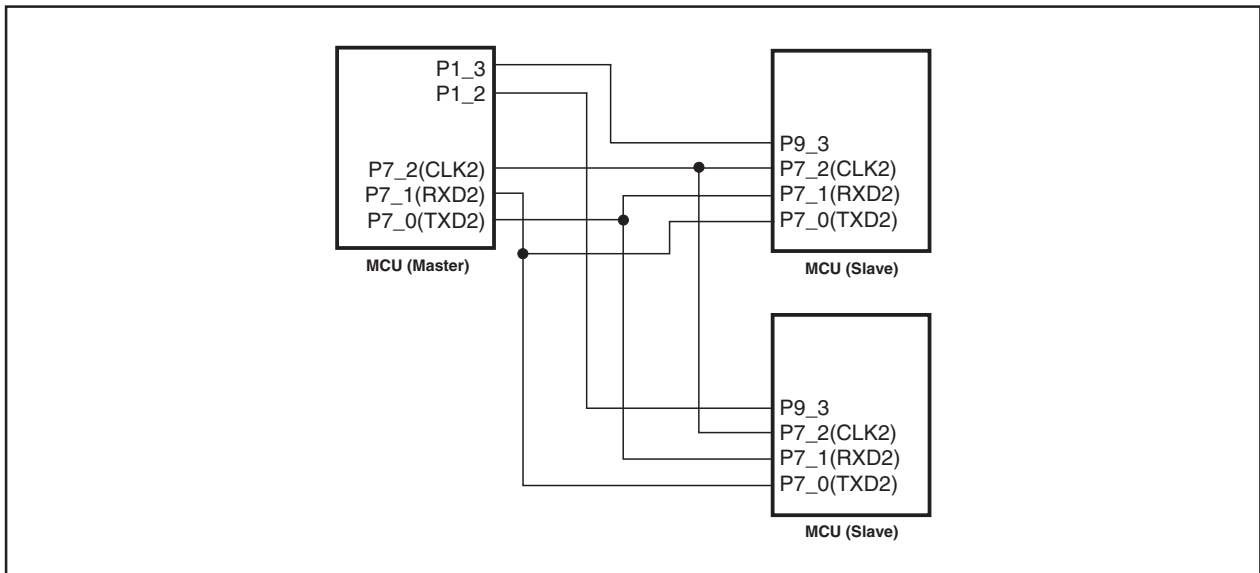


Figure 15.27 Serial Bus Communication Control Example (UART2)

Table 15.15 Registers to Be Used and Settings in Special Mode 2

| Register | Bit | Function |
|---------------------|--|---|
| UiTB ⁽¹⁾ | 0 to 7 | Set transmit data |
| UiRB ⁽¹⁾ | 0 to 7 | Receive data can be read |
| | OER | Overrun error flag |
| UiBRG | 0 to 7 | Set a bit rate |
| UiMR ⁽¹⁾ | SMD2 to SMD0 | Set to 001b |
| | CKDIR | Set this bit to 0 for master mode or 1 for slave mode |
| | IOPOL | Set to 0 |
| UiC0 | CLK1 to CLK0 | Select the count source for the UiBRG register |
| | CRS | Invalid because the CRD bit = 1 |
| | TXEPT | Transmit register empty flag |
| | CRD | Set to 1 |
| | NCH | Select TXDi pin output format |
| | CKPOL | Clock phases can be set in combination with the CKPH bit in the UiSMR3 register |
| | UFORM | Set to 0 |
| 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 UART2 transmit interrupt source |
| | U2RRM ⁽²⁾ , UiLCH, UiERE | Set to 0 |
| UiSMR | 0 to 7 | Set to 0 |
| UiSMR2 | 0 to 7 | Set to 0 |
| UiSMR3 | CKPH | Clock phases can be set in combination with the CKPOL bit in the UiC0 register |
| | NODC | Set to 0 |
| | 0, 2, 4 to 7 | Set to 0 |
| UiSMR4 | 0 to 7 | Set to 0 |
| UCON | U0IRS, U1IRS | Select the UART0 and UART1 transmit interrupt source |
| | U0RRM, U1RRM | Set to 0 |
| | CLKMD0 | Invalid because the CLKMD1 bit = 0 |
| | CLKMD1, RCSP, 7 | Set to 0 |

i = 0 to 2

NOTES:

1. Not all register bits are described above. Set those bits to 0 when writing to the registers in Special Mode 2.
2. Set bits 4 and 5 in registers U0C1 and U1C1 to 0. Bits U0IRS, U1IRS, U0RRM, and U1RRM are in the UCON register.

15.1.4.1 Clock Phase Setting Function

One of four combinations of transfer clock phases and polarities can be selected using the CKPH bit in the UiSMR3 register and the CKPOL bit in the UiC0 register.

Make sure the transfer clock polarity and phase are the same for the master and slaves to be communicated.

Figure 15.28 shows the Transmission and Reception Timing in Master Mode (internal clock).

Figure 15.29 shows the Transmission and Reception Timing (CKPH = 0) in Slave Mode (external clock).

Figure 15.30 shows the Transmission and Reception Timing (CKPH = 1) in Slave Mode (external clock).

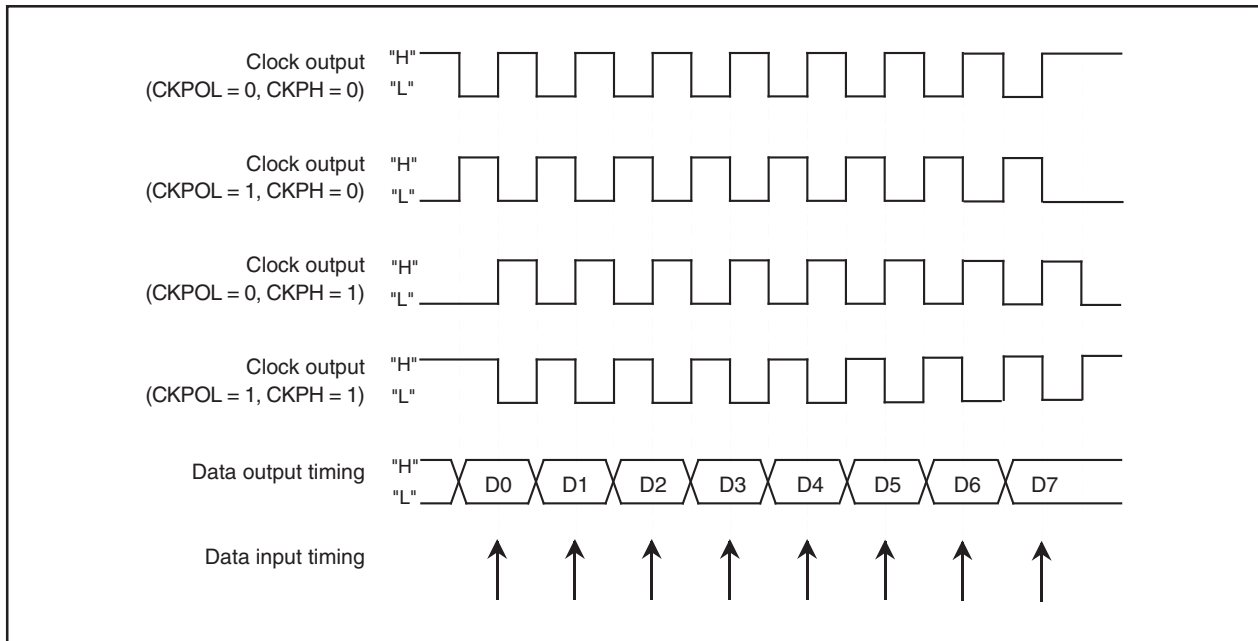


Figure 15.28 Transmission and Reception Timing in Master Mode (Internal Clock)

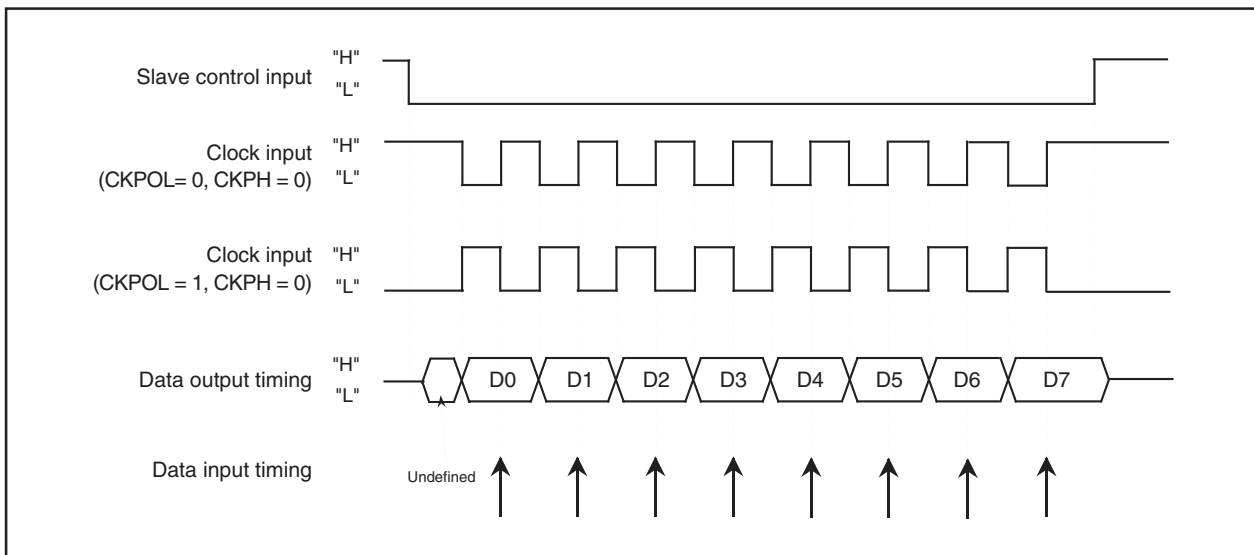


Figure 15.29 Transmission and Reception Timing (CKPH = 0) in Slave Mode (External Clock)

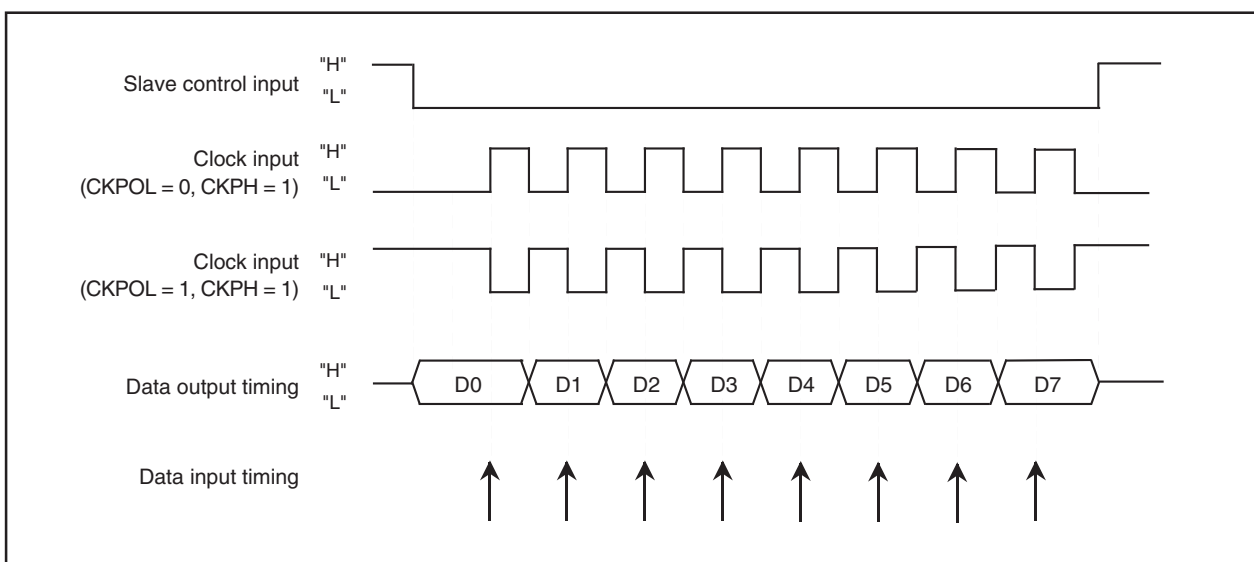


Figure 15.30 Transmission and Reception Timing (CKPH = 1) in Slave Mode (External Clock)

15.1.5 Special Mode 3 (IE Mode)

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

Table 15.16 lists the Registers to be Used and Settings in IE mode. Figure 15.31 shows the Bus Collision Detect Function-Related Bits.

If the TXDi pin (i = 0 to 2) output level and RXDi pin input level do not match, a UARTi bus collision detect interrupt request is generated.

Use bits IFSR06 and IFSR07 in the IFSR0 register to enable the UART0/UART1 bus collision detect function.

Table 15.16 Registers to Be Used and Settings in IE Mode

| Register | Bit | Function |
|---------------------|--|---|
| UiTB | 0 to 8 | Set transmit data |
| UiRB ⁽¹⁾ | 0 to 8 | Receive data can be read |
| | OER, FER, PER, SUM | Error flag |
| UiBRG | 0 to 7 | Set a bit rate |
| UiMR | SMD2 to SMD0 | Set to 110b |
| | CKDIR | Select the internal clock or external clock |
| | STPS | Set to 0 |
| | PRY | Invalid because the PRYE bit = 0 |
| | PRYE | Set to 0 |
| | IOPOL | Select the TXD/RXD input/output polarity |
| UiC0 | CLK1 to CLK0 | Select the count source for the UiBRG register |
| | CRS | Invalid because the CRD bit = 1 |
| | TXEPT | Transmit register empty flag |
| | CRD | Set to 1 |
| | NCH | Select TXDi pin output mode |
| | CKPOL | Set to 0 |
| | UFORM | Set to 0 |
| 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 UART2 transmit interrupt source |
| | U2RRM ⁽²⁾ , UiLCH, UiERE | Set to 0 |
| UiSMR | 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 |
| UiSMR2 | 0 to 7 | Set to 0 |
| UiSMR3 | 0 to 7 | Set to 0 |
| UiSMR4 | 0 to 7 | Set to 0 |
| IFSR0 | IFSR06, IFSR07 | Set to 1 |
| UCON | U0IRS, U1IRS | Select the UART0/UART1 transmit interrupt source |
| | U0RRM, U1RRM | Set to 0 |
| | CLKMD0 | Invalid because the CLKMD1 bit = 0 |
| | CLKMD1, RCSP, 7 | Set to 0 |

i= 0 to 2

NOTES:

1. Not all register bits are described above. Set those bits to 0 when writing to the registers in IE mode.
2. Set bits 4 and 5 in registers U0C1 and U1C1 to 0. Bits U0IRS, U1IRS, U0RRM, and U1RRM are in the UCON register.

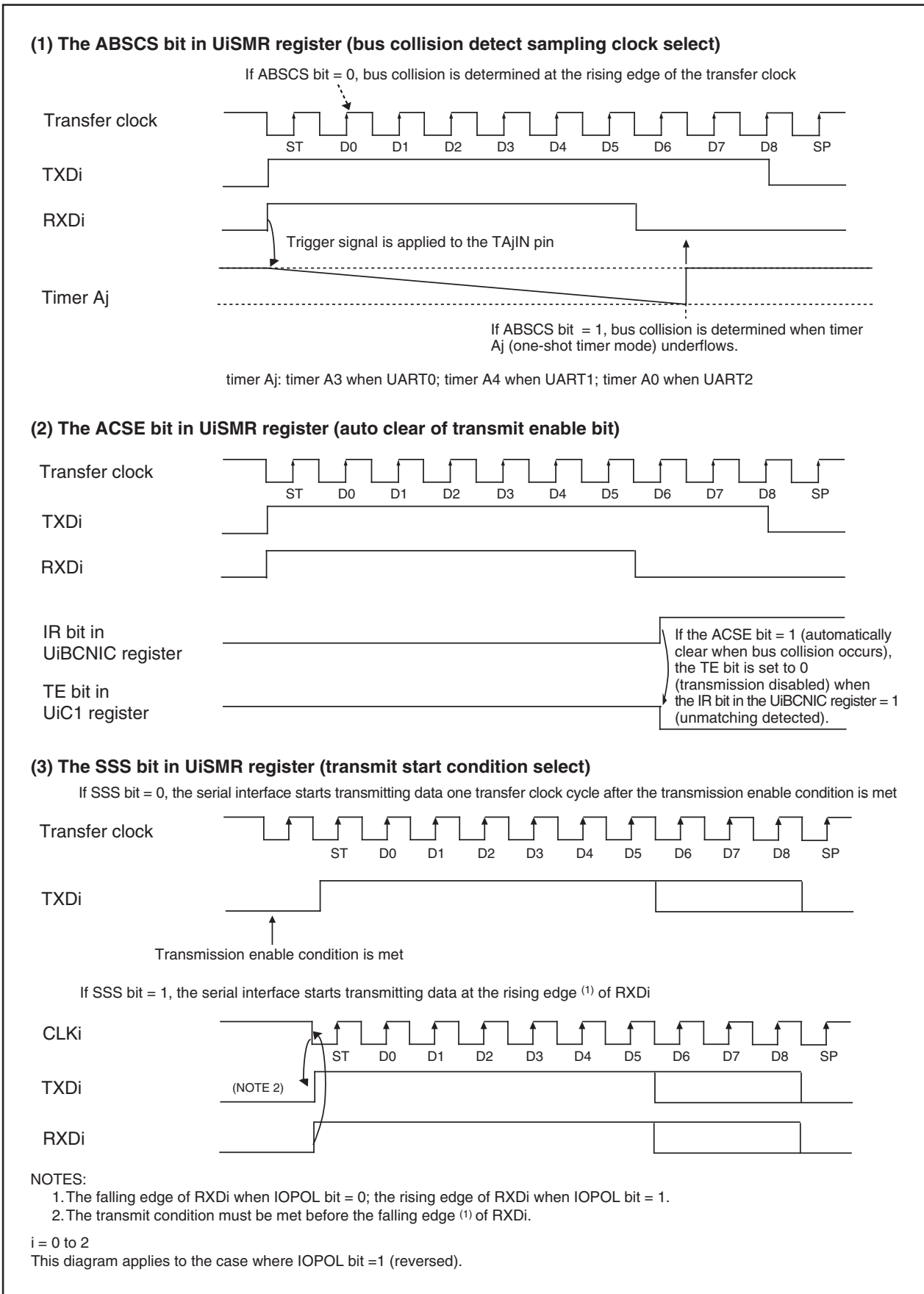


Figure 15.31 Bus Collision Detect Function-Related Bits

15.1.6 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 to output a low from the TXD2 pin when a parity error is detected.

Table 15.17 lists the SIM Mode Specifications. Table 15.18 lists the Registers to be Used and Settings in SIM Mode. Figure 15.32 shows the Transmit and Receive Timing in SIM Mode.

Table 15.17 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 = 0 (internal clock) : $f_i/(16(n+1))$ $f_i = f1SIO, f2SIO, f8SIO, f32SIO$. n: Setting value of the U2BRG register 00h to FFh • The CKDIR bit = 1 (external clock) : $fEXT/(16(n+1))$ $fEXT$: Input from CLK2 pin. n: Setting value of the U2BRG register 00h to FFh |
| Transmit start condition | <p>Before transmission can start, meet the following requirements</p> <ul style="list-style-type: none"> • The TE bit in the U2C1 register = 1 (transmission enabled) • The TI bit in the U2C1 register = 0 (data present in the U2TB register) |
| Receive start condition | <p>Before reception can start, meet the following requirements</p> <ul style="list-style-type: none"> • The RE bit in the U2C1 register = 1 (reception enabled) • Start bit detection |
| Interrupt request generation timing ⁽²⁾ | <ul style="list-style-type: none"> • For transmission When the serial interface 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 interface started receiving the next data before reading the U2RB register and received the bit one before the last stop bit of the next data • Framing error ⁽³⁾ This error occurs when the number of stop bits set is not detected • Parity error ⁽³⁾ 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, the value of the U2RB register will be 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 completed) and U2ERE bit in the U2C1 register to 1 (error signal output) after reset. Therefore, when using SIM mode, set the IR bit to 0 (interrupt not requested) after setting these bits.
3. The timing at which the framing error flag and the parity error flag are set is detected when data is transferred from the UARTi receive register to the UIRB register.

Table 15.18 Registers to Be Used and Settings in SIM Mode

| Register | Bit | Function |
|----------------------|-----------------|---|
| U2TB ⁽¹⁾ | 0 to 7 | Set transmit data |
| U2RB ⁽¹⁾ | 0 to 7 | Receive data can be read |
| | OER,FER,PER,SUM | Error flag |
| U2BRG | 0 to 7 | Set a bit rate |
| U2MR | SMD2 to SMD0 | Set to 101b |
| | 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 to CLK0 | Select the count source for the U2BRG register |
| | CRS | Invalid because the CRD bit = 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.

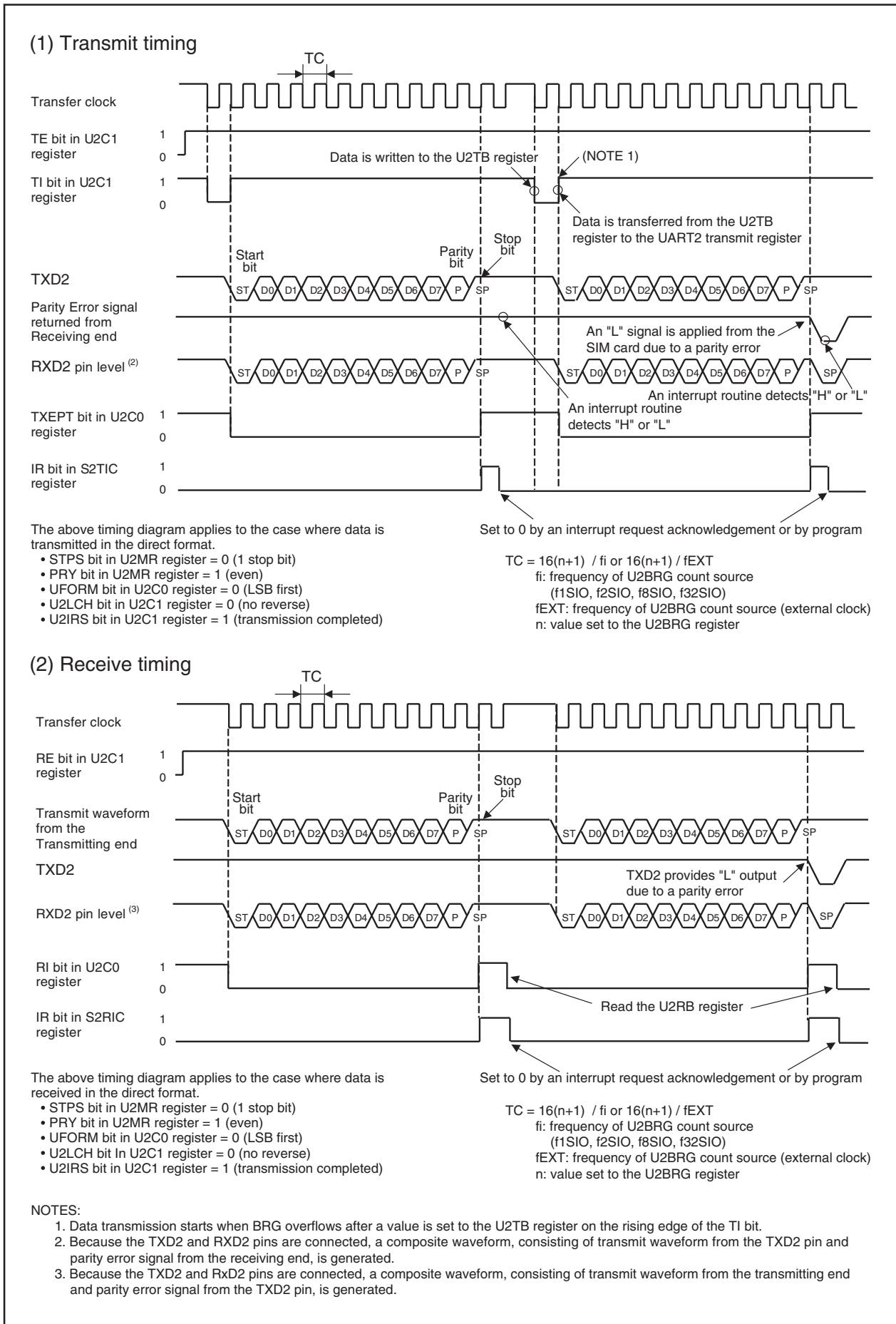


Figure 15.32 Transmit and Receive Timing in SIM Mode

Figure 15.33 shows the SIM Interface Connection. Connect TXD2 and RXD2 and apply pull-up.

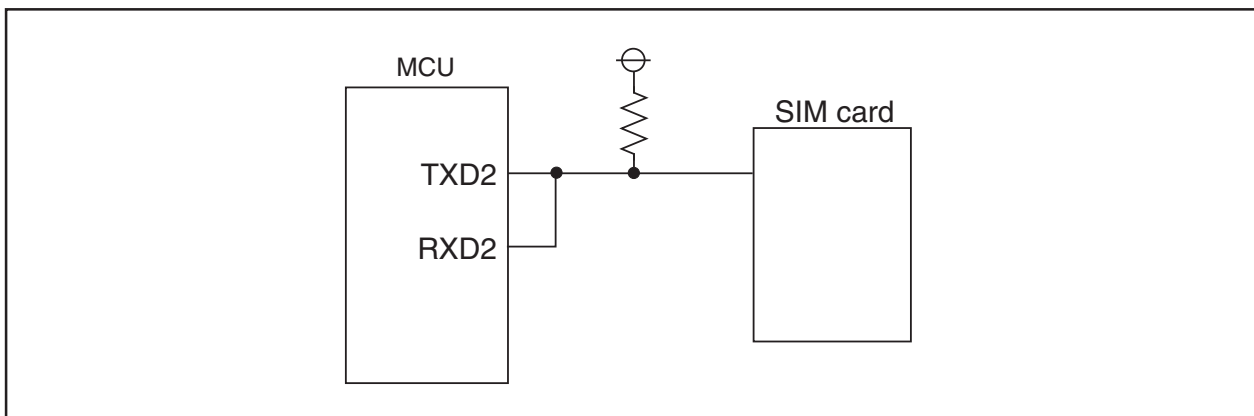


Figure 15.33 SIM Interface Connection

15.1.6.1 Parity Error Signal Output

The parity error signal is enabled by setting the U2ERE bit in the U2C1 register to 1 (output enabled). 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 15.32. If the U2RB register is read while outputting a parity error signal, the PER bit in the U2RB register is set to 0 (no parity error) 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 UXD2 pin in a transmission-finished interrupt routine.

Figure 15.34 shows the output timing of the parity error signal

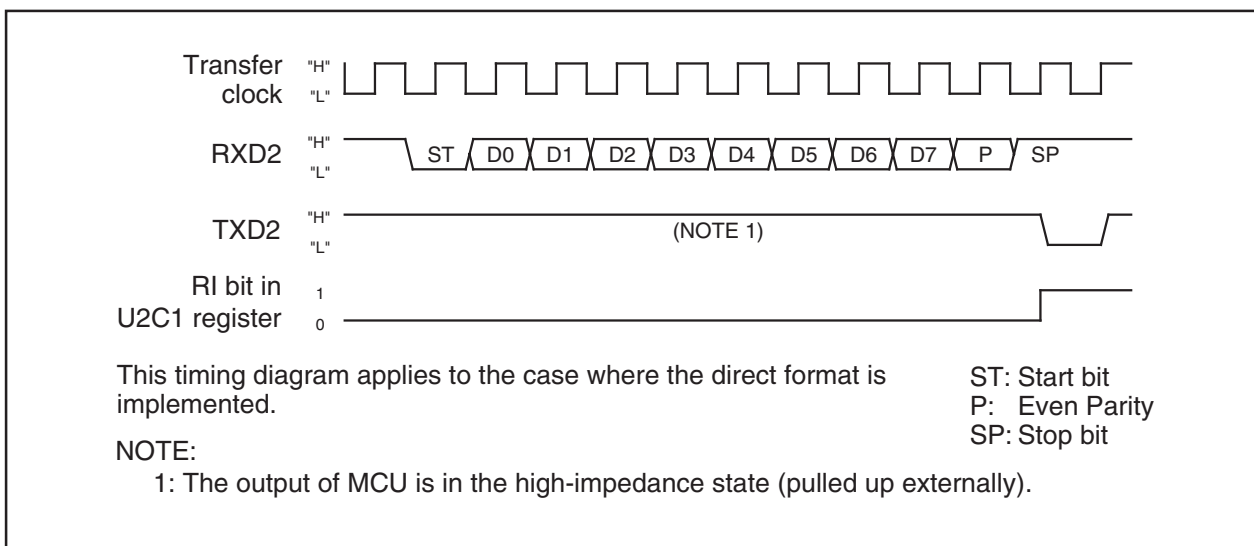


Figure 15.34 Parity Error Signal Output Timing

15.1.6.2 Format

When direct format, set the PRYE bit in the U2MR register to 1, the PRY bit to 1, the UFORM bit in the U2C0 register to 0 and the U2LCH bit in the U2C1 register to 0. When data are transmitted, data set in the U2TB register are transmitted with the even-numbered parity, starting from D0. When data are received, received data are stored in the U2RB register, starting from D0. The even-numbered parity determines whether a parity error occurs.

When inverse format, set the PRYE bit to 1, the PRY bit to 0, the UFORM bit to 1 and the U2LCH bit to 1. When data are transmitted, values set in the U2TB register are logically inversed and are transmitted with the odd-numbered parity, starting from D7. When data are received, received data are logically inversed to be stored in the U2RB register, starting from D7. The odd-numbered parity determines whether a parity error occurs.

Figure 15.35 shows the SIM Interface Format.

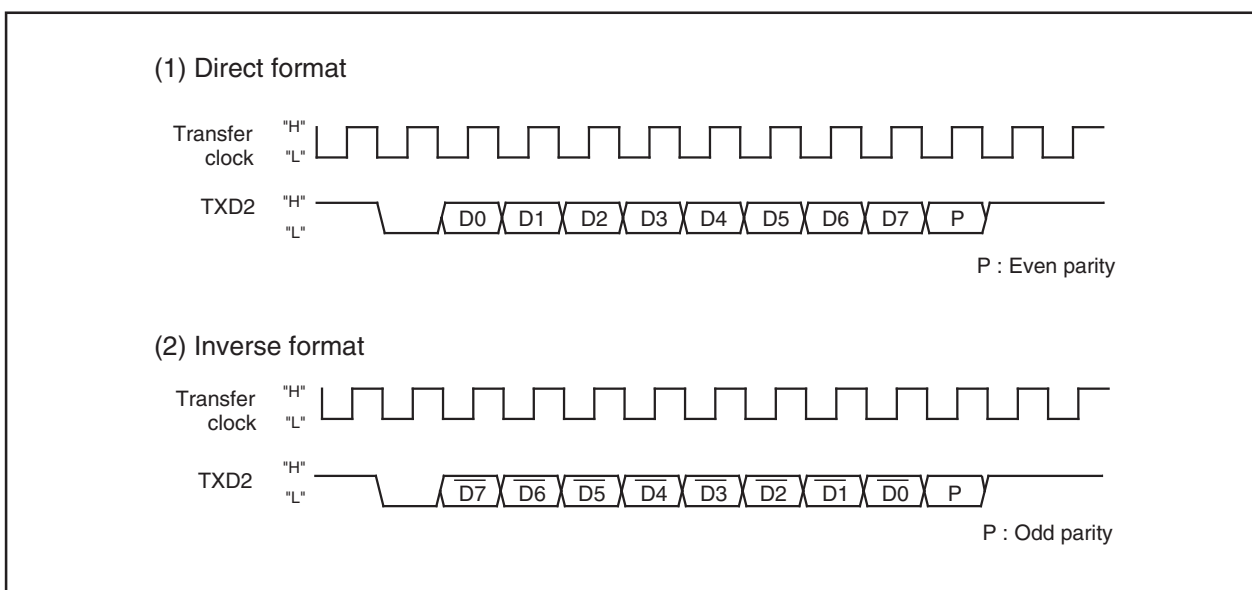


Figure 15.35 SIM Interface Format

15.2 SI/O3

SI/O3 is exclusive clock-synchronous serial I/Os.

Figure 15.36 shows the SI/O3 Block Diagram, and Figure 15.37 shows the SI/O3-related registers. Table 15.19 lists the SI/O3 Specifications.

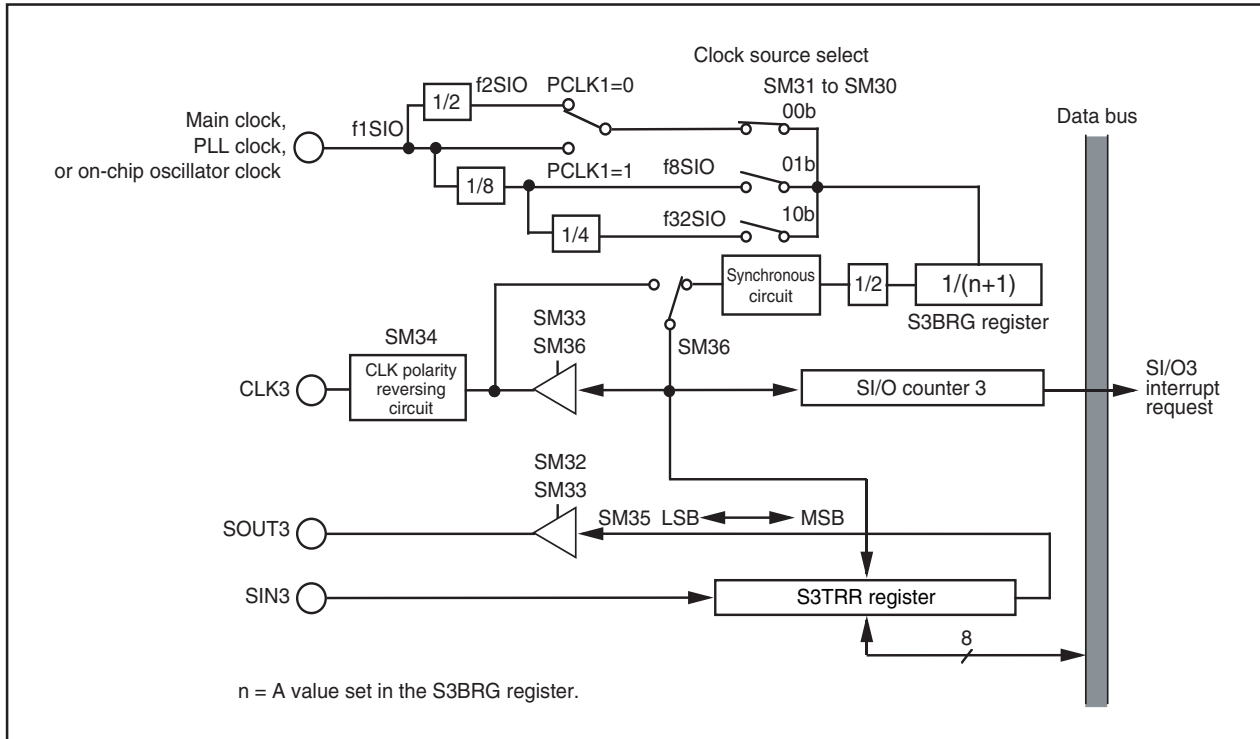


Figure 15.36 SI/O3 Block Diagram

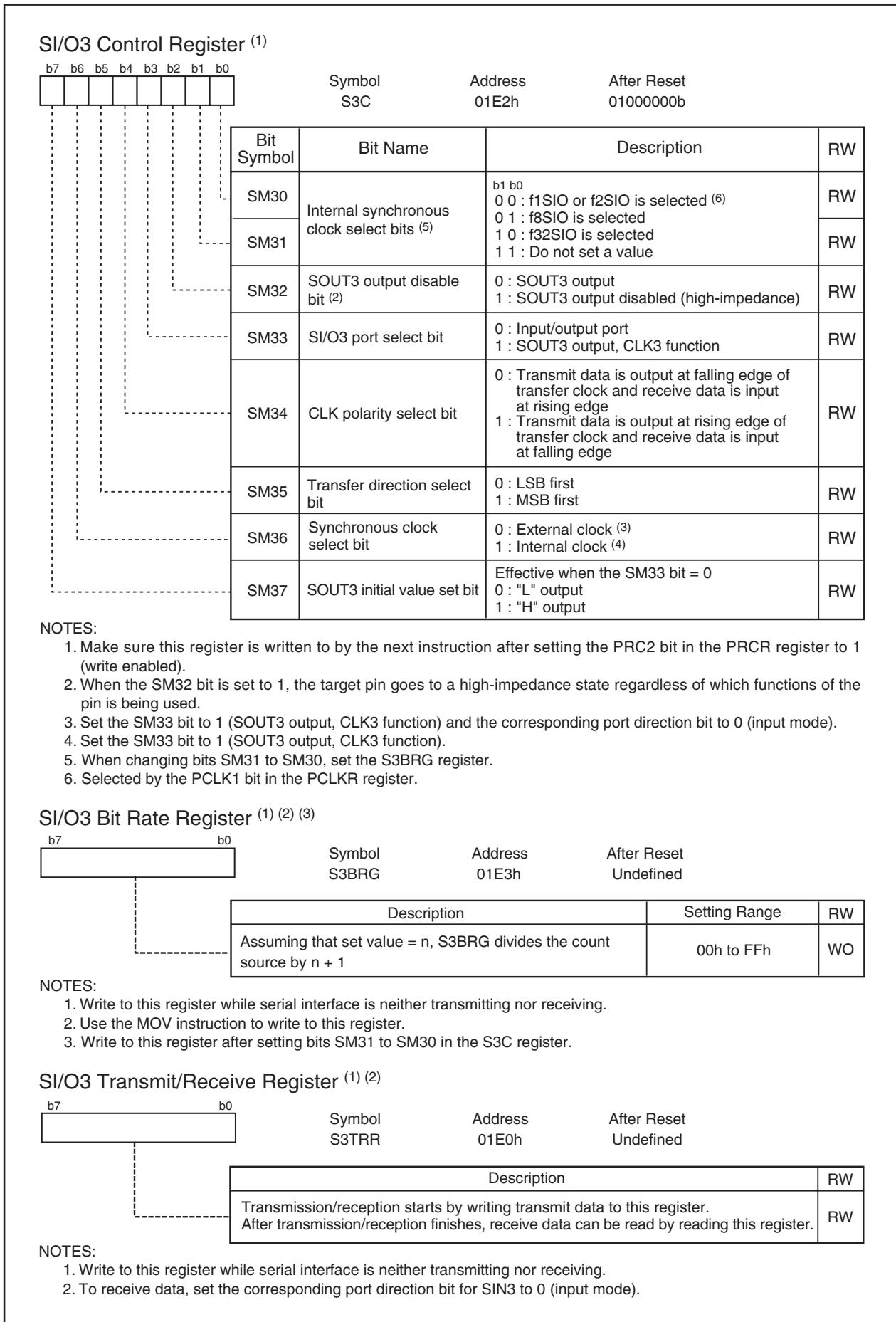


Figure 15.37 Registers S3C, S3BRG, and S3TRR

Table 15.19 SI/O3 Specifications

| Item | Specification |
|-------------------------------------|---|
| Transfer data format | Transfer data length: 8 bits |
| Transfer clock | <ul style="list-style-type: none"> • SM36 bit in S3C register = 1 (internal clock) : $f_j/(2(n+1))$ $f_j = f1SIO, f8SIO, f32SIO$. n = Setting value of S3BRG register 00h to FFh • SM36 bit = 0 (external clock) : Input from CLK3 pin ⁽¹⁾ |
| Transmit/receive start condition | Before transmission/reception can start, meet the following requirements Write transmit data to the S3TRR register ^{(2) (3)} |
| Interrupt request generation timing | <ul style="list-style-type: none"> • When SM34 bit in S3C register = 0 The rising edge of the last transfer clock pulse ⁽⁴⁾ • When SM34 bit = 1 The falling edge of the last transfer clock pulse ⁽⁴⁾ |
| CLK3 pin function | I/O port, transfer clock input, transfer clock output |
| SOUT3 pin function | I/O port, transmit data output, high-impedance |
| SIN3 pin function | I/O port, receive data input |
| Select function | <ul style="list-style-type: none"> • LSB first or MSB first selection Whether to start transmitting or receiving data begins with bit 0 or begins with bit 7 can be selected • Function for setting an SOUT3 initial value set function When the SM36 bit in the S3C register = 0 (external clock), the SOUT3 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. |

NOTES:

- To set the SM36 bit in the S3C register to 0 (external clock), follow the procedure described below.
 - If the SM34 bit in the S3C register = 0, write transmit data to the S3TRR register while input on the CLK3 pin is high. The same applies when rewriting the SM37 bit in the S3C register.
 - If the SM34 bit = 1, write transmit data to the S3TRR register while input on the CLK3 pin is low. The same applies when rewriting the SM37 bit.
 - Because shift operation continues as long as the transfer clock is supplied to the SI/O3 circuit, stop the transfer clock after supplying eight pulses. If the SM36 bit = 1 (internal clock), the transfer clock automatically stops.
- Unlike UART0 to UART2, SI/O3 is not separated between the transfer register and buffer. Therefore, do not write the next transmit data to the S3TRR register during transmission.
- When the SM36 bit = 1 (internal clock), SOUT3 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 S3TRR register during this period, SOUT3 immediately goes to a high-impedance state, with the data hold time thereby reduced.
- When the SM36 bit = 1 (internal clock), the transfer clock stops in the high state if the SM34 bit = 0, or stops in the low state if the SM34 bit = 1.

15.2.1 SI/O3 Operation Timing

Figure 15.38 shows the SI/O3 Operation Timing.

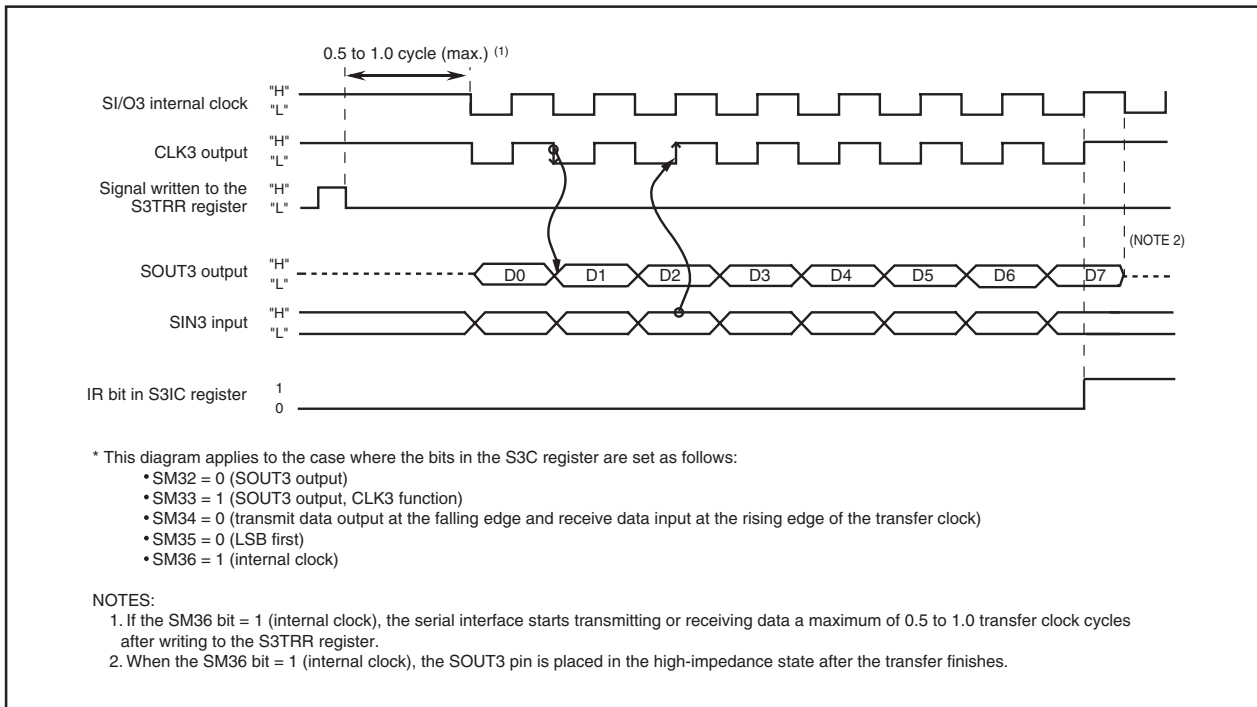


Figure 15.38 SI/O3 Operation Timing

15.2.2 CLK Polarity Selection

The SM34 bit in the S3C register allows selection of the polarity of the transfer clock.

Figure 15.39 shows the Polarity of Transfer Clock.

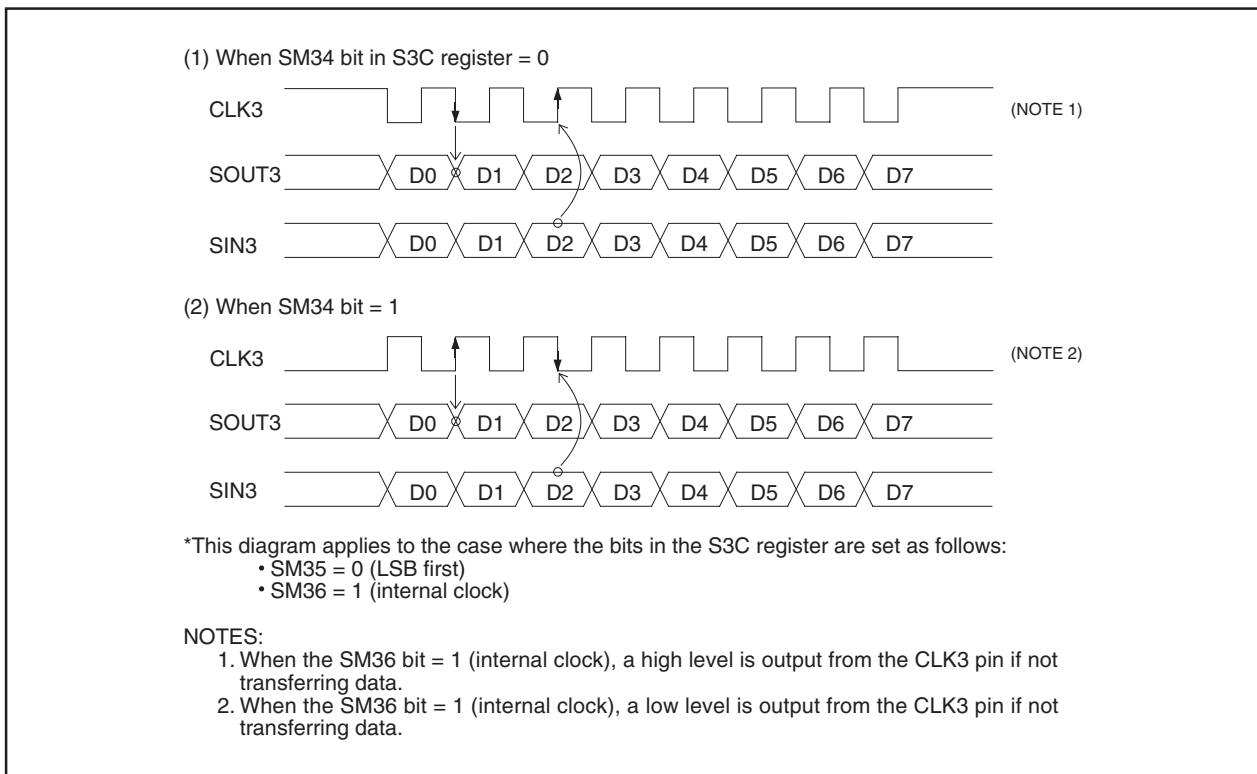


Figure 15.39 Polarity of Transfer Clock

15.2.3 Functions for Setting SOUT3 Initial Value

If the SM36 bit in the S3C register is set to 0 (external clock), the SOUT3 pin output can be fixed high or low when not transferring. However, the last bit value of the former data is retained between data and data when transmitting the continuous data.

Figure 15.40 shows the timing chart and how to set it for the SOUT3's Initial Value Setting.

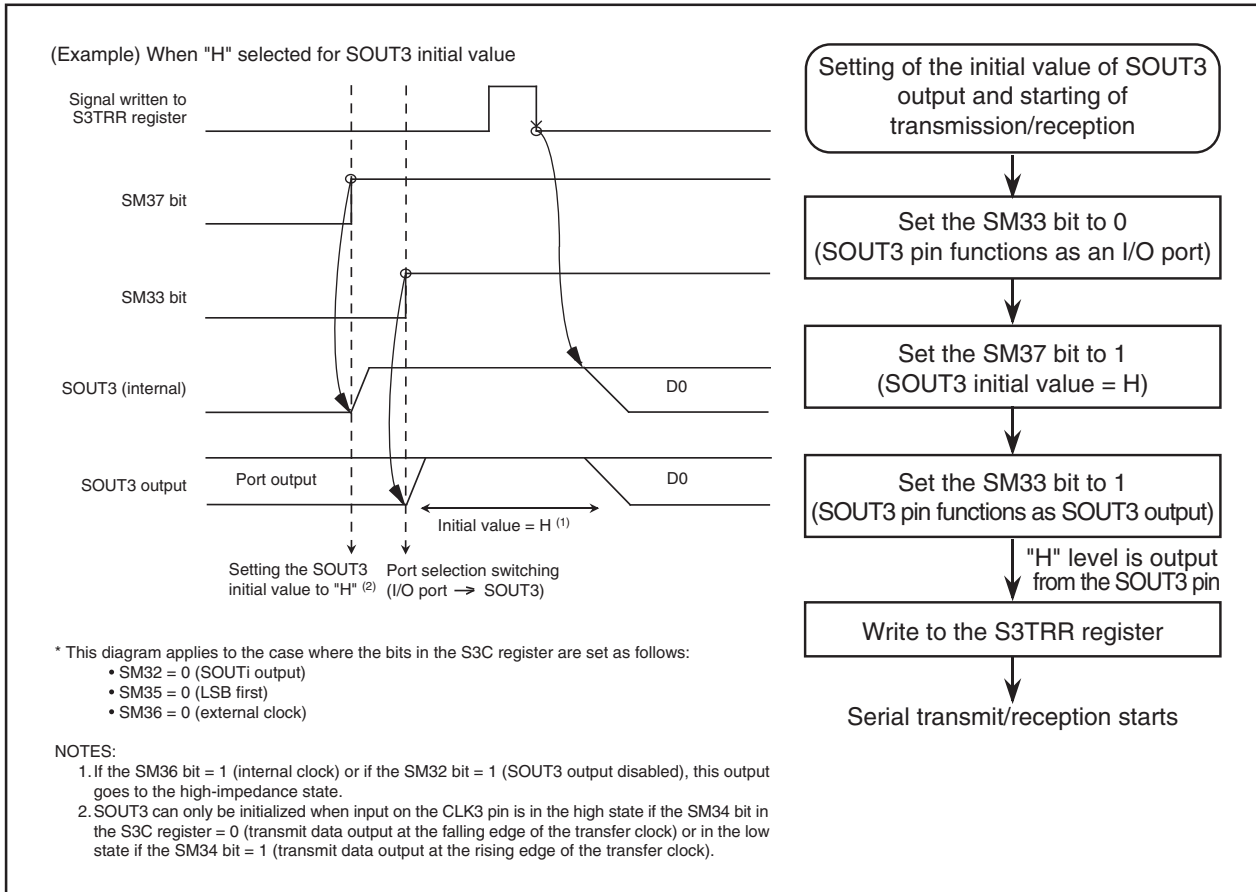


Figure 15.40 SOUT3's Initial Value Setting

16. A/D Converter

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 P10_0 to P10_7, P9_5, P9_6, P0_0 to P0_7, and P2_0 to P2_7. Similarly, ADTRG input shares the pin with P9_7. 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 bits in the ADi register for pins ANi, AN0_i, and AN2_i (i = 0 to 7). Table 16.1 shows the A/D Converter Performance. Figure 16.1 shows the A/D Converter Block Diagram, and Figures 16.2 and 16.3 show the A/D converter-related registers.

Table 16.1 A/D Converter Performance

| Item | Performance |
|--|--|
| Method of A/D conversion | Successive approximation (capacitive coupling amplifier) |
| Analog input voltage ⁽¹⁾ | 0 V to AVCC (VCC) |
| Operating clock ϕ_{AD} ⁽²⁾ | fAD, divide-by-2 of fAD, divide-by-3 of fAD, divide-by-4 of fAD, divide-by-6 of fAD, divide-by-12 of fAD |
| Resolution | 8 bits or 10 bits (selectable) |
| Integral nonlinearity error | When AVCC = VREF = 5 V <ul style="list-style-type: none"> • With 8-bit resolution: ± 2 LSB • With 10-bit resolution: ± 3 LSB When external operation amp connection mode is selected: ± 7 LSB When AVCC = VREF = 3.3 V <ul style="list-style-type: none"> • With 8-bit resolution: ± 2 LSB • With 10-bit resolution: ± 5 LSB When external operation amp connection mode is selected: ± 7 LSB |
| Operating modes | One-shot mode, repeat mode, single sweep mode, repeat sweep mode 0, and repeat sweep mode 1 |
| Analog input pins | 8 pins (AN0 to AN7) + 2 pins (ANEX0 and ANEX1) + 8 pins (AN0_0 to AN0_7) + 8 pins (AN2_0 to AN2_7) |
| A/D conversion start condition | <ul style="list-style-type: none"> • Software trigger The ADST bit in the ADCON0 register is set to 1 (A/D conversion starts) • External trigger (retriggerable) Input on the ADTRG pin changes state from high to low after the ADST bit is set to 1 (A/D conversion starts) |
| Conversion speed per pin | <ul style="list-style-type: none"> • Without sample and hold 8-bit resolution: 49 ϕ_{AD} cycles, 10-bit resolution: 59 ϕ_{AD} cycles • With sample and hold 8-bit resolution: 28 ϕ_{AD} cycles, 10-bit resolution: 33 ϕ_{AD} cycles |

NOTES:

1. Does not depend on use of sample and hold.
2. ϕ_{AD} frequency must be 10 MHz or less.
When sample and hold is disabled, ϕ_{AD} frequency must be 250 kHz or more.
When sample and hold is enabled, ϕ_{AD} frequency must be 1 MHz or more.

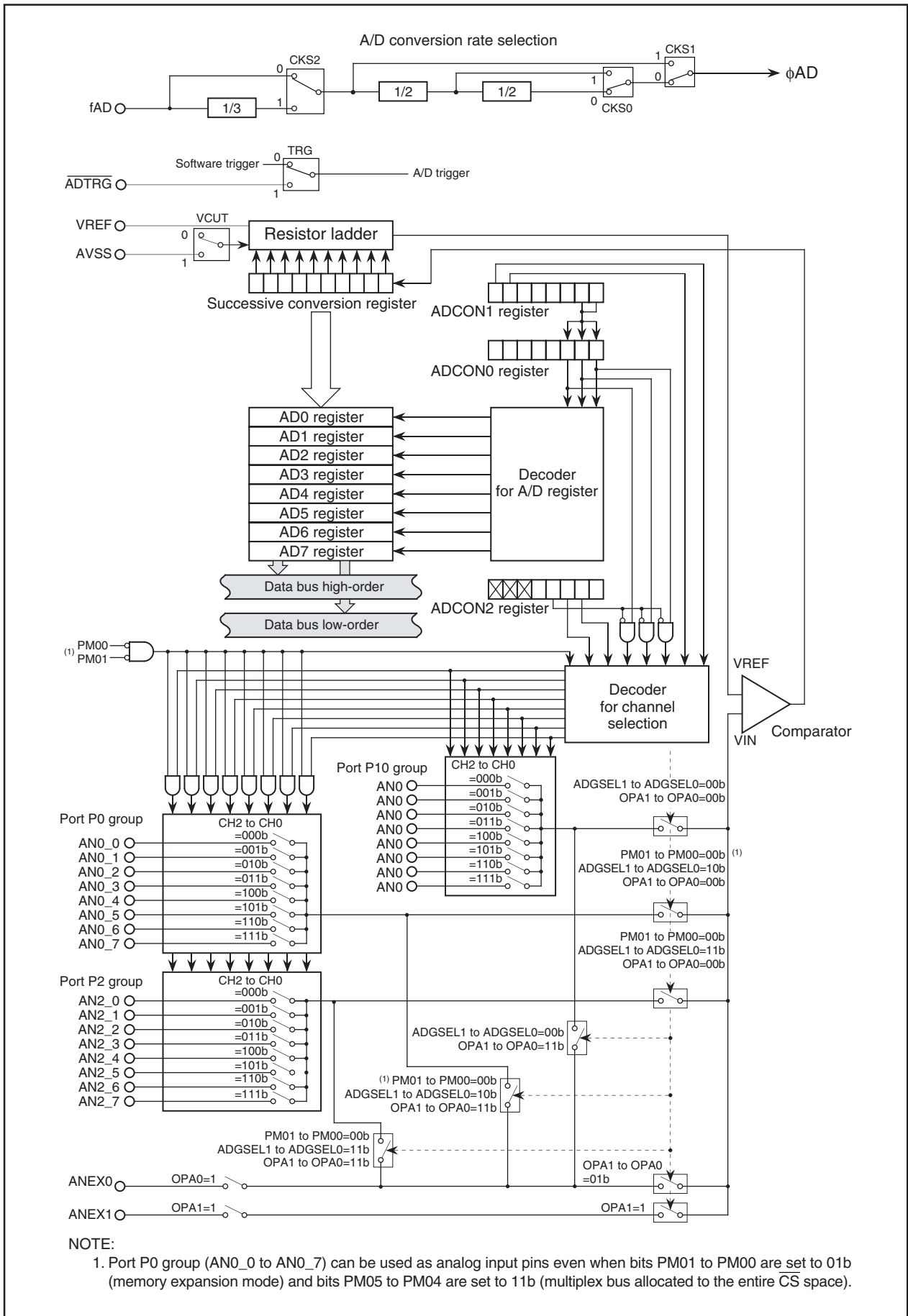


Figure 16.1 A/D Converter Block Diagram

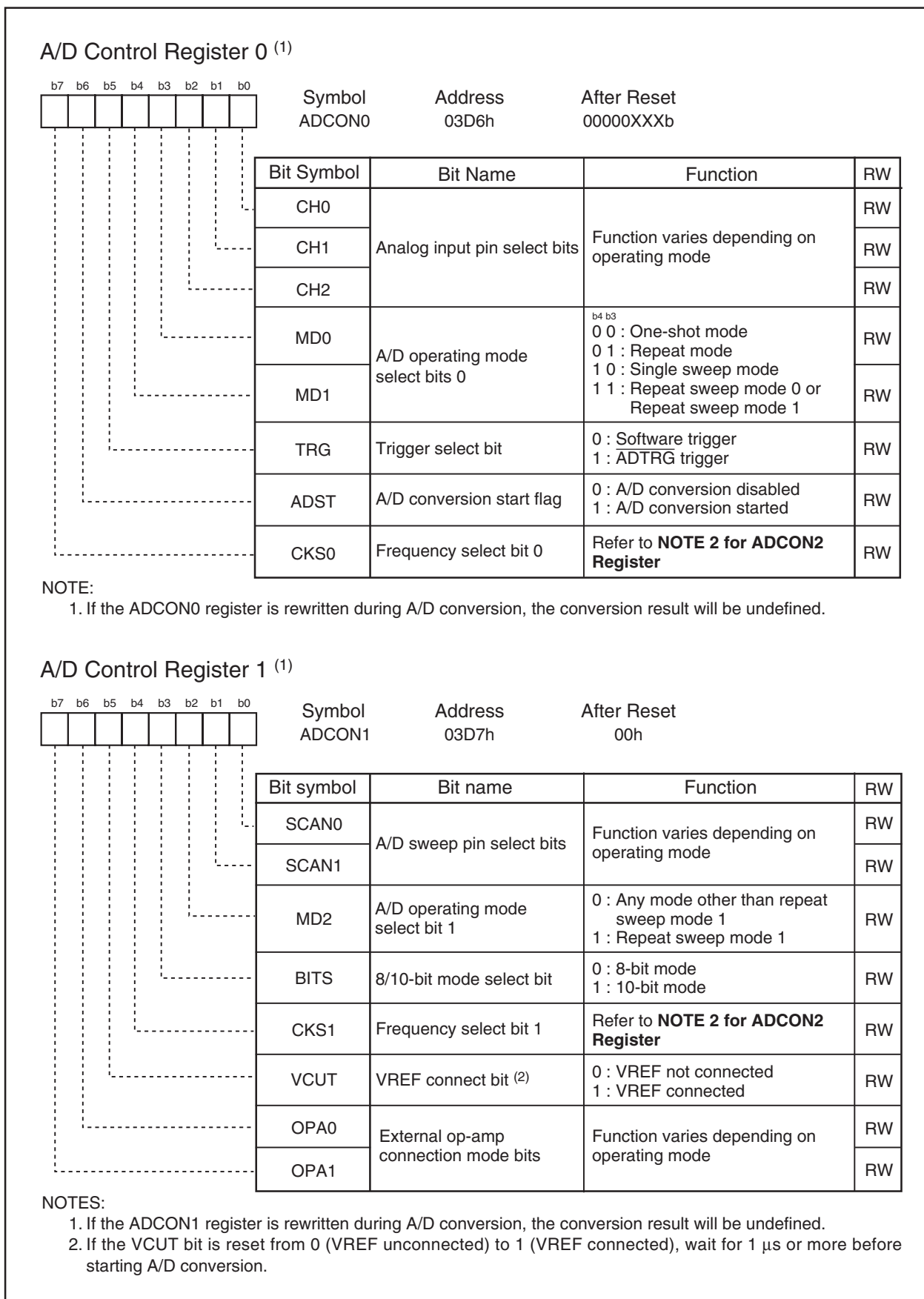


Figure 16.2 Registers ADCON0 and ADCON1

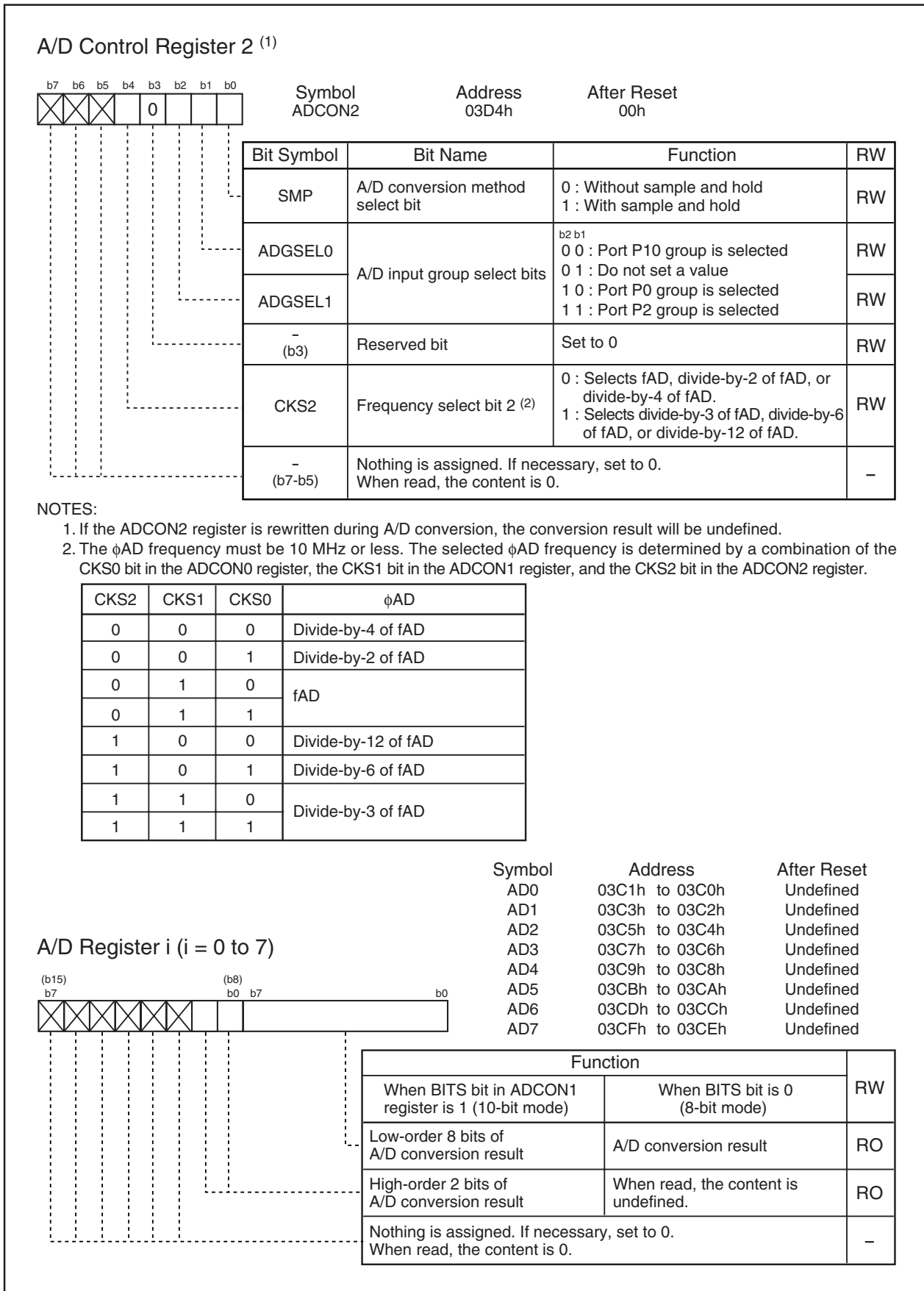


Figure 16.3 Registers ADCON2, and AD0 to AD7

16.1 Mode Description

16.1.1 One-shot Mode

In one-shot mode, analog voltage applied to a selected pin is converted to a digital code once.

Table 16.2 lists the One-shot Mode Specifications. Figure 16.4 shows Registers ADCON0 and ADCON1 in One-shot Mode.

Table 16.2 One-shot Mode Specifications

| Item | Specification |
|-------------------------------------|---|
| Function | Bits CH2 to CH0 in the ADCON0 register, bits ADGSEL1 to ADGSEL0 in the ADCON2 register, and bits OPA1 to OPA0 in the ADCON1 register select a pin Analog voltage applied to the pin is converted to a digital code once. |
| A/D conversion start condition | <ul style="list-style-type: none"> • When the TRG bit in the ADCON0 register is 0 (software trigger) The ADST bit in the ADCON0 register is set to 1 (A/D conversion starts) • When the TRG bit is 1 ($\overline{\text{ADTRG}}$ trigger) Input on the $\overline{\text{ADTRG}}$ pin changes state from high to low after the ADST bit is set to 1 (A/D conversion starts) |
| A/D conversion stop condition | <ul style="list-style-type: none"> • Completion of A/D conversion (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 | Completion of A/D conversion |
| Analog input pin | Select one pin from AN0 to AN7, AN0_0 to AN0_7, AN2_0 to AN2_7, ANEX0 to ANEX1 |
| Reading of result of A/D converter | Read one of registers AD0 to AD7 that corresponds to the selected pin |

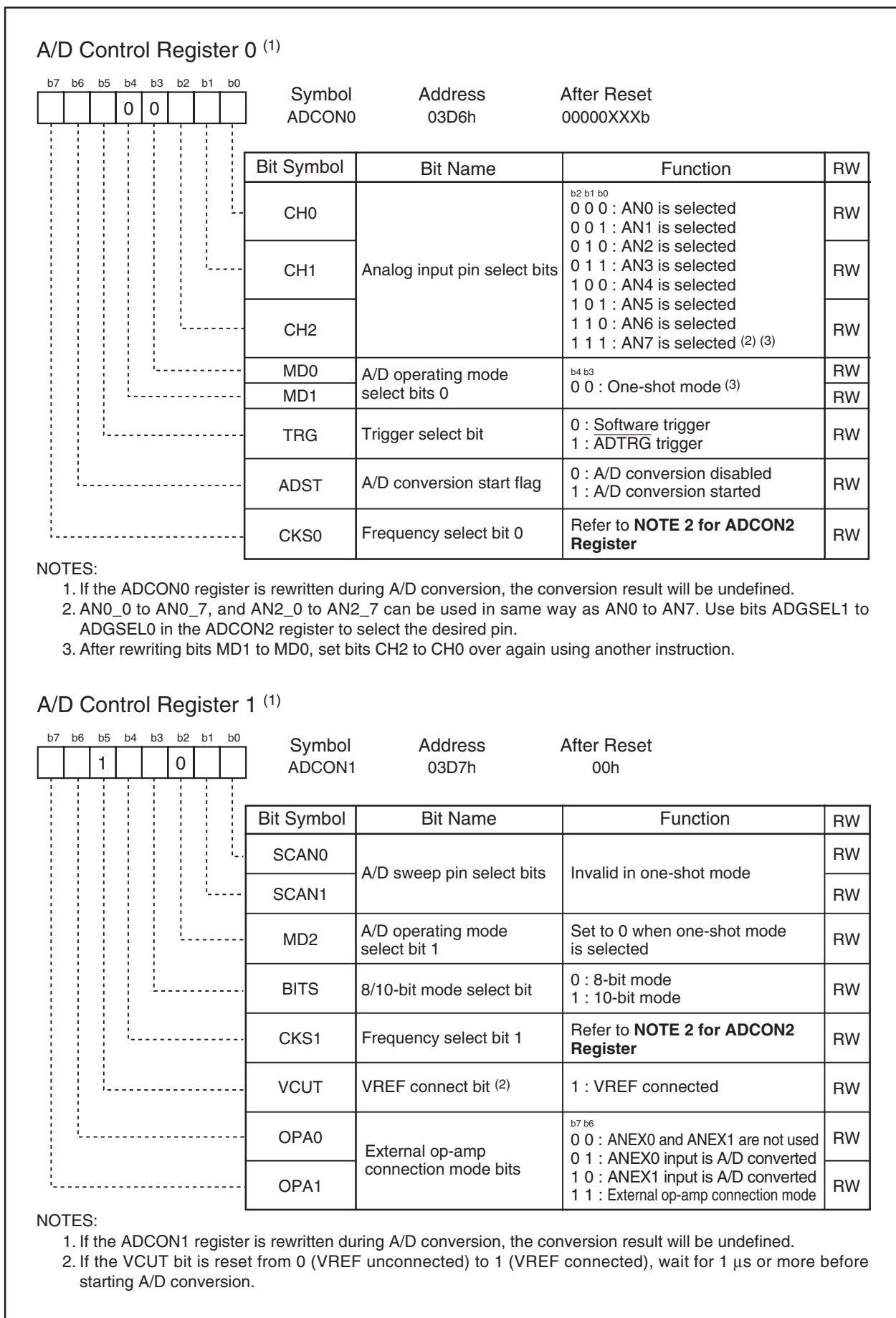


Figure 16.4 Registers ADCON0 and ADCON1 in One-shot Mode

16.1.2 Repeat Mode

In repeat mode, analog voltage applied to a selected pin is repeatedly converted to a digital code.

Table 16.3 lists the Repeat Mode Specifications. Figure 16.5 shows Registers ADCON0 and ADCON1 in Repeat Mode.

Table 16.3 Repeat Mode Specifications

| Item | Specification |
|-------------------------------------|--|
| Function | Bits CH2 to CH0 in the ADCON0 register, bits ADGSEL1 to ADGSEL0 in the ADCON2 register, and bits OPA1 to OPA0 in the ADCON1 register select a pin. Analog voltage applied to this 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) The ADST bit in the ADCON0 register is set to 1 (A/D conversion starts) • When the TRG bit is 1 (ADTRG trigger) Input on the $\overline{\text{ADTRG}}$ pin changes state from high to low after the ADST bit is set to 1 (A/D conversion starts) |
| 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, AN0_0 to AN0_7, AN2_0 to AN2_7, ANEX0 to ANEX1 |
| Reading of result of A/D converter | Read one of registers AD0 to AD7 that corresponds to the selected pin |

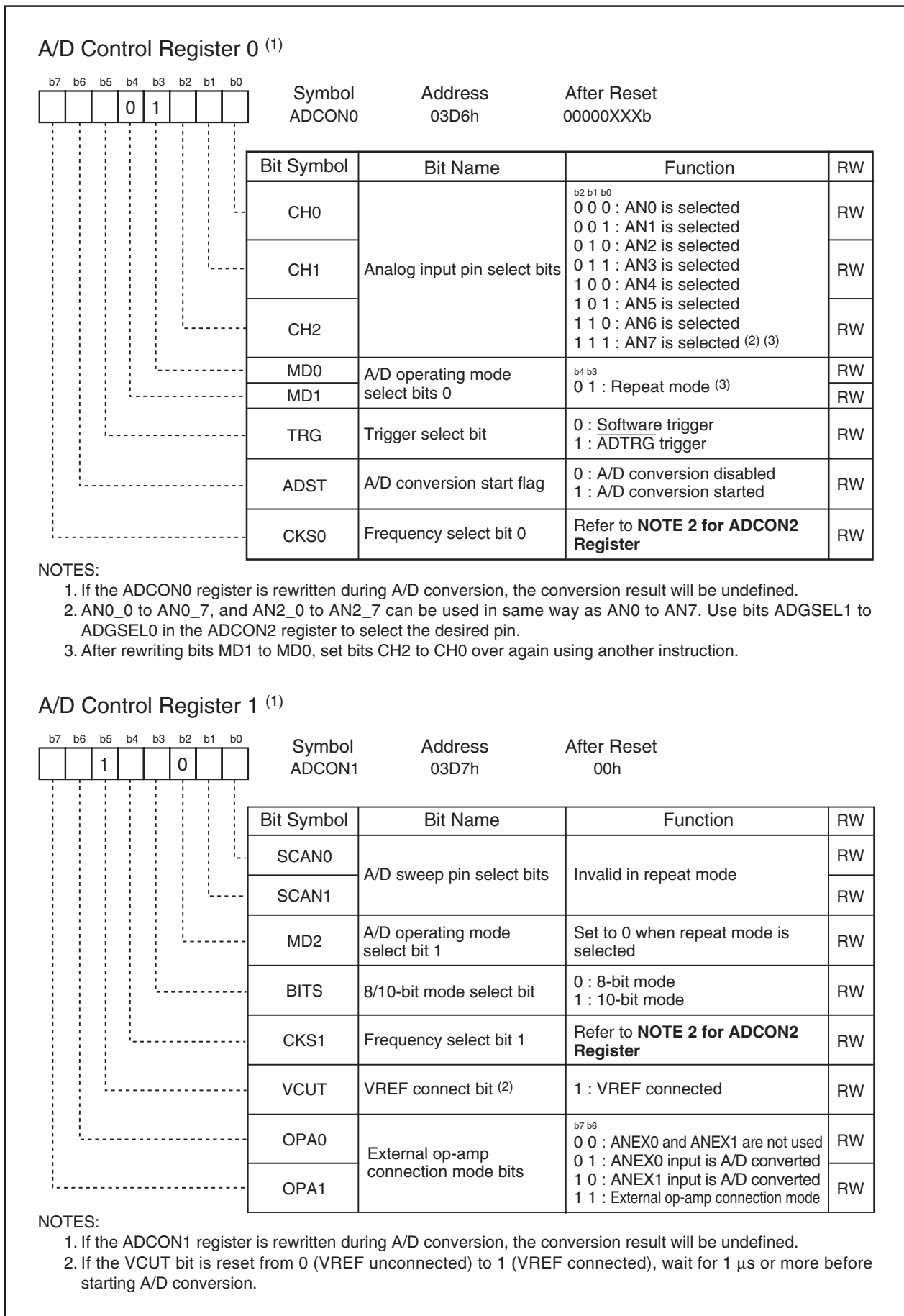


Figure 16.5 Registers ADCON0 and ADCON1 in Repeat Mode

16.1.3 Single Sweep Mode

In single sweep mode, analog voltage that is applied to selected pins is converted one-by-one to a digital code. Table 16.4 lists the Single Sweep Mode Specifications. Figure 16.6 shows Registers ADCON0 and ADCON1 in Single Sweep Mode.

Table 16.4 Single Sweep Mode Specifications

| Item | Specification |
|-------------------------------------|---|
| Function | Bits SCAN1 to SCAN0 in the ADCON1 register and bits ADGSEL1 to ADGSEL0 in the ADCON2 register select pins. Analog voltage applied to this 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) The ADST bit in the ADCON0 register is set to 1 (A/D conversion starts) When the TRG bit is 1 ($\overline{\text{ADTRG}}$ trigger) Input on the $\overline{\text{ADTRG}}$ pin changes state from high to low after the ADST bit is set to 1 (A/D conversion starts) |
| A/D conversion stop condition | <ul style="list-style-type: none"> Completion of A/D conversion (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 | Completion of A/D conversion |
| 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⁾ |
| Reading of result of A/D converter | Read one of registers AD0 to AD7 that corresponds to the selected pin |

NOTE:

1. AN0_0 to AN0_7, and AN2_0 to AN2_7 can be used in the same way as AN0 to AN7.

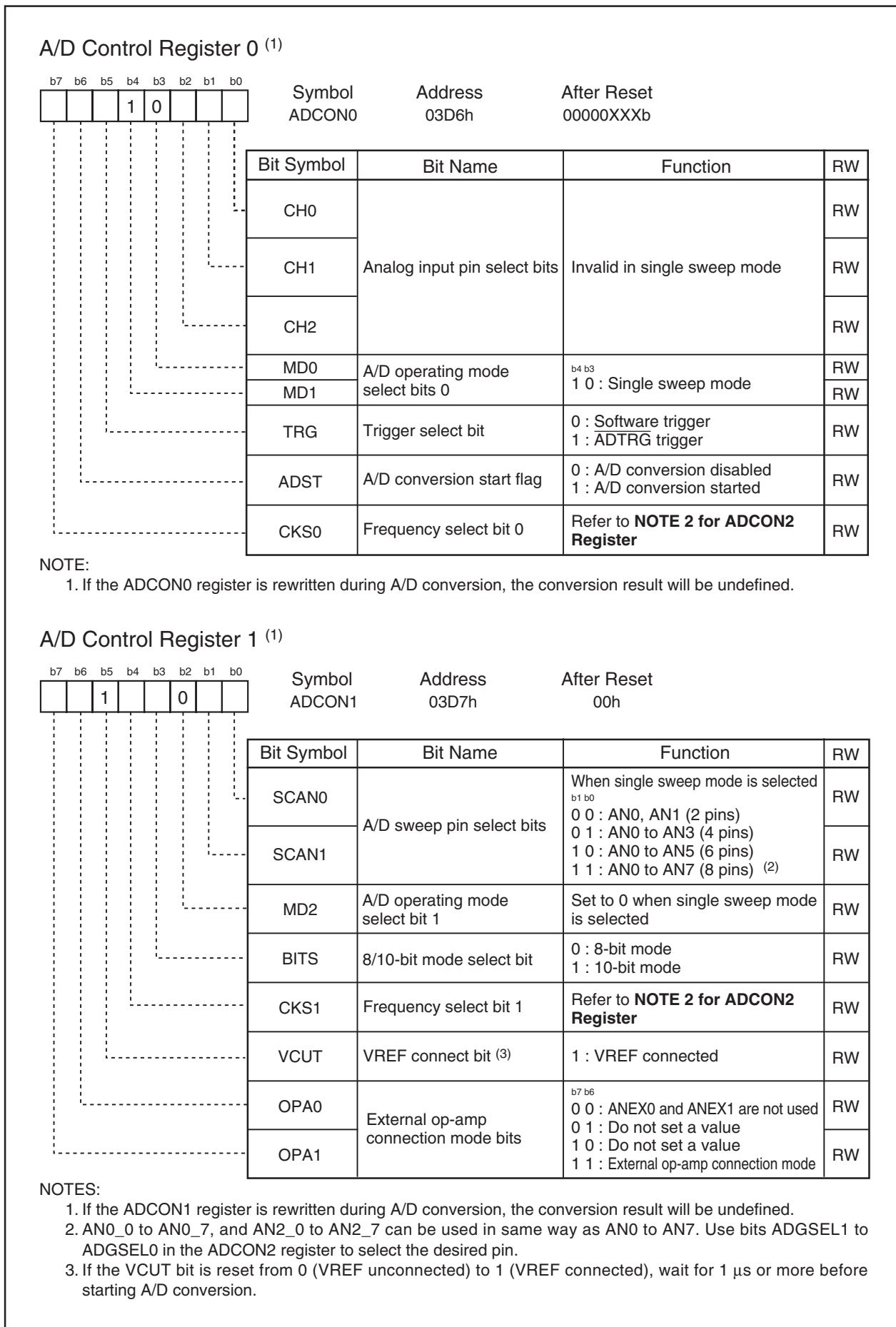


Figure 16.6 Registers ADCON0 and ADCON1 in Single Sweep Mode

16.1.4 Repeat Sweep Mode 0

In repeat sweep mode 0, analog voltage applied to selected pins is repeatedly converted to a digital code. Table 16.5 lists the Repeat Sweep Mode 0 Specifications. Figure 16.7 shows Registers ADCON0 and ADCON1 in Repeat Sweep Mode 0.

Table 16.5 Repeat Sweep Mode 0 Specifications

| Item | Specification |
|-------------------------------------|---|
| Function | Bits SCAN1 to SCAN0 in the ADCON1 register and bits ADGSEL1 to ADGSEL0 in the ADCON2 register select pins. Analog voltage applied to the 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) The ADST bit in the ADCON0 register is set to 1 (A/D conversion starts) • When the TRG bit is 1 (ADTRG trigger) Input on the ADTRG pin changes state from high to low after the ADST bit is set to 1 (A/D conversion starts) |
| 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⁾ |
| Reading of result of A/D converter | Read one of registers AD0 to AD7 that corresponds to the selected pin |

NOTE:

1. AN0_0 to AN0_7, and AN2_0 to AN2_7 can be used in the same way as AN0 to AN7.

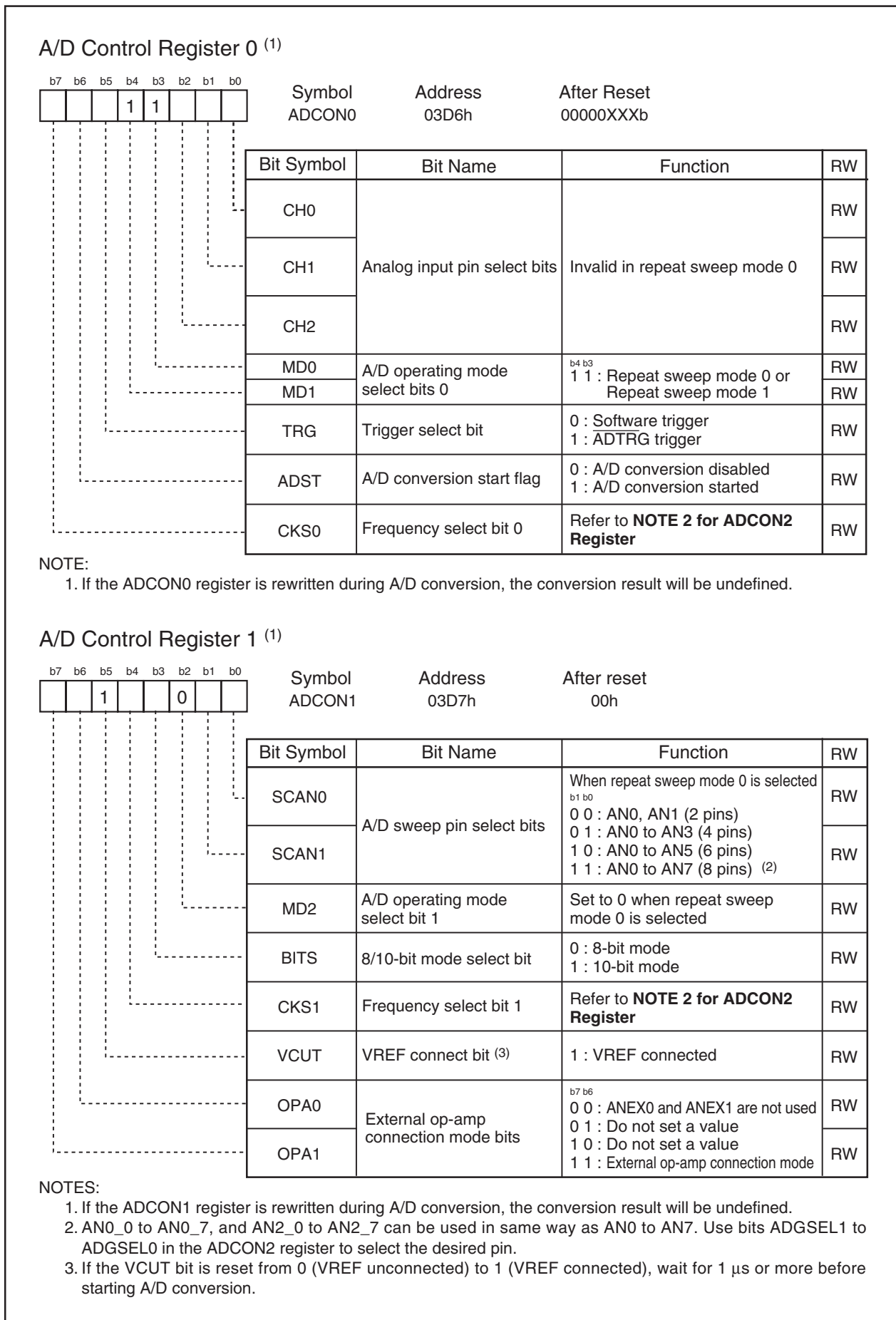


Figure 16.7 Registers ADCON0 and ADCON1 in Repeat Sweep Mode 0

16.1.5 Repeat Sweep Mode 1

In repeat sweep mode 1, analog voltage selectively applied to all pins is repeatedly converted to a digital code. Table 16.6 lists the Repeat Sweep Mode 1 Specifications. Figure 16.8 shows Registers ADCON0 and ADCON1 in Repeat Sweep Mode 1.

Table 16.6 Repeat Sweep Mode 1 Specifications

| Item | Specification |
|---|---|
| Function | The input voltages on all pins selected by bits ADGSEL1 to ADGSEL0 in the ADCON2 register are A/D converted repeatedly, with priority given to pins selected by bits SCAN1 to SCAN0 in the ADCON1 register and bits ADGSEL1 to ADGSEL0. Example : If AN0 selected, input voltages are A/D converted in order of 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) The ADST bit in the ADCON0 register is set to 1 (A/D conversion starts) When the TRG bit is 1 ($\overline{\text{ADTRG}}$ trigger) Input on the $\overline{\text{ADTRG}}$ pin changes state from high to low after the ADST bit is set to 1 (A/D conversion starts) |
| A/D conversion stop condition | Set the ADST bit to 0 (A/D conversion halted) |
| Interrupt request generation timing | None generated |
| Analog input pins to be given priority when A/D converted | Select from AN0 (1 pin), AN0 to AN1 (2 pins), AN0 to AN2 (3 pins), AN0 to AN3 (4 pins) ⁽¹⁾ |
| Reading of result of A/D converter | Read one of registers AD0 to AD7 that corresponds to the selected pin |

NOTE:

1. AN0_0 to AN0_7, and AN2_0 to AN2_7 can be used in the same way as AN0 to AN7.

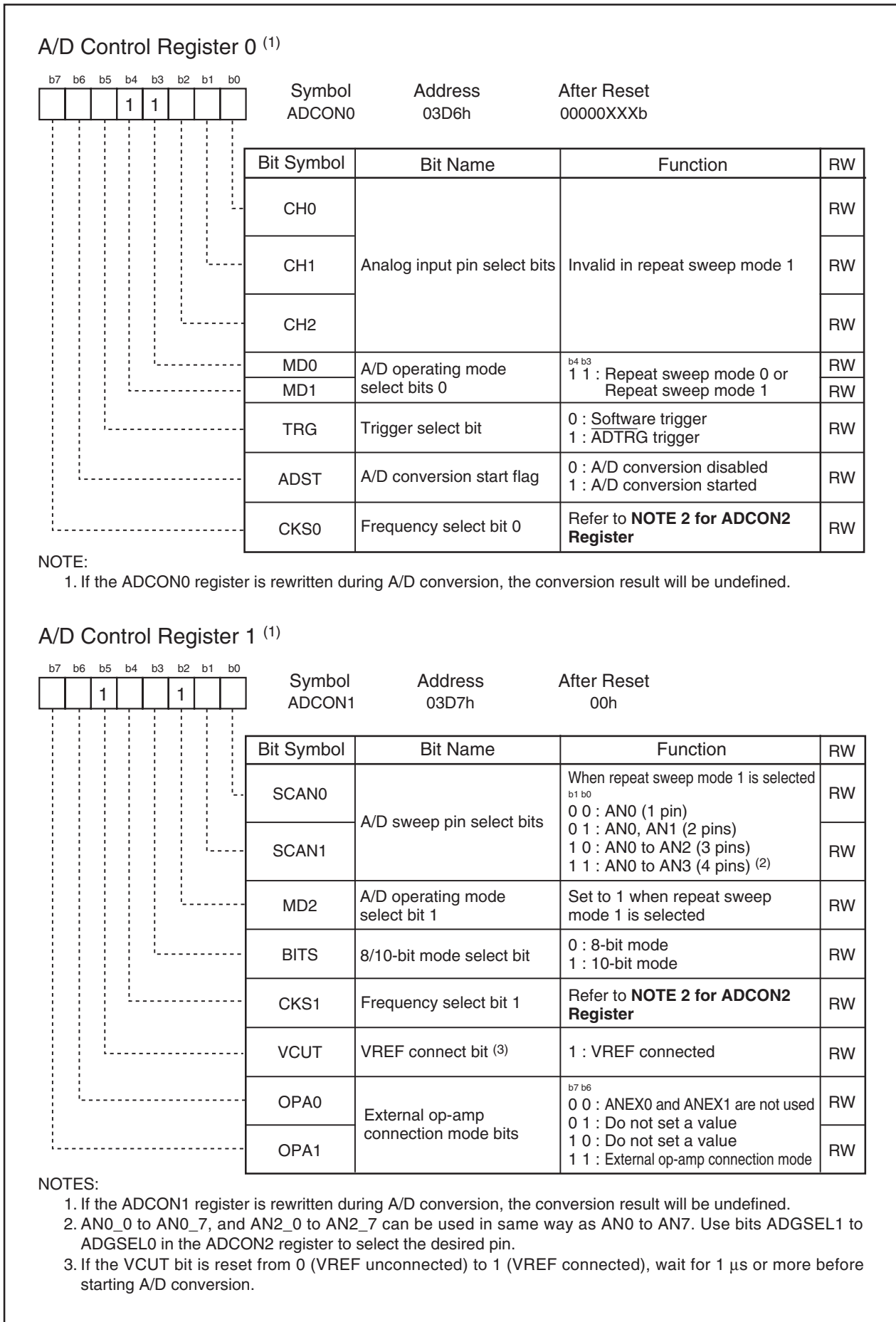


Figure 16.8 Registers ADCON0 and ADCON1 in Repeat Sweep Mode 1

16.2 Function

16.2.1 Resolution Select Function

The desired resolution can be selected using the BITS bit in the ADCON1 register. If the BITS bit is set to 1 (10-bit conversion accuracy), the A/D conversion result is stored in the bits 0 to 9 in the ADi register ($i = 0$ to 7). If the BITS bit is set to 0 (8-bit conversion accuracy), the A/D conversion result is stored in the bits 0 to 7 in the ADi register.

16.2.2 Sample and Hold

If the SMP bit in the ADCON2 register is set to 1 (with sample and hold), the conversion speed per pin is increased to $28 \phi_{AD}$ cycles for 8-bit resolution or $33 \phi_{AD}$ cycles for 10-bit resolution. Sample and hold is effective in all operating modes. Select whether or not to use the sample and hold function before starting A/D conversion.

16.2.3 Extended Analog Input Pins

In one-shot and repeat modes, pins ANEX0 and ANEX1 can be used as analog input pins. Use bits OPA1 to OPA0 in the ADCON1 register to select whether or not use ANEX0 and ANEX1.

The A/D conversion results of ANEX0 and ANEX1 inputs are stored in registers AD0 and AD1, respectively.

16.2.4 External Operation Amplifier (Op-Amp) Connection Mode

Multiple analog inputs can be amplified using a single external op-amp via pins ANEX0 and ANEX1.

Set bits OPA1 to OPA0 in the ADCON1 register to 11b (external op-amp connection mode). The inputs from ANi ($i = 0$ to 7)⁽¹⁾ are output from the ANEX0 pin. Amplify this output with an external op-amp before sending it back to the ANEX1 pin. The A/D conversion result is stored in the corresponding ADi register. The A/D conversion speed depends on the response characteristics of the external op-amp.

Figure 16.9 shows an External Op-Amp Connection.

NOTE:

1. AN0_i and AN2_i can be used the same as ANi.

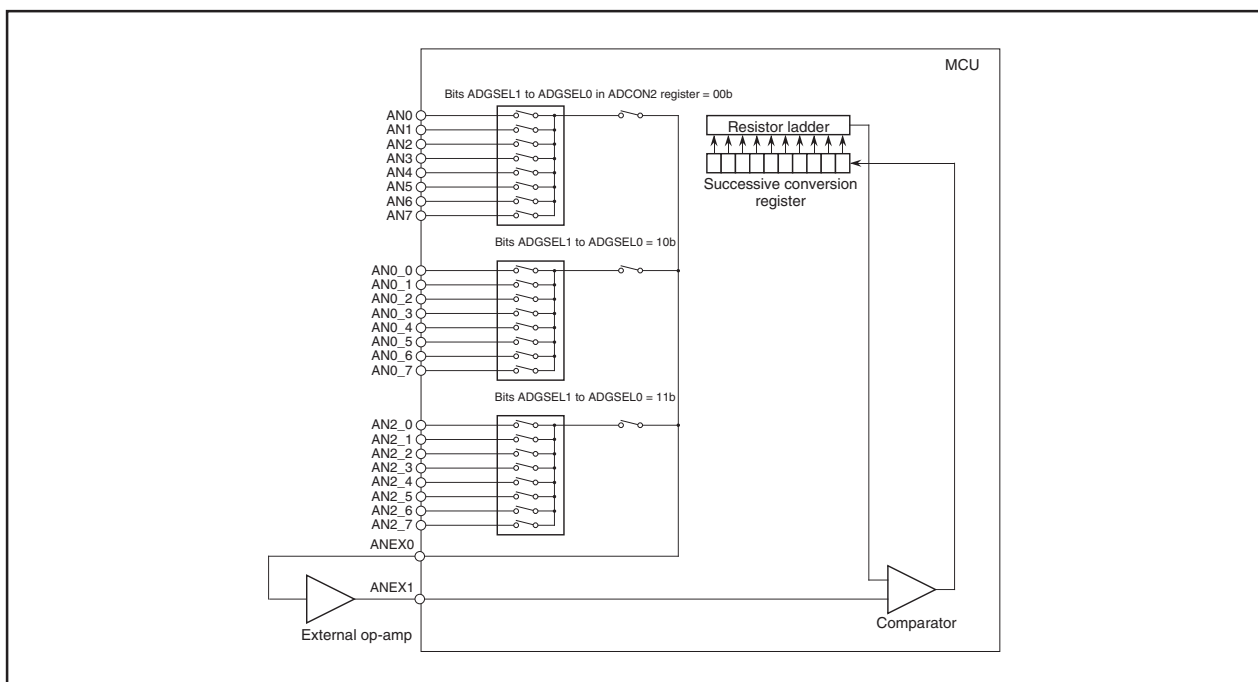


Figure 16.9 External Op-Amp Connection

16.2.5 Current Consumption Reducing Function

When not using the A/D converter, its resistor ladder and reference voltage input pin (VREF) can be separated using the VCUT bit in the ADCON1 register. When separated, no current will flow from the VREF pin into the resistor ladder, helping to reduce the power consumption of the chip.

To use the A/D converter, set the VCUT bit to 1 (VREF connected) and then set the ADST bit in the ADCON0 register to 1 (A/D conversion start). The VCUT and ADST bits cannot be set to 1 at the same time.

Nor can the VCUT bit be set to 0 (VREF unconnected) during A/D conversion.

Note that this does not affect VREF for the D/A converter (irrelevant).

16.2.6 Output Impedance of Sensor under A/D Conversion

To carry out A/D conversion properly, charging the internal capacitor C shown in Figure 16.10 has to be completed within a specified period of time. T (sampling time) as the specified time. Let output impedance of sensor equivalent circuit be R0, internal resistance of MCU be R, precision (error) of the A/D converter be X, and the resolution of A/D converter be Y (Y is 1024 in 10-bit mode, and 256 in 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 16.10 shows the 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 time T. (0.1/1024) means that A/D precision drop due to insufficient capacitor charge is held to 0.1 LSB at time of A/D conversion in 10-bit mode. Actual error however is the value of absolute precision added to 0.1 LSB.

When f(φAD) = 10 MHz, T = 0.3 μs 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 = 0.3 μs, R = 7.8 kΩ, C = 1.5 pF, X = 0.1, and Y = 1024. Hence,

$$R0 = -\frac{0.3 \times 10^{-6}}{1.5 \times 10^{-12} \cdot \ln \frac{0.1}{1024}} - 7.8 \times 10^3 = 13.9 \times 10^3$$

Thus, the allowable output impedance of the sensor equivalent circuit, making the precision (error) 0.1 LSB or less, is approximately 13.9 kΩ. maximum.

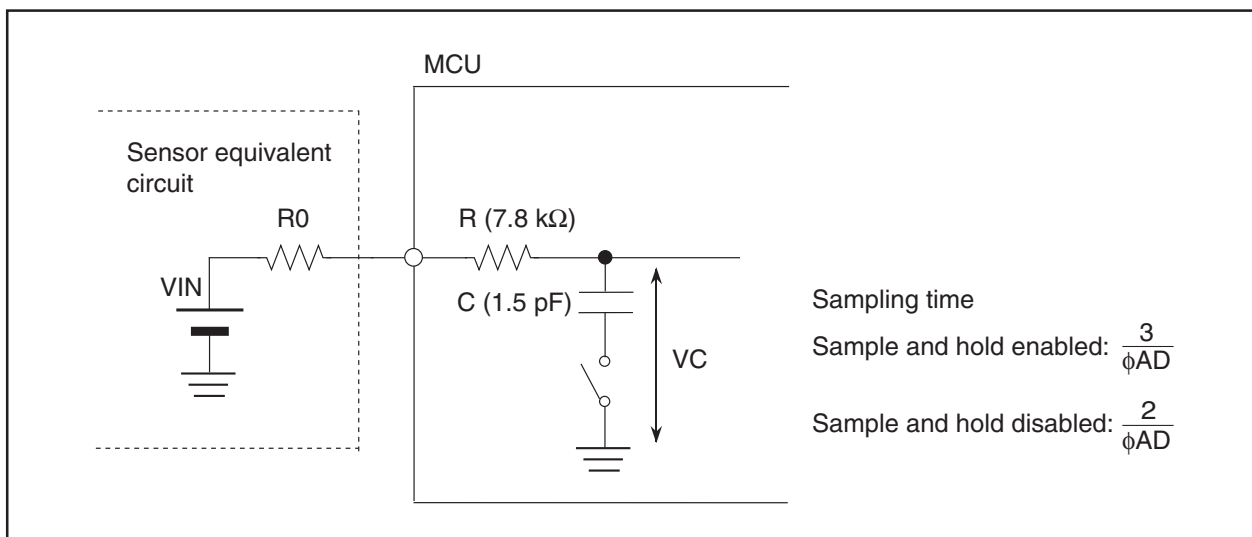


Figure 16.10 Analog Input Pin and External Sensor Equivalent Circuit

17. D/A Converter

This is an 8-bit, R-2R type D/A converter. These are two independent D/A converters.

D/A conversion is performed by writing to the DA_i register (i = 0, 1). To output the result of conversion, set the DA_iE bit in the DACON register to 1 (output enabled). Before D/A conversion can be used, the corresponding port direction bit is set to 0 (input mode). Setting the DA_iE bit to 1 removes a pull-up from the corresponding port.

Output analog voltage (V) is determined by a set value (n : decimal) in the DA_i register.

$$V = V_{REF} \times n / 256 \quad (n = 0 \text{ to } 255)$$

V_{REF} : reference voltage

Table 17.1 lists the D/A converter Performance. Figure 17.1 shows the D/A Converter Block Diagram.

Figure 17.2 shows the D/A converter-related registers. Figure 17.3 shows the D/A Converter Equivalent Circuit.

Table 17.1 D/A Converter Performance

| Item | Performance |
|-----------------------|--------------------------|
| D/A conversion method | R-2R method |
| Resolution | 8 bits |
| Analog output pin | 2 channels (DA0 and DA1) |

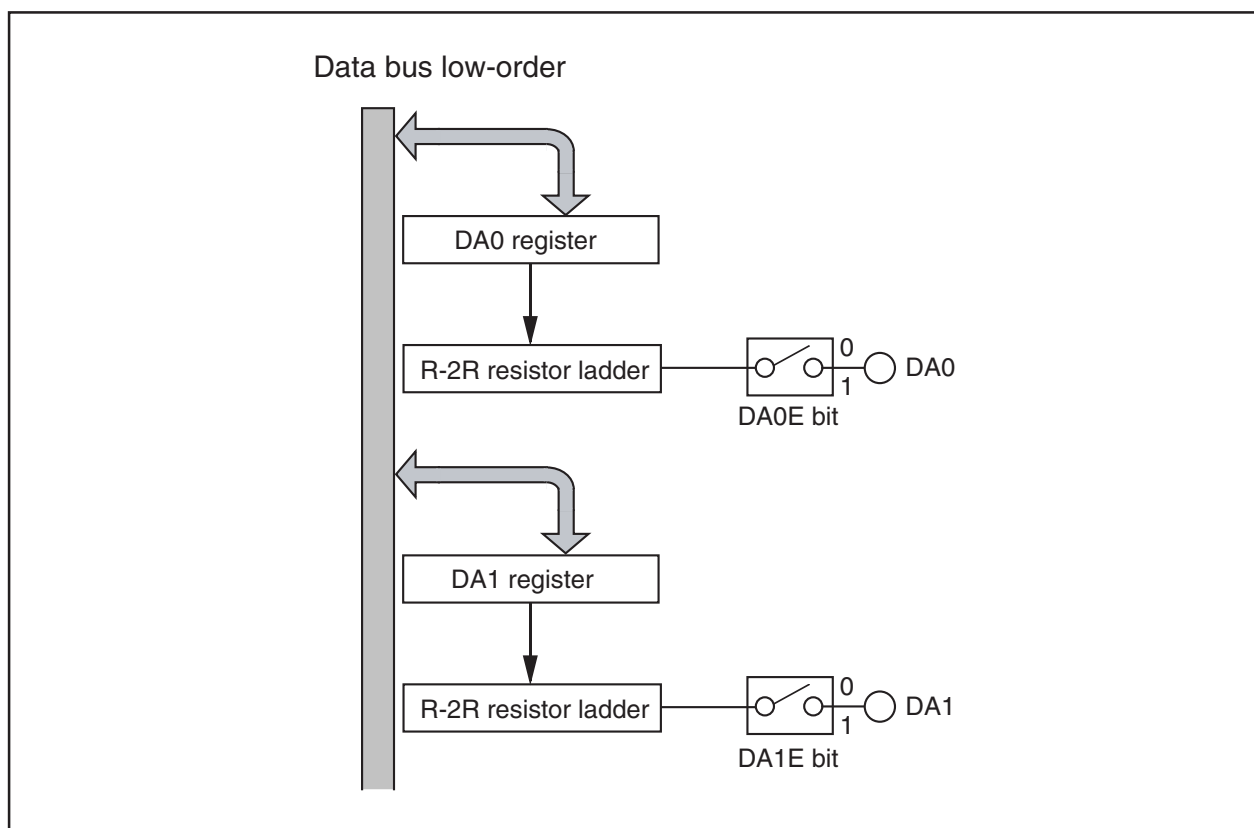


Figure 17.1 D/A Converter Block Diagram

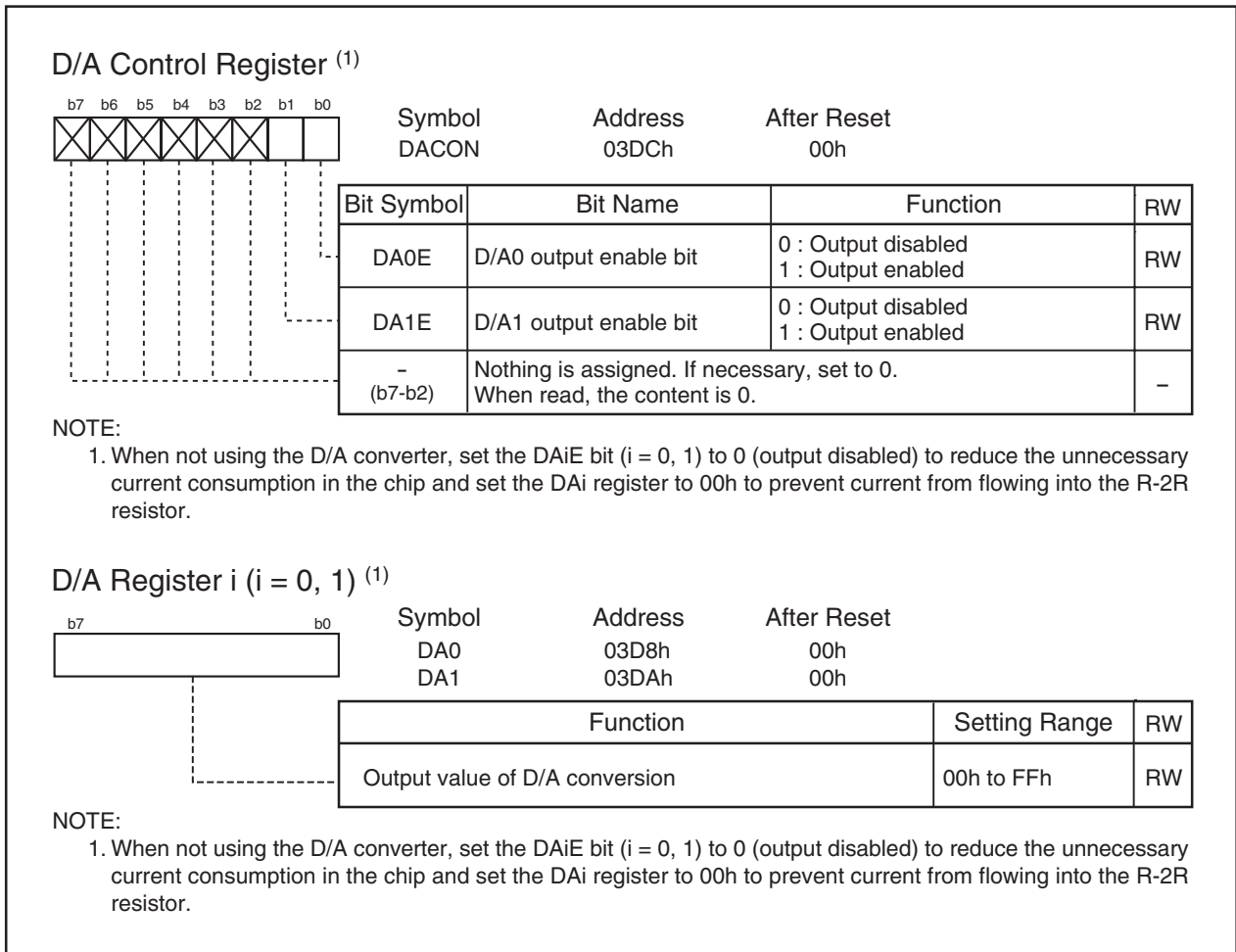


Figure 17.2 Registers DACON, DA0, and DA1

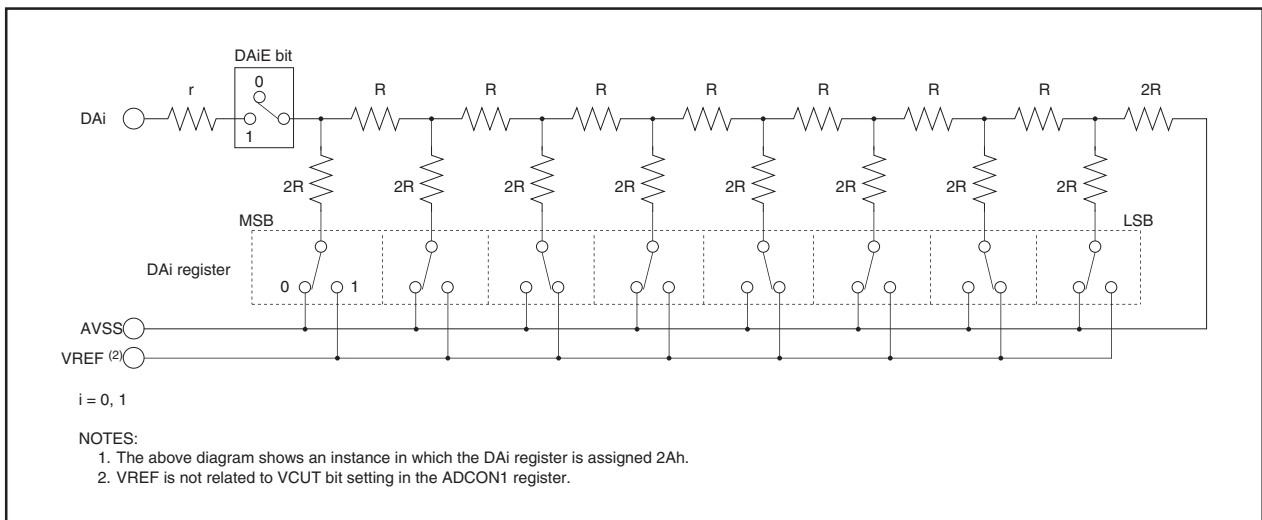


Figure 17.3 D/A Converter Equivalent Circuit

18. CRC Calculation

The Cyclic Redundancy Check (CRC) operation detects an error in data blocks. The MCU uses a generator polynomial of CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$) to generate CRC code.

The CRC code consists of 16 bits which are generated for each data block in given length, separated in 8-bit unit. After the initial value is set in the CRCD register, the CRC code is set in that register each time one byte of data is written to the CRCIN register. CRC code generation for one-byte data is finished in two cycles.

Figure 18.1 shows the CRC Circuit Block Diagram. Figure 18.2 shows the CRC-related registers. Figure 18.3 shows the calculation example using the CRC operation.

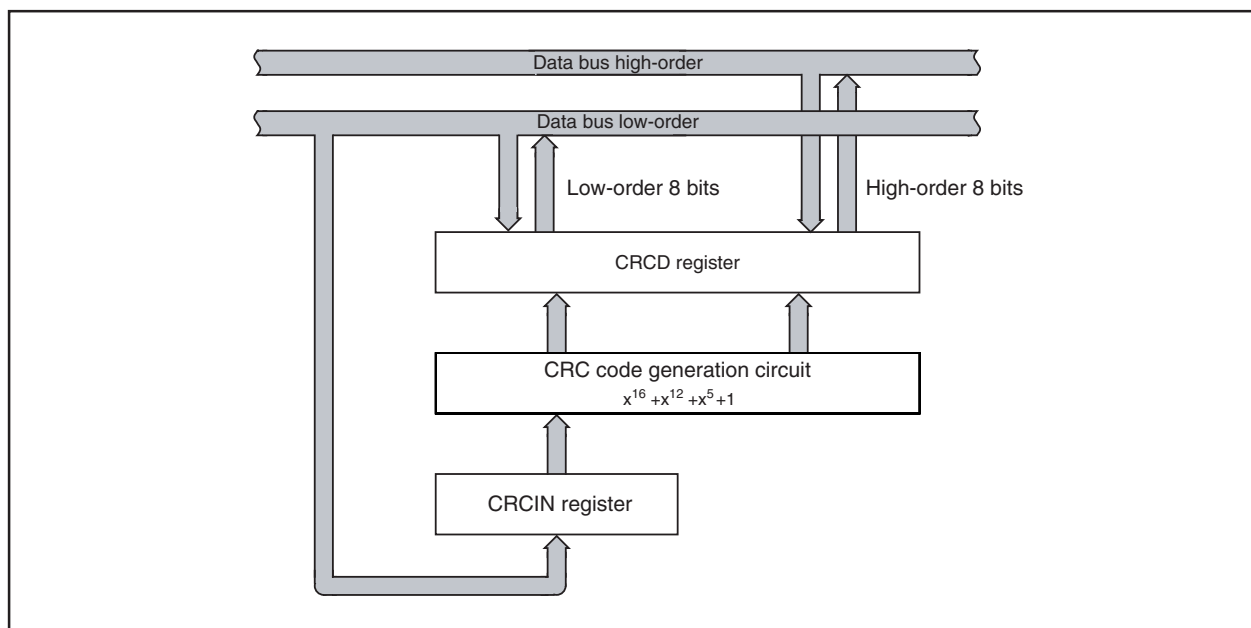


Figure 18.1 CRC Circuit Block Diagram

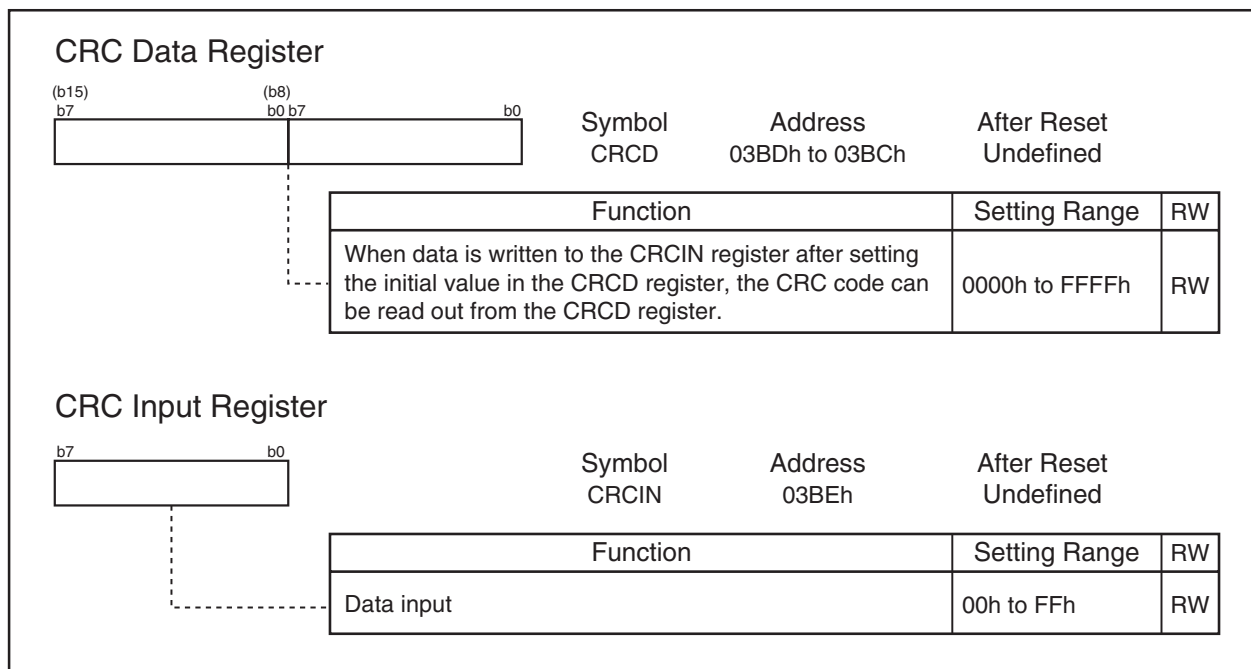


Figure 18.2 Registers CRCD and CRCIN

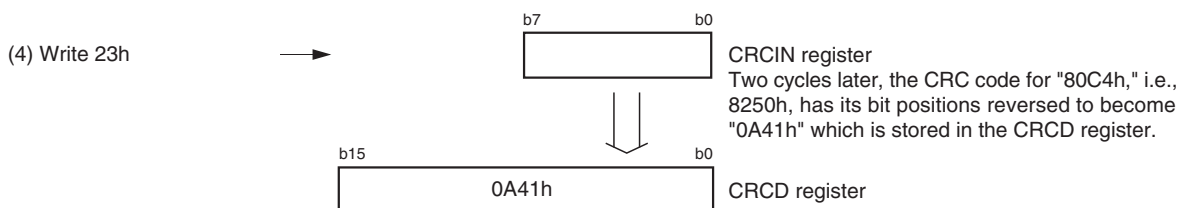
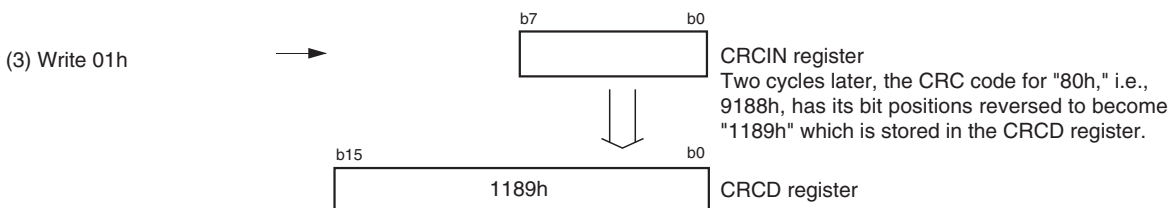
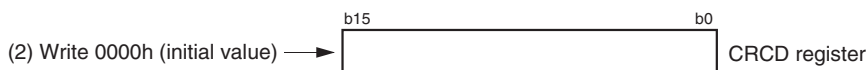
Setup procedure and CRC operation when generating CRC code "80C4h"

- CRC operation performed by the M16C

CRC code: Remainder of a division in which the value written to the CRCIN register with its bit positions reversed is divided by the generator polynomial
 Generator polynomial: $X^6 + X^{12} + X^5 + 1(1\ 0001\ 0000\ 0010\ 0001b)$

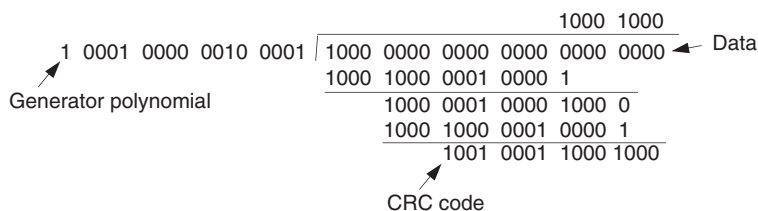
- Setting procedure

(1) Reverse the bit positions of the value "80C4h" by program in 1-byte unit.
 "80h" → "01h", "C4h" → "23h"



- Details of CRC operation

As shown in (3) above, bit position of "01h" (00000001b) written to the CRCIN register is inversed and becomes "10000000b". Add "1000 0000 0000 0000 0000 0000b", as "10000000b" plus 16 digits, to "0000 0000 0000 0000 0000 0000b", as "0000 0000 0000 0000b" plus 8 digits as the default value of the CRCD register to perform the modulo-2 division.



Modulo-2 operation is operation that complies with the law given below.

0 + 0 = 0
 0 + 1 = 1
 1 + 0 = 1
 1 + 1 = 0
 -1 = 1

"0001 0001 1000 1001b (1189h)", the remainder "1001 0001 1000 1000b (9188h)" with inversed bit position, can be read from the CRCD register.

When going on to (4) above, "23h (00100011b)" written in the CRCIN register is inversed and becomes "11000100b". Add "1100 0100 0000 0000 0000 0000b", as "11000100b" plus 16 digits, to "1001 0001 1000 1000 0000 0000b", as "1001 0001 1000 1000b" plus 8 digits as a remainder of (3) left in the CRCD register to perform the modulo-2 division. "0000 1010 0100 0001b (0A41h)", the remainder with inversed bit position, can be read from CRCD register.

Figure 18.3 CRC Calculation

19. CAN Module

The CAN (Controller Area Network) module for the M16C/6N Group (M16C/6N4) of MCUs is a communication controller implementing the CAN 2.0B protocol. The M16C/6N Group (M16C/6N4) contains two CAN modules which can transmit and receive messages in both standard (11-bit) ID and extended (29-bit) ID formats.

Figure 19.1 shows the CAN Module Block Diagram.

External CAN bus driver and receiver are required.

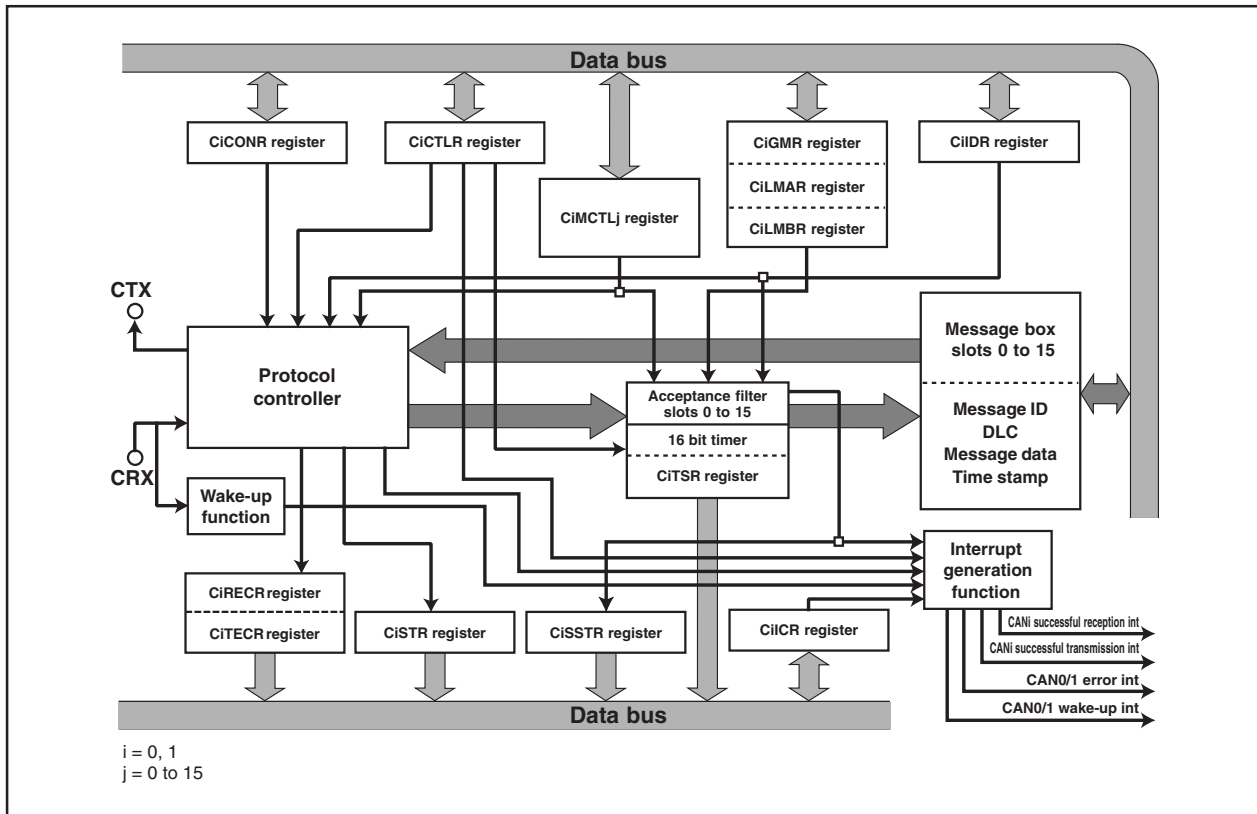


Figure 19.1 CAN Module Block Diagram

| | |
|--------------------------------|---|
| 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 CiGMR register (i = 0, 1), the CiLMAR register, or the CiLMBR 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: | CAN0/1 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. CANi successful reception interrupt, CANi successful transmission interrupt, CAN0/1 error interrupt, and CAN0/1 wake-up interrupt. |

19.1 CAN Module-Related Registers

The CAN_i (i = 0, 1) module has the following registers.

19.1.1 CAN_i Message Box (i = 0, 1)

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.

19.1.2 Acceptance Mask Registers

A CAN module is equipped with 3 masks for the acceptance filter.

- CAN_i global mask register (i = 0, 1) (CiGMR register: 6 bytes)
Configuration of the masking condition for acceptance filtering processing to slots 0 to 13
- CAN_i local mask A register (CiLMAR register: 6 bytes)
Configuration of the masking condition for acceptance filtering processing to slot 14
- CAN_i local mask B register (CiLMBR register: 6 bytes)
Configuration of the masking condition for acceptance filtering processing to slot 15

19.1.3 CAN SFR Registers

- CAN_i message control register j (i = 0, 1, j = 0 to 15) (CiMCTL_j register: 8 bits × 16)
Control of transmission and reception of a corresponding slot
- CAN_i control register (CiCTLR register: 16 bits)
Control of the CAN protocol
- CAN_i status register (CiSTR register: 16 bits)
Indication of the protocol status
- CAN_i slot status register (CiSSTR register: 16 bits)
Indication of the status of contents of each slot
- CAN_i interrupt control register (CiICR register: 16 bits)
Selection of “interrupt enabled or disabled” for each slot
- CAN_i extended ID register (CiIDR register: 16 bits)
Selection of ID format (standard or extended) for each slot
- CAN_i configuration register (CiCONR register: 16 bits)
Configuration of the bus timing
- CAN_i receive error count register (CiRECR 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.
- CAN_i transmit error count register (CiTECR 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.
- CAN_i time stamp register (CiTSR register: 16 bits)
Indication of the value of the time stamp counter
- CAN_i acceptance filter support register (CiAFS register: 16 bits)
Decoding the received ID for use by the acceptance filter support unit

Explanation of each register is given below.

19.2 CANi Message Box (i = 0, 1)

Table 19.1 shows the CANi Message Box Memory Mapping.

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 in the CiCTRL register.

Table 19.1 CANi Message Box Memory Mapping

| Address | | Message Content (Memory Mapping) | |
|---------------------|---------------------|----------------------------------|----------------------------|
| CAN0 | CAN1 | Byte Access (8 bits) | Word Access (16 bits) |
| 0060h + n × 16 + 0 | 0260h + n × 16 + 0 | SID10 to SID6 | SID5 to SID0 |
| 0060h + n × 16 + 1 | 0260h + n × 16 + 1 | SID5 to SID0 | SID10 to SID6 |
| 0060h + n × 16 + 2 | 0260h + n × 16 + 2 | EID17 to EID14 | EID13 to EID6 |
| 0060h + n × 16 + 3 | 0260h + n × 16 + 3 | EID13 to EID6 | EID17 to EID14 |
| 0060h + n × 16 + 4 | 0260h + n × 16 + 4 | EID5 to EID0 | Data length code (DLC) |
| 0060h + n × 16 + 5 | 0260h + n × 16 + 5 | Data length code (DLC) | EID5 to EID0 |
| 0060h + n × 16 + 6 | 0260h + n × 16 + 6 | Data byte 0 | Data byte 1 |
| 0060h + n × 16 + 7 | 0260h + n × 16 + 7 | Data byte 1 | Data byte 0 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 0060h + n × 16 + 13 | 0260h + n × 16 + 13 | Data byte 7 | Data byte 6 |
| 0060h + n × 16 + 14 | 0260h + n × 16 + 14 | Time stamp high-order byte | Time stamp low-order byte |
| 0060h + n × 16 + 15 | 0260h + n × 16 + 15 | Time stamp low-order byte | Time stamp high-order byte |

i = 0, 1

n = 0 to 15: the number of the slot

Figures 19.2 and 19.3 show the Bit Mapping in Byte Access and Word Access. The content of each slot remains unchanged unless transmission or reception of a new message is performed.

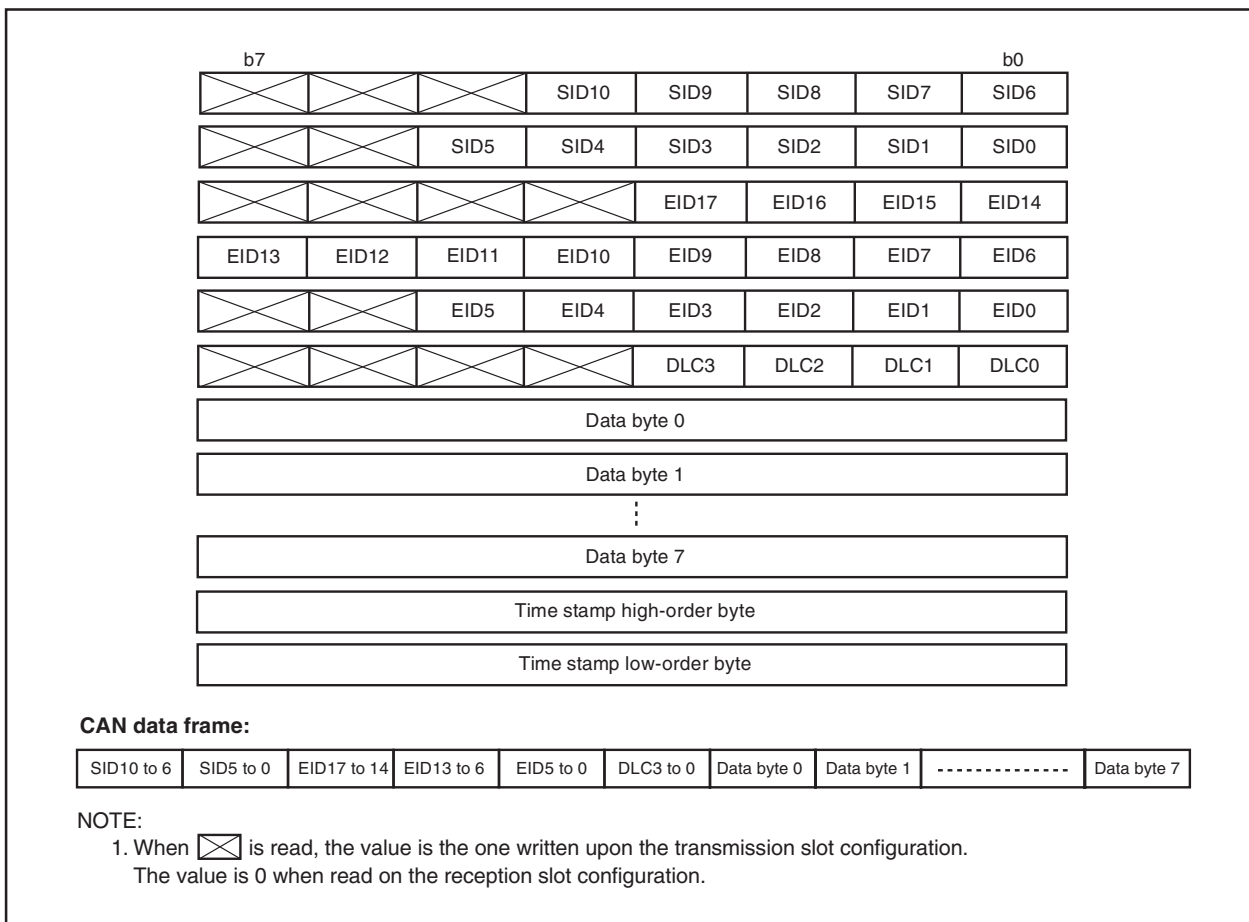


Figure 19.2 Bit Mapping in Byte Access

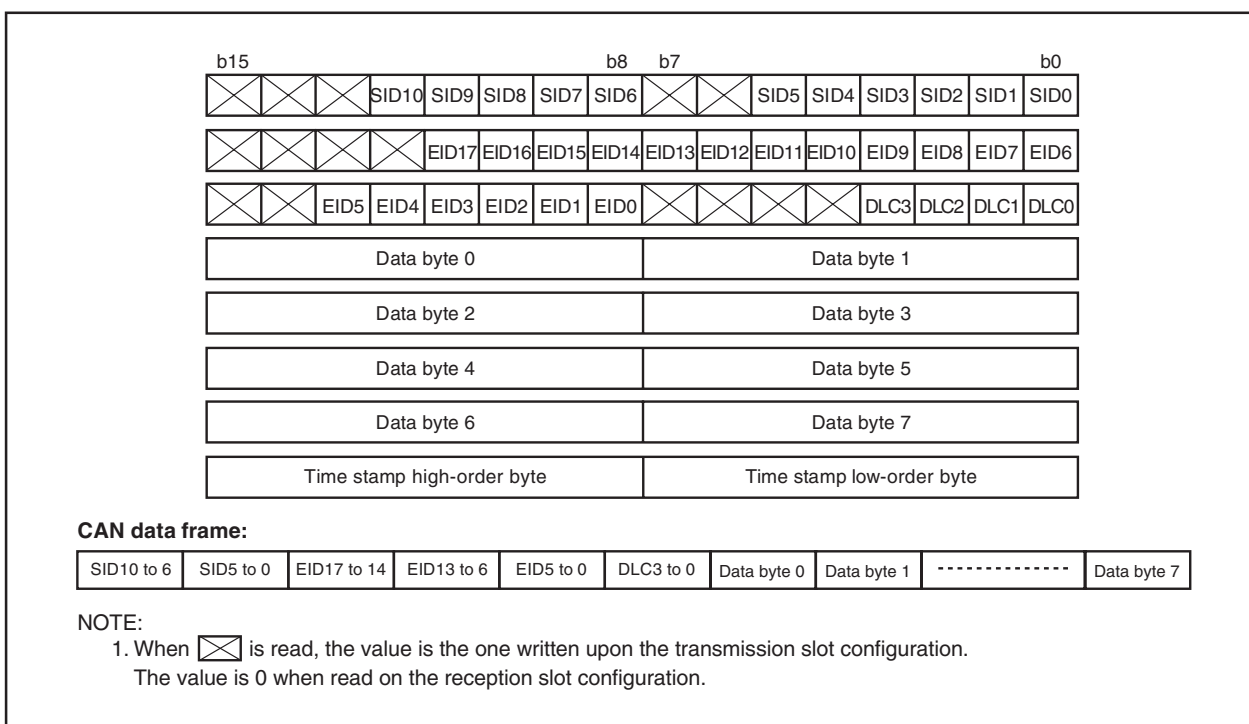


Figure 19.3 Bit Mapping in Word Access

19.3 Acceptance Mask Registers

Figures 19.4 and 19.5 show the Mask registers Bit Mapping (registers CiGMR (i = 0, 1), CiLMAR, and CiLMBR) in Byte Access and Word Access.

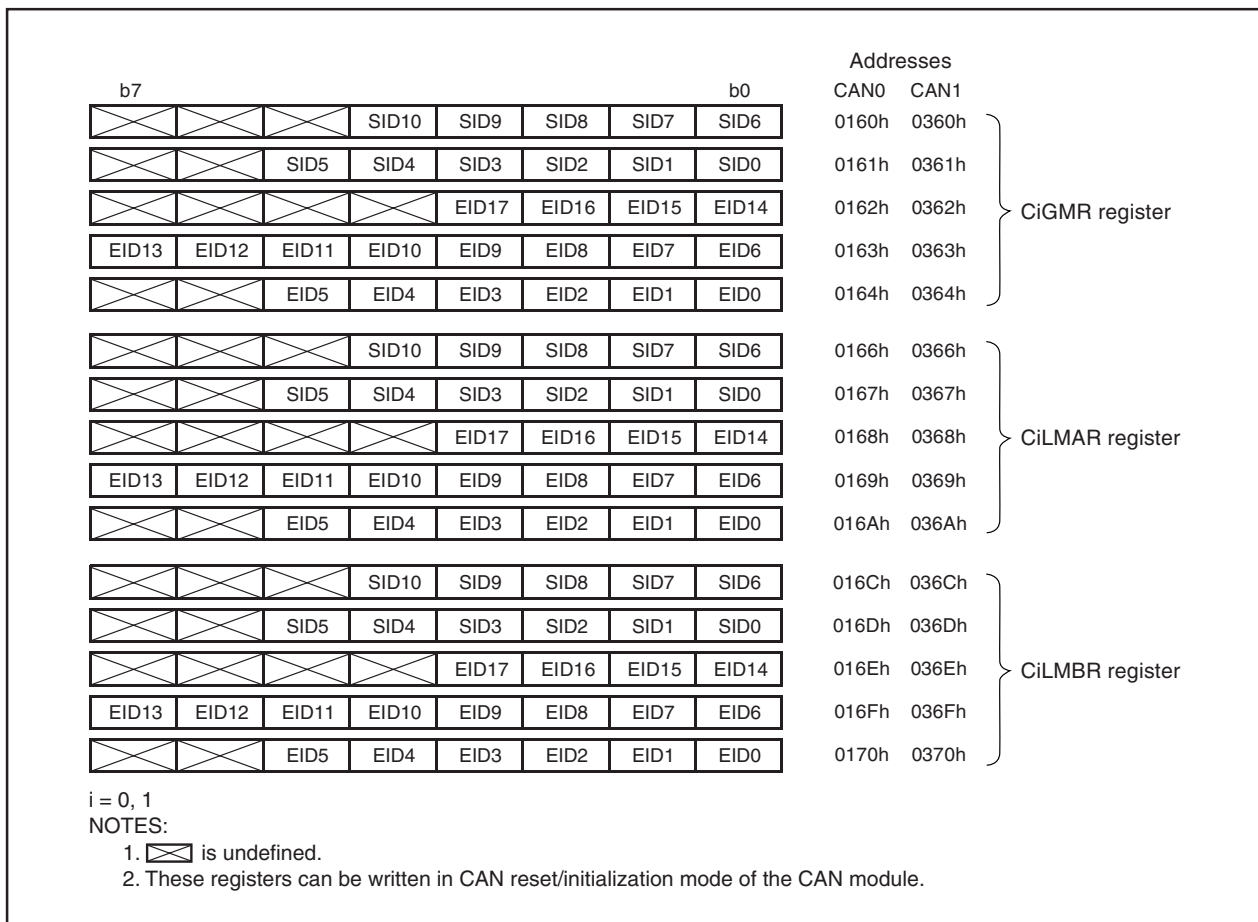


Figure 19.4 Mask Registers Bit Mapping in Byte Access

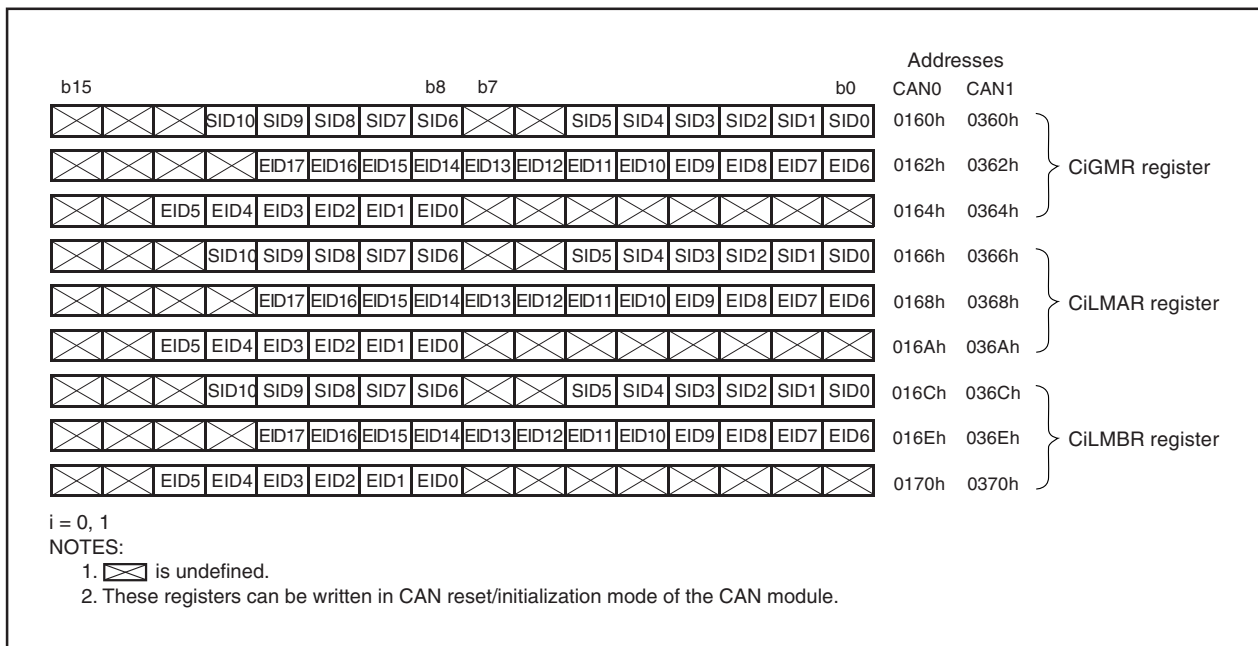


Figure 19.5 Mask Registers Bit Mapping in Word Access

19.4 CAN SFR Registers

Figures 19.6 to 19.11 show the CAN SFR registers.

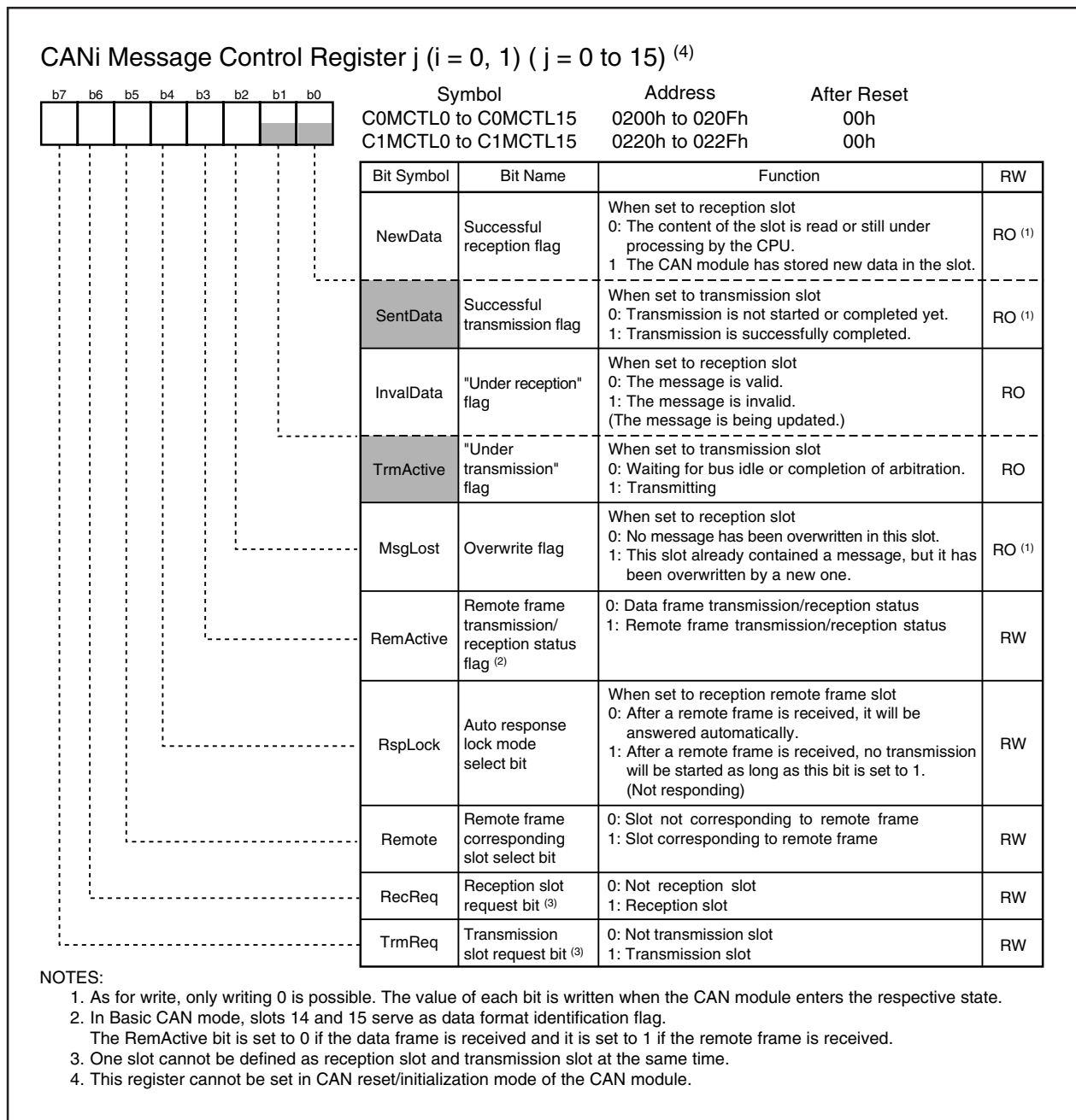


Figure 19.6 Registers C0MCTLj and C1MCTLj

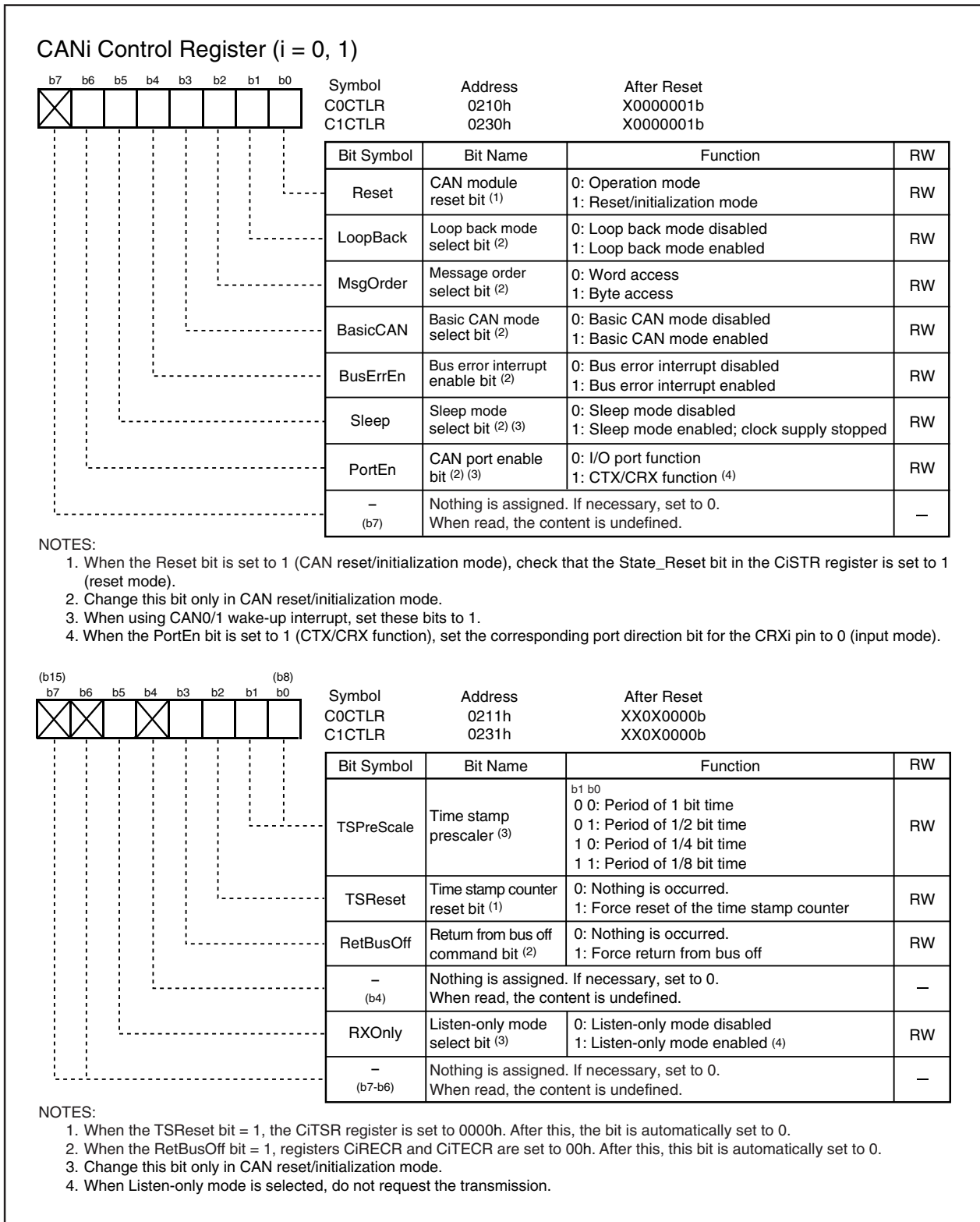


Figure 19.7 Registers C0CTLR and C1CTLR

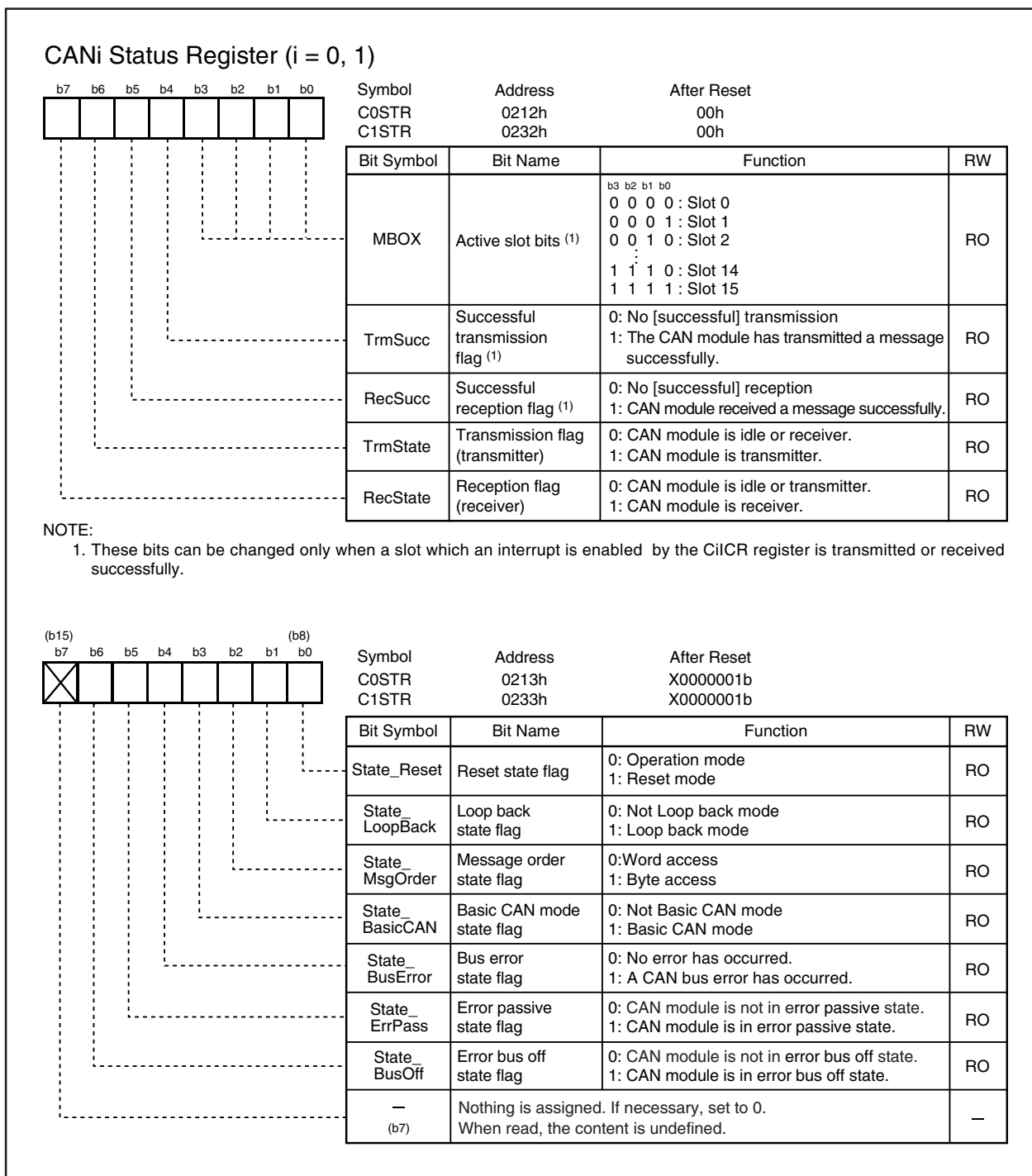


Figure 19.8 Registers C0STR and C1STR

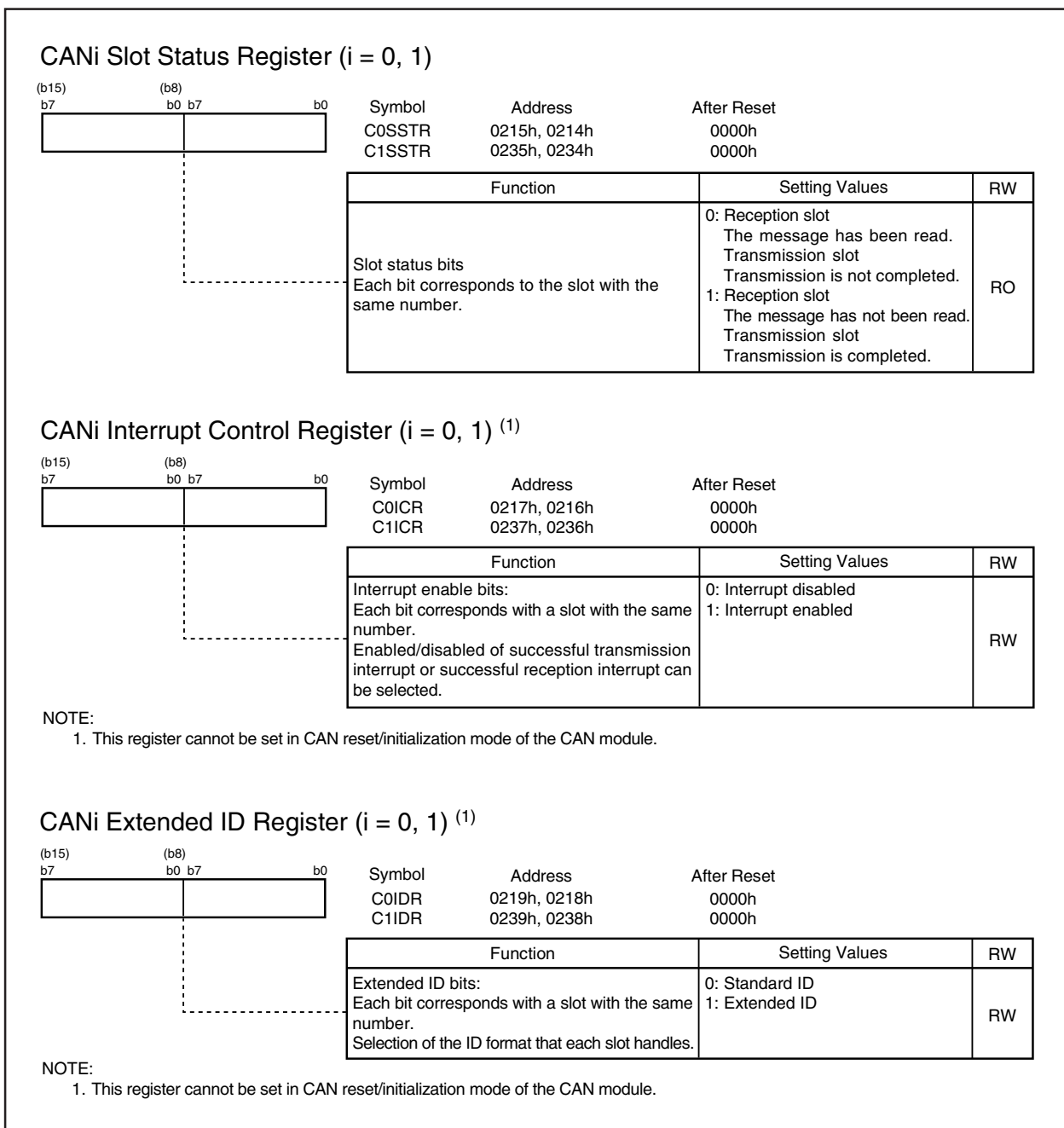


Figure 19.9 Registers C0SSTR, C1SSTR, C0ICR, C1ICR, C0IDR, and C1IDR

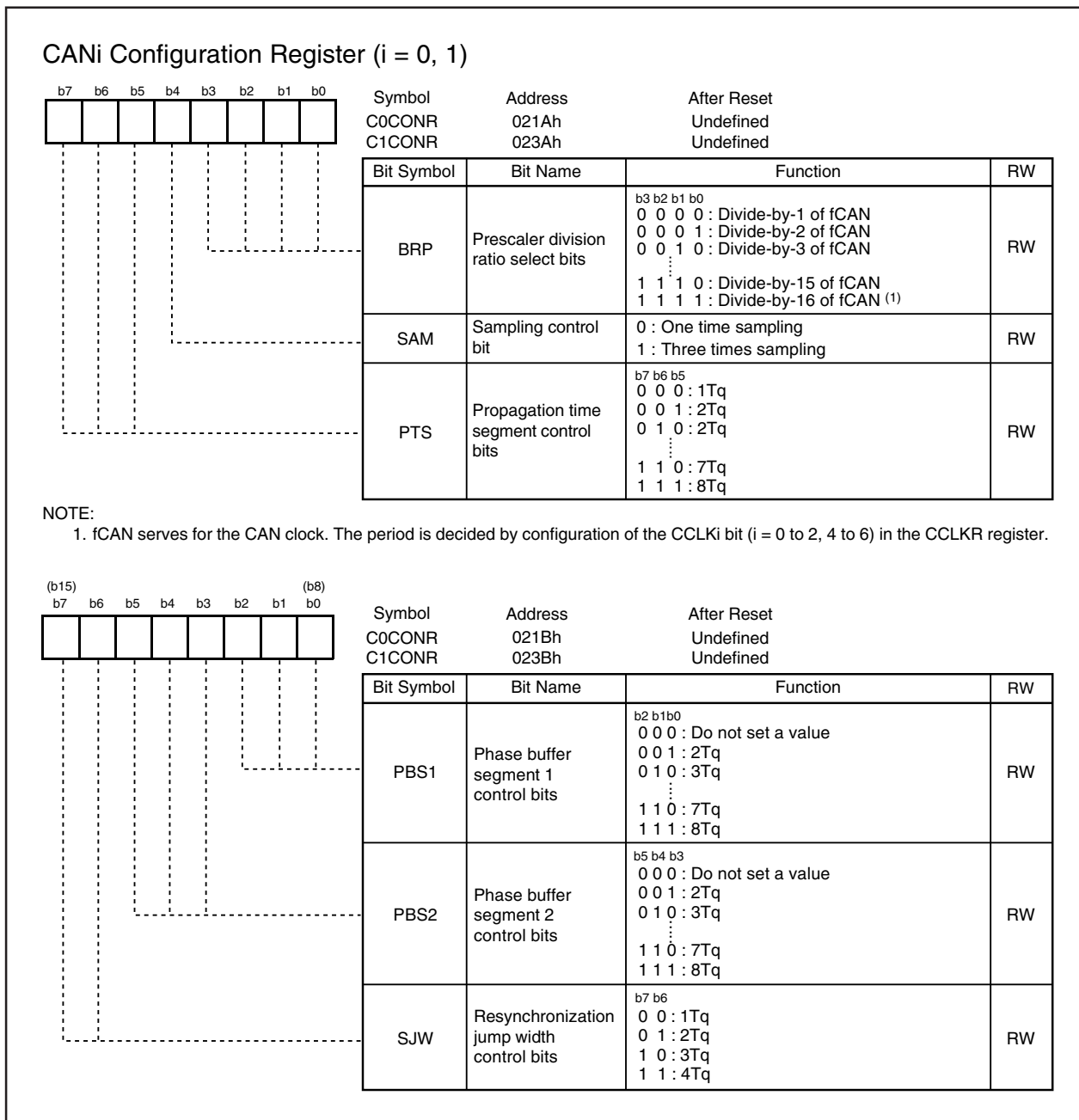


Figure 19.10 Registers C0CONR and C1CONR

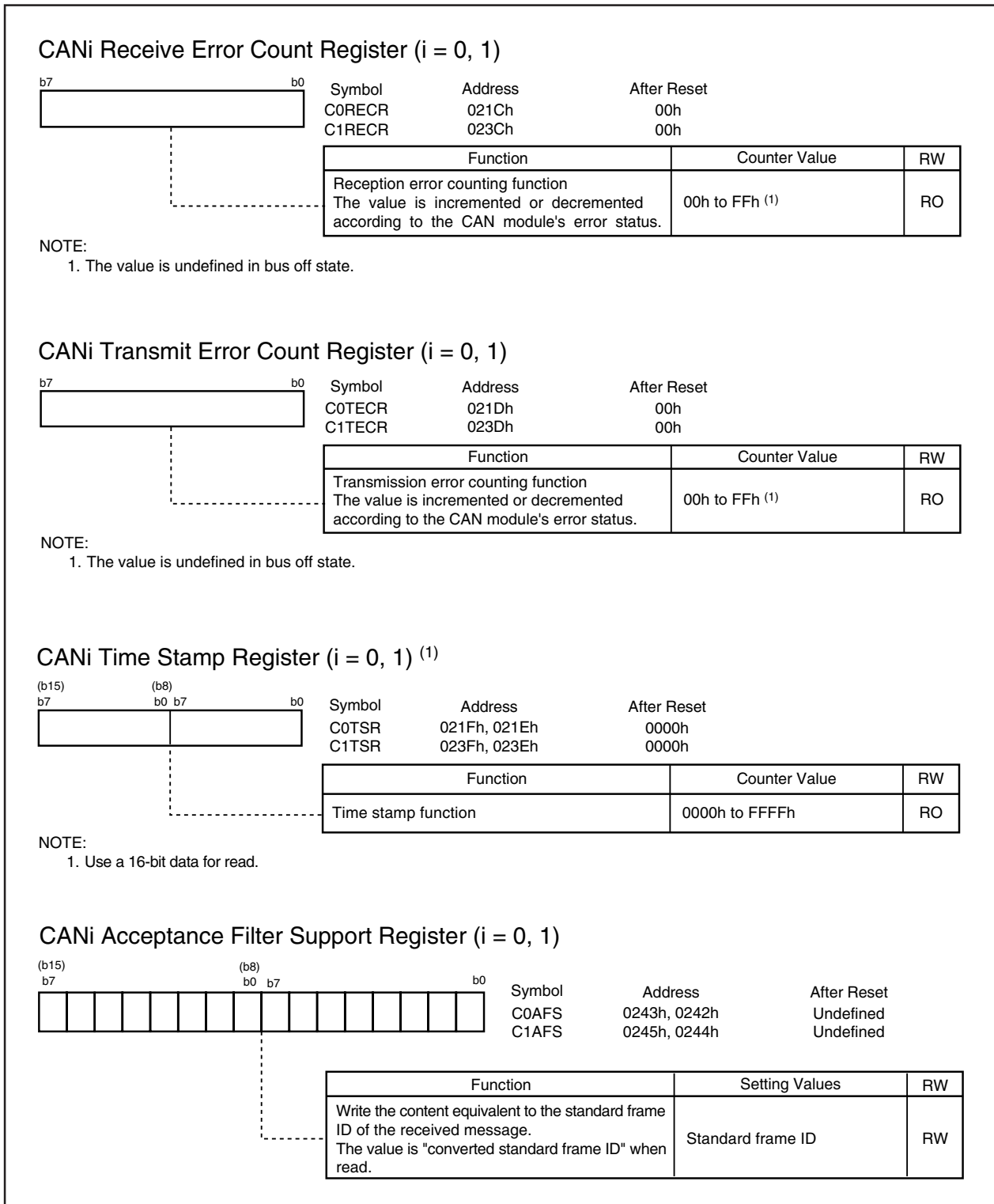


Figure 19.11 Registers C0RECR, C1RECR, C0TECR, C1TECR, C0TSR, C1TSR, C0AFS, and C1AFS

19.5 Operational Modes

The CAN module has the following four operational modes.

- CAN reset/initialization mode
- CAN operation mode
- CAN sleep mode
- CAN interface sleep mode

Figure 19.12 shows the Transition between Operational Modes.

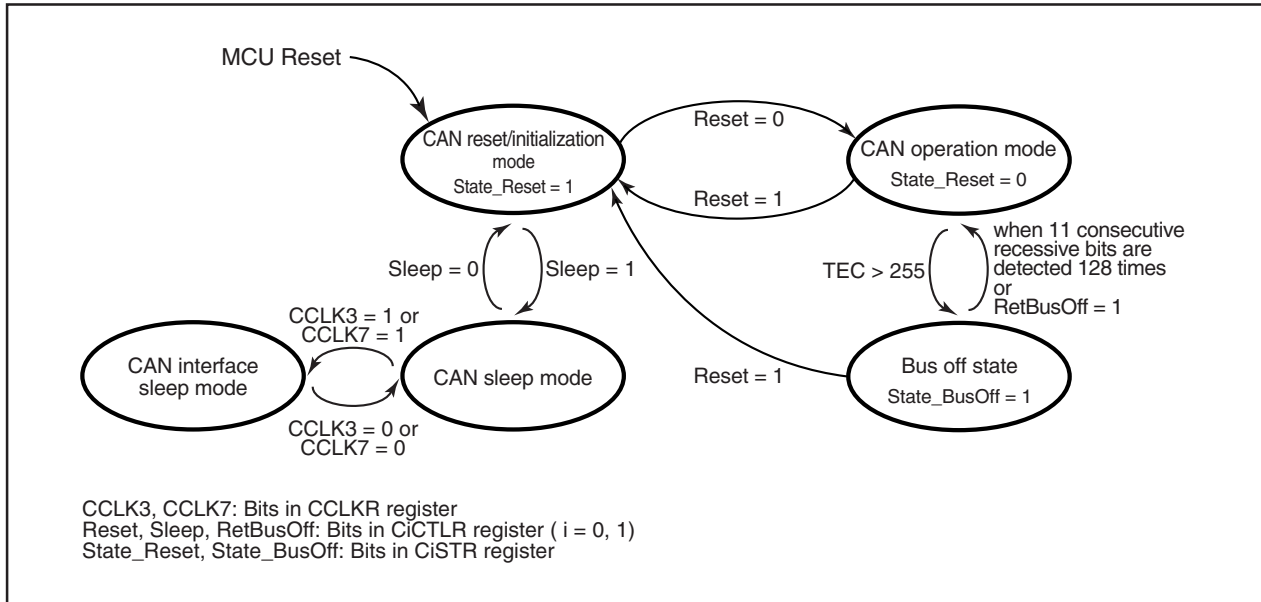


Figure 19.12 Transition between Operational Modes

19.5.1 CAN Reset/Initialization Mode

CAN reset/initialization mode is activated upon MCU reset or by setting the Reset bit in the CiCTLR register (i = 0, 1) to 1. If the Reset bit is set to 1, check that the State_Reset bit in the CiSTR register is set to 1.

Entering CAN reset/initialization mode initiates the following functions by the module:

- CAN communication is impossible.
- When 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 CAN reset/initialization mode is activated.
- Registers CiMCTLj (j = 0 to 15), CiSTR, CiICR, CiIDR, CiRECR, CiTECR, and CiTSR are initialized. All these registers are locked to prevent CPU modification.
- Registers CiCTLR, CiCONR, CiGMR, CiLMAR, and CiLMBR, and the CANi message box retain their contents and are available for CPU access.

19.5.2 CAN Operation Mode

CAN operation mode is activated by setting the Reset bit in the CiCTLR register ($i = 0, 1$) to 0. If the Reset bit is set to 0, check that the State_Reset bit in the CiSTR register is set to 0.

If 11 consecutive recessive bits are detected after entering CAN operation 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 CAN operation mode depending on the error counts.

Within CAN operation 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 CiCTLR register = 1 (Loop back mode enabled).

Figure 19.13 shows the Sub Modes of CAN Operation Mode.

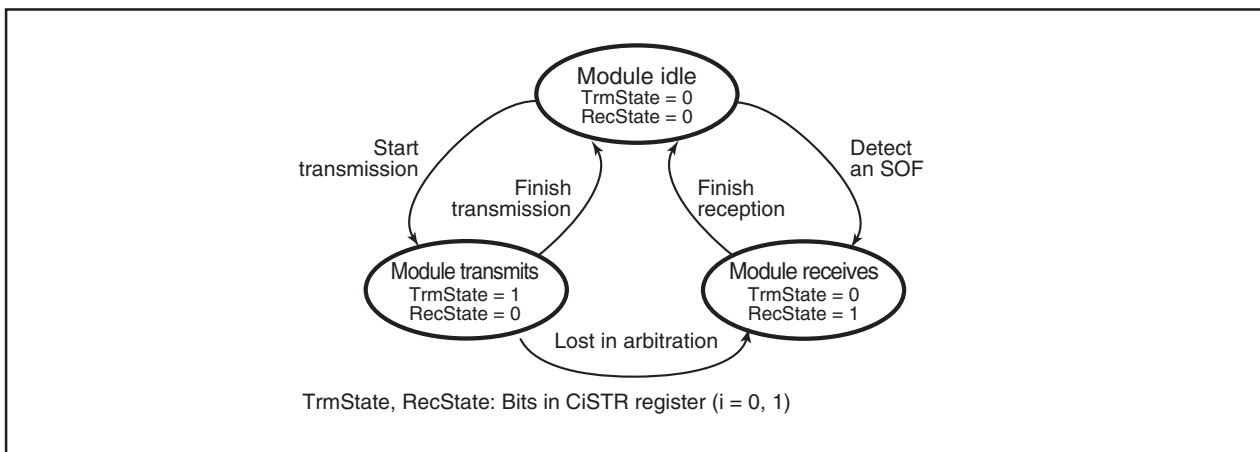


Figure 19.13 Sub Modes of CAN Operation Mode

19.5.3 CAN Sleep Mode

CAN sleep mode is activated by setting the Sleep bit to 1 in the CiCTLR register. It should never be activated from CAN operation mode but only via CAN reset/initialization mode.

Entering CAN sleep mode instantly stops the clock supply to the module and thereby reduces power dissipation.

19.5.4 CAN Interface Sleep Mode

CAN interface sleep mode is activated by setting the CCLK3 or CCLK7 bit in the CCLKR register to 1. It should never be activated but only via CAN sleep mode.

Entering CAN interface sleep mode instantly stops the clock supply to the CPU Interface in the module and thereby reduces power dissipation.

19.5.5 Bus Off State

The bus off state is entered according to the fault confinement rules of the CAN specification. When returning to CAN operation mode from the bus off state, the module has the following two cases. In this time, the value of any CAN registers, except registers CiSTR, CiRECR, and CiTECR, 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 CiCTRL 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.

19.6 CAN Module System Clock Configuration

The M16C/6N Group (M16C/6N4) 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 CiCONR register ($i = 0, 1$).

For the CCLKR register, refer to **8. Clock Generation Circuit**.

Figure 19.14 shows the CAN Module System Clock Generation Circuit Block Diagram.

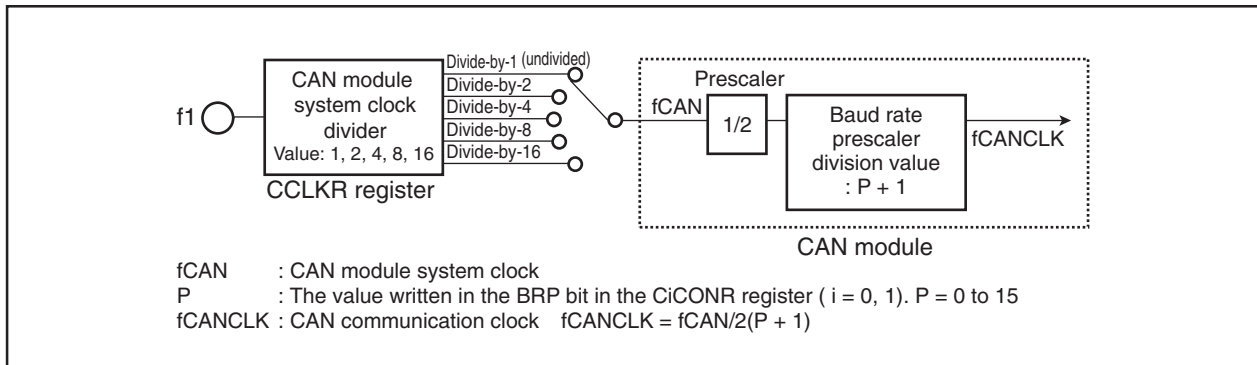


Figure 19.14 CAN Module System Clock Generation Circuit Block Diagram

19.7 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 19.15 shows the Bit Timing.

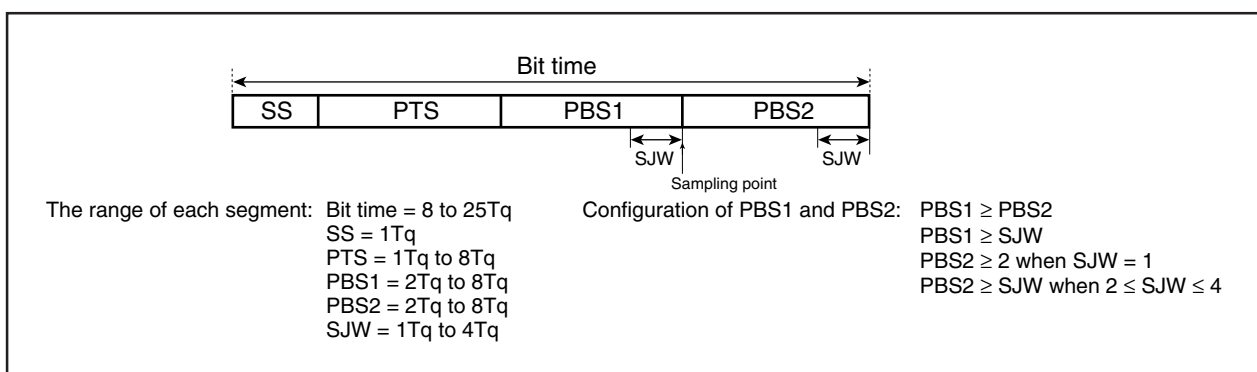


Figure 19.15 Bit Timing

19.8 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 T_q of one bit.

Table 19.2 shows the Examples of Bit-rate.

Table 19.2 Examples of Bit-rate

| Bit-rate | 24 MHz ⁽²⁾ | 20 MHz | 16 MHz | 10 MHz | 8 MHz |
|-----------|-----------------------|---------------|---------------|---------------|---------------|
| 1 Mbps | 12 T_q (1) | 10 T_q (1) | 8 T_q (1) | - | - |
| 500 kbps | 8 T_q (3) | 10 T_q (2) | 8 T_q (2) | 10 T_q (1) | 8 T_q (1) |
| | 12 T_q (2) | 20 T_q (1) | 16 T_q (1) | - | - |
| | 24 T_q (1) | - | - | - | - |
| 125 kbps | 8 T_q (12) | 8 T_q (10) | 8 T_q (8) | 8 T_q (5) | 8 T_q (4) |
| | 12 T_q (8) | 10 T_q (8) | 16 T_q (4) | 10 T_q (4) | 16 T_q (2) |
| | 16 T_q (6) | 16 T_q (5) | - | 20 T_q (2) | - |
| | 24 T_q (4) | 20 T_q (4) | - | - | - |
| 83.3 kbps | 8 T_q (18) | 8 T_q (15) | 8 T_q (12) | 10 T_q (6) | 8 T_q (6) |
| | 12 T_q (12) | 10 T_q (12) | 16 T_q (6) | 20 T_q (3) | 16 T_q (3) |
| | 16 T_q (9) | 20 T_q (6) | - | - | - |
| | 24 T_q (6) | - | - | - | - |
| 33.3 kbps | 10 T_q (36) | 10 T_q (30) | 8 T_q (30) | 10 T_q (15) | 8 T_q (15) |
| | 12 T_q (30) | 20 T_q (15) | 10 T_q (24) | - | 10 T_q (12) |
| | 20 T_q (18) | - | 16 T_q (15) | - | 20 T_q (6) |
| | 24 T_q (15) | - | 20 T_q (12) | - | - |

NOTES:

1. The number in () indicates a value of “fCAN division value” multiplied by “baud rate prescaler division value”.
2. 24 MHz is available Normal-ver. only.

19.8.1 Calculation of Bit-rate

$$\frac{f_1}{2 \times \text{“fCAN division value”}^{(1)} \times \text{“baud rate prescaler division value”}^{(2)} \times \text{“number of } T_q \text{ of one bit”}}$$

NOTES:

1. fCAN division value = 1, 2, 4, 8, 16
fCAN division value: a value selected in the CCLKR register
2. Baud rate prescaler division value = P + 1 (P: 0 to 15)
P: a value selected in the BRP bit in the CiCONR register (i = 0, 1)

19.9 Acceptance Filtering Function and Masking Function

These functions serve the users to select and receive a facultative message. Registers CiGMR ($i = 0, 1$), CiLMAR, and CiLMBR can perform masking to the standard ID and the extended ID of 29 bits. The CiGMR register corresponds to slots 0 to 13, the CiLMAR register corresponds to slot 14, and the CiLMBR 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 CiIDR register upon acceptance filtering operation. When the masking function is employed, it is possible to receive a certain range of IDs.

Figure 19.16 shows the Correspondence of Mask Registers to Slots, Figure 19.17 shows the Acceptance Function.

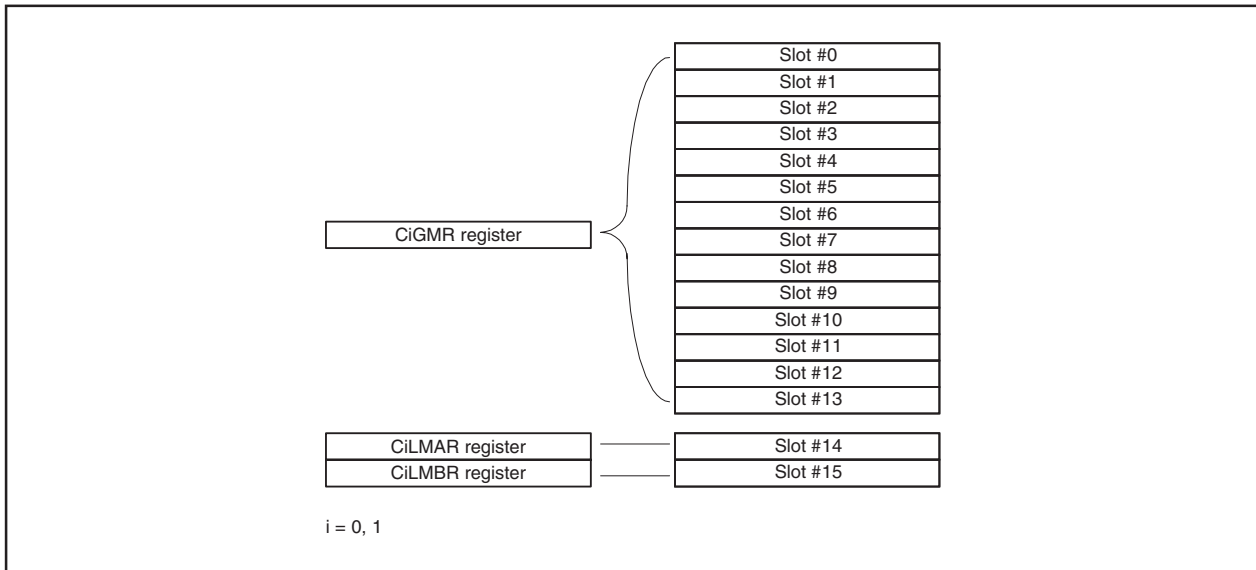


Figure 19.16 Correspondence of Mask Registers to Slots

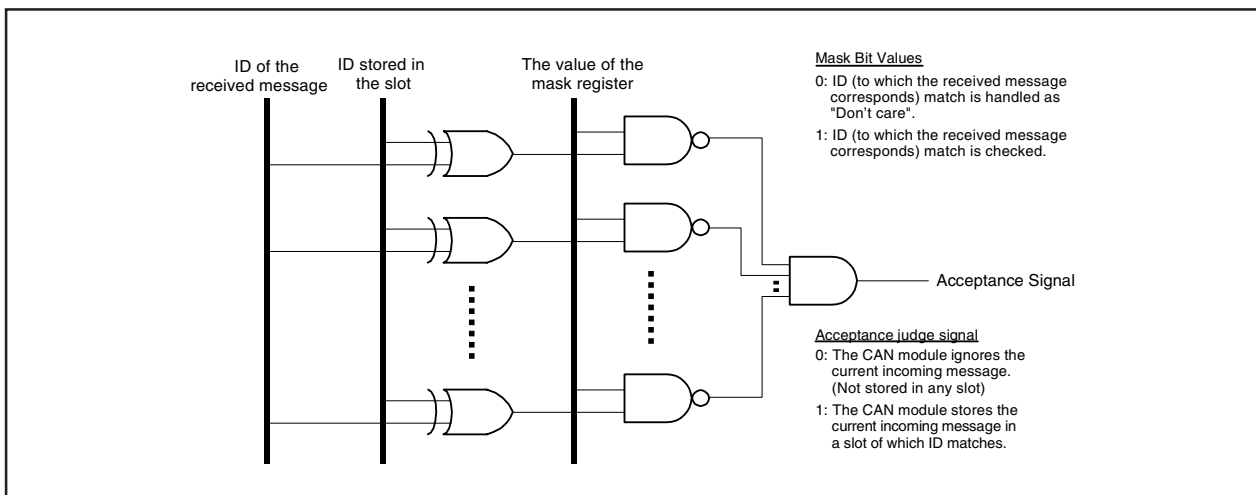


Figure 19.17 Acceptance Function

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.

19.10 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 CiAFS register ($i = 0, 1$), 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: 078h, 087h, 111h
- When there are too many IDs to receive; it would take too much time to filter them by software.

Figure 19.18 shows the Write/Read of CiAFS Register in Word Access.

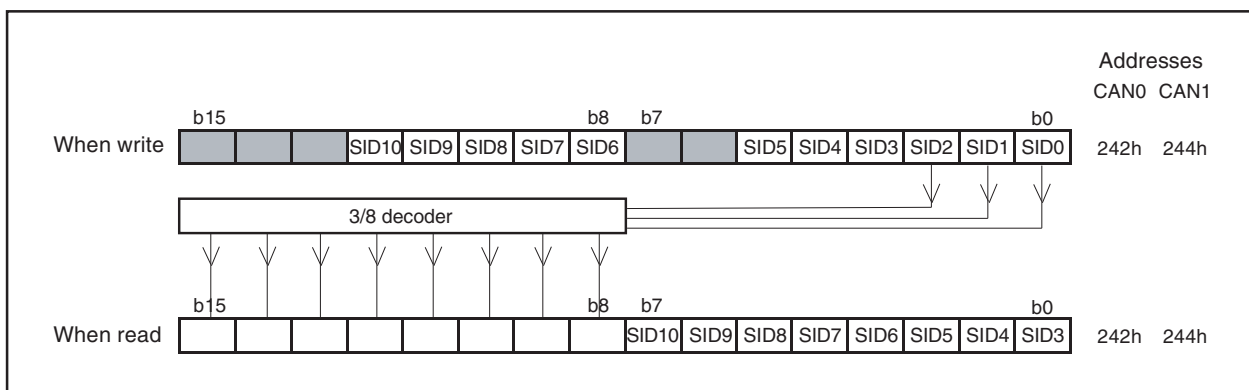


Figure 19.18 Write/read of CiAFS Register in Word Access

19.11 Basic CAN Mode

When the BasicCAN bit in the CiCTLR register ($i = 0, 1$) is set to 1 (Basic CAN mode enabled), slots 14 and 15 correspond to Basic CAN mode. In normal operation mode, each slot can handle only one type message at a time, either a data frame or a remote frame by setting CiMCTLj register ($j = 0$ to 15). However, in Basic CAN mode, slots 14 and 15 can receive both types of message at the same time.

When slots 14 and 15 are defined as reception slots in Basic CAN mode, received messages are stored in slots 14 and 15 alternately.

Which type of message has been received can be checked by the RemActive bit in the CiMCTLj register.

Figure 19.19 shows the Slots 14 and 15 Operation in Basic CAN Mode.

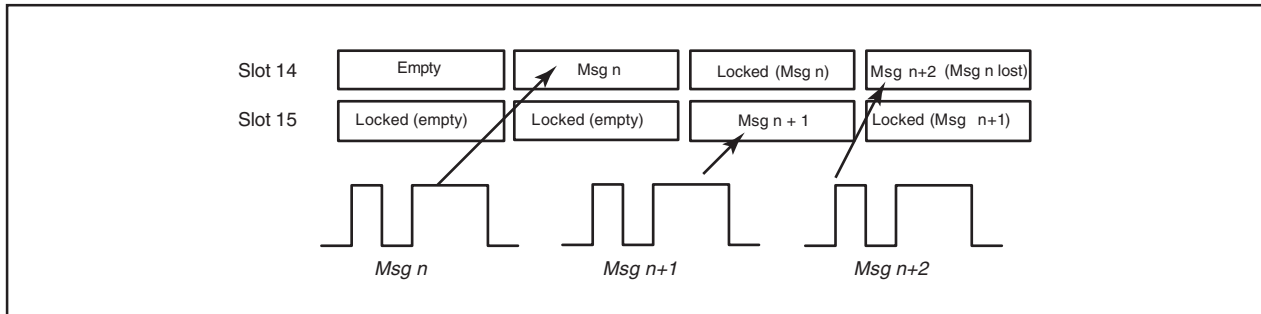


Figure 19.19 Slots 14 and 15 Operation in Basic CAN Mode

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 registers CiLMAR and CiLMBR 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 operation mode.

19.12 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 CiCTRL register ($i = 0, 1$) 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 CiRECR and CiTECR are initialized and the State_BusOff bit in the CiSTR register is set to 0 (CAN module is not in error bus off state). However, registers of the CAN module such as CiCONR register and the content of each slot are not initialized.

19.13 Time Stamp Counter and Time Stamp Function

When the CiTSR register ($i = 0, 1$) 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 CiCONR 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 CiCTRL 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.

19.14 Listen-Only Mode

When the RXOnly bit in the CiCTRL register ($i = 0, 1$) is set to 1, the module enters Listen-only mode.

In Listen-only mode, no transmission, such as data frames, error frames, and ACK response, is performed to bus.

When Listen-only mode is selected, do not request the transmission.

19.15 Reception and Transmission

Table 19.3 lists the CAN Reception and Transmission Mode Configuration.

Table 19.3 CAN Reception and Transmission Mode Configuration

| TrmReq | RecReq | Remote | RspLock | Communication Mode of 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 CiMCTLj register (i = 0, 1, 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 CiMCTLj register to 00h.
- (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 operation 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 CiMCTLj registers to 00h.
- (2) Set the TrmReq bit in the CiMCTLj 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 CiMCTLj register is 1 (transmitting).

If it is rewritten, an undefined data will be transmitted.

19.15.1 Reception

Figure 19.20 shows the Timing of Receive Data Frame Sequence. Figure 19.20 shows the behavior of the module when receiving two consecutive CAN messages, that fit into the slot of the shown CiMCTLj register ($i = 0, 1, j = 0$ to 15) and leads to losing/overwriting of the first message.

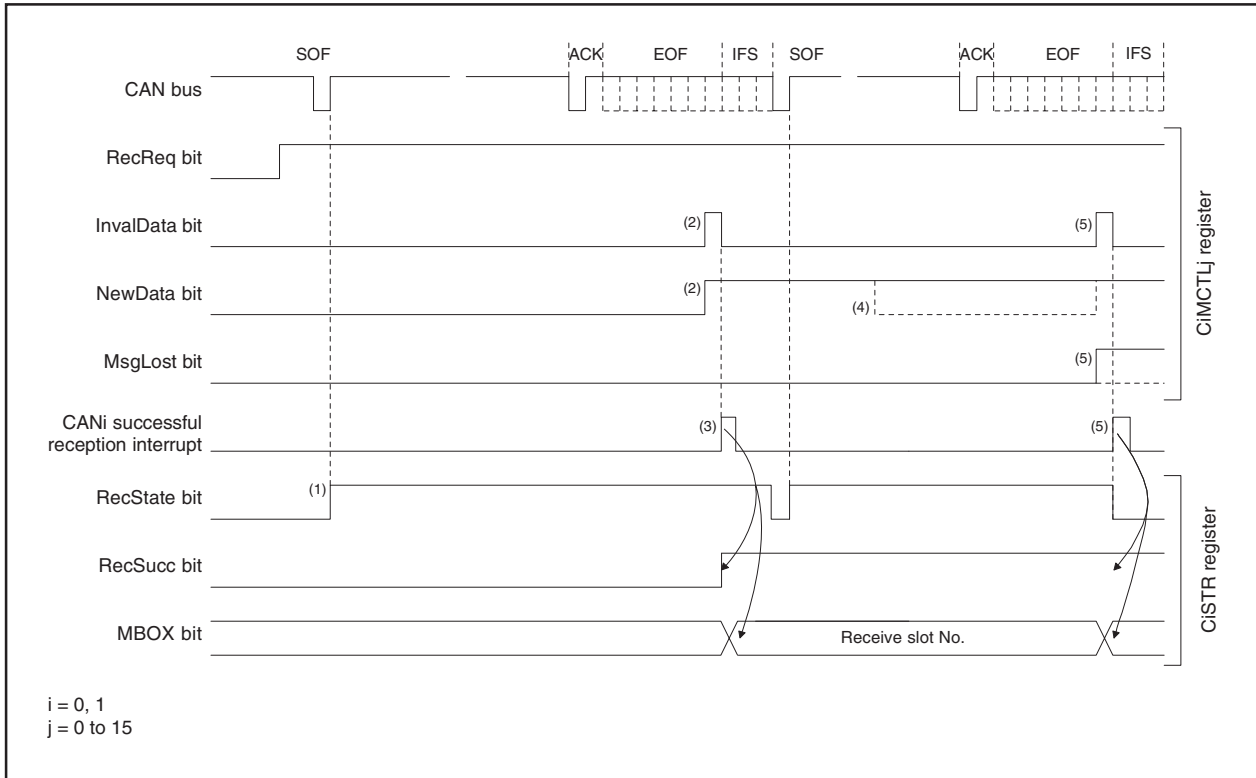


Figure 19.20 Timing of Receive Data Frame Sequence

- (1) On monitoring a SOF on the CAN bus the RecState bit in the CiSTR 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 CiMCTLj register of the receiving slot becomes 1 (stored new data in slot). The InvalData bit in the CiMCTLj 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 CiICR register of the receiving slot = 1 (interrupt enabled), the CANi successful reception interrupt request is generated and the MBOX bit in the CiSTR register is changed. It shows the slot number where the message was stored and the RecSucc bit in the CiSTR 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 a program.
- (5) When next CAN message is received before the NewData bit is set to 0 by a program or a receive request to a slot is canceled, the MsgLost bit in the CiMCTLj 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 CiSTR register are same as in 3).

19.15.2 Transmission

Figure 19.21 shows the Timing of Transmit Sequence.

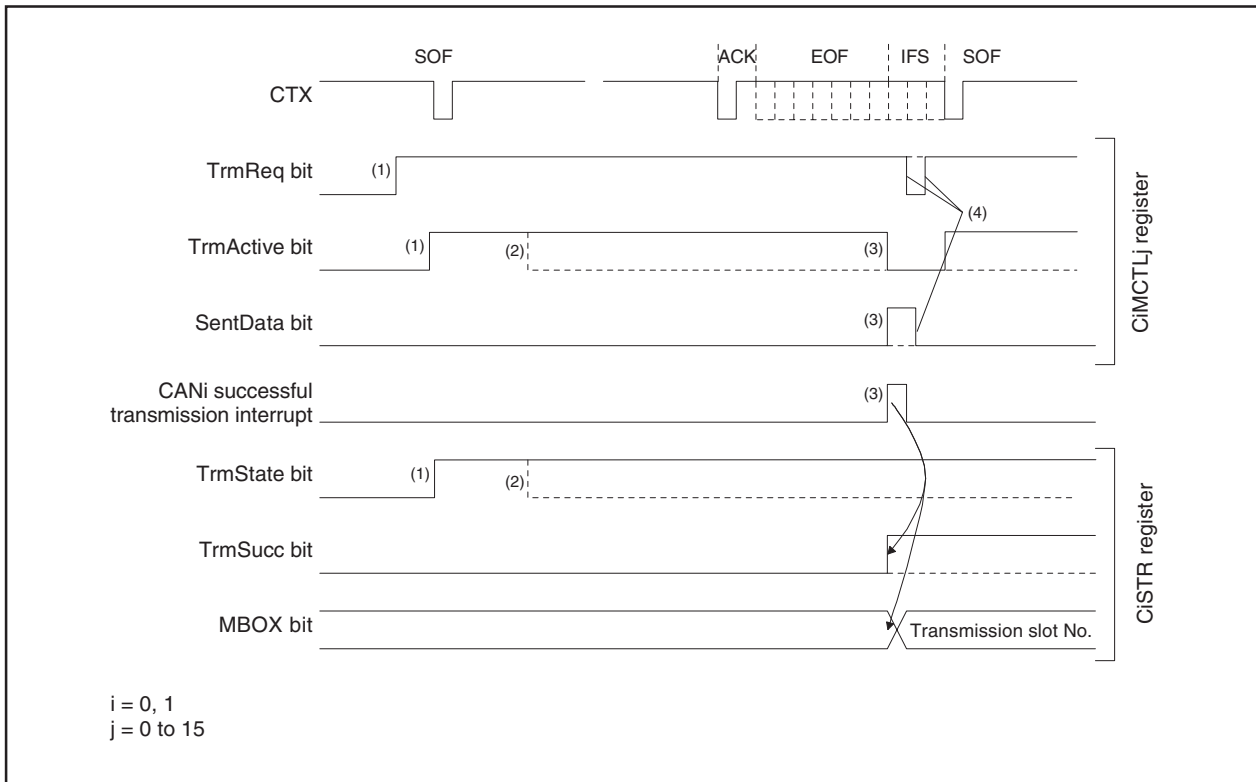


Figure 19.21 Timing of Transmit Sequence

- (1) If the TrmReq bit in the CiMCTLj register ($i = 0, 1, j = 0 \text{ to } 15$) is set to 1 (transmission slot) in the bus idle state, the TrmActive bit in the CiMCTLj register and the TrmState bit in the CiSTR 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, bits TrmActive and TrmState are set to 0.
- (3) If the transmission has been successful without lost in arbitration, the SentData bit in the CiMCTLj register is set to 1 (transmission is successfully completed) and TrmActive bit is set to 0 (waiting for bus idle or completion of arbitration). And when the interrupt enable bits in the CiICR register = 1 (interrupt enabled), CANi successful transmission interrupt request is generated and the MBOX (the slot number which transmitted the message) and TrmSucc bit in the CiSTR 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.

19.16 CAN Interrupt

The CAN module provides the following CAN interrupts.

- CANi successful reception interrupt (i = 0, 1)
- CANi successful transmission interrupt
- CAN0/1 error Interrupt: Error passive state
Error bus off state
Bus error (this feature can be disabled separately)
- CAN0/1 wake-up interrupt

When the CPU detects the CANi successful reception/transmission interrupt request, the MBOX bit in the CiSTR register must be read to determine which slot has generated the interrupt request.

20. Programmable I/O Ports

The programmable input/output ports (hereafter referred to simply as I/O ports) consist of 87 lines P0 to P10 (except P8_5). 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 every 4 lines. P8_5 is an input-only port and does not have a pull-up resistor. Port P8_5 shares the pin with NMI, so that the NMI input level can be read from the P8_5 bit in the P8 register.

Figures 20.1 to 20.5 show the I/O Ports. Figure 20.6 shows the I/O Pins.

Each pin functions as an I/O port, a peripheral function input/output pin or a bus control pin.

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 or D/A converter output pin, set the direction bit for that pin to 0 (input mode). Any pin used as an output pin for peripheral functions other than the D/A converter is directed for output no matter how the corresponding direction bit is set.

When using any pin as a bus control pin, refer to **7.2 Bus Control**.

20.1 PDi Register (i = 0 to 10)

Figure 20.7 shows the PDi Register.

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.

During memory expansion and microprocessor modes, the PDi registers for the pins functioning as bus control pins (A0 to A19, D0 to D15, CS0 to CS3, RD, WRL/WR, WRH/BHE, ALE, RDY, HOLD, HLDA, and BCLK) cannot be modified.

No direction register bit for P8_5 is available.

20.2 Pi Register (i = 0 to 10)

Figure 20.8 shows the Pi Register.

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 input/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.

During memory expansion and microprocessor modes, the Pi registers for the pins functioning as bus control pins (A0 to A19, D0 to D15, CS0 to CS3, RD, WRL/WR, WRH/BHE, ALE, RDY, HOLD, HLDA, and BCLK) cannot be modified.

20.3 PURj Register (j = 0 to 2)

Figure 20.9 shows the PURj Register.

The PURj register bits can be used to select whether or not to pull the corresponding port high in 4-bit unit. The port selected to be pulled high has a pull-up resistor connected to it when the direction bit is set for input mode.

However, the pull-up control register has no effect on P0 to P3, P4_0 to P4_3, and P5 during memory expansion and microprocessor modes. Although the register contents can be modified, no pull-up resistors are connected.

20.4 PCR Register

Figure 20.10 shows the PCR 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.

Tables 20.1 lists the Unassigned Pin Handling in Single-chip Mode and 20.2 lists the Unassigned Pin Handling in Memory Expansion Mode and Microprocessor Mode. Figure 20.11 shows the Unassigned Pin Handling.

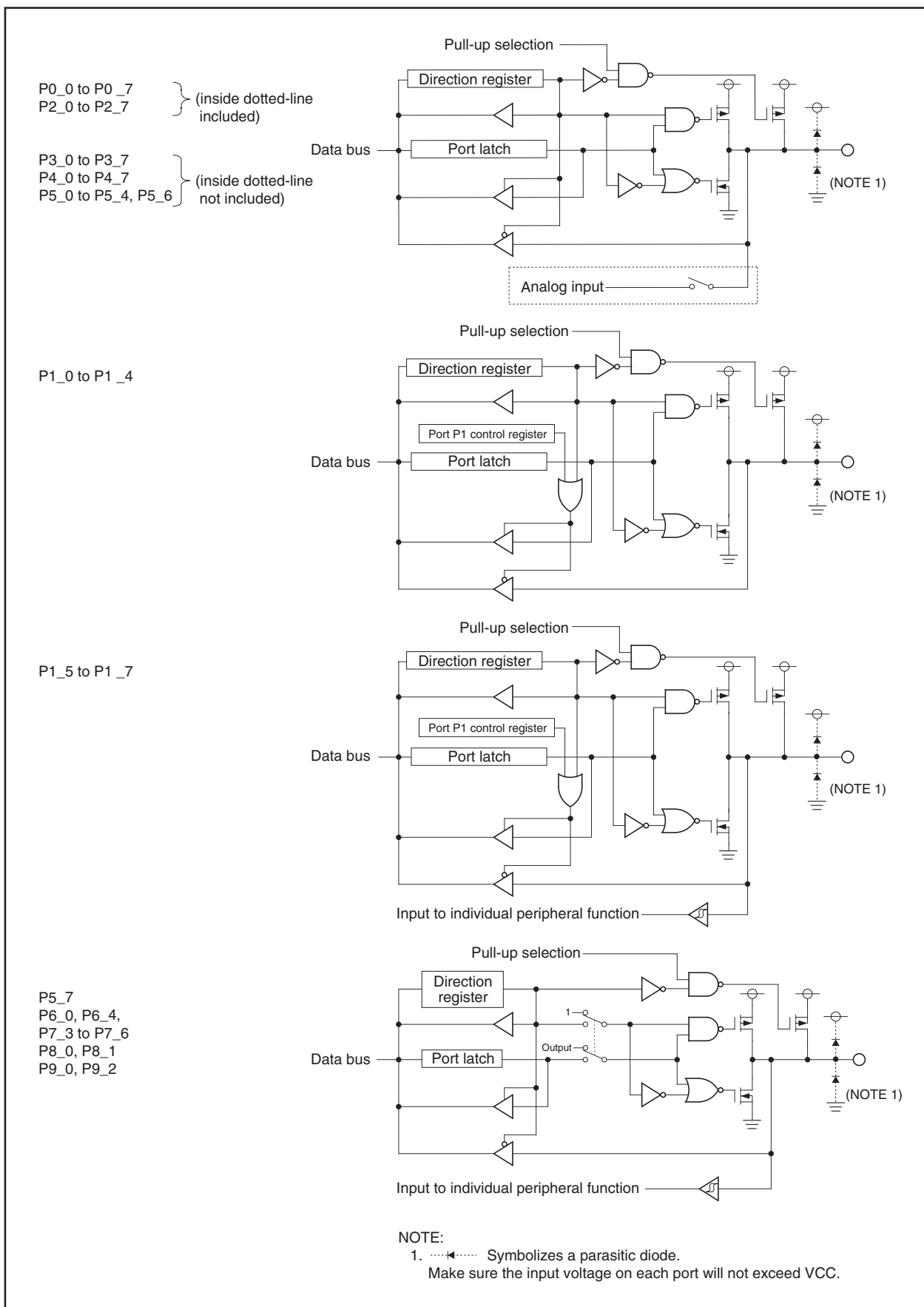


Figure 20.1 I/O Ports (1)

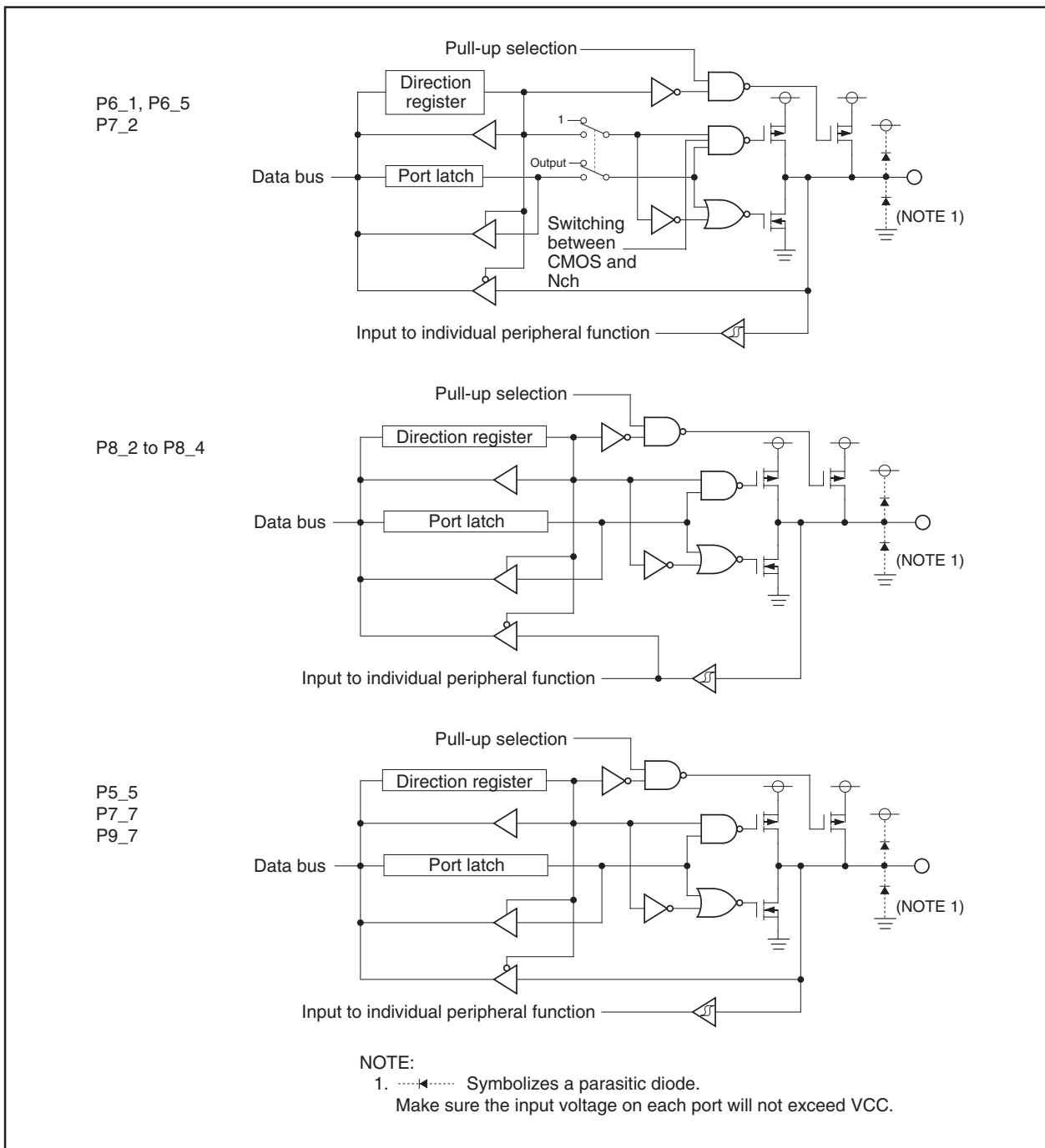


Figure 20.2 I/O Ports (2)

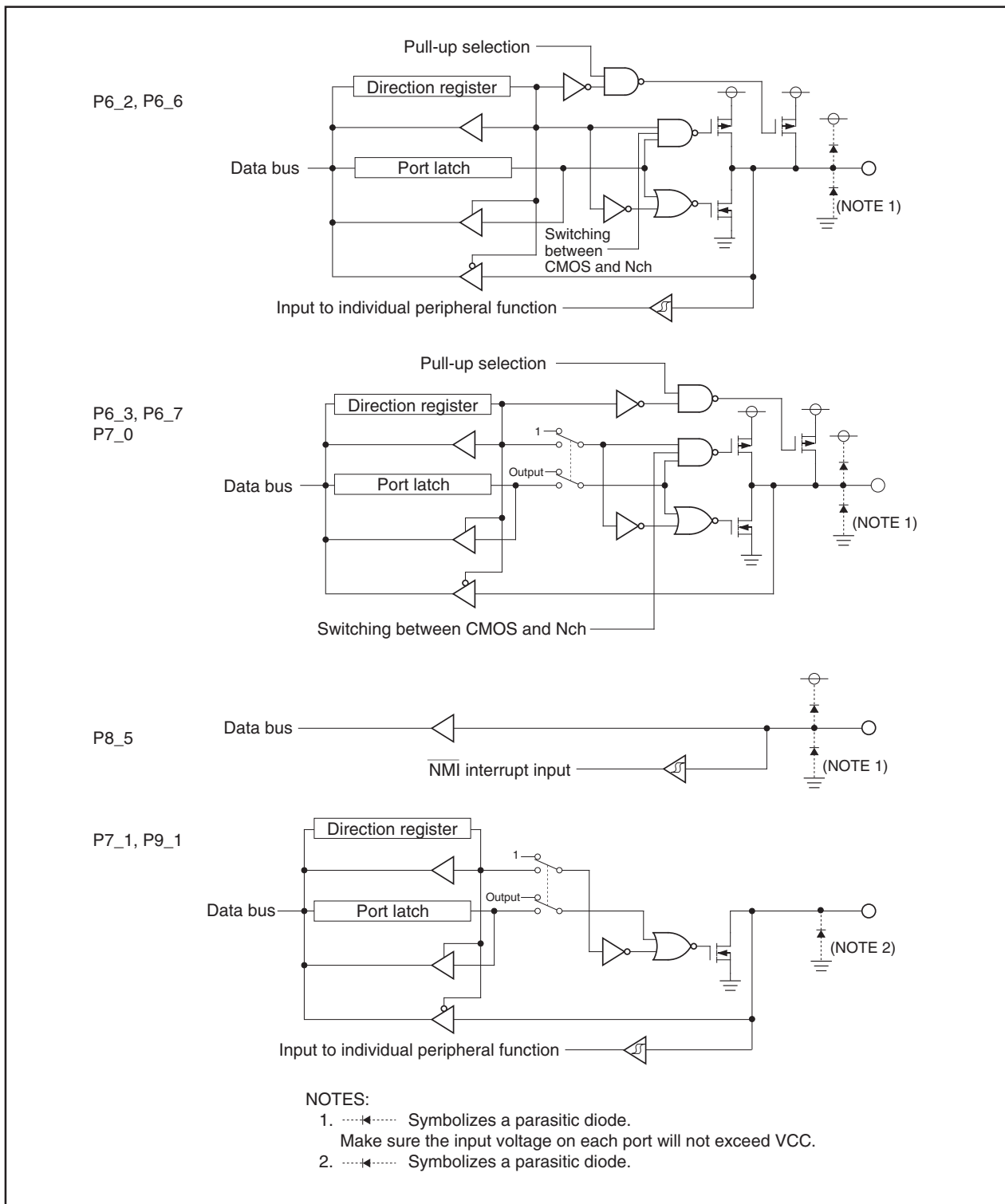


Figure 20.3 I/O Ports (3)

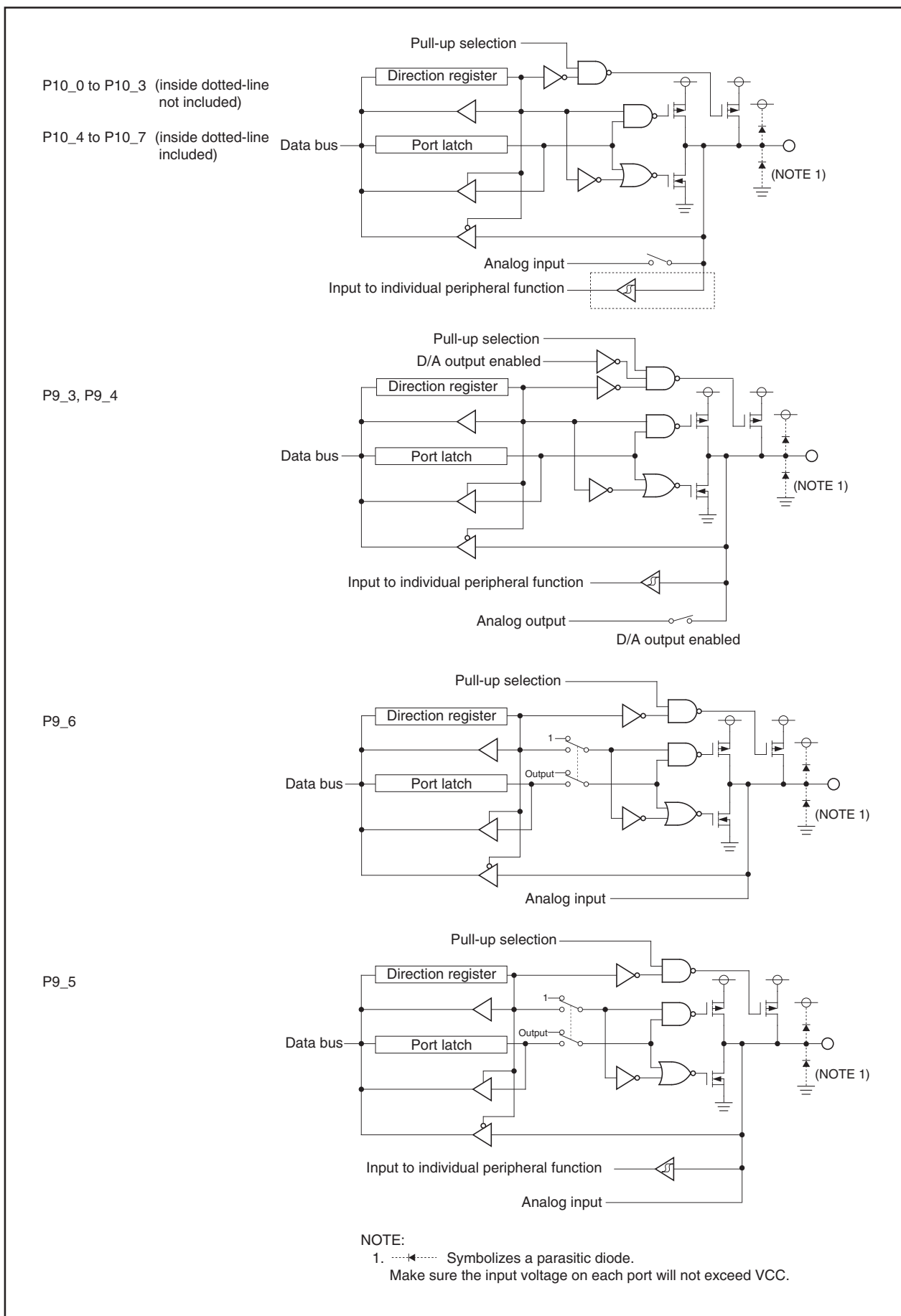


Figure 20.4 I/O Ports (4)

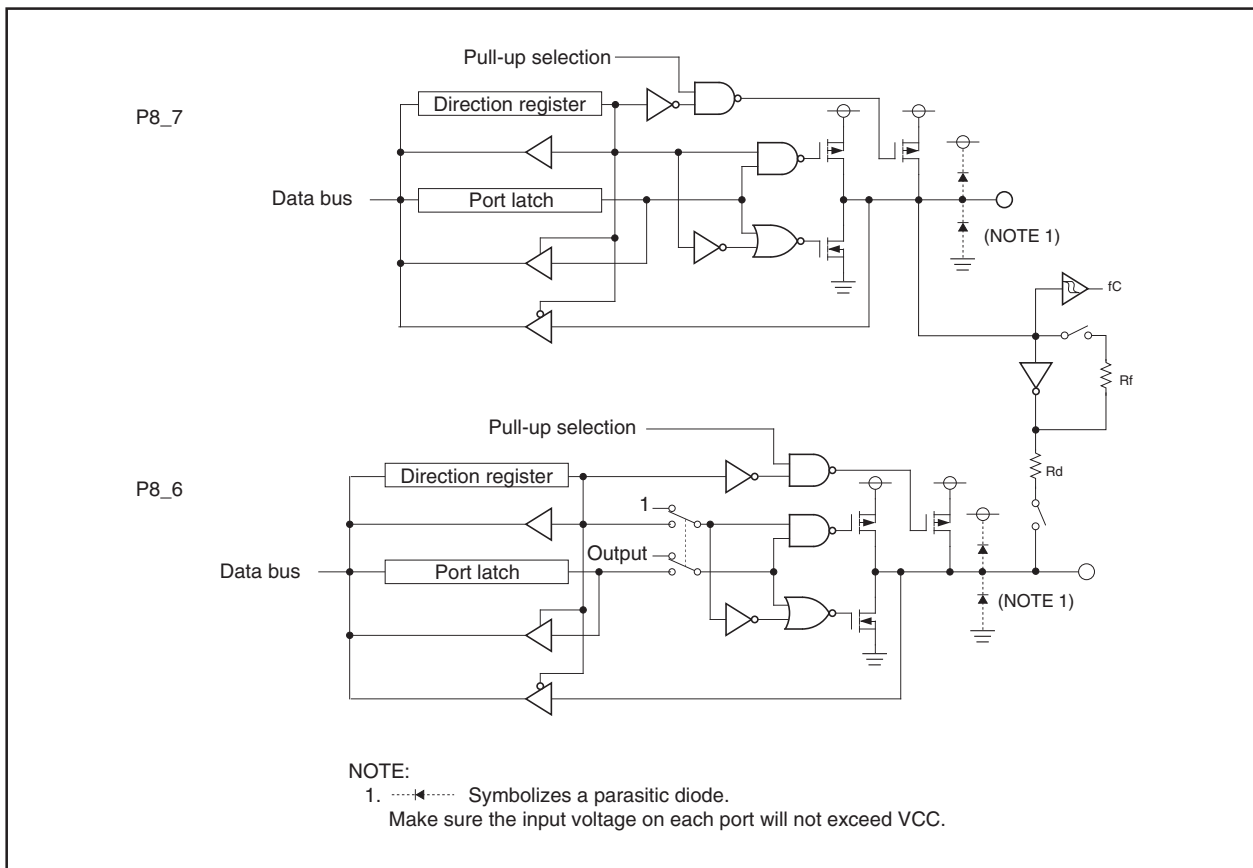


Figure 20.5 I/O Ports (5)

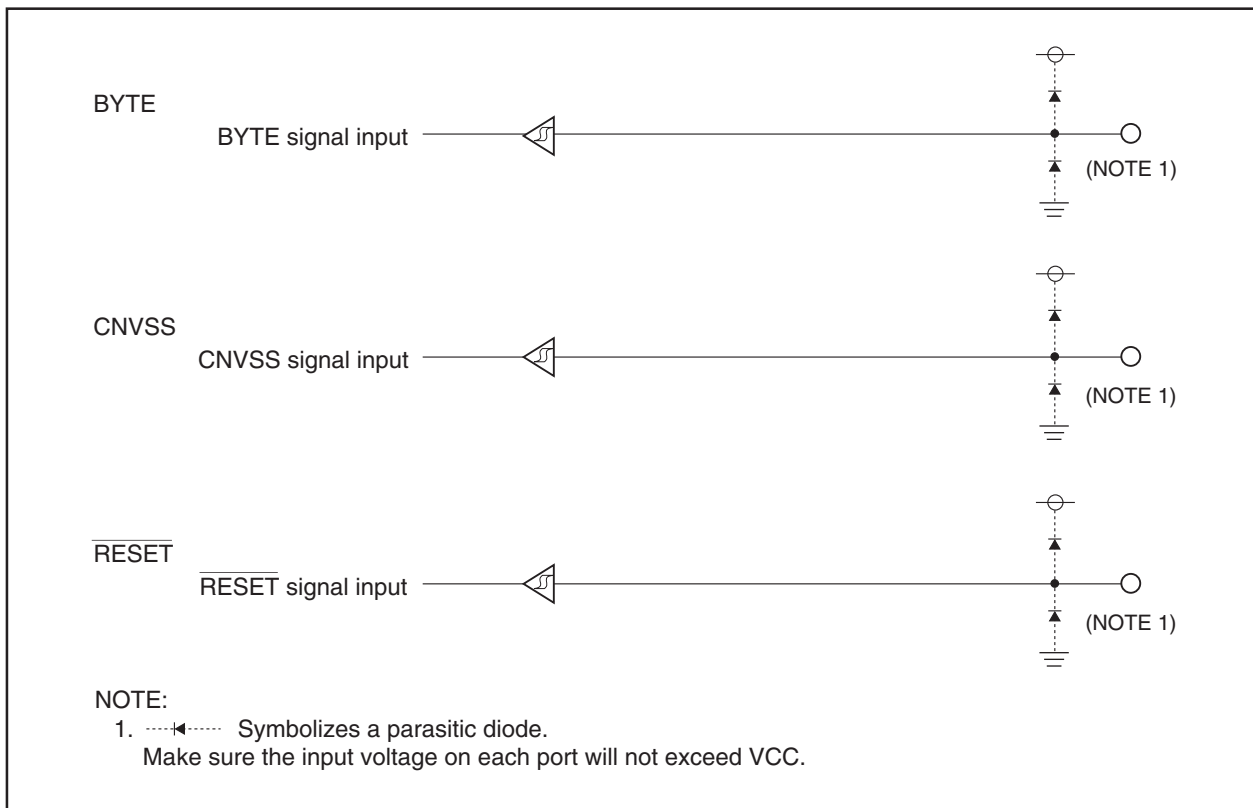


Figure 20.6 I/O Pins

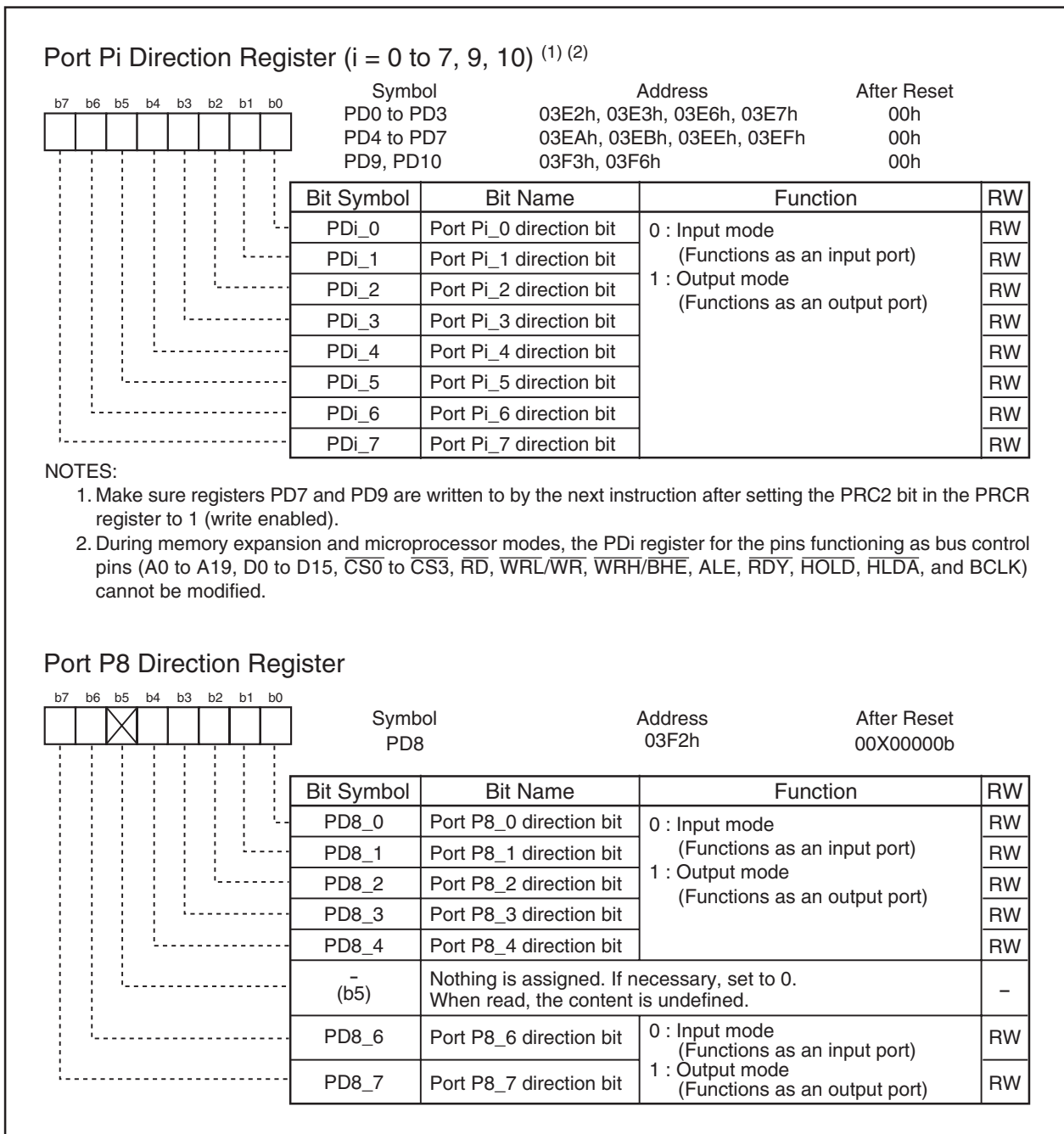


Figure 20.7 Registers PD0 to PD10

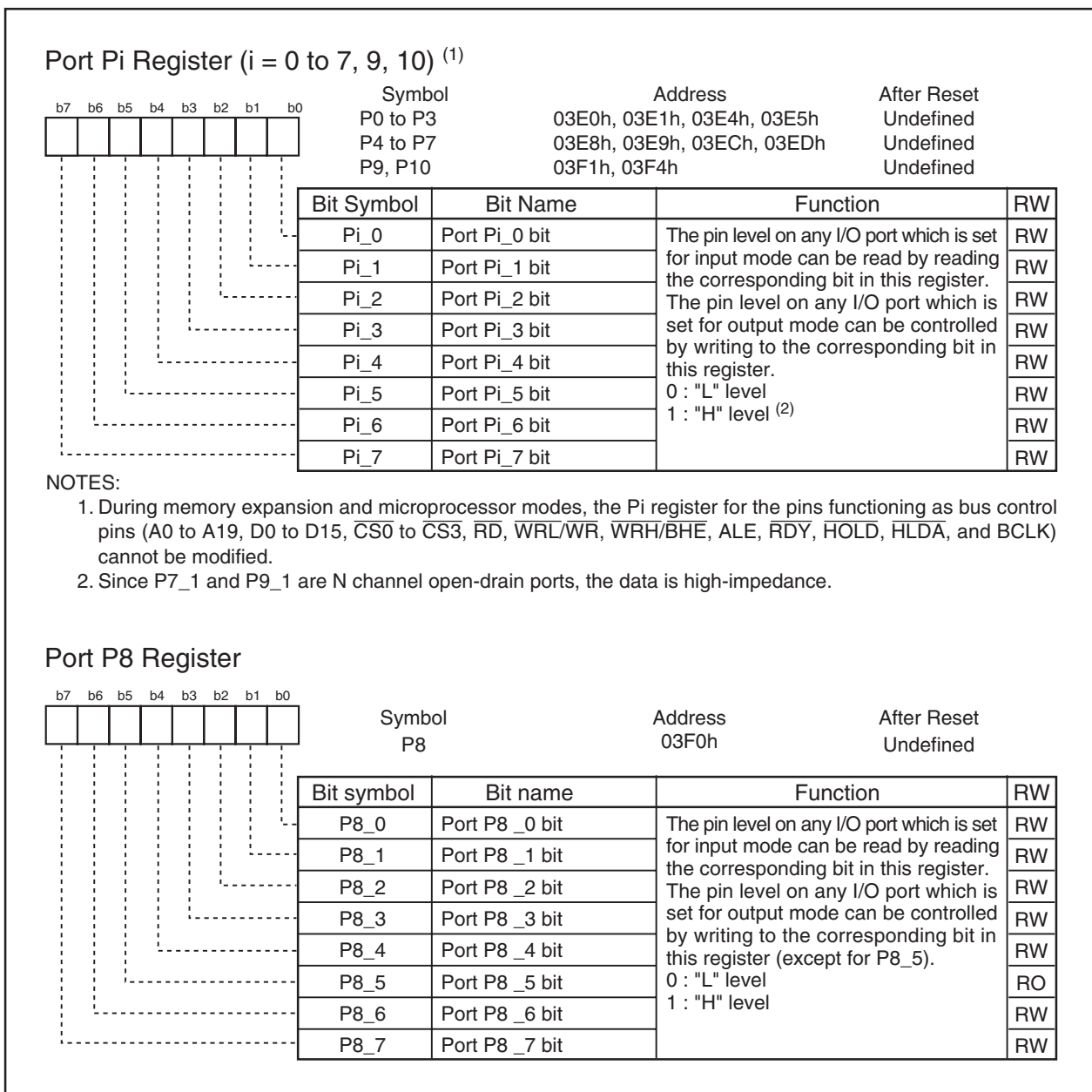


Figure 20.8 Registers P0 to P10

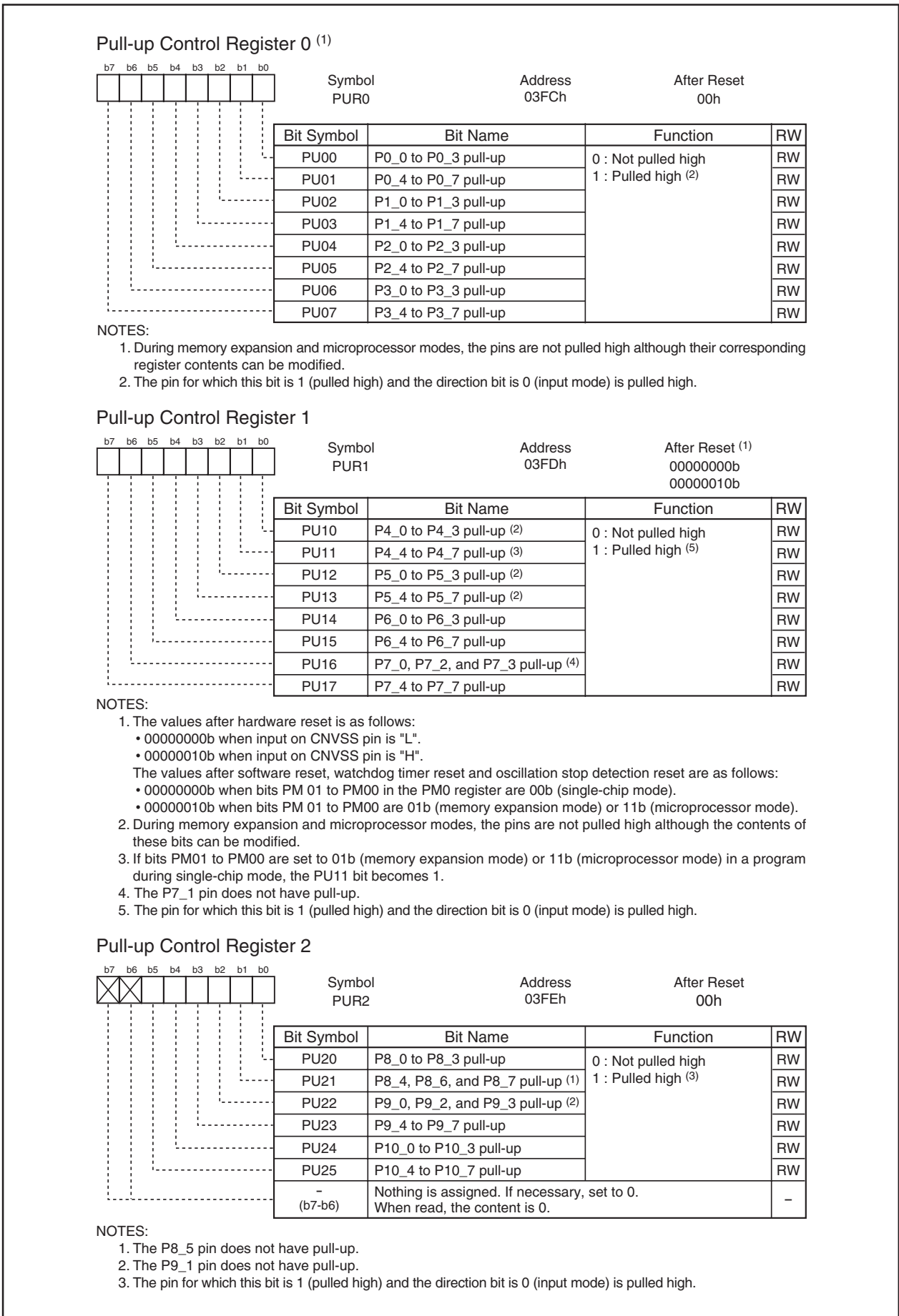


Figure 20.9 Registers PUR0, PUR1, and PUR2

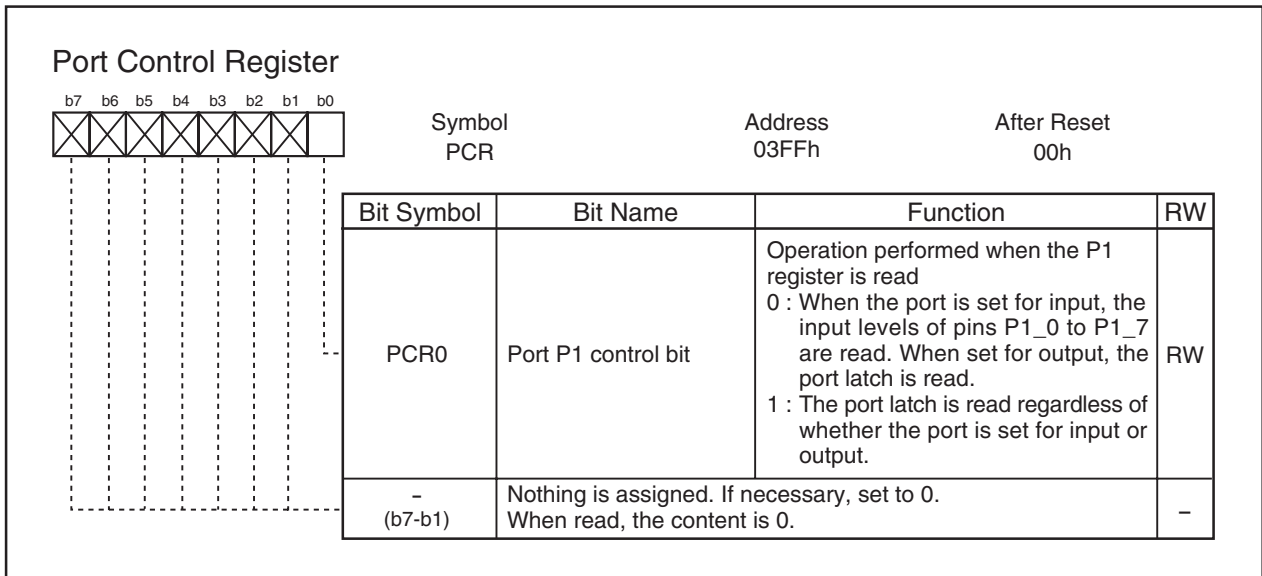


Figure 20.10 PCR Register

Table 20.1 Unassigned Pin Handling in Single-chip Mode

| Pin Name | Connection |
|---|--|
| Ports P0 to P7, P8_0 to P8_4, P8_6, P8_7, P9, P10 | After setting for input mode, connect every pin to VSS via a resistor (pull-down); or after setting for output mode, leave these pins open. ^{(1) (2) (3)} |
| XOUT ⁽⁴⁾ | Open |
| NMI(P8_5) | Connect via resistor to VCC (pull-up) |
| AVCC | Connect to VCC |
| AVSS, VREF, BYTE | Connect to VSS |

NOTES:

1. When setting the port for output mode and leave it open, be aware that the port remains in input mode until it is switched to output mode in a program after reset. For this reason, the voltage level on the pin becomes undefined, causing the power supply current to increase while the port remains in input mode.
Furthermore, by considering a possibility that the contents of the direction registers may change due to noise or program runaway caused by noise, it is recommended that the contents of the direction registers be periodically reset in software, for the increased reliability of the program.
2. Make sure the unused pins are processed with the shortest possible wiring from the MCU pins (2 cm or less).
3. When the ports P7_1 and P9_1 are set for output mode, make sure a low-level signal is output from the pins.
The ports P7_1 and P9_1 are N-channel open-drain outputs.
4. With external clock input to XIN pin.

Table 20.2 Unassigned Pin Handling in Memory Expansion Mode and Microprocessor Mode

| Pin Name | Connection |
|---|--|
| Ports P6, P7, P8_0 to P8_4, P8_6, P8_7, P9, P10 | After setting for input mode, connect every pin to VSS via a resistor (pull-down); or after setting for output mode, leave these pins open. ^{(1) (2) (3) (4)} |
| P4_5/CS1 to P4_7/CS3 | Connect to VCC via a resistor (pulled high) by setting the corresponding direction bit in the PD4 register for \overline{CS}_i ($i = 1$ to 3) to 0 (input mode) and the \overline{CS}_i bit in the CSR register to 0 (chip select disabled). |
| \overline{BHE} , \overline{ALE} , \overline{HLDA} , XOUT ⁽⁵⁾ , BCLK ⁽⁶⁾ | Open |
| HOLD, \overline{RDY} , \overline{NMI} (P8_5) | Connect via resistor to VCC (pull-up) |
| AVCC | Connect to VCC |
| AVSS, VREF | Connect to VSS |

NOTES:

1. When setting the port for output mode and leave it open, be aware that the port remains in input mode until it is switched to output mode in a program after reset. For this reason, the voltage level on the pin becomes undefined, causing the power supply current to increase while the port remains in input mode.
Furthermore, by considering a possibility that the contents of the direction registers may change due to noise or program runaway caused by noise, it is recommended that the contents of the direction registers be periodically reset in software, for the increased reliability of the program.
2. Make sure the unused pins are processed with the shortest possible wiring from the MCU pins (2 cm or less).
3. If the CNVSS pin has the VSS level applied to it, these pins are set for input ports until the processor mode is switched over in a program after reset. For this reason, the voltage levels on these pins become undefined, causing the power supply current to increase while they remain set for input ports.
4. When the ports P7_1 and P9_1 are set for output mode, make sure a low-level signal is output from the pins.
The ports P7_1 and P9_1 are N-channel open-drain outputs.
5. With external clock input to XIN pin.
6. If the PM07 bit in the PM0 register is set to 1 (BCLK not output), connect this pin to VCC via a resistor (pulled high).

21. Flash Memory Version

Aside from the on-chip flash memory, the flash memory version MCU has the same functions as the masked ROM version.

In the flash memory version, the flash memory can perform in four rewrite mode: CPU rewrite mode, standard serial I/O mode, parallel I/O mode, and CAN I/O mode.

Table 21.1 lists the Flash Memory Version Specifications. See **Table 1.1 Functions and Specifications**, for the items not listed in Table 21.1. Table 21.2 shows the Flash Memory Rewrite Modes Overview.

Table 21.1 Flash Memory Version Specifications

| Item | Specifications |
|--|---|
| Flash memory rewrite mode | 4 modes (CPU rewrite, standard serial I/O, parallel I/O, CAN I/O) |
| Erase block | User ROM area |
| | Boot ROM area |
| Program method | In units of word, in units of byte ⁽²⁾ |
| Erase method | Collective erase, block erase |
| Program and erase control method | Program and erase controlled by software command |
| Protect method | Lock bit protects each block |
| Number of commands | 8 commands |
| Programming and erasure endurance ⁽³⁾ | 100 times |
| ROM code protection | Parallel I/O, standard serial I/O, and CAN I/O modes are supported. |

NOTES:

1. The boot ROM area contains standard serial I/O mode and CAN I/O mode rewrite control program which is stored in it when shipped from the factory. This area can only be rewritten in parallel I/O mode.
2. Can be programmed in byte units in only parallel I/O mode.
3. Definition of programming and erasure endurance
The programming and erasure endurance is defined to be per-block erasure endurance. For example, assume a case where a 4K-byte block A is programmed in 2,048 operations by writing one word at a time and erased thereafter. In this case, the block is reckoned as having been programmed and erased once.
If a product is 100 times of programming and erasure endurance, each block in it can be erased up to 100 times.

Table 21.2 Flash Memory Rewrite Modes Overview

| Flash Memory Rewrite Mode | CPU Rewrite Mode ⁽¹⁾ | 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 commands. EW0 mode: Rewrite in areas other than flash memory ⁽²⁾ EW1 mode: Can be rewritten in the 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 ⁽³⁾ | The boot ROM and user ROM areas are rewritten using a dedicated parallel programmer. | The user ROM area is rewritten using a dedicated CAN programmer. |
| Areas which can be rewritten | User ROM area | User ROM area | User ROM area Boot ROM area | User ROM area |
| Operating mode | Single-chip mode Memory expansion mode (EW0 mode) Boot mode (EW0 mode) | Boot mode | Parallel I/O mode | Boot mode |
| ROM programmer | None | Serial programmer | Parallel programmer | CAN programmer |

NOTES:

1. The PM13 bit remains set to 1 while the FMR01 bit in the FMR0 register = 1 (CPU rewrite mode enabled). The PM13 bit is reverted to its original value by setting the FMR01 bit to 0 (CPU rewrite mode disabled). However, if the PM13 bit is changed during CPU rewrite mode, its changed value is not reflected until after the FMR01 bit is set to 0.
2. When in CPU rewrite mode, bits PM10 and PM13 in the PM1 register are set to 1. The rewrite control program can only be executed in the internal RAM or in an external area that is enabled for use when the PM13 bit = 1.
3. When using standard serial I/O mode 2, make sure a main clock input oscillation frequency is set to 5 MHz, 10 MHz, or 16 MHz.

21.1 Memory Map

The flash memory contains the user ROM area and the boot ROM area. The user ROM area has space to store the MCU operating program in single-chip mode or memory expansion mode and a separate 4-Kbyte space as the block A.

Figure 21.1 shows the Flash Memory Block Diagram.

The user ROM area is divided into several blocks, each of which can be protected (locked) against programming or erasure. The user ROM area can be rewritten in CPU rewrite, standard serial I/O mode, parallel I/O mode, and CAN I/O mode. Block A is enabled for use by setting the PM10 bit in the PM1 register to 1 (block A enabled. $\overline{CS2}$ area at addresses 10000h to 26FFFh).

The boot ROM area is located at the same addresses as the user ROM area. It can only be rewritten in parallel I/O mode (refer to **21.1.1 Boot Mode**). A program in the boot ROM area is executed after a hardware reset occurs while an “H” signal is applied to pins CNVSS and P5_0 and an “L” signal is applied to the P5_5 pin (refer to **21.1.1 Boot Mode**). A program in the user ROM area is executed after a hardware reset occurs while an “L” signal is applied to the CNVSS pin. However, the boot ROM area cannot be read.

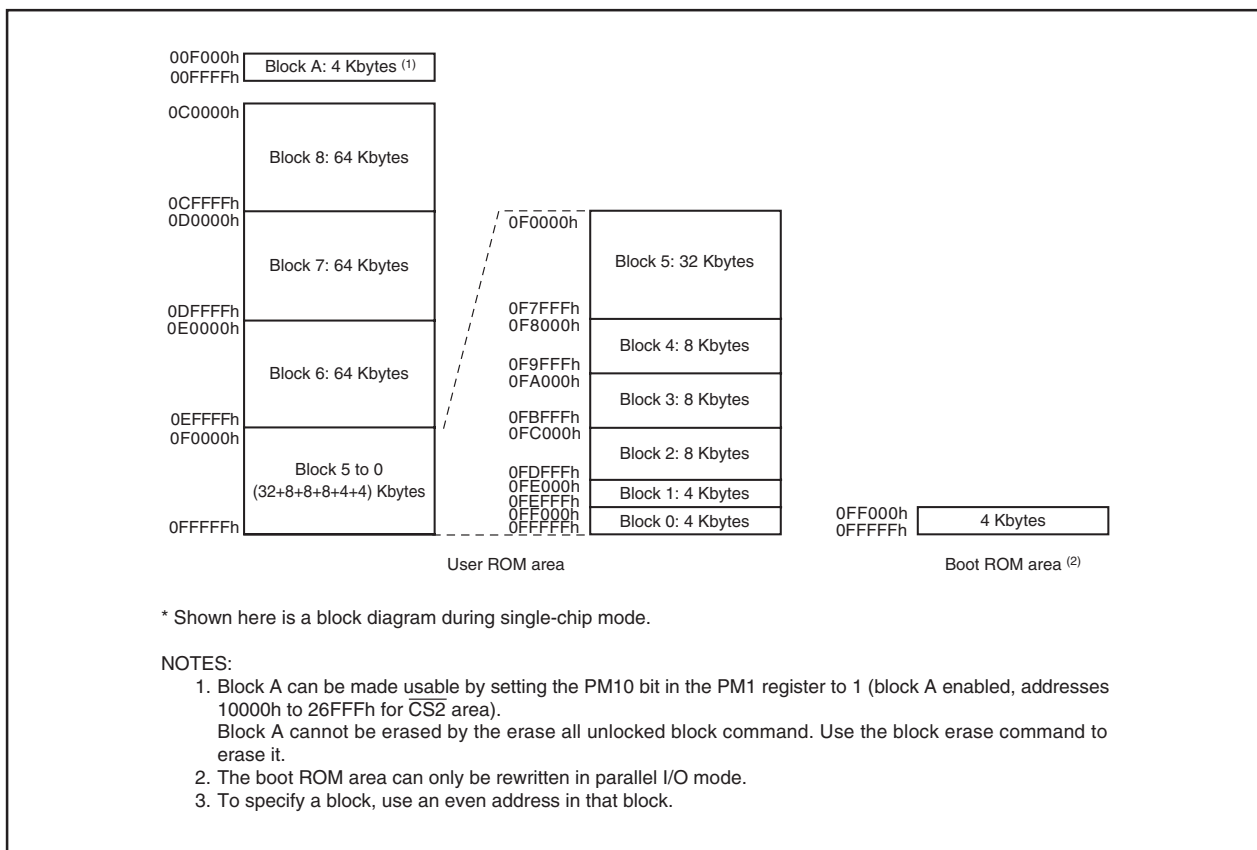


Figure 21.1 Flash Memory Block Diagram

21.1.1 Boot Mode

The MCU enters boot mode when a hardware reset occurs while an “H” signal is applied to pins CNVSS and P5_0 and an “L” signal is applied to the P5_5 pin. A program in the boot ROM area is executed.

In boot mode, the FMR05 bit in the FMR0 register selects access to the boot ROM area or the user ROM area. The rewrite control program for standard serial I/O mode is stored in the boot ROM area before shipment. The boot ROM area can be rewritten in parallel I/O mode only. If given rewrite control program using erase-write mode (EW0 mode) is written in the boot ROM area, the flash memory can be rewritten according to the system implemented.

21.2 Functions to Prevent Flash Memory from Rewriting

The flash memory has the ROM code protect function for parallel I/O mode and the ID code check function for standard serial I/O mode and CAN I/O mode to prevent the flash memory from reading or rewriting.

21.2.1 ROM Code Protect Function

The ROM code protect function inhibits the flash memory from being read or rewritten during parallel I/O mode. Figure 21.2 shows the ROMCP Register. The ROMCP register is located in the user ROM area. The ROM code protect function is enabled when the ROMCR bits are set to other than 11b. In this case, set the bit 5 to bit 0 to 111111b.

When exiting ROM code protect, erase the block including the ROMCP register by CPU rewrite mode, standard serial I/O mode, or CAN I/O mode.

21.2.2 ID Code Check Function

Use the ID code check function in standard serial I/O mode and CAN I/O mode. The ID code sent from the serial programmer is compared with the ID code written in the flash memory for a match. If the ID codes do not match, commands sent from the serial programmer are not accepted. However, if the four bytes of the reset vector are FFFFFFFFh, ID codes are not compared, allowing all commands to be accepted.

The ID codes are 7-byte data stored consecutively, starting with the first byte, into addresses 0FFFDfH, 0FFFE3h, 0FFFEbH, 0FFFEfH, 0FFFF3h, 0FFFF7h, and 0FFFFbH. The flash memory must have a program with the ID codes set in these addresses.

Figure 21.3 shows the Addresses for ID Code Stored.

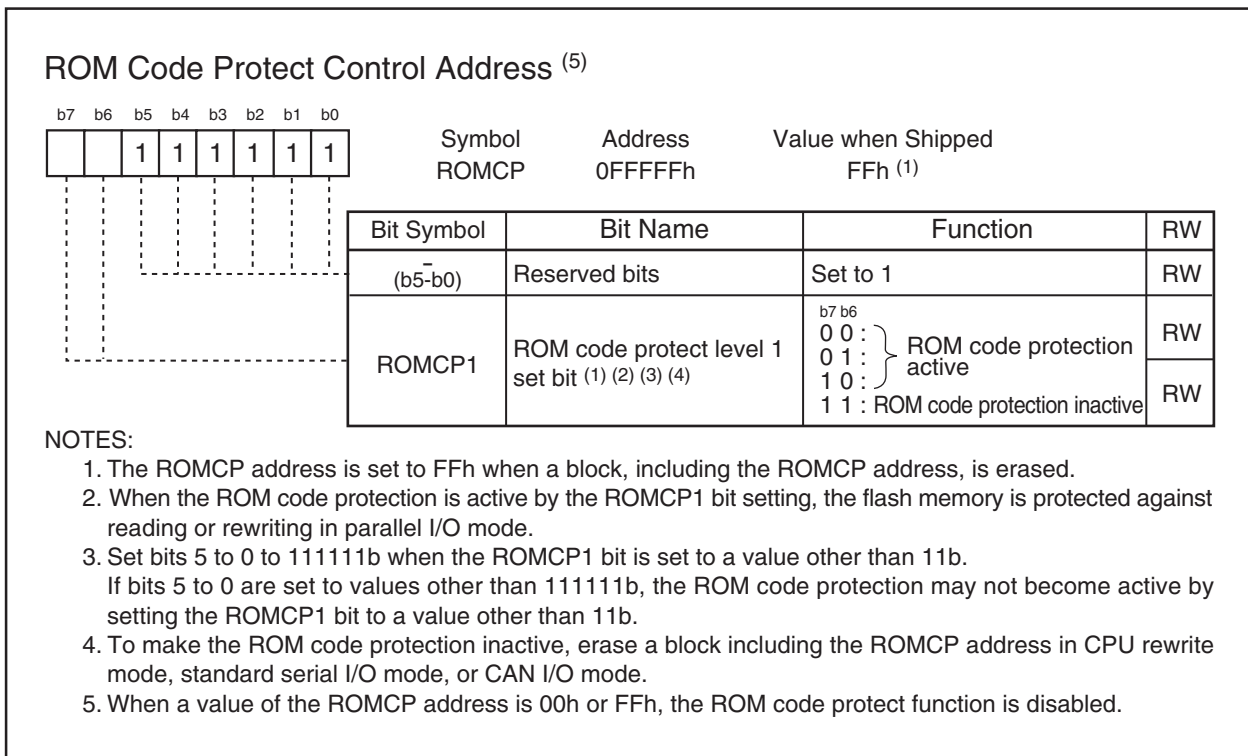


Figure 21.2 ROMCP Register

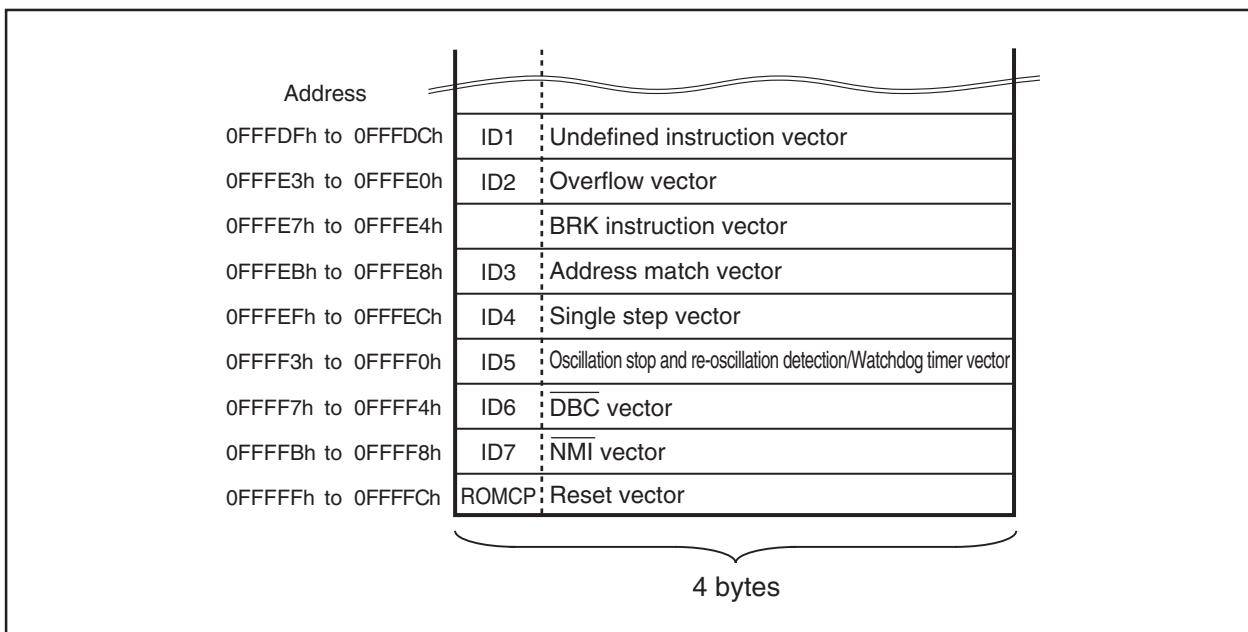


Figure 21.3 Address for ID Code Stored

21.3 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 the MCU is mounted on a board without using a parallel, serial or CAN programmer.

In CPU rewrite mode, only the user ROM area shown in Figure 21.1 can be rewritten. The boot ROM area cannot be rewritten. Program and the block erase command are executed only in the user ROM area.

Erase-write 0 (EW0) mode and erase-write 1 (EW1) mode are provided as CPU rewrite mode.

Table 21.3 lists the differences between EW0 and EW1 Modes.

Table 21.3 EW0 Mode and EW1 Mode

| Item | EW0 Mode | EW1 Mode |
|---|---|--|
| Operating mode | <ul style="list-style-type: none"> • Single-chip mode • Memory expansion mode • Boot mode | Single-chip mode |
| Space where rewrite control program can be placed | <ul style="list-style-type: none"> • User ROM area • Boot ROM area | User ROM area |
| Space where rewrite control program can be executed | The rewrite control program must be transferred to any space other than the flash memory (e.g., RAM) before being executed ⁽²⁾ | The rewrite control program can be executed in the user ROM area |
| Space which can be rewritten | User ROM area | User ROM area However, this excludes blocks with the rewrite control program |
| Software command restriction | None | <ul style="list-style-type: none"> • Program and block erase commands cannot be executed in a block having the rewrite control program. • Erase all unlocked block command cannot be executed when the lock bit in a block having the rewrite control program is set to 1 (unlocked) or when the FMR02 bit in the FMR0 register is set to 1 (lock bit disabled). • Read status register command cannot be used. |
| Modes after program or erasing | Read status register mode | Read array mode |
| CPU status during auto-programming and auto-erase | Operating | Maintains hold state (I/O ports maintains the state before the command was executed) ⁽¹⁾ |
| Flash memory status detection | <ul style="list-style-type: none"> • Read bits FMR00, FMR06, and FMR07 in the FMR0 register by program • Execute the read status register command to read bits SR7, SR5, and SR4 in the status register | Read bits FMR00, FMR06, and FMR07 in the FMR0 register by program |

NOTES:

1. Do not generate an interrupts (except $\overline{\text{NMI}}$ interrupt) and DMA transfer.
2. When in CPU rewrite mode, bits PM10 and PM13 in the PM1 register are set to 1. The rewrite control program can only be executed in the internal RAM or in an external area that is enabled for use when the PM13 bit = 1.

21.3.1 EW0 Mode

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 commands. EW0 mode 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 program or erase operation is completed as expected or not.

21.3.2 EW1 Mode

EW1 mode is selected by setting FMR11 bit to 1 (by writing 0 and then 1 in succession) after setting the FMR01 bit to 1 (by writing 0 and then 1 in succession). (Both bits must be set to 0 first before setting to 1.) The FMR0 register indicates whether or not a program or erase operation has been completed as expected. The status register cannot be read in EW1 mode.

When an erase/program operation is initiated the CPU halts all program execution until the operation is completed or erase-suspend is requested.

21.3.3 Registers FMR0 and FMR1

Figure 21.4 shows Registers FMR0 and FMR1.

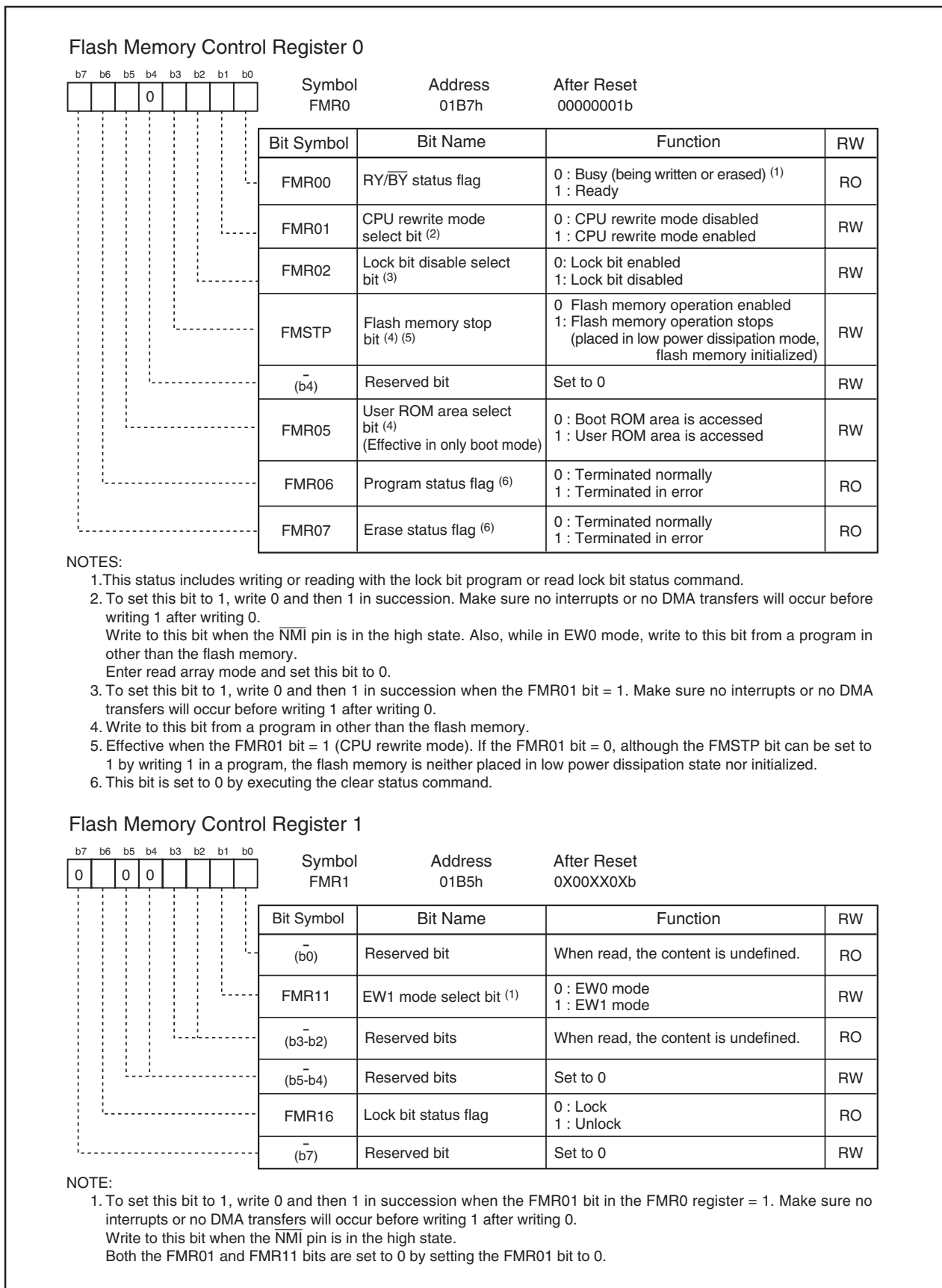


Figure 21.4 Registers FMR0 and FMR1

21.3.3.1 FMR00 Bit

This bit indicates the operating status of the flash memory. It is set to 0 while the program, block erase, erase all unlocked block, lock bit program, or read lock bit status command is being executed; otherwise, it is set to 1.

21.3.3.2 FMR01 Bit

The MCU can accept commands when the FMR01 bit is set to 1 (CPU rewrite mode). Set the FMR05 bit to 1 (user ROM area access) as well if in boot mode.

21.3.3.3 FMR02 Bit

The lock bit is disabled by setting the FMR02 bit to 1 (lock bit disabled). (Refer to **21.3.6 Data Protect Function**.) The lock bit is enabled by setting the FMR02 bit to 0 (lock bit enabled).

The FMR02 bit does not change the lock bit status but disables the lock bit function. If the block erase or erase all unlocked block command is executed when the FMR02 bit is set to 1, the lock bit status changes 0 (locked) to 1 (unlocked) after command execution is completed.

21.3.3.4 FMSTP Bit

The FMSTP bit resets the flash memory control circuits and minimizes power consumption in the flash memory. Access to the flash memory is disabled when the FMSTP bit is set to 1 (flash memory operation stops). Set the FMSTP bit by program in a space other than the flash memory.

Set the FMSTP bit to 1 if one of the followings occurs:

- A flash memory access error occurs while erasing or programming in EW0 mode (FMR00 bit does not switch back to 1 (ready))
- Low power dissipation mode or on-chip oscillator low power dissipation mode is entered

Use the following the procedure to change the FMSTP bit setting.

- (1) Set the FMSTP bit to 1
- (2) Set tps (the wait time to stabilize flash memory circuit)
- (3) Set the FMSTP bit to 0
- (4) Set tps (the wait time to stabilize flash memory circuit)

Figure 21.7 shows the Processing Before and After Low Power Dissipation Mode or On-chip Oscillator Low Power Dissipation Mode. Follow the procedure on this flow chart.

When entering stop or wait mode, the flash memory is automatically turned off. When exiting stop or wait mode, the flash memory is turned back on. The FMR0 register does not need to be set.

21.3.3.5 FMR05 Bit

This bit selects the boot ROM or user ROM area in boot mode. Set to 0 to access (read) the boot ROM area or to 1 (user ROM access) to access (read, write or erase) the user ROM area.

21.3.3.6 FMR06 Bit

This is a read-only bit indicating the status of an auto-program operation. The FMR06 bit is set to 1 when a program error occurs; otherwise, it is set to 0. Refer to **21.3.8 Full Status Check**.

21.3.3.7 FMR07 Bit

This is a read-only bit indicating the status of an auto-erase operation. The FMR07 bit is set to 1 when an erase error occurs; otherwise, it is set to 0. For details, refer to **21.3.8 Full Status Check**.

21.3.3.8 FMR11 Bit

EW0 mode is entered by setting the FMR11 bit to 0 (EW0 mode).

EW1 mode is entered by setting the FMR11 bit to 1 (EW1 mode).

21.3.3.9 FMR16 Bit

This is a read-only bit indicating the execution result of the read lock bit status command. When the block, where the read lock bit status command is executed, is locked, the FMR16 bit is set to 0.

When the block, where the read lock bit status command is executed, is unlocked, the FMR16 bit is set to 1.

Figure 21.5 shows the Setting and Resetting of EW0 Mode. Figure 21.6 show the Setting and Resetting of EW1 Mode.

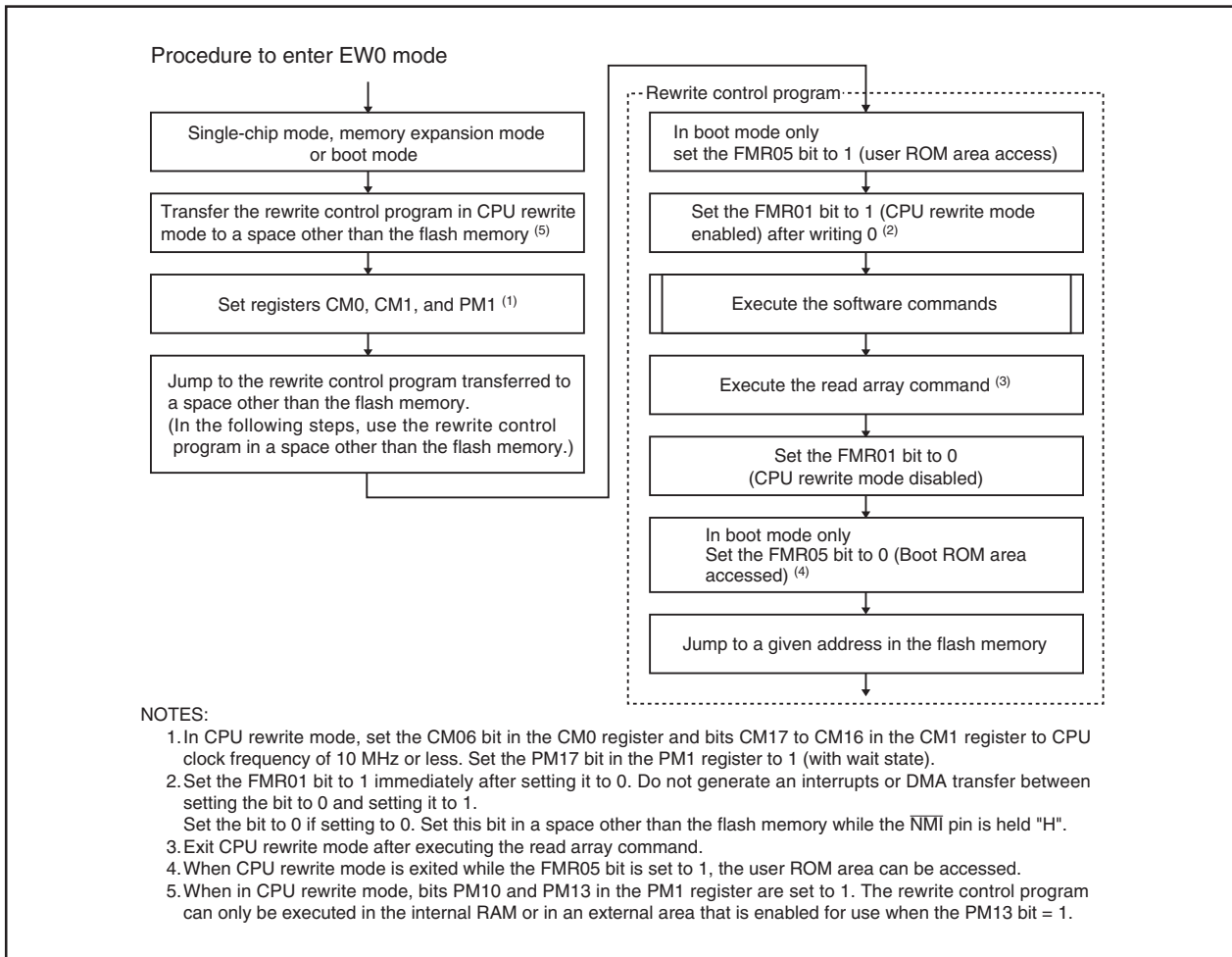


Figure 21.5 Setting and Resetting of EW0 Mode

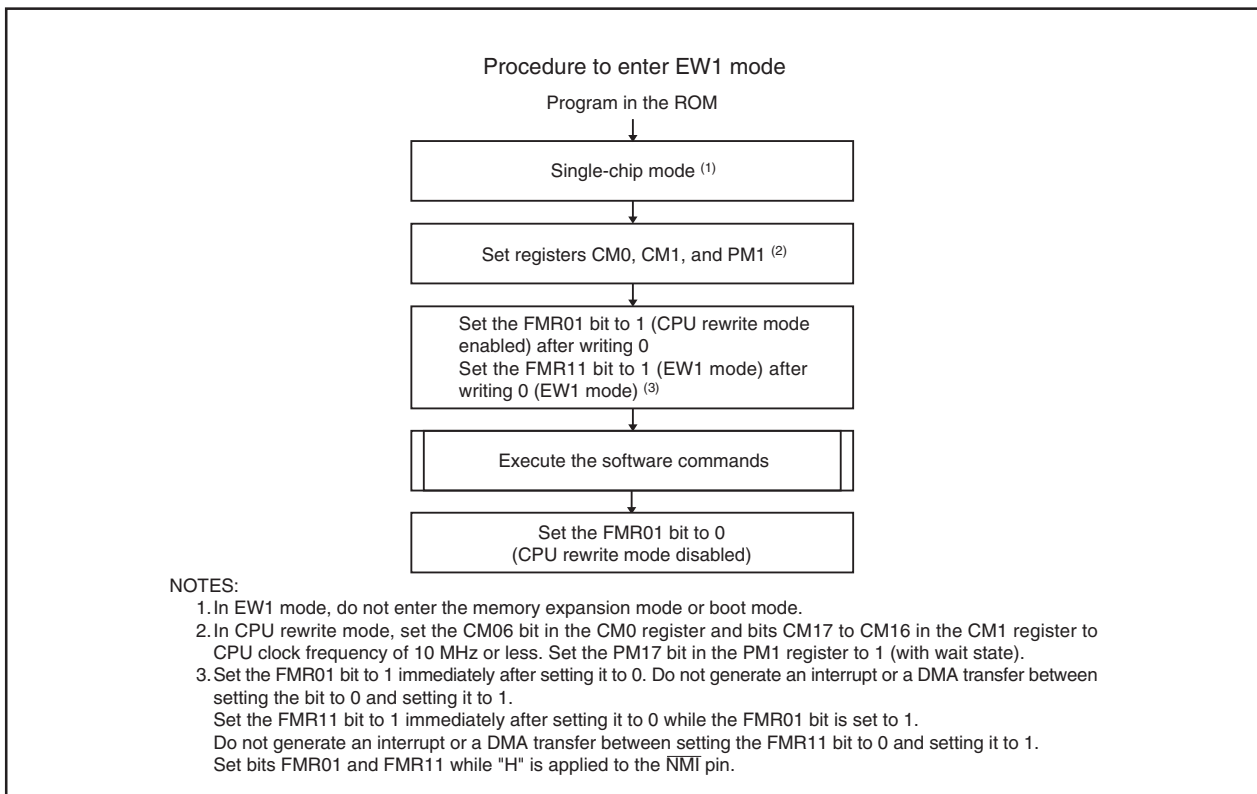


Figure 21.6 Setting and Resetting of EW1 Mode

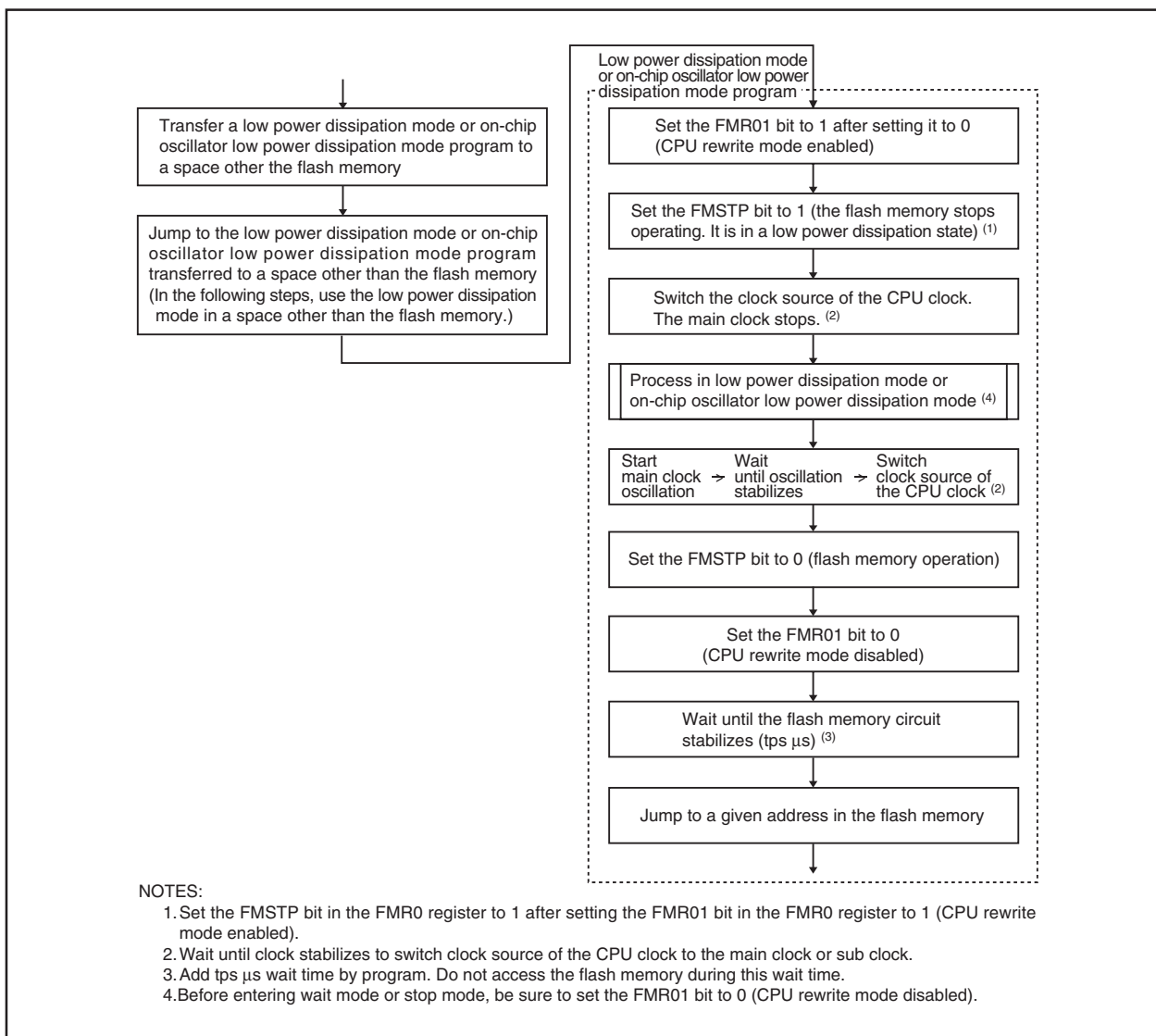


Figure 21.7 Processing Before and After Low Power Dissipation Mode or On-chip Oscillator Low Power Dissipation Mode

21.3.4 Notes on CPU Rewrite Mode

21.3.4.1 Operating Speed

Before entering CPU rewrite mode (EW0 or EW1 mode), set the CM11 bit in the CM1 register to 0 (main clock), select 10 MHz or less for CPU clock using the CM06 bit in the CM0 register and bits CM17 to CM16 in the CM1 register. Also, set the PM17 bit in the PM1 register to 1 (with wait state).

21.3.4.2 Prohibited Instructions

The following instructions cannot be used in EW0 mode because the CPU tries to read data in flash memory: the UND instruction, INTO instruction, JMPS instruction, JSRS instruction, and BRK instruction

21.3.4.3 Interrupts (EW0 Mode)

- 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 request is generated. Allocate the jump addresses for each interrupt service routines to the fixed vector table. Flash memory rewrite operation is suspended when the $\overline{\text{NMI}}$ or watchdog timer interrupt request is generated. Execute the rewrite program again after exiting the interrupt routine.
- The address match interrupt is not available since the CPU tries to read data in the flash memory.

21.3.4.4 Interrupts (EW1 Mode)

- Do not acknowledge any interrupts with vectors in the relocatable vector table or address match interrupt during auto-programming or auto-erasure.
- Do not use the watchdog timer interrupt.
- The $\overline{\text{NMI}}$ interrupt is available since registers FMR0 and FMR1 are forcibly reset when the interrupt request is generated. Allocate the jump address for the interrupt service routine to the fixed vector table. Flash memory rewrite operation is suspended when the $\overline{\text{NMI}}$ interrupt request is generated. Execute the rewrite program again after exiting the interrupt service routine.

21.3.4.5 How to Access

To set the FMR01, FMR02 or FMR11 bit to 1, write 1 after first setting the bit 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 while an "H" signal is applied to the $\overline{\text{NMI}}$ pin.

21.3.4.6 Rewriting in User ROM Area (EW0 Mode)

If the supply voltage drops while rewriting the block where the rewrite control program is stored, the flash memory cannot be rewritten because the rewrite control program is not correctly rewritten. If this error occurs, rewrite the user ROM area while in standard serial I/O mode, parallel I/O mode, or CAN I/O mode.

21.3.4.7 Rewriting in User ROM Area (EW1 Mode)

Avoid rewriting any block in which the rewrite control program is stored.

21.3.4.8 DMA Transfer

In EW1 mode, do not perform a DMA transfer while the FMR00 bit in the FMR0 register is set to 0 (auto-programming or auto-erasure).

21.3.4.9 Writing Command and Data

Write commands and data to even addresses in the user ROM area.

21.3.4.10 Wait Mode

When entering wait mode, set the FMR01 bit in the FMR0 register to 0 (CPU rewrite mode disabled) before executing the WAIT instruction.

21.3.4.11 Stop Mode

When entering 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.

21.3.4.12 Low Power Dissipation Mode and On-chip Oscillator Low Power Dissipation Mode

If the CM05 bit is set to 1 (main clock stopped), do not execute the following commands:

- Program
- Block erase
- Erase all unlocked blocks
- Lock bit program
- Read lock bit status

21.3.5 Software Commands

Software commands are described below. The command code and data must be read and written in 16-bit unit, to and from even addresses in the user ROM area. When writing command code, the high-order 8 bits (D15 to D8) are ignored.

Table 21.4 lists the Software Commands.

Table 21.4 Software Commands

| Software Command | First Bus Cycle | | | Second Bus Cycle | | |
|---|-----------------|---------|------------------|------------------|---------|------------------|
| | Mode | Address | Data (D15 to D0) | Mode | Address | Data (D15 to D0) |
| Read array | Write | X | xxFFh | - | - | - |
| Read status register | Write | X | xx70h | Read | X | SRD |
| Clear status register | Write | X | xx50h | - | - | - |
| Program | Write | WA | xx40h | Write | WA | WD |
| Block erase | Write | X | xx20h | Write | BA | xxD0h |
| Erase all unlocked block ⁽¹⁾ | Write | X | xxA7h | Write | X | xxD0h |
| Lock bit program | Write | BA | xx77h | Write | BA | xxD0h |
| Read lock bit status | Write | X | xx71h | Write | BA | xxD0h |

SRD: data in the SRD register (D7 to D0)

WA: Address to be written (The address specified in the first bus cycle is the same even address as the address specified in the second bus cycle.)

WD: 16-bit write data

BA: Highest-order block address (must be an even address)

X: Given even address in the user ROM area

xx: High-order 8 bits of command code (ignored)

NOTE:

1. Blocks 0 to 8 can be erased by the erase all unlocked block command.

Block A cannot be erased. The block erase command must be used to erase the block A.

21.3.5.1 Read Array Command (FFh)

The read array command reads the flash memory.

By writing command code xxFFh in the first bus cycle, read array mode is entered. 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 from multiple addresses can be read consecutively.

21.3.5.2 Read Status Register Command (70h)

The read status register command reads the status register (refer to **21.3.7 Status Register (SRD Register)** for detail).

By writing command code xx70h in the first bus cycle, the status register can be read in the second bus cycle. Read an even address in the user ROM area.

Do not execute this command in EW1 mode.

21.3.5.3 Clear Status Register Command (50h)

The clear status register command clears the status register.

By writing xx50h in the first bus cycle, bits FMR07 to FMR06 in the FMR0 register are set to 00b and bits SR5 to SR4 in the status register are set to 00b.

21.3.5.4 Program Command (40h)

The program command writes 2-byte data to the flash memory.

By writing xx40h in the first bus cycle and data to the write address in the second bus cycle, an auto-program operation (data program and verify) will start. 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-program operation has been completed. The FMR00 bit is set to 0 (busy) during auto-programming and to 1 (ready) when an auto-program operation is completed.

After the completion of an auto-program operation, the FMR06 bit in the FMR0 register indicates whether or not the auto-program operation has been completed as expected. (Refer to **21.3.8 Full Status Check**.)

An address that is already written cannot be altered or rewritten.

Figure 21.8 shows a flow chart of the Program Command.

The lock bit protects each block from being programmed inadvertently. (Refer to **21.3.6 Data Protect Function**.)

In EW1 mode, do not execute this command on the block where the rewrite control program is allocated. In EW0 mode, the MCU enters read status register mode as soon as an auto-program operation starts. The status register can be read. The SR7 bit in the status register is set to 0 at the same time an auto-program operation starts. It is set to 1 when auto-program operation is completed. The MCU remains in read status register mode until the read array command is written. After completion of an auto-program operation, the status register indicates whether or not the auto-program operation has been completed as expected.

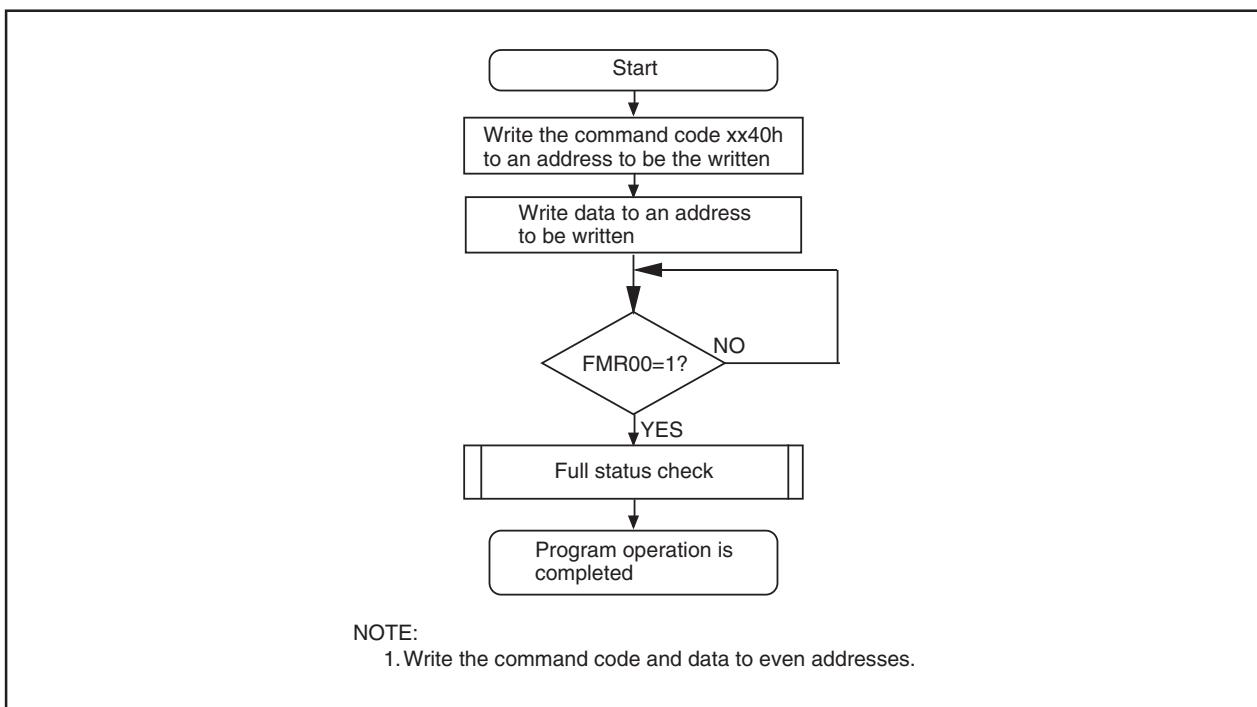


Figure 21.8 Program Command

21.3.5.5 Block Erase Command

The block erase command erases each block.

By writing xx20h in the first bus cycle and xxD0h to the highest-order even address of a block in the second bus cycle, an auto-erase operation (erase and verify) will start in the specified block.

The FMR00 bit in the FMR0 register indicates whether an auto-erase operation has been completed.

The FMR00 bit is set to 0 (busy) during auto-erasure and to 1 (ready) when the auto-erase operation is completed.

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 completed as expected. (Refer to **21.3.8 Full Status Check**.)

Figure 21.9 shows a flow chart of the Block Erase Command.

The lock bit protects each block from being programmed inadvertently. (Refer to **21.3.6 Data Protect Function**.)

In EW1 mode, do not execute this command on the block where the rewrite control program is allocated.

In EW0 mode, the MCU enters read status register mode as soon as an auto-erase operation starts. The status register can be read. The SR7 bit in the status register is set to 0 at the same time an auto-erase operation starts.

It is set to 1 when an auto-erase operation is completed. The MCU remains in read status register mode until the read array command or read lock bit status command is written. Also execute the clear status register command and block erase command at least 3 times until an erase error is not generated when an erase error is generated.

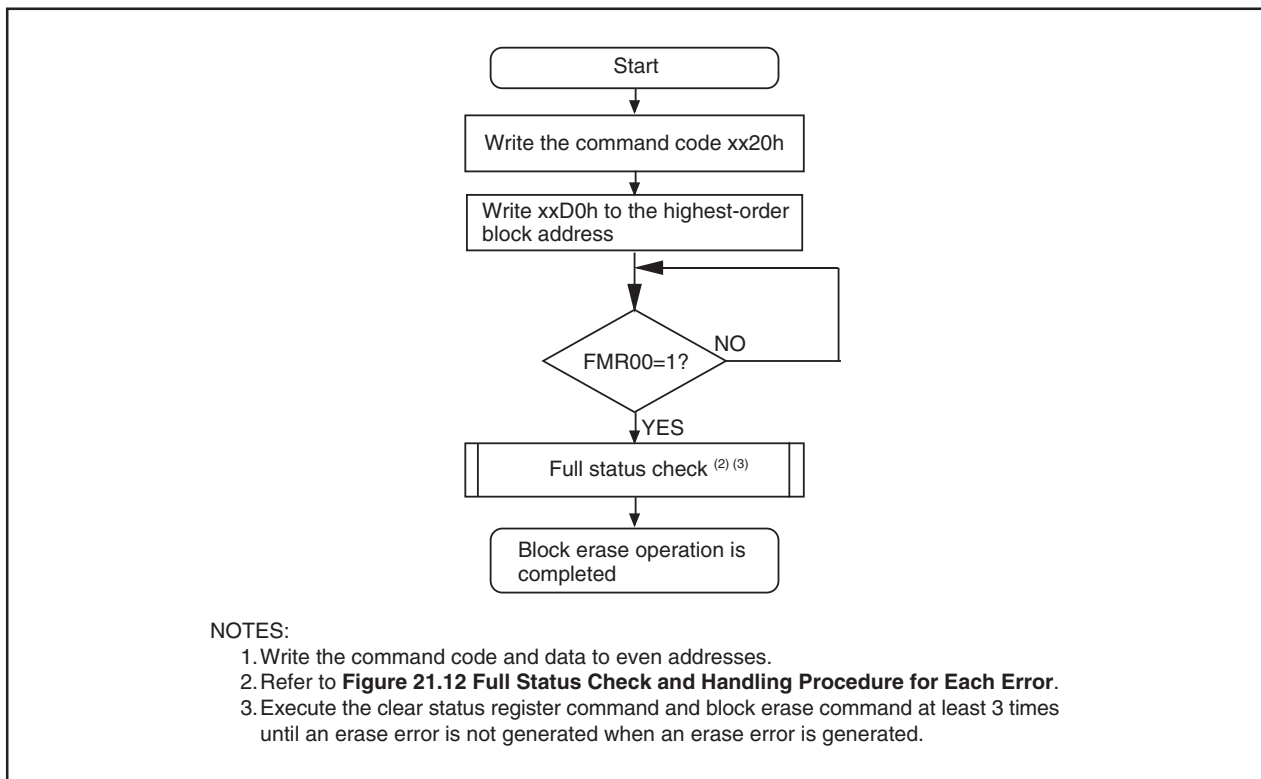


Figure 21.9 Block Erase Command

21.3.5.6 Erase All Unlocked Block

The erase all unlocked block command erases all blocks except the block A.

By writing xxA7h in the first bus cycle and xxD0h in the second bus cycle, an auto-erase (erase and verify) operation will run continuously in all blocks except the block A.

The FMR00 bit in the FMR0 register indicates whether an auto-erase operation has been completed.

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 completed as expected.

The lock bit can protect each block from being programmed inadvertently. (Refer to **21.3.6 Data Protect Function**.)

In EW1 mode, do not execute this command when the lock bit for any block storing the rewrite control program is set to 1 (unlocked) or when the FMR02 bit in the FMR0 register is set to 1 (lock bit disabled).

In EW0 mode, the MCU enters read status register mode as soon as an auto-erase operation starts. The status register can be read. The SR7 bit in the status register is set to 0 (busy) at the same time an auto-erase operation starts. It is set to 1 (ready) when an auto-erase operation is completed. The MCU remains in read status register mode until the read array command or read lock bit status command is written.

Only blocks 0 to 8 can be erased by the erase all unlocked block command. The block A cannot be erased. Use the block erase command to erase the block A.

21.3.5.7 Lock Bit Program Command

The lock bit program command sets the lock bit for a specified block to 0 (locked).

By writing xx77h in the first bus cycle and xxD0h to the highest-order even address of a block in the second bus cycle, the lock bit for the specified block is set to 0. The address value specified in the first bus cycle must be the same highest-order even address of a block specified in the second bus cycle.

Figure 21.10 shows a flow chart of the Lock Bit Program Command. Execute read lock bit status command to read lock bit state (lock bit data).

The FMR00 bit in the FMR0 register indicates whether a lock bit program operation is completed.

Refer to **21.3.6 Data Protect Function** for details on lock bit functions and how to set it to 1 (unlocked).

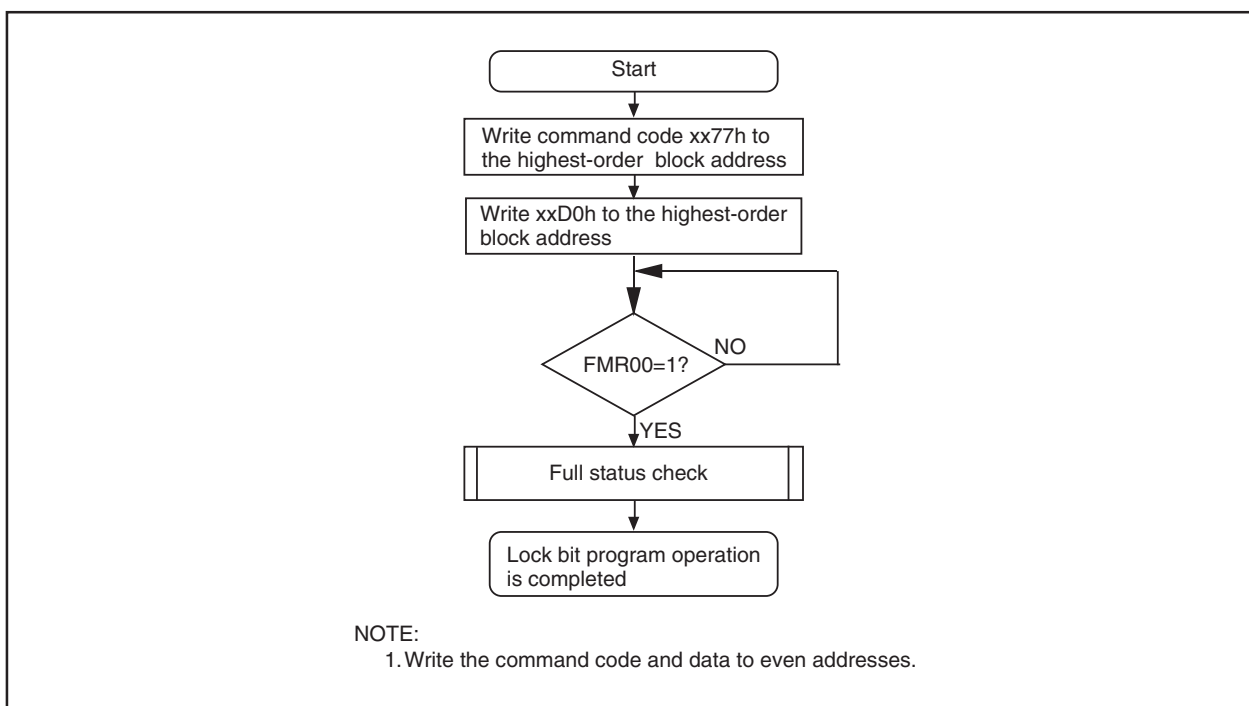


Figure 21.10 Lock Bit Program Command

21.3.5.8 Read Lock Bit Status Command (71h)

The read lock bit status command reads the lock bit state of a specified block.

By writing xx71h in the first bus cycle and xxD0h to the highest-order even address of a block in the second bus cycle, the FMR16 bit in the FMR1 register stores information on whether or not the lock bit of a specified block is locked. Read the FMR16 bit after the FMR00 bit in the FMR0 register is set to 1 (ready).

Figure 21.11 shows a flow chart of the Read Lock Bit Status Command.

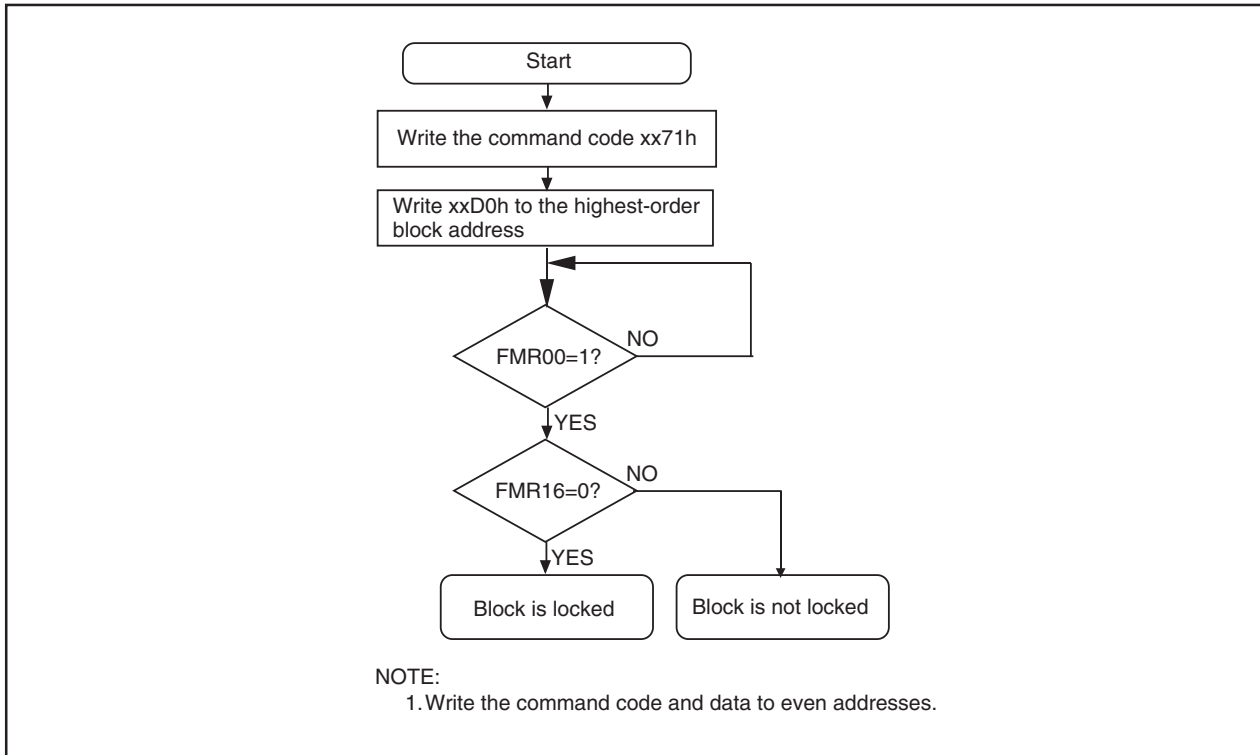


Figure 21.11 Read Lock Bit Status Command

21.3.6 Data Protect Function

Each block in the flash memory has a nonvolatile lock bit. The lock bit is enabled by setting the FMR02 bit in the FMR0 register to 0 (lock bit enabled). The lock bit allows each block to be individually protected (locked) against program and erase. This helps prevent data from being inadvertently written to or erased from the flash memory.

- When the lock bit status is set to 0, the block is locked (block is protected against program and erase).
- When the lock bit status is set to 1, the block is not locked (block can be programmed or erased).

The lock bit status is set to 0 (locked) by executing the lock bit program command and to 1 (unlocked) by erasing the block. The lock bit status cannot be set to 1 by any commands.

The lock bit status can be read by the read lock bit status command.

The lock bit function is disabled by setting the FMR02 bit to 1 (lock bit disabled). All blocks are unlocked. However, individual lock bit status remains unchanged. The lock bit function is enabled by setting the FMR02 bit to 0. Lock bit status is retained.

If the block erase or erase all unlocked block command is executed while the FMR02 bit is set to 1, the target block or all blocks are erased regardless of lock bit status. The lock bit status of each block are set to 1 after an erase operation is completed.

Refer to **21.3.5 Software Commands** for details on each command.

21.3.7 Status Register (SRD Register)

The status register indicates the operating status of the flash memory and whether or not an erase or program operation is completed as expected. Bits FMR00, FMR06, and FMR07 in the FMR0 register indicate status register states.

Table 21.5 shows the Status Register.

In EW0 mode, the status register can be read when the followings occur.

- Given even address in the user ROM area is read after writing the read status register command.
- Given even address in the user ROM area is read from when the program, block erase, erase all unlocked block, or lock bit program command is executed until when the read array command is executed.

21.3.7.1 Sequencer Status (Bits SR7 and FMR00)

The sequencer status indicates the operating status of the flash memory. It is set to 0 while the program, block erase, erase all unlocked block, lock bit program, or read lock bit status command is being executed; otherwise, it is set to 1.

21.3.7.2 Erase Status (Bits SR5 and FMR07)

Refer to **21.3.8 Full Status Check**.

21.3.7.3 Program Status (Bits SR4 and FMR06)

Refer to **21.3.8 Full Status Check**.

Table 21.5 Status Register

| Bits in Status Register | Bits in FMR0 Register | Status Name | Contents | | Value after Reset |
|-------------------------|-----------------------|------------------|---------------------|---------------------|-------------------|
| | | | 0 | 1 | |
| SR0 (D0) | - | Reserved | - | - | - |
| SR1 (D1) | - | Reserved | - | - | - |
| SR2 (D2) | - | Reserved | - | - | - |
| SR3 (D3) | - | Reserved | - | - | - |
| SR4 (D4) | FMR06 | Program status | Terminated normally | Terminated in error | 0 |
| SR5 (D5) | FMR07 | Erase status | Terminated normally | Terminated in error | 0 |
| SR6 (D6) | - | Reserved | - | - | - |
| SR7 (D7) | FMR00 | Sequencer status | Busy | Ready | 1 |

D0 to D7: These data bus are read when the read status register command is executed.

NOTE:

1. Bits FMR06 (SR4) and FMR07 (SR5) are set to 0 by executing the clear status register command. When the FMR06 bit (SR4) or FMR07 bit (SR5) is set to 1, the program, block erase, erase all unlocked block and lock bit program commands are not accepted.

21.3.8 Full Status Check

If an error occurs when a program or erase operation is completed, the FMR06, FMR07 bits in the FMR0 register are set to 1, indicating a specific error. Therefore, execution results can be confirmed by checking these bits (full status check).

Table 21.6 lists the Errors and FMR0 Register Status. Figure 21.12 shows a flow chart of the Full Status Check and Handling Procedure for Each Error.

Table 21.6 Errors and FMR0 Register Status

| FRM00 Register (Status Register) Status | | Error | Error Occurrence Conditions |
|---|--------------------|------------------------|--|
| FMR07 Bit (SR5) | FMR06 Bit (SR4) | | |
| 1 | 1 | Command Sequence error | <ul style="list-style-type: none"> • Command is written incorrectly • A value other than xxD0h or xxFFh is written in the second bus cycle of the lock bit program, block erase or erase all unlocked block command ⁽¹⁾ |
| 1 | 0 | Erase error | <ul style="list-style-type: none"> • The block erase command is executed on a locked block ⁽²⁾ • The block erase or erase all unlocked block command is executed on an unlock block and auto-erase operation is not completed as expected |
| 0 | 1 | Program error | <ul style="list-style-type: none"> • The program command is executed on locked blocks ⁽²⁾ • The program command is executed on unlocked blocks and auto-program operation is not completed as expected • The lock bit program command is executed but program operation is not completed as expected |

NOTES:

1. The flash memory enters read array mode by writing command code xxFFh in the second bus cycle of these commands. The command code written in the first bus cycle becomes invalid.
2. When the FMR02 bit in the FMR0 register is set to 1 (lock bit disabled), no error occurs even under the conditions above.

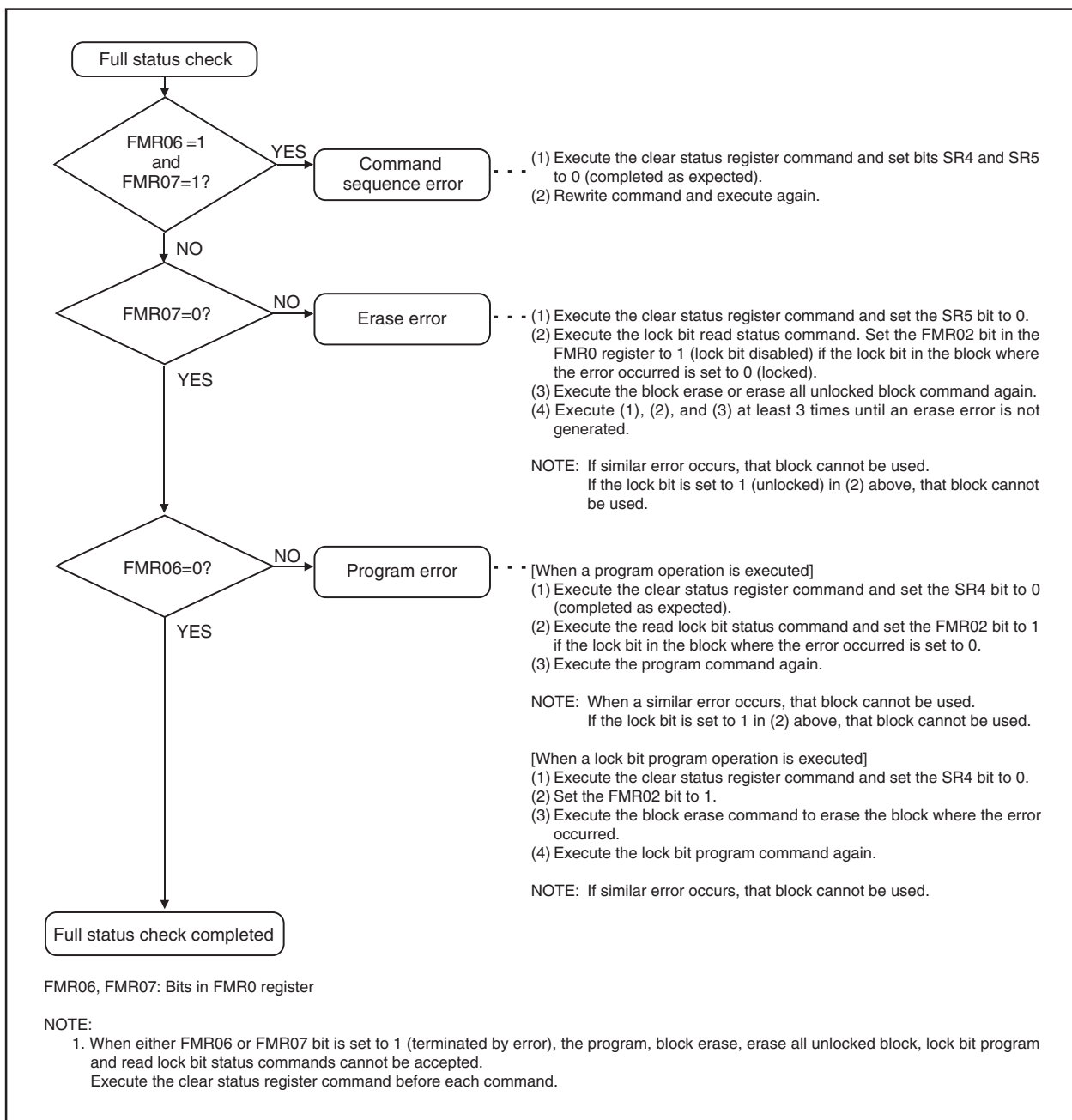


Figure 21.12 Full Status Check and Handling Procedure for Each Error

21.4 Standard Serial I/O Mode

In standard serial I/O mode, the serial programmer supporting the M16C/6N Group (M16C/6N4) can be used to rewrite the flash memory user ROM area in the MCU 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 instructions.

Table 21.7 lists the Pin Functions in Standard Serial I/O Mode. Figures 21.13 and 21.14 show the Pin Connections in Standard Serial I/O Mode.

21.4.1 ID Code Check Function

The ID code check function determines whether the ID codes sent from the serial programmer matches those written in the flash memory. (Refer to **21.2 Functions to Prevent Flash Memory from Rewriting.**)

Table 21.7 Pin Functions in Standard Serial I/O Mode

| Pin | Name | I/O | Description |
|-----------------------------|---------------------------|-----|---|
| VCC1, VCC2, VSS | Power supply input | | Apply the Flash Program, Erase Voltage to VCC1 pin and VCC2 to VCC2 pin. The VCC apply condition is that VCC2 = VCC1. Apply 0 V to VSS pin. |
| CNVSS | CNVSS | I | Connect to VCC1 pin. |
| RESET | Reset input | I | Reset input pin. While RESET pin is "L" level, input 20 cycles or longer clock to XIN pin. |
| 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 | |
| BYTE | BYTE | I | Connect this pin to VCC1 or VSS. |
| AVCC, AVSS | Analog power supply input | | Connect AVCC to VCC1 and AVSS to VSS, respectively. |
| VREF | Reference voltage input | I | Enter the reference voltage for A/D and D/A converters from this pin. |
| P0_0 to P0_7 | Input port P0 | I | Input "H" or "L" level signal or open. |
| P1_0 to P1_7 | Input port P1 | I | Input "H" or "L" level signal or open. |
| P2_0 to P2_7 | Input port P2 | I | Input "H" or "L" level signal or open. |
| P3_0 to P3_7 | Input port P3 | I | Input "H" or "L" level signal or open. |
| P4_0 to P4_7 | Input port P4 | I | Input "H" or "L" level signal or open. |
| P5_0 | CE input | I | Input "H" level signal. |
| P5_1 to P5_4, P5_6, P5_7 | Input port P5 | I | Input "H" or "L" level signal or open. |
| P5_5 | EPM input | I | Input "L" level signal. |
| P6_0 to P6_3 | Input port P6 | I | Input "H" or "L" level signal or open. |
| P6_4/RTS1 | BUSY output | O | Standard serial I/O mode 1: BUSY signal output pin Standard serial I/O mode 2: Monitors the boot program operation check signal output pin. |
| P6_5/CLK1 | SCLK input | I | Standard serial I/O mode 1: Serial clock input pin. Standard serial I/O mode 2: Input "L". |
| P6_6/RXD1 | RXD input | I | Serial data input pin |
| P6_7/TXD1 | TXD output | O | Serial data output pin ⁽¹⁾ |
| P7_0 to P7_7 | Input port P7 | I | Input "H" or "L" level signal or open. |
| P8_0 to P8_3, P8_6, P8_7 | Input port P8 | I | Input "H" or "L" level signal or open. |
| P8_4 | P8_4 input | I | Input "L" level signal. ⁽²⁾ |
| P8_5/NMI | NMI input | I | Connect this pin to VCC1. |
| P9_0 to P9_4, P9_7 | Input port P9 | I | Input "H" or "L" level signal or open. |
| P9_5/CRX0 | CRX input | I | Input "H" or "L" level signal or connect to a CAN transceiver. |
| P9_6/CTX0 | CTX output | O | Input "H" level signal, open or connect to a CAN transceiver. |
| P10_0 to P10_7 | Input port P10 | I | Input "H" or "L" level signal or open. |

NOTES:

1. When using standard serial I/O mode, It is necessary to input "H" to the TXD1(P6_7) pin while the RESET pin is "L". Therefore, the internal pull-up is enabled for the TXD1(P6_7) pin while the RESET pin is "L".
2. When using standard serial I/O mode, pins P0_0 to P0_7, P1_0 to P1_7 may become undefined while the P8_4 pin is "H" and the RESET pin is "L". If this causes a problem, apply "L" to the P8_4 pin.

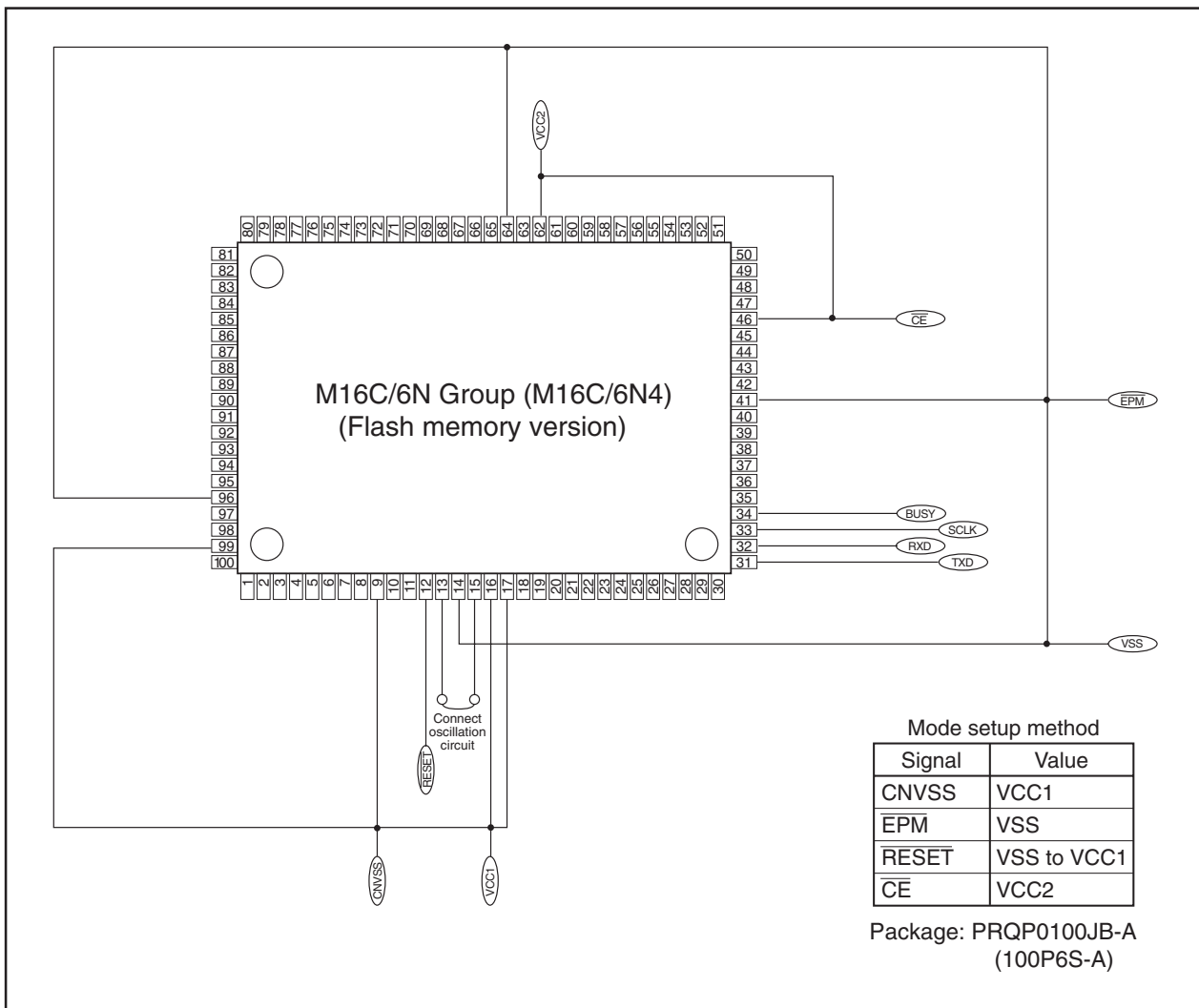


Figure 21.13 Pin Connections in Standard Serial I/O Mode (1)

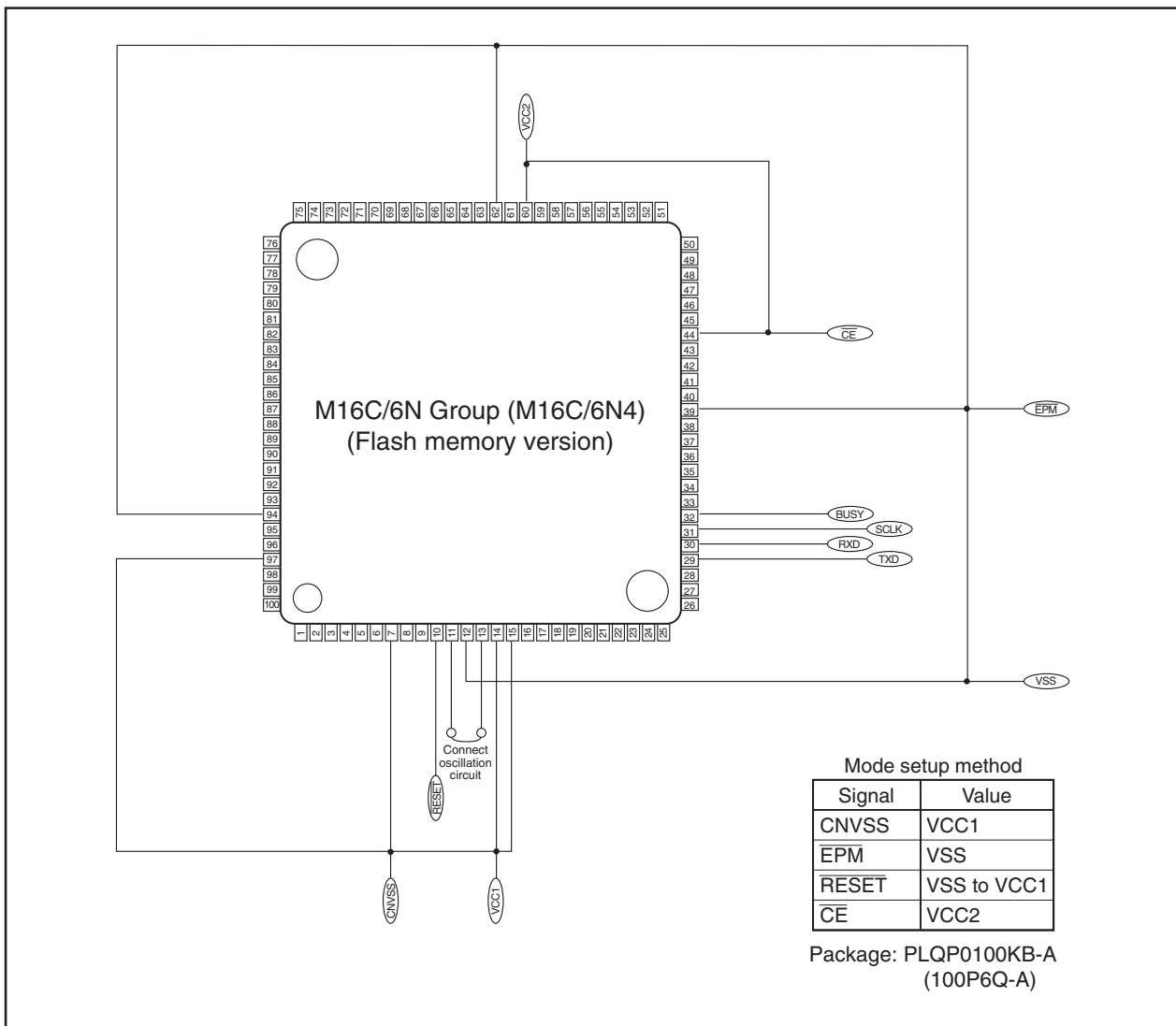


Figure 21.14 Pin Connections in Standard Serial I/O Mode (2)

21.4.2 Example of Circuit Application in Standard Serial I/O Mode

Figures 21.15 and 21.16 show the Circuit Application in Standard Serial I/O Mode 1 and Mode 2. Refer to the user's manual of your serial programmer to handle pins controlled by a serial programmer.

Note that when using standard serial I/O mode 2, make sure a main clock input oscillation frequency is set to 5 MHz, 10 MHz, or 16 MHz.

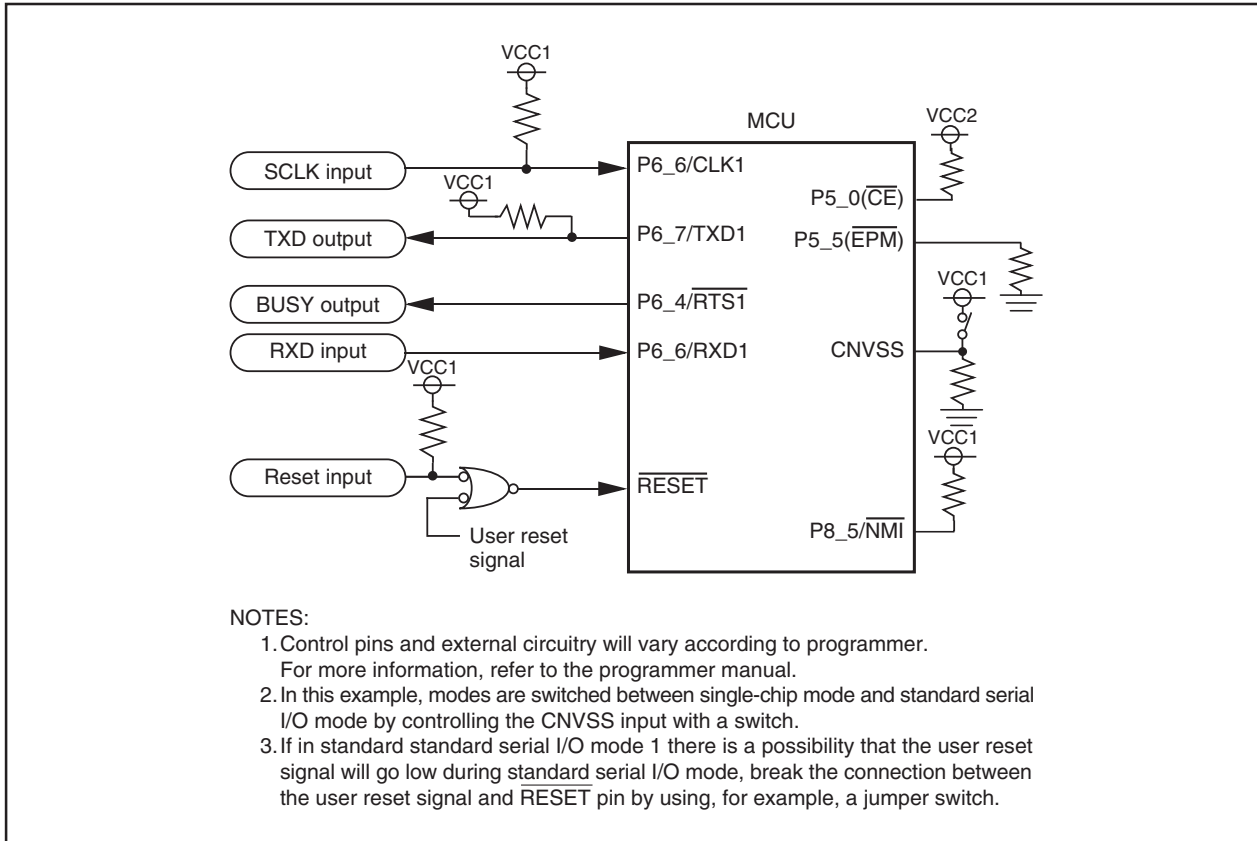


Figure 21.15 Circuit Application in Standard Serial I/O Mode 1

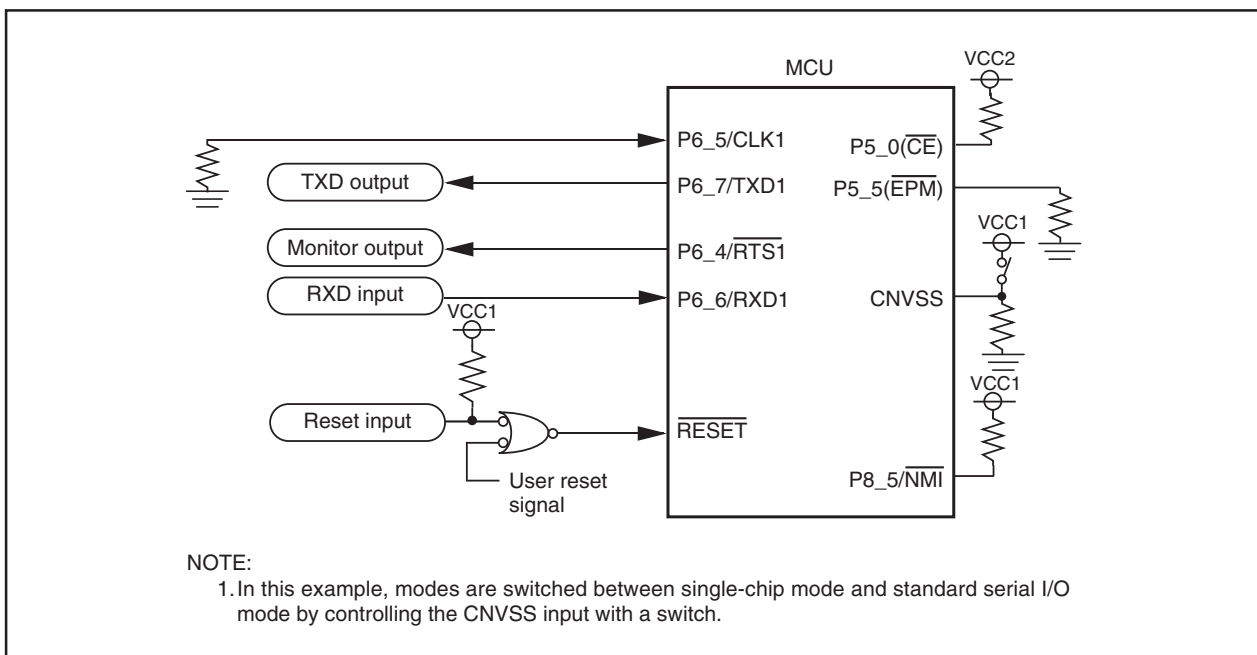


Figure 21.16 Circuit Application in Standard Serial I/O Mode 2

21.5 Parallel I/O Mode

In parallel I/O mode, the user ROM area and the boot ROM area can be rewritten by a parallel programmer supporting the M16C/6N Group (M16C/6N4). 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.

21.5.1 User ROM and Boot ROM Areas

An erase block operation in the boot ROM area is applied to only one 4-Kbyte block. The rewrite control program in standard serial I/O and CAN I/O modes are written in the boot ROM area before shipment. Do not rewrite the boot ROM area if using the serial programmer.

In parallel I/O mode, the boot ROM area is located in addresses 0FF000h to 0FFFFFFh. Rewrite this address range only if rewriting the boot ROM area. (Do not access addresses other than addresses 0FF000h to 0FFFFFFh.)

21.5.2 ROM Code Protect Function

The ROM code protect function prevents the flash memory from being read and rewritten in parallel I/O mode. (Refer to **21.2 Functions to Prevent Flash Memory from Rewriting.**)

21.6 CAN I/O Mode

In CAN I/O mode, the CAN programmer supporting the M16C/6N Group (M16C/6N4) can be used to rewrite the flash memory user ROM area in the MCU mounted on a board. For more information about the CAN programmer, contact your CAN programmer manufacturer. Refer to the user's manual included with your CAN programmer for instructions.

Table 21.8 lists pin functions in CAN I/O mode. Figures 21.17 and 21.18 show pin connections in CAN I/O mode.

21.6.1 ID Code Check Function

The ID code check function determines whether the ID codes sent from the CAN programmer matches those written in the flash memory. (Refer to **21.2 Functions to Prevent Flash Memory from Rewriting.**)

Table 21.8 Pin Functions in CAN I/O Mode

| Pin | Name | I/O | Description |
|-----------------------------|---------------------------|-----|---|
| VCC1, VCC2, VSS | Power supply input | | Apply the Flash Program, Erase Voltage to VCC1 pin and VCC2 to VCC2 pin. The VCC apply condition is that VCC2 = VCC1. Apply 0 V to VSS pin. |
| CNVSS | CNVSS | I | Connect to VCC1 pin. |
| RESET | Reset input | I | Reset input pin. While RESET pin is "L" level, input 20 cycles or longer clock to XIN pin. |
| 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 | |
| BYTE | BYTE | I | Connect this pin to VCC1 or VSS. |
| AVCC, AVSS | Analog power supply input | | Connect AVCC to VCC1 and AVSS to VSS, respectively. |
| VREF | Reference voltage input | I | Enter the reference voltage for A/D and D/A converters from this pin. |
| P0_0 to P0_7 | Input port P0 | I | Input "H" or "L" level signal or open. |
| P1_0 to P1_7 | Input port P1 | I | Input "H" or "L" level signal or open. |
| P2_0 to P2_7 | Input port P2 | I | Input "H" or "L" level signal or open. |
| P3_0 to P3_7 | Input port P3 | I | Input "H" or "L" level signal or open. |
| P4_0 to P4_7 | Input port P4 | I | Input "H" or "L" level signal or open. |
| P5_0 | CE input | I | Input "H" level signal. |
| P5_1 to P5_4, P5_6, P5_7 | Input port P5 | I | Input "H" or "L" level signal or open. |
| P5_5 | EPM input | I | Input "L" level signal. |
| P6_0 to P6_4, P6_6 | Input port P6 | I | Input "H" or "L" level signal or open. |
| P6_5/CLK1 | SCLK input | I | Input "L" level signal. |
| P6_7/TXD1 | TXD output | O | Input "H" level signal. |
| P7_0 to P7_7 | Input port P7 | I | Input "H" or "L" level signal or open. |
| P8_0 to P8_3, P8_6, P8_7 | Input port P8 | I | Input "H" or "L" level signal or open. |
| P8_4 | P8_4 Input | I | Input "L" level signal. ⁽¹⁾ |
| P8_5/NMI | NMI input | I | Connect this pin to VCC1. |
| P9_0 to P9_4, P9_7 | Input port P9 | I | Input "H" or "L" level signal or open. |
| P9_5/CRX0 | CRX input | I | Connect to a CAN transceiver. |
| P9_6/CTX0 | CTX output | O | Connect to a CAN transceiver. |
| P10_0 to P10_7 | Input port P10 | I | Input "H" or "L" level signal or open |

NOTE:

1. When using CAN I/O mode, pins P0_0 to P0_7, P1_0 to P1_7 may become undefined while the P8_4 pin is "H" and the RESET pin is "L". If this causes a problem, apply "L" to the P8_4 pin.

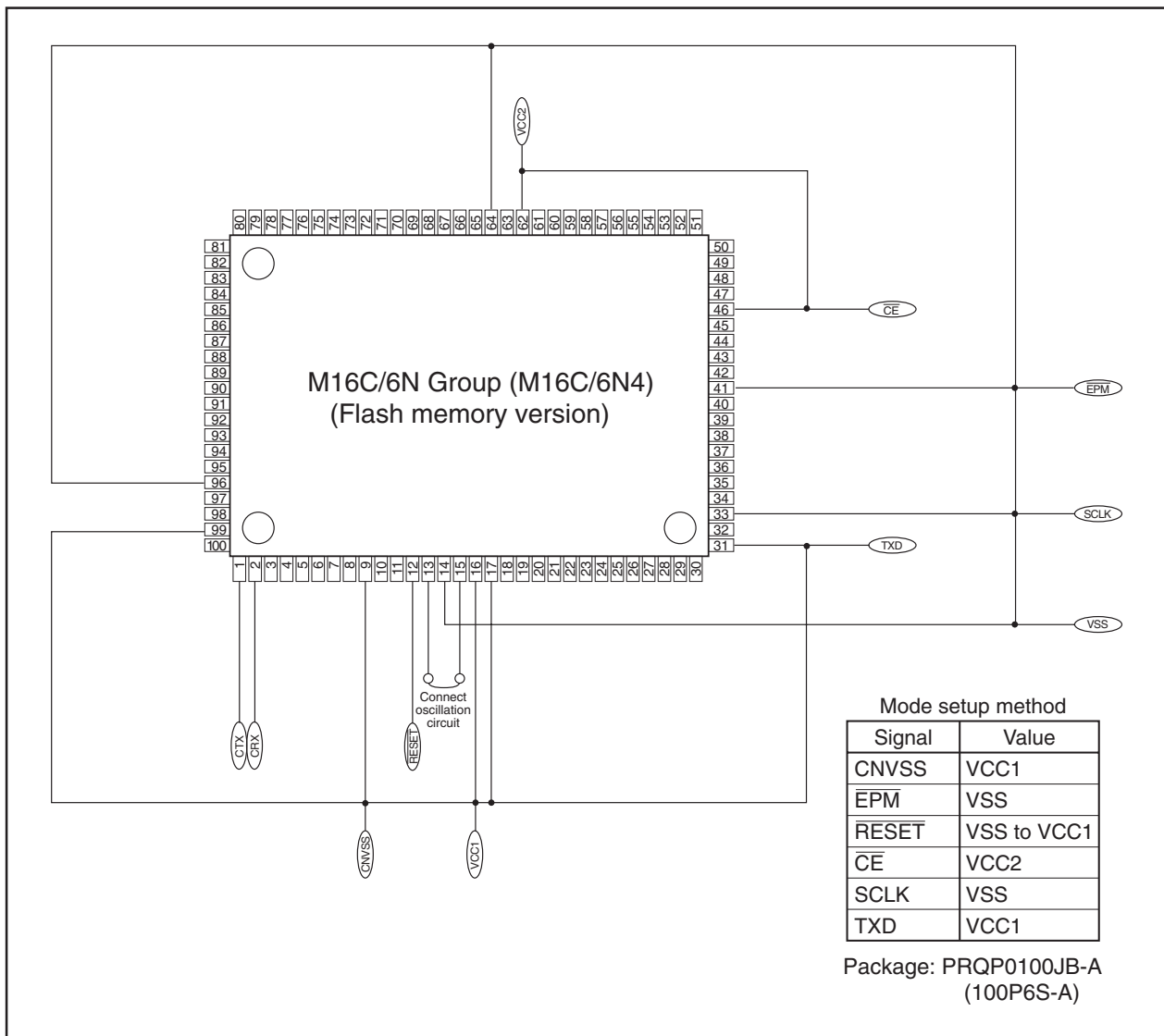


Figure 21.17 Pin Connections in CAN I/O Mode (1)

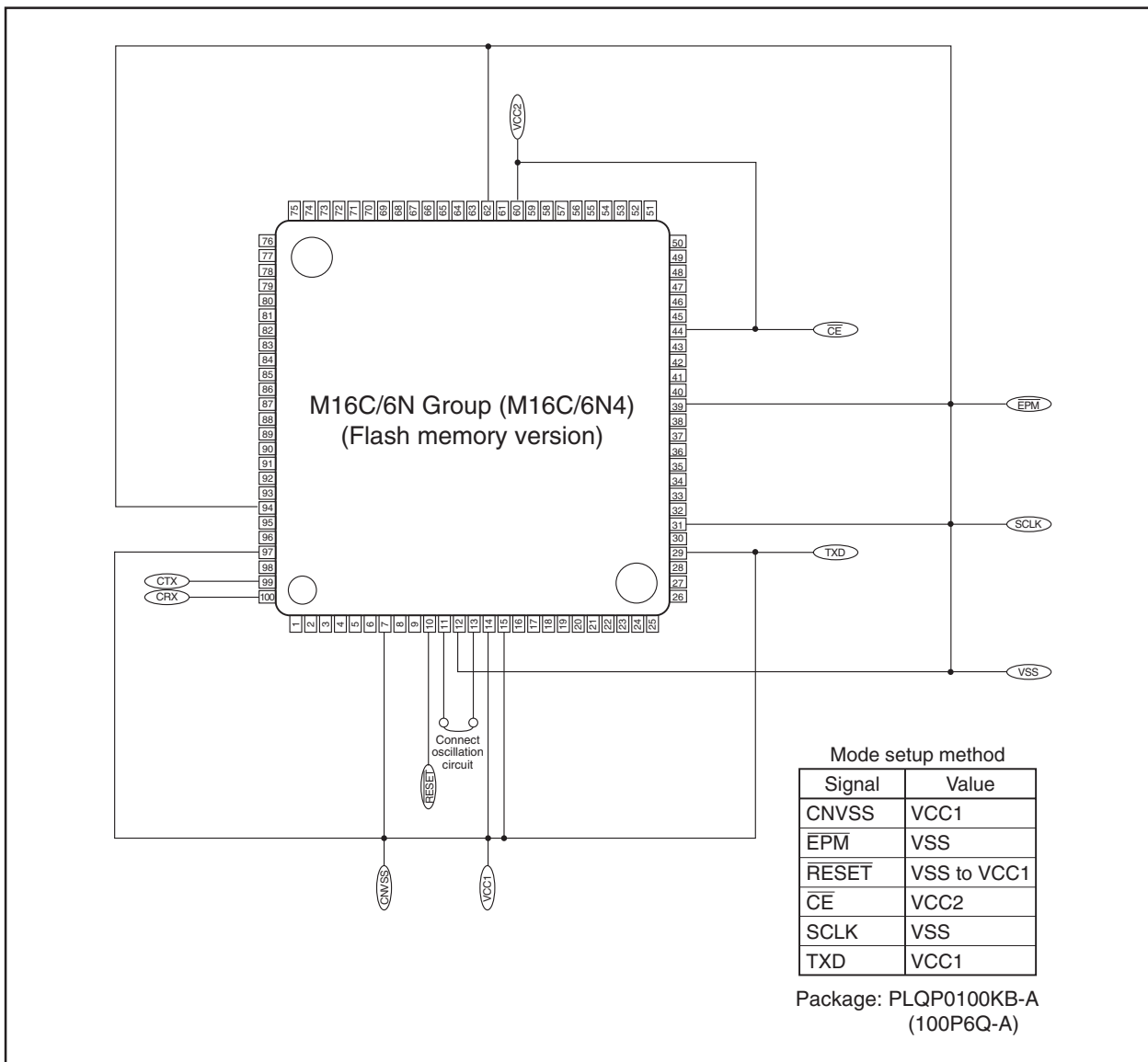


Figure 21.18 Pin Connections in CAN I/O Mode (2)

21.6.2 Example of Circuit Application in CAN I/O Mode

Figure 21.19 shows the Circuit Application in CAN I/O Mode. Refer to the user's manual of your CAN programmer to handle pins controlled by a CAN programmer.

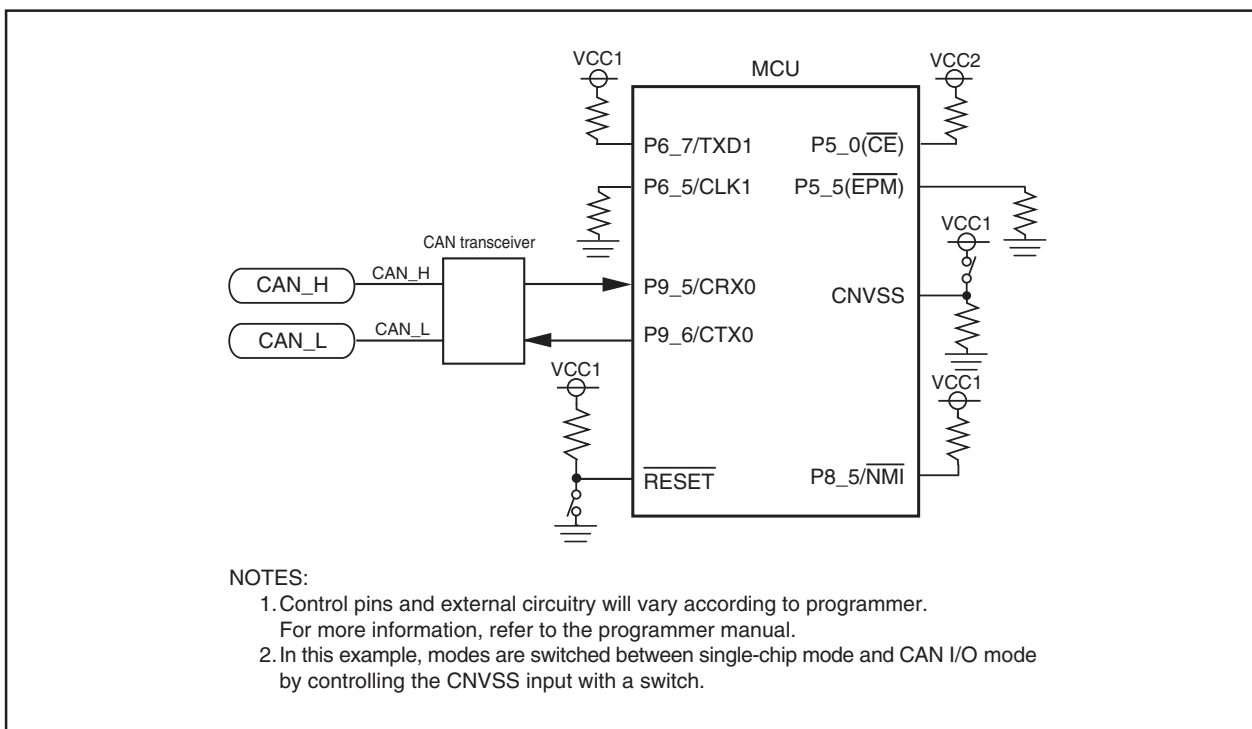


Figure 21.19 Circuit Application in CAN I/O Mode

21.7 Electrical Characteristics

21.7.1 Electrical Characteristics (T/V-ver.)

Table 21.9 lists the Flash Memory Electrical Characteristics. Table 21.10 lists the Flash Memory Version Program/Erase Voltage and Read Operation Voltage Characteristics.

Table 21.9 Flash Memory Version Electrical Characteristics ⁽¹⁾

| Symbol | Parameter | Standard | | | Unit |
|--------|--|----------------|------|----------------------|-------|
| | | Min. | Typ. | Max. | |
| - | Programming and erasure endurance ⁽²⁾ | 100 | | | cycle |
| - | Word program time (VCC = 5.0 V) | | 25 | 200 | μs |
| - | Lock bit program time | | 25 | 200 | μs |
| - | Block erase time (VCC = 5.0 V) | 4-Kbyte block | 0.3 | 4 | s |
| | | 8-Kbyte block | 0.3 | 4 | s |
| | | 32-Kbyte block | 0.5 | 4 | s |
| | | 64-Kbyte block | 0.8 | 4 | s |
| - | Erase all unlocked blocks time | | | 4 × n ⁽³⁾ | s |
| tps | Flash memory circuit stabilization wait time | | | 15 | μs |

NOTES:

1. Referenced to VCC = 4.5 to 5.5 V, Topr = 0 to 60°C unless otherwise specified.
2. Programming and erasure endurance refers to the number of times a block erase can be performed. If the programming and erasure endurance is n (n = 100), each block can be erased n times. For example, if a 4-Kbyte block A is erased after writing 1 word data 2,048 times, each to a different address, this counts as one programming and erasure endurance. Data cannot be written to the same address more than once without erasing the block (rewrite prohibited).
3. n denotes the number of block erases.

**Table 21.10 Flash Memory Version Program/Erase Voltage and Read Operation Voltage Characteristics
(at Topr = 0 to 60 °C)**

| Flash Program, Erase Voltage | Flash Read Operation Voltage |
|------------------------------|------------------------------|
| VCC = 5.0 ± 0.5 V | VCC = 4.2 to 5.5 V |

21.7.2 Electrical Characteristics (Normal-ver.)

Table 21.11 lists the Flash Memory Electrical Characteristics. Table 21.12 lists the Flash Memory Version Program/Erase Voltage and Read Operation Voltage Characteristics.

Table 21.11 Flash Memory Version Electrical Characteristics ⁽¹⁾

| Symbol | Parameter | Standard | | | Unit |
|--------|--|----------------|------|----------------------|-------|
| | | Min. | Typ. | Max. | |
| - | Programming and erasure endurance ⁽²⁾ | 100 | | | cycle |
| - | Word program time (VCC = 5.0 V) | | 25 | 200 | μs |
| - | Lock bit program time | | 25 | 200 | μs |
| - | Block erase time (VCC = 5.0 V) | 4-Kbyte block | 0.3 | 4 | s |
| | | 8-Kbyte block | 0.3 | 4 | s |
| | | 32-Kbyte block | 0.5 | 4 | s |
| | | 64-Kbyte block | 0.8 | 4 | s |
| - | Erase all unlocked blocks time | | | 4 × n ⁽³⁾ | s |
| tps | Flash memory circuit stabilization wait time | | | 15 | μs |

NOTES:

1. Referenced to VCC = 4.5 to 5.5 V, 3.0 to 3.6 V, Topr = 0 to 60°C unless otherwise specified.
2. Programming and erasure endurance refers to the number of times a block erase can be performed. If the programming and erasure endurance is n (n = 100), each block can be erased n times. For example, if a 4-Kbyte block A is erased after writing 1 word data 2,048 times, each to a different address, this counts as one programming and erasure endurance. Data cannot be written to the same address more than once without erasing the block (rewrite prohibited).
3. n denotes the number of block erases.

Table 21.12 Flash Memory Version Program/Erase Voltage and Read Operation Voltage Characteristics (at Topr = 0 to 60 °C)

| Flash Program, Erase Voltage | Flash Read Operation Voltage |
|----------------------------------|------------------------------|
| VCC = 3.3 ± 0.3 V or 5.0 ± 0.5 V | VCC = 3.0 to 5.5 V |

22. Electrical Characteristics

22.1 Electrical Characteristics (T/V-ver.)

Table 22.1 Absolute Maximum Ratings

| Symbol | Parameter | | Condition | Rated Value | Unit |
|------------------|-------------------------------|--|-------------------------|--|------|
| V _{CC} | Supply voltage (VCC1 = VCC2) | | VCC = AVCC | -0.3 to 6.5 | V |
| AV _{CC} | Analog supply voltage | | VCC = AVCC | -0.3 to 6.5 | V |
| V _I | Input voltage | RESET, CNVSS, BYTE, P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7, VREF, XIN | | -0.3 to VCC+0.3 | V |
| | | P7_1, P9_1 | | -0.3 to 6.5 | 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, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7, XOUT | | -0.3 to VCC+0.3 | V |
| | | P7_1, P9_1 | | -0.3 to 6.5 | V |
| P _d | Power dissipation | | T _{opr} = 25°C | 700 | mW |
| T _{opr} | Operating ambient temperature | During MCU operation | | T version: -40 to 85 V version: -40 to 125 (option) | °C |
| | | During flash memory program and erase operation | | 0 to 60 | |
| T _{stg} | Storage temperature | | | -65 to 150 | °C |

option: All options are on request basis.

Table 22.2 Recommended Operating Conditions (1) ⁽¹⁾

| Symbol | Parameter | | Standard | | | Unit |
|-----------------------|---|--|---------------------|-----------------|----------------------|------|
| | | | Min. | Typ. | Max. | |
| V _{CC} | Supply voltage (V _{CC1} = V _{CC2}) | | 4.2 | 5.0 | 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} | HIGH input voltage | P3_1 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7, XIN, RESET, CNVSS, BYTE | 0.8 V _{CC} | | V _{CC} | V |
| | | P7_1, P9_1 | 0.8 V _{CC} | | 6.5 | V |
| | | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 (During single-chip mode) | 0.8 V _{CC} | | V _{CC} | V |
| | | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 (Data input during memory expansion and microprocessor modes) | 0.5 V _{CC} | | V _{CC} | V |
| V _{IL} | LOW input voltage | P3_1 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, XIN, RESET, CNVSS, BYTE | 0 | | 0.2 V _{CC} | V |
| | | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 (During single-chip mode) | 0 | | 0.2 V _{CC} | V |
| | | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 (Data input during memory expansion and microprocessor modes) | 0 | | 0.16 V _{CC} | V |
| I _{OH(peak)} | HIGH peak output current | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7 | | | -10.0 | mA |
| I _{OH(avg)} | HIGH average output current | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7 | | | -5.0 | mA |
| I _{OL(peak)} | LOW peak output current | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7 | | | 10.0 | mA |
| I _{OL(avg)} | LOW average output current | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7 | | | 5.0 | mA |

NOTES:

1. Referenced to V_{CC} = 4.2 to 5.5 V at T_{opr} = -40 to 85°C unless otherwise specified.
2. Average output current values during 100 ms period.
3. The total I_{OL(peak)} for ports P0, P1, P2, P8_6, P8_7, P9, and P10 must be 80 mA max.
The total I_{OL(peak)} for ports P3, P4, P5, P6, P7, and P8_0 to P8_4 must be 80 mA max.
The total I_{OH(peak)} for ports P0, P1, and P2 must be -40 mA max.
The total I_{OH(peak)} for ports P3, P4, and P5 must be -40 mA max.
The total I_{OH(peak)} for ports P6, P7, and P8_0 to P8_4 must be -40 mA max.
The total I_{OH(peak)} for ports P8_6, P8_7, P9, and P10 must be -40 mA max.

Table 22.3 Recommended Operating Conditions (2) ⁽¹⁾

| Symbol | Parameter | | | Standard | | | Unit | |
|--------------------------|---|---------|--|--------------------|------|--------|------|------|
| | | | | Min. | Typ. | Max. | | |
| f(XIN) | Main clock input oscillation frequency ^{(2) (3) (4)} | No wait | Mask ROM version Flash memory version | VCC = 4.2 to 5.5 V | 0 | | 16 | MHz |
| f(XCIN) | Sub clock oscillation frequency | | | | | 32.768 | 50 | kHz |
| f(Ring) | On-chip oscillation frequency | | | | | 1 | | MHz |
| f(PLL) | PLL clock oscillation frequency | | | | 16 | | 20 | MHz |
| f(BCLK) | CPU operation clock | | | VCC = 4.2 to 5.5 V | 0 | | 20 | MHz |
| t _{su(PLL)} | PLL frequency synthesizer stabilization wait time | | | | | | 20 | ms |
| f _(ripple) | Power supply ripple allowable frequency (VCC) | | | | | | 10 | kHz |
| V _{P-P(ripple)} | Power supply ripple allowable amplitude voltage | | | VCC = 5 V | | | 0.5 | V |
| V _{CC(ΔV/ΔT)} | Power supply ripple rising/falling gradient | | | VCC = 5 V | | | 0.3 | V/ms |

NOTES:

1. Referenced to VCC = 4.2 to 5.5 V at Topr = -40 to 85°C unless otherwise specified.
2. Relationship between main clock oscillation frequency and supply voltage is shown right.
3. Execute program/erase of flash memory by VCC = 5.0 ± 0.5 V.
4. When using over 16 MHz, use PLL clock. PLL clock oscillation frequency which can be used is 16 MHz or 20 MHz.

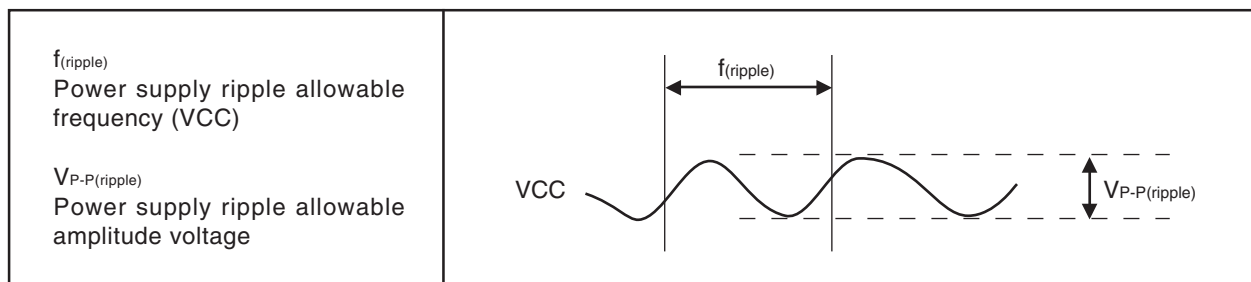
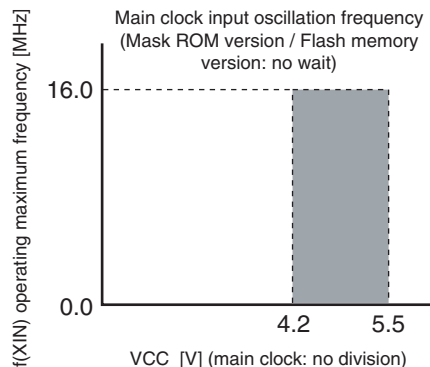


Figure 22.1 Voltage Fluctuation Timing

Table 22.4 Electrical Characteristics (1) ⁽¹⁾

| Symbol | Parameter | | Measuring Condition | Standard | | | Unit | |
|----------------------------------|-----------------------|--|---------------------------|---------------------------|------|-----------------|------|---|
| | | | | Min. | Typ. | Max. | | |
| V _{OH} | HIGH output voltage | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7 | I _{OH} = -5 mA | V _{CC} -2.0 | | V _{CC} | V | |
| V _{OH} | HIGH output voltage | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7 | I _{OH} = -200 μA | V _{CC} -0.3 | | V _{CC} | V | |
| V _{OH} | HIGH output voltage | XOUT | HIGHPOWER | I _{OH} = -1 mA | 3.0 | V _{CC} | V | |
| | | | LOWPOWER | I _{OH} = -0.5 mA | 3.0 | V _{CC} | | |
| | HIGH output voltage | XCOU _T | HIGHPOWER | With no load applied | | 2.5 | | V |
| | | | LOWPOWER | With no load applied | | 1.6 | | |
| V _{OL} | LOW output voltage | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7 | I _{OL} = 5 mA | | | 2.0 | V | |
| V _{OL} | LOW output voltage | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7 | I _{OL} = 200 μA | | | 0.45 | V | |
| V _{OL} | LOW output voltage | XOUT | HIGHPOWER | I _{OL} = 1 mA | | 2.0 | V | |
| | | | LOWPOWER | I _{OL} = 0.5 mA | | 2.0 | | |
| | LOW output voltage | XCOU _T | HIGHPOWER | With no load applied | | 0 | | V |
| | | | LOWPOWER | With no load applied | | 0 | | |
| V _{T+} -V _{T-} | Hysteresis | HOLD, RDY, TA0IN to TA4IN, TB0IN to TB5IN, INT0 to INT5, NMI, ADTRG, CTS0 to CTS2, SCL0 to SCL2, SDA0 to SDA2, CLK0 to CLK3, TA0OUT to TA4OUT, K _{I0} to K _{I3} , RXD0 to RXD2, SIN3 | | 0.2 | | 1.0 | V | |
| V _{T+} -V _{T-} | Hysteresis | RESET | | 0.2 | | 2.5 | V | |
| I _{IH} | HIGH input current | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, XIN, RESET, CNVSS, BYTE | V _I = 5 V | | | 5.0 | μA | |
| I _{IL} | LOW input current | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, XIN, RESET, CNVSS, BYTE | 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, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7 | V _I = 0 V | 30 | 50 | 170 | kΩ | |
| R _{IXIN} | Feedback resistance | XIN | | | 1.5 | | MΩ | |
| R _{IXCIN} | Feedback resistance | XCIN | | | 15 | | MΩ | |
| V _{RAM} | RAM retention voltage | | At stop mode | 2.0 | | | V | |

NOTES:

1. Referenced to V_{CC} = 4.2 to 5.5 V, V_{SS} = 0 V at T_{opr} = -40 to 85°C, f(BCLK) = 20 MHz unless otherwise specified.

Table 22.5 Electrical Characteristics (2) ⁽¹⁾

| Symbol | Parameter | | Measuring Condition | | Standard | | | Unit |
|-----------------|--|---|----------------------|---|--------------------------------|------|------|------|
| | | | | | Min. | Typ. | Max. | |
| I _{cc} | Power supply current (VCC = 4.2 to 5.5 V) | In single-chip mode, the output pins are open and other pins are VSS. | Mask ROM | f(BCLK) = 20 MHz, PLL operation, No division | | 18 | 32 | mA |
| | | | | On-chip oscillation, No division | | 1 | | mA |
| | | | Flash memory | f(BCLK) = 20 MHz, PLL operation, No division | | 20 | 34 | mA |
| | | | | On-chip oscillation, No division | | 1.8 | | mA |
| | | | Flash memory program | f(BCLK) = 10 MHz, VCC = 5 V | | 15 | | mA |
| | | | Flash memory erase | f(BCLK) = 10 MHz, VCC = 5 V | | 25 | | mA |
| | | | Mask ROM | f(BCLK) = 32kHz, Low power dissipation mode, ROM ⁽²⁾ | | 25 | | μA |
| | | | Flash memory | f(BCLK) = 32 kHz, Low power dissipation mode, RAM ⁽²⁾ | | 25 | | μA |
| | | | | f(BCLK) = 32 kHz, Low power dissipation mode, Flash memory ⁽²⁾ | | 420 | | μA |
| | | | | Mask ROM Flash memory | On-chip oscillation, Wait mode | | 50 | |
| | | | | f(BCLK) = 32 kHz, Wait mode ⁽³⁾ , Oscillation capacity High | | 8.5 | | μA |
| | | | | f(BCLK) = 32 kHz, Wait mode ⁽³⁾ , Oscillation capacity Low | | 3.0 | | μA |
| | | | | Stop mode, Topr = 25°C | | 0.8 | 3.0 | μA |

NOTES:

1. Referenced to VCC = 4.2 to 5.5 V, VSS = 0 V at Topr = -40 to 85°C, f(BCLK) = 20 MHz unless otherwise specified.
2. This indicates the memory in which the program to be executed exists.
3. With one timer operated using fC32.

Table 22.6 A/D Conversion Characteristics ⁽¹⁾

| Symbol | Parameter | | Measuring Condition | | Standard | | | Unit |
|---------------------|---|---------|--------------------------------|--|--|------|------------------|------|
| | | | | | Min. | Typ. | Max. | |
| – | Resolution | | VREF = VCC | | | | 10 | Bit |
| INL | Integral nonlinearity error | 10 bits | VREF = VCC = 5 V | ANEX0, ANEX1 input, AN0 to AN7 input, AN0_0 to AN0_7 input, AN2_0 to AN2_7 input | | | ±3 | LSB |
| | | 8 bits | | VREF = AVCC = VCC = 5 V | External operation amp connection mode | | | ±7 |
| – | Absolute accuracy | 10 bits | VREF = VCC = 5 V | ANEX0, ANEX1 input, AN0 to AN7 input, AN0_0 to AN0_7 input, AN2_0 to AN2_7 input | | | ±3 | LSB |
| | | 8 bits | | VREF = AVCC = VCC = 5 V | External operation amp connection mode | | | ±7 |
| DNL | Differential nonlinearity error | | | | | | ±1 | LSB |
| – | Offset error | | | | | | ±3 | LSB |
| – | Gain error | | | | | | ±3 | LSB |
| R _{LADDER} | Resistor ladder | | VREF = VCC | | 10 | | 40 | kΩ |
| t _{CONV} | 10-bit conversion time, sample & hold available | | VREF = VCC = 5 V, φAD = 10 MHz | | 3.3 | | | μs |
| | 8-bit conversion time, sample & hold available | | VREF = VCC = 5 V, φAD = 10 MHz | | 2.8 | | | μs |
| t _{SAMP} | Sampling time | | | | 0.3 | | | μs |
| V _{REF} | Reference voltage | | | | 2.0 | | V _{CC} | V |
| V _{IA} | Analog input voltage | | | | 0 | | V _{REF} | V |

NOTES:

1. Referenced to VCC = AVCC = VREF = 4.2 to 5.5 V, VSS = AVSS = 0 V, –40 to 85°C unless otherwise specified.
2. φAD frequency must be 10 MHz or less.
3. When sample & hold is disabled, φAD frequency must be 250kHz or more in addition to a limit of NOTE 2.
When sample & hold is enabled, φAD frequency must be 1MHz or more in addition to a limit of NOTE 2.

Table 22.7 D/A conversion Characteristics ⁽¹⁾

| Symbol | Parameter | | Measuring condition | | Standard | | | Unit |
|-------------------|--------------------------------------|--|---------------------|--|----------|------|------|------|
| | | | | | Min. | Typ. | Max. | |
| – | Resolution | | | | | | 8 | Bits |
| – | Absolute accuracy | | | | | | 1.0 | % |
| t _{su} | Setup time | | | | | | 3 | μs |
| R _o | Output resistance | | | | 4 | 10 | 20 | kΩ |
| I _{VREF} | Reference power supply input current | | (NOTE 2) | | | | 1.5 | mA |

NOTES:

1. Referenced to VCC = AVCC = VREF = 4.2 to 5.5 V, VSS = AVSS = 0 V, –40 to 85°C unless otherwise specified.
2. This applies when using one D/A converter, with the DAi register (i = 0, 1) for the unused D/A converter set to 00h.
The resistor ladder of the A/D converter is not included. Also, the I_{VREF} will flow even if VREF is disconnected by the ADCON1 register.

Table 22.8 Power Supply Circuit Timing Characteristics

| Symbol | Parameter | Measuring Condition | Standard | | | Unit |
|--------------|---|---------------------|----------|------|------|---------|
| | | | Min. | Typ. | Max. | |
| $t_{d(P-R)}$ | Time for internal power supply stabilization during powering-on | VCC = 4.2 to 5.5 V | | | 2 | ms |
| $t_{d(R-S)}$ | STOP release time | | | | 150 | μ s |
| $t_{d(W-S)}$ | Low power dissipation mode wait mode release time | | | | 150 | μ s |

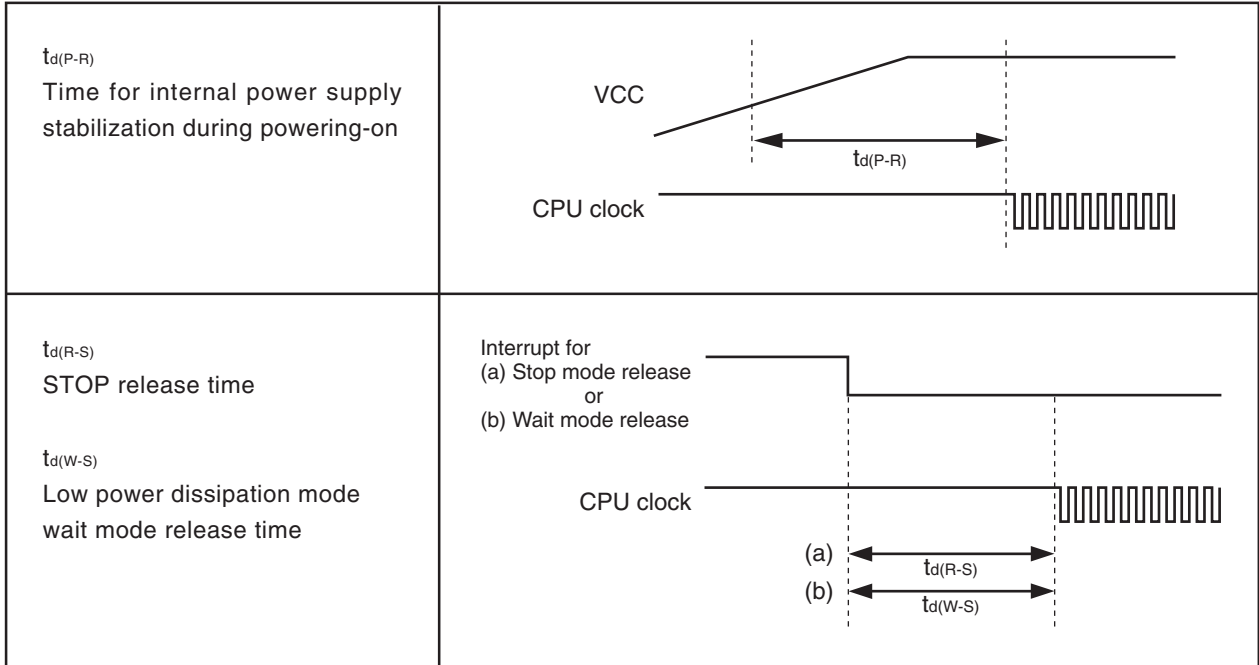


Figure 22.2 Power Supply Circuit Timing Diagram

Timing Requirements**VCC = 5 V****(Referenced to VCC = 5 V, VSS = 0 V, at Topr = –40 to 85°C unless otherwise specified)****Table 22.9 External Clock Input (XIN Input)**

| Symbol | Parameter | Standard | | Unit |
|-------------------|---------------------------------------|----------|------|------|
| | | Min. | Max. | |
| t _c | External clock input cycle time | 62.5 | | ns |
| t _{w(H)} | External clock input HIGH pulse width | 25 | | ns |
| t _{w(L)} | External clock input LOW pulse width | 25 | | ns |
| t _r | External clock rise time | | 15 | ns |
| t _f | External clock fall time | | 15 | ns |

Table 22.10 Memory Expansion Mode and Microprocessor Mode

| Symbol | Parameter | Standard | | Unit |
|----------------------------|--|----------|----------|------|
| | | Min. | Max. | |
| t _{ac1(RD-DB)} | Data input access time (for setting with no wait) | | (NOTE 1) | ns |
| t _{ac2(RD-DB)} | Data input access time (for setting with wait) | | (NOTE 2) | ns |
| t _{ac3(RD-DB)} | Data input access time (when accessing multiplexed bus area) | | (NOTE 3) | ns |
| t _{su(DB-RD)} | Data input setup time | 40 | | ns |
| t _{su(RDY-BCLK)} | RDY input setup time | 30 | | ns |
| t _{su(HOLD-BCLK)} | HOLD input setup time | 40 | | ns |
| t _{h(RD-DB)} | Data input hold time | 0 | | ns |
| t _{h(BCLK-RDY)} | RDY input hold time | 0 | | ns |
| t _{h(BCLK-HOLD)} | HOLD input hold time | 0 | | ns |

NOTES:

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 45 \text{ [ns]}$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{(n - 0.5) \times 10^9}{f(\text{BCLK})} - 45 \text{ [ns]} \quad n \text{ is "2" for 1-wait setting, "3" for 2-wait setting and "4" for 3-wait setting.}$$

3. Calculated according to the BCLK frequency as follows:

$$\frac{(n - 0.5) \times 10^9}{f(\text{BCLK})} - 45 \text{ [ns]} \quad n \text{ is "2" for 2-wait setting, "3" for 3-wait setting.}$$

Timing Requirements**VCC = 5 V****(Referenced to VCC = 5 V, VSS = 0 V, at Topr = -40 to 85°C unless otherwise specified)****Table 22.11 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 22.12 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 22.13 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 22.14 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 22.15 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 22.16 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 |

Timing Requirements**VCC = 5 V****(Referenced to VCC = 5 V, VSS = 0 V, at Topr = -40 to 85°C unless otherwise specified)****Table 22.17 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 22.18 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 22.19 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 22.20 A/D Trigger Input

| Symbol | Parameter | Standard | | Unit |
|--------------|--|----------|------|------|
| | | Min. | Max. | |
| $t_{c(AD)}$ | ADTRG $\bar{}$ input cycle time (trigger able minimum) | 1000 | | ns |
| $t_{w(ADL)}$ | ADTRG $\bar{}$ input LOW pulse width | 125 | | ns |

Table 22.21 Serial Interface

| 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 22.22 External Interrupt INTi Input

| Symbol | Parameter | Standard | | Unit |
|--------------|-----------------------------|----------|------|------|
| | | Min. | Max. | |
| $t_{w(INH)}$ | INTi input HIGH pulse width | 250 | | ns |
| $t_{w(INL)}$ | INTi input LOW pulse width | 250 | | ns |

Switching Characteristics**VCC = 5 V****(Referenced to VCC = 5 V, VSS = 0 V, at Topr = -40 to 85 °C unless otherwise specified)****Table 22.23 Memory Expansion Mode and Microprocessor Mode (for setting with no wait)**

| Symbol | Parameter | Measuring Condition | Standard | | Unit |
|--------------------|--|---------------------|----------|------|------|
| | | | Min. | Max. | |
| $t_{d(BCLK-AD)}$ | Address output delay time | Figure 22.3 | | 25 | ns |
| $t_{h(BCLK-AD)}$ | Address output hold time (in relation to BCLK) | | 4 | | ns |
| $t_{h(RD-AD)}$ | Address output hold time (in relation to RD) | | 0 | | ns |
| $t_{h(WR-AD)}$ | Address output hold time (in relation to WR) | | (NOTE 1) | | ns |
| $t_{d(BCLK-CS)}$ | Chip select output delay time | | | 25 | ns |
| $t_{h(BCLK-CS)}$ | Chip select output hold time (in relation to BCLK) | | 4 | | ns |
| $t_{d(BCLK-ALE)}$ | ALE signal output delay time | | | 15 | ns |
| $t_{h(BCLK-ALE)}$ | ALE signal output hold time | | -4 | | ns |
| $t_{d(BCLK-RD)}$ | RD signal output delay time | | | 25 | ns |
| $t_{h(BCLK-RD)}$ | RD signal output hold time | | 0 | | ns |
| $t_{d(BCLK-WR)}$ | WR signal output delay time | | | 25 | ns |
| $t_{h(BCLK-WR)}$ | WR signal output hold time | | 0 | | ns |
| $t_{d(BCLK-DB)}$ | Data output delay time (in relation to BCLK) | | | 40 | ns |
| $t_{h(BCLK-DB)}$ | Data output hold time (in relation to BCLK) ⁽³⁾ | | 4 | | ns |
| $t_{d(DB-WR)}$ | Data output delay time (in relation to WR) | | (NOTE 2) | | ns |
| $t_{h(WR-DB)}$ | Data output hold time (in relation to WR) ⁽³⁾ | | (NOTE 1) | | ns |
| $t_{d(BCLK-HLDA)}$ | HLDA output delay time | | 40 | ns | |

NOTES:

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 10 \text{ [ns]}$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 40 \text{ [ns]} \quad f(\text{BCLK}) \text{ is 12.5 MHz or less.}$$

3. This standard value shows the timing when the output is off, and does not show hold time of data bus.

Hold time of data bus varies with capacitor volume and pull-up (pull-down) resistance value.

Hold time of data bus is expressed in

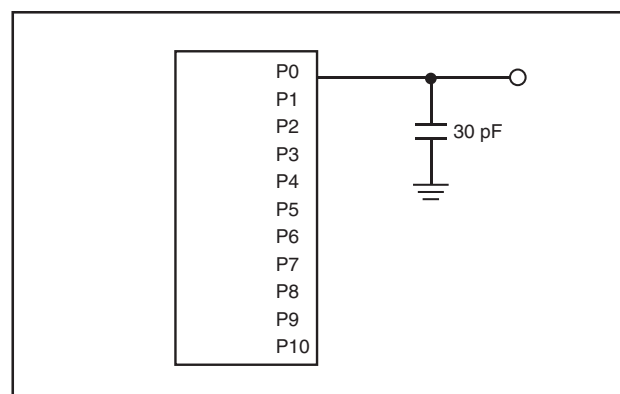
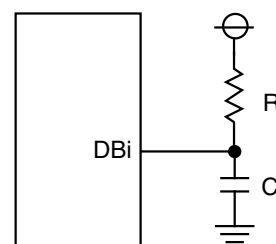
$$t = -CR \times \ln(1 - V_{OL} / V_{CC})$$

by a circuit of the right figure.

For example, when $V_{OL} = 0.2 V_{CC}$, $C = 30 \text{ pF}$,

$R = 1 \text{ k}\Omega$, hold time of output "L" level is

$$t = -30 \text{ pF} \times 1 \text{ k}\Omega \times \ln(1 - 0.2 V_{CC} / V_{CC}) = 6.7 \text{ ns.}$$

**Figure 22.3 Port P0 to P10 Measurement Circuit**

Switching Characteristics**VCC = 5 V****(Referenced to VCC = 5 V, VSS = 0 V, at Topr = -40 to 85 °C unless otherwise specified)****Table 22.24 Memory Expansion Mode and Microprocessor Mode (for 1- to 3-wait setting and external area access)**

| Symbol | Parameter | Measuring Condition | Standard | | Unit |
|----------------------------|--|---------------------|----------|------|------|
| | | | Min. | Max. | |
| t _d (BCLK-AD) | Address output delay time | Figure 22.3 | | 25 | ns |
| t _h (BCLK-AD) | Address output hold time (in relation to BCLK) | | 4 | | ns |
| t _h (RD-AD) | Address output hold time (in relation to RD) | | 0 | | ns |
| t _h (WR-AD) | Address output hold time (in relation to WR) | | (NOTE 1) | | ns |
| t _d (BCLK-CS) | Chip select output delay time | | | 25 | ns |
| t _h (BCLK-CS) | Chip select output hold time (in relation to BCLK) | | 4 | | ns |
| t _d (BCLK-ALE) | ALE signal output delay time | | | 15 | ns |
| t _h (BCLK-ALE) | ALE signal output hold time | | -4 | | ns |
| t _d (BCLK-RD) | RD signal output delay time | | | 25 | ns |
| t _h (BCLK-RD) | RD signal output hold time | | 0 | | ns |
| t _d (BCLK-WR) | WR signal output delay time | | | 25 | ns |
| t _h (BCLK-WR) | WR signal output hold time | | 0 | | ns |
| t _d (BCLK-DB) | Data output delay time (in relation to BCLK) | | | 40 | ns |
| t _h (BCLK-DB) | Data output hold time (in relation to BCLK) ⁽³⁾ | | 4 | | ns |
| t _d (DB-WR) | Data output delay time (in relation to WR) | | (NOTE 2) | | ns |
| t _h (WR-DB) | Data output hold time (in relation to WR) ⁽³⁾ | | (NOTE 1) | | ns |
| t _d (BCLK-HLDA) | HLDA output delay time | | 40 | ns | |

NOTES:

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 10 \text{ [ns]}$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{(n - 0.5) \times 10^9}{f(\text{BCLK})} - 40 \text{ [ns]}$$

n is "1" for 1-wait setting, "2" for 2-wait setting and "3" for 3-wait setting.
When n = 1, f(BCLK) is 12.5 MHz or less.

3. This standard value shows the timing when the output is off, and does not show hold time of data bus.

Hold time of data bus varies with capacitor volume and pull-up (pull-down) resistance value.

Hold time of data bus is expressed in

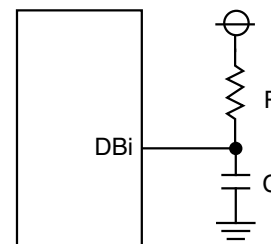
$$t = -CR \times \ln(1 - V_{OL} / V_{CC})$$

by a circuit of the right figure.

For example, when $V_{OL} = 0.2 V_{CC}$, $C = 30 \text{ pF}$,

$R = 1 \text{ k}\Omega$, hold time of output "L" level is

$$t = -30 \text{ pF} \times 1 \text{ k}\Omega \times \ln(1 - 0.2 V_{CC} / V_{CC}) = 6.7 \text{ ns.}$$



Switching Characteristics**VCC = 5 V****(Referenced to VCC = 5 V, VSS = 0 V, at Topr = –40 to 85 °C unless otherwise specified)****Table 22.25 Memory Expansion Mode and Microprocessor Mode
(for 2- to 3-wait setting, external area access and multiplexed bus selection)**

| Symbol | Parameter | Measuring Condition | Standard | | Unit |
|----------------------------|---|---------------------|----------|------|------|
| | | | Min. | Max. | |
| t _d (BCLK-AD) | Address output delay time | Figure 22.3 | | 25 | ns |
| t _h (BCLK-AD) | Address output hold time (in relation to BCLK) | | 4 | | ns |
| t _h (RD-AD) | Address output hold time (in relation to RD) | | (NOTE 1) | | ns |
| t _h (WR-AD) | Address output hold time (in relation to WR) | | (NOTE 1) | | ns |
| t _d (BCLK-CS) | Chip select output delay time | | | 25 | ns |
| t _h (BCLK-CS) | Chip select output hold time (in relation to BCLK) | | 4 | | ns |
| t _h (RD-CS) | Chip select output hold time (in relation to RD) | | (NOTE 1) | | ns |
| t _h (WR-CS) | Chip select output hold time (in relation to WR) | | (NOTE 1) | | ns |
| t _d (BCLK-RD) | RD signal output delay time | | | 25 | ns |
| t _h (BCLK-RD) | RD signal output hold time | | 0 | | ns |
| t _d (BCLK-WR) | WR signal output delay time | | | 25 | ns |
| t _h (BCLK-WR) | WR signal output hold time | | 0 | | ns |
| t _d (BCLK-DB) | Data output delay time (in relation to BCLK) | | | 40 | ns |
| t _h (BCLK-DB) | Data output hold time (in relation to BCLK) | | 4 | | ns |
| t _d (DB-WR) | Data output delay time (in relation to WR) | | (NOTE 2) | | ns |
| t _h (WR-DB) | Data output hold time (in relation to WR) | | (NOTE 1) | | ns |
| t _d (BCLK-HLDA) | HLDA output delay time | | | 40 | ns |
| t _d (BCLK-ALE) | ALE signal output delay time (in relation to BCLK) | | | 15 | ns |
| t _h (BCLK-ALE) | ALE signal output hold time (in relation to BCLK) | | –4 | | ns |
| t _d (AD-ALE) | ALE signal output delay time (in relation to Address) | | (NOTE 3) | | ns |
| t _h (ALE-AD) | ALE signal output hold time (in relation to Address) | (NOTE 4) | | ns | |
| t _d (AD-RD) | RD signal output delay from the end of Address | 0 | | ns | |
| t _d (AD-WR) | WR signal output delay from the end of Address | 0 | | ns | |
| t _{dZ} (RD-AD) | Address output floating start time | | 8 | ns | |

NOTES:

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 10 \text{ [ns]}$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{(n - 0.5) \times 10^9}{f(\text{BCLK})} - 40 \text{ [ns]} \quad n \text{ is "2" for 2-wait setting, "3" for 3-wait setting.}$$

3. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 25 \text{ [ns]}$$

4. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 15 \text{ [ns]}$$

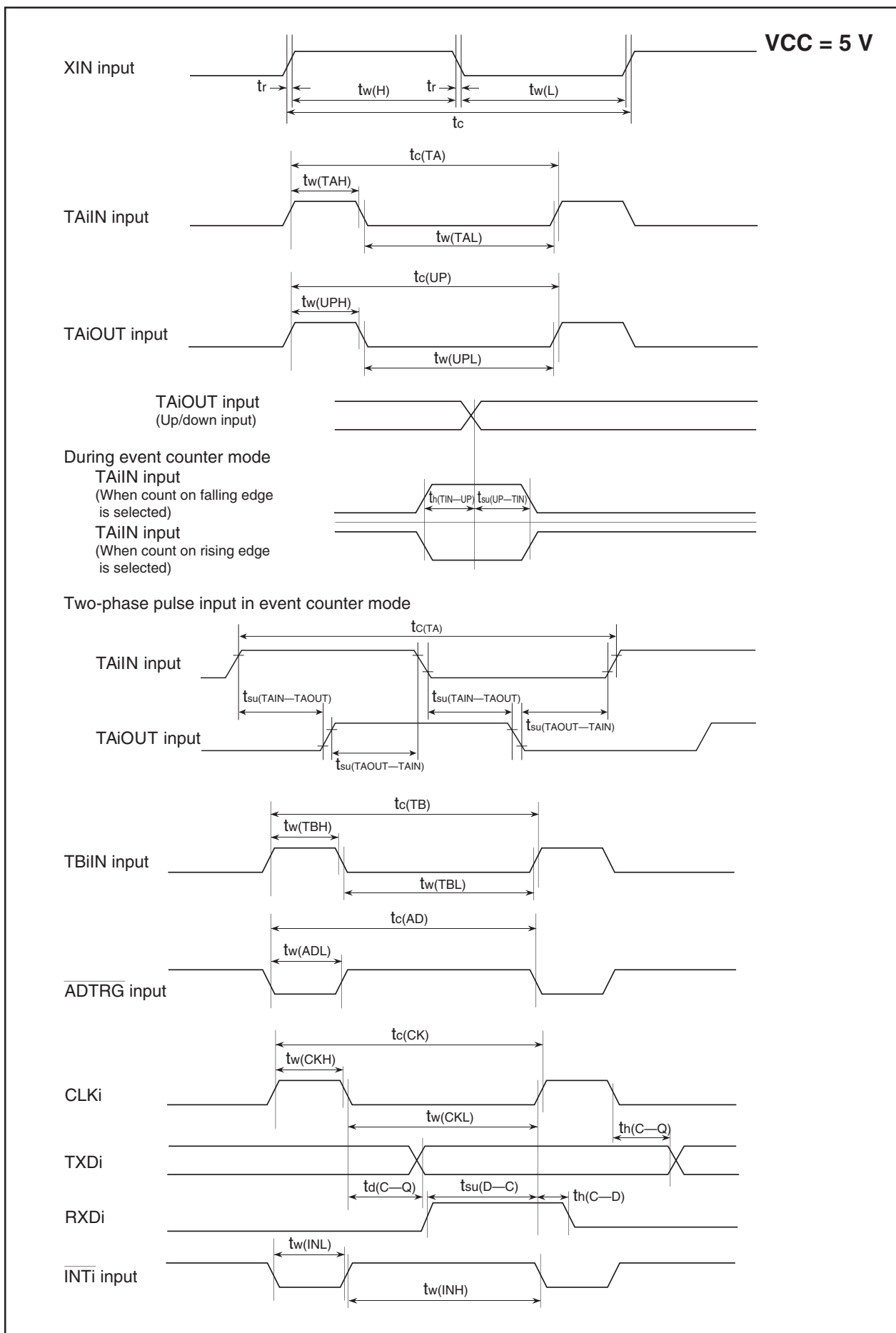


Figure 22.4 Timing Diagram (1)

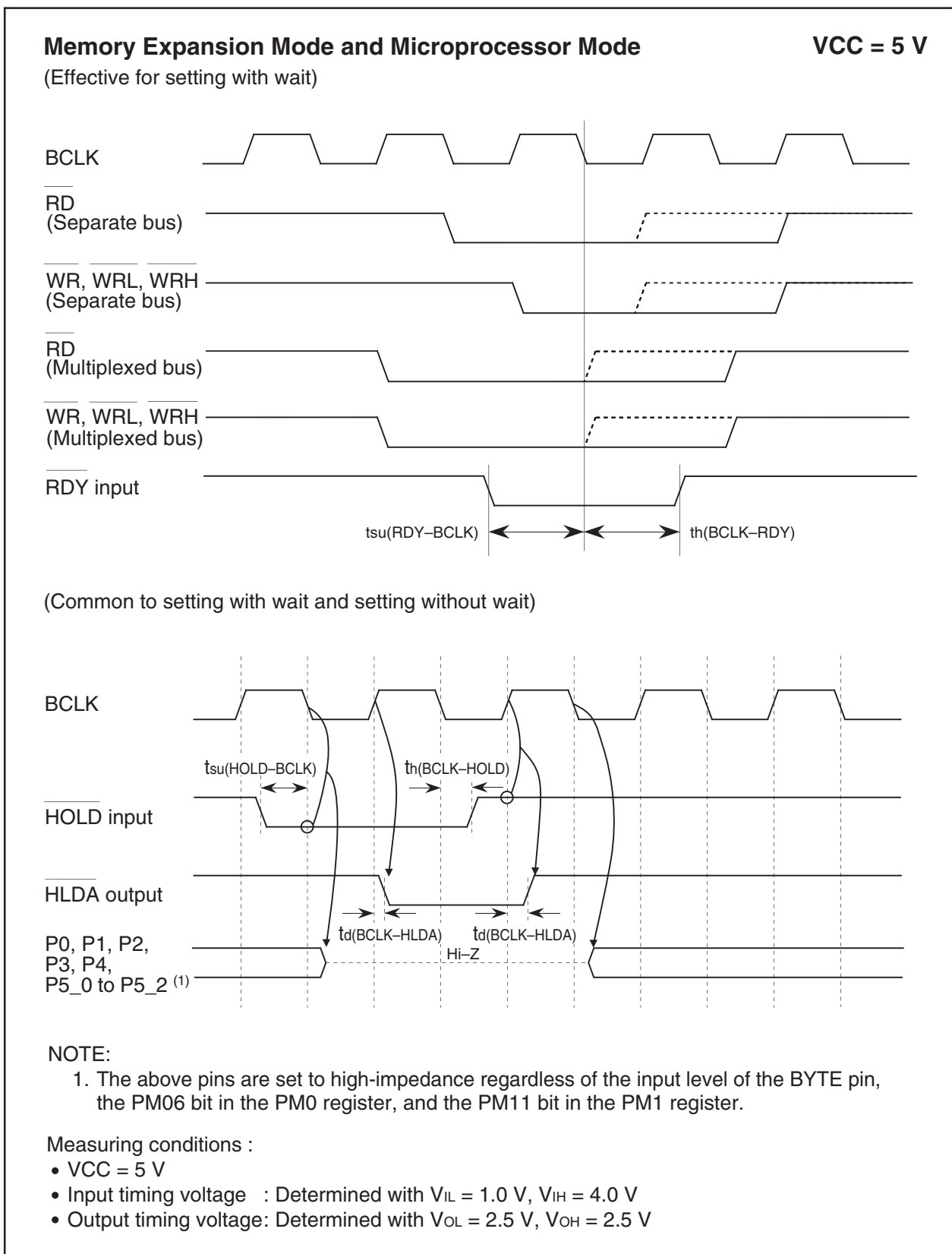


Figure 22.5 Timing Diagram (2)

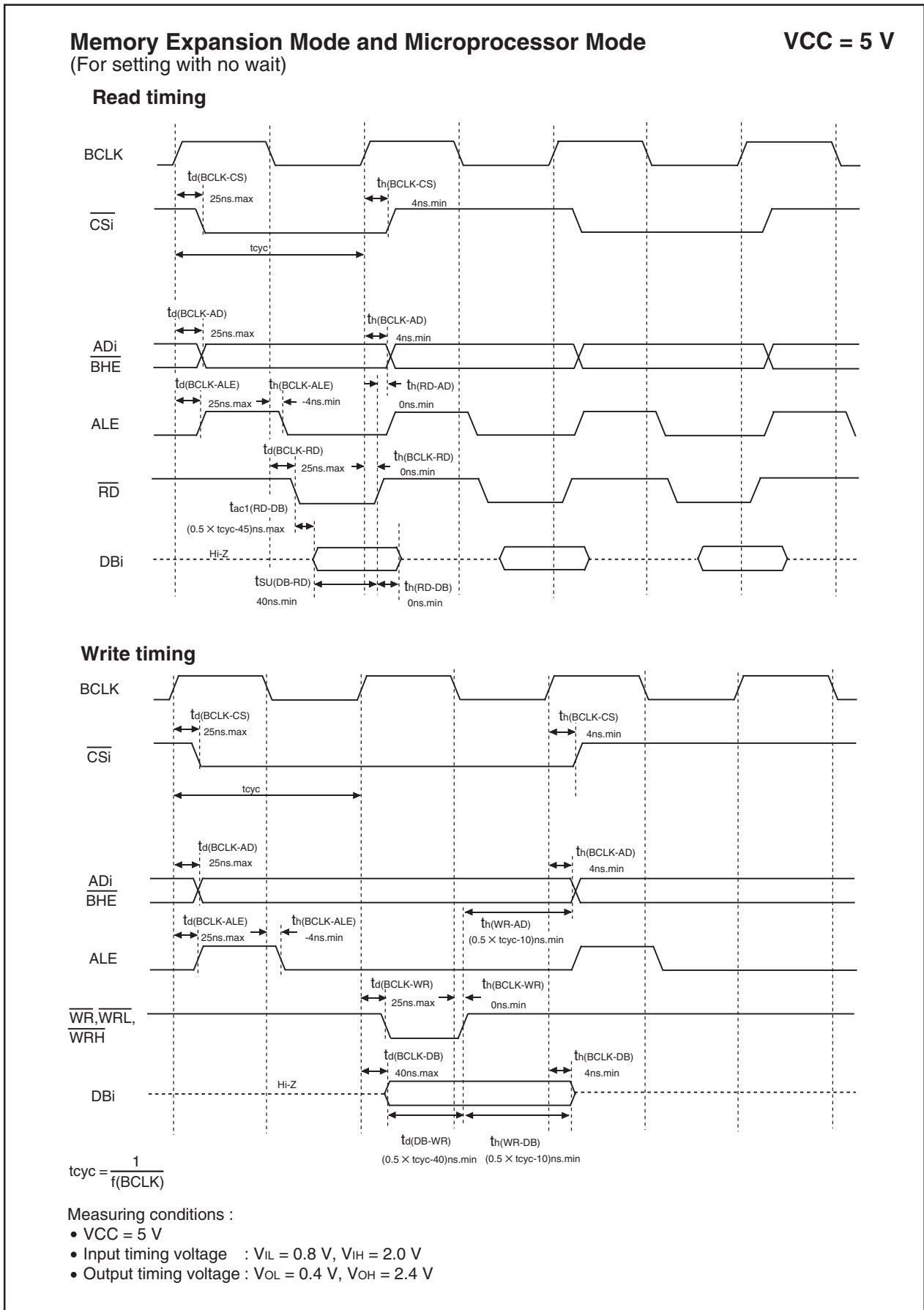


Figure 22.6 Timing Diagram (3)

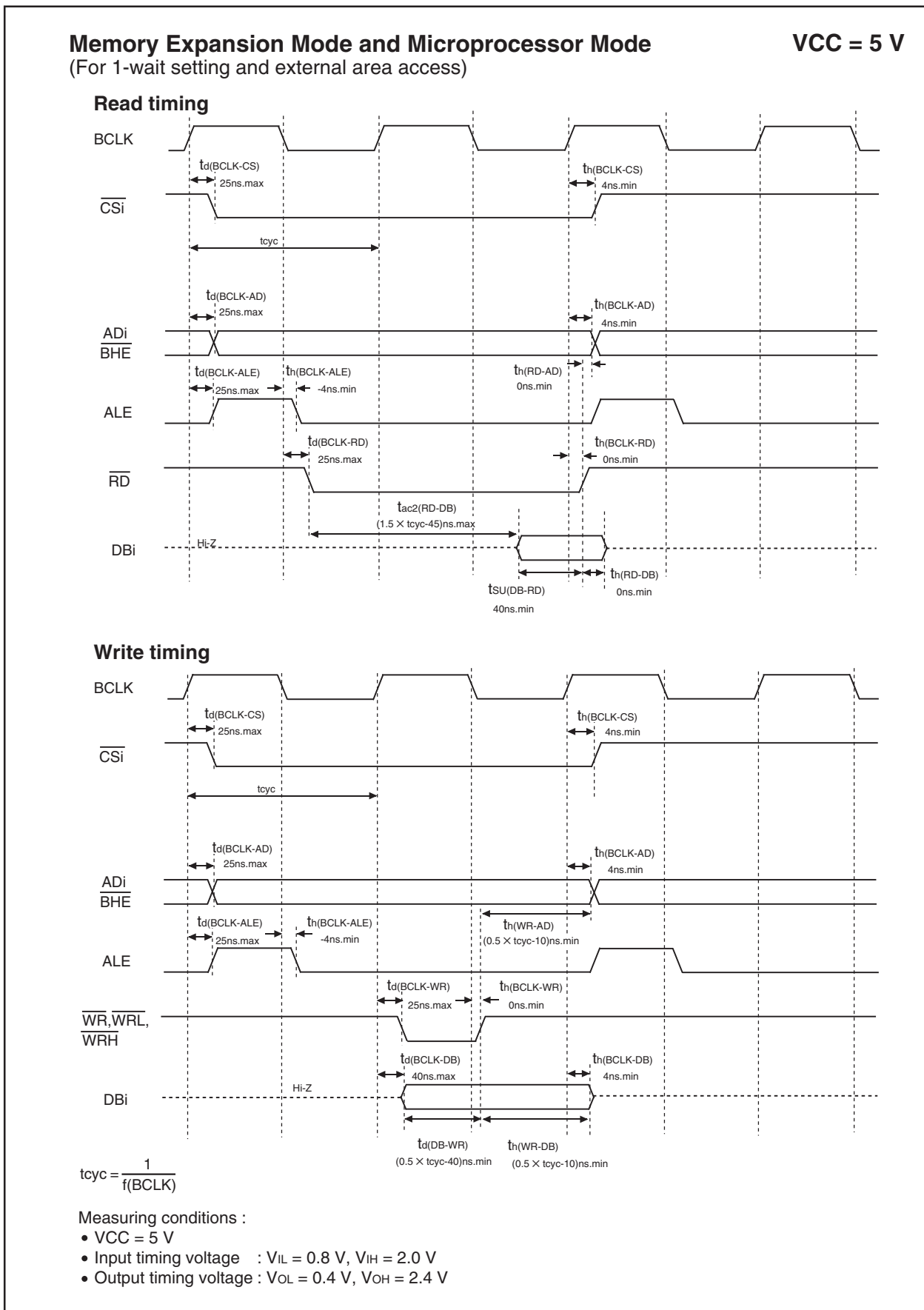


Figure 22.7 Timing Diagram (4)

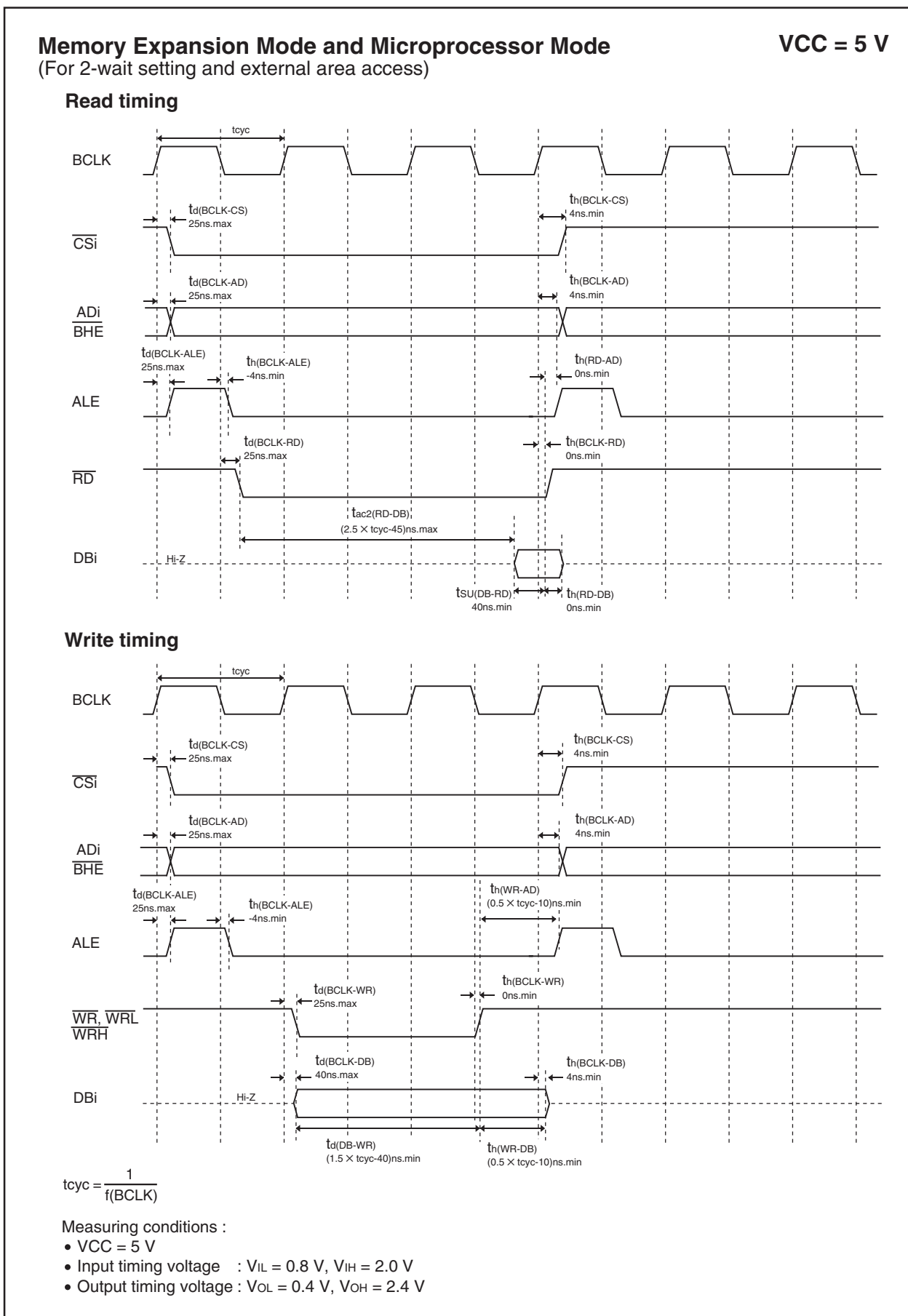


Figure 22.8 Timing Diagram (5)

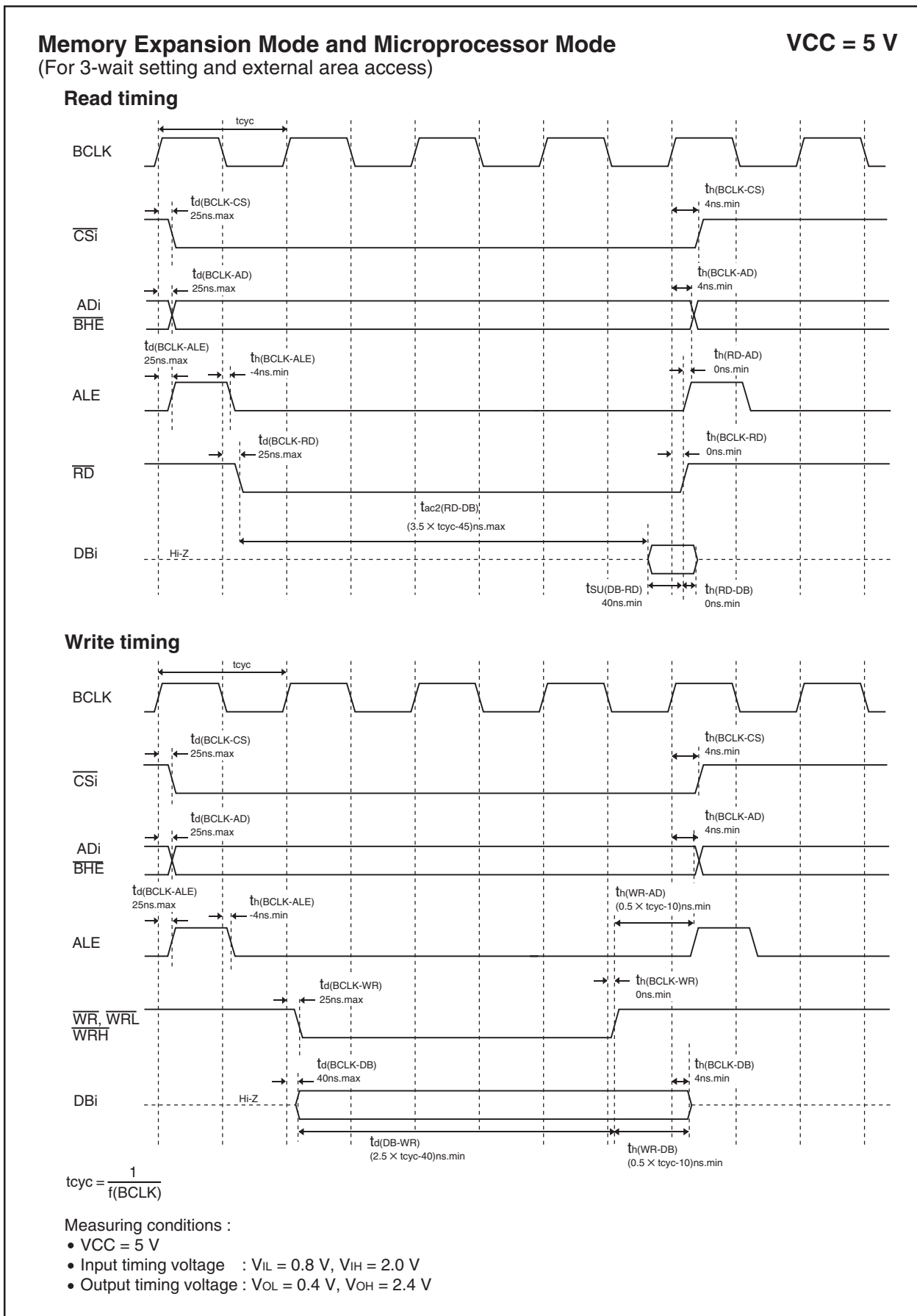


Figure 22.9 Timing Diagram (6)

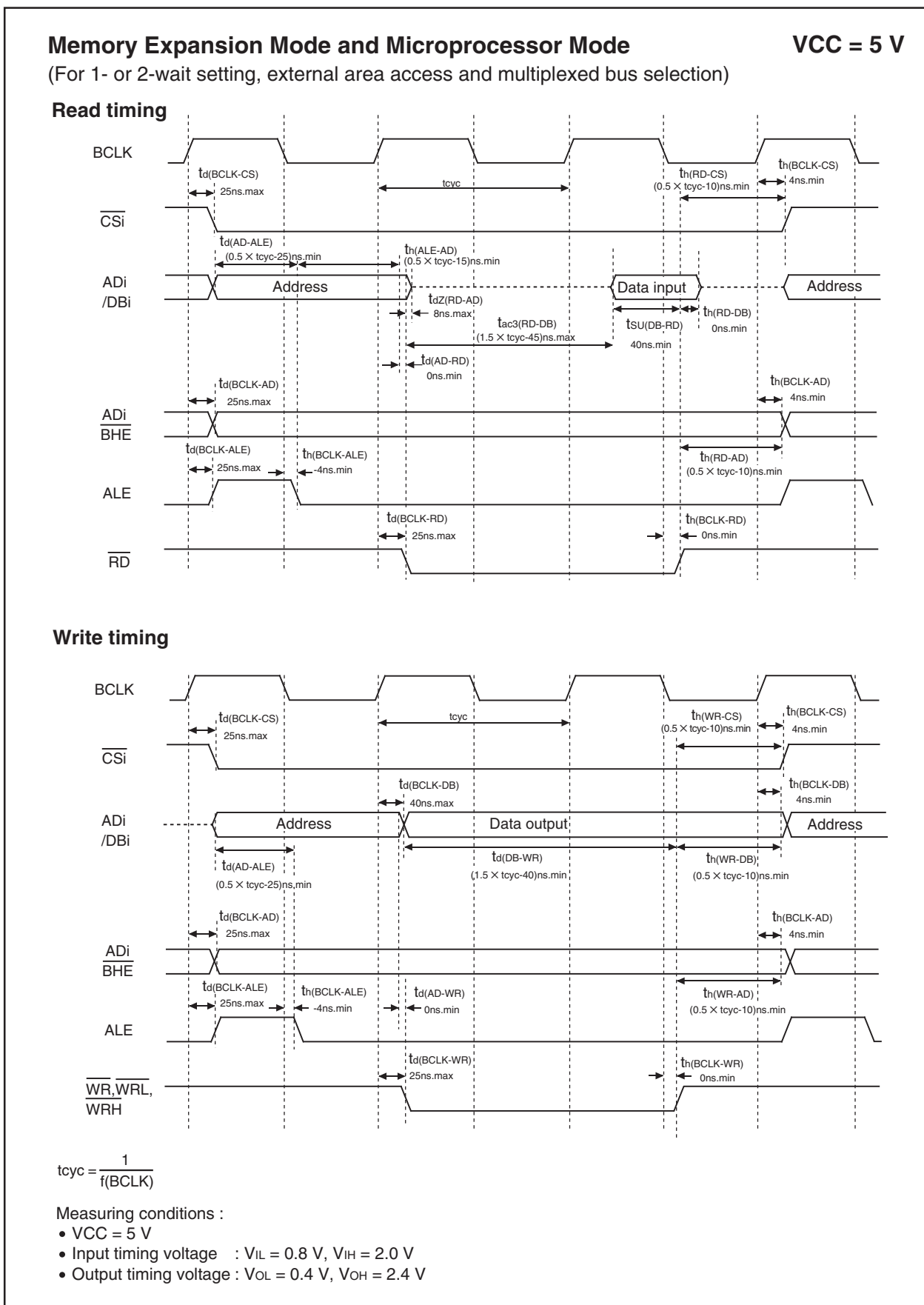


Figure 22.10 Timing Diagram (7)

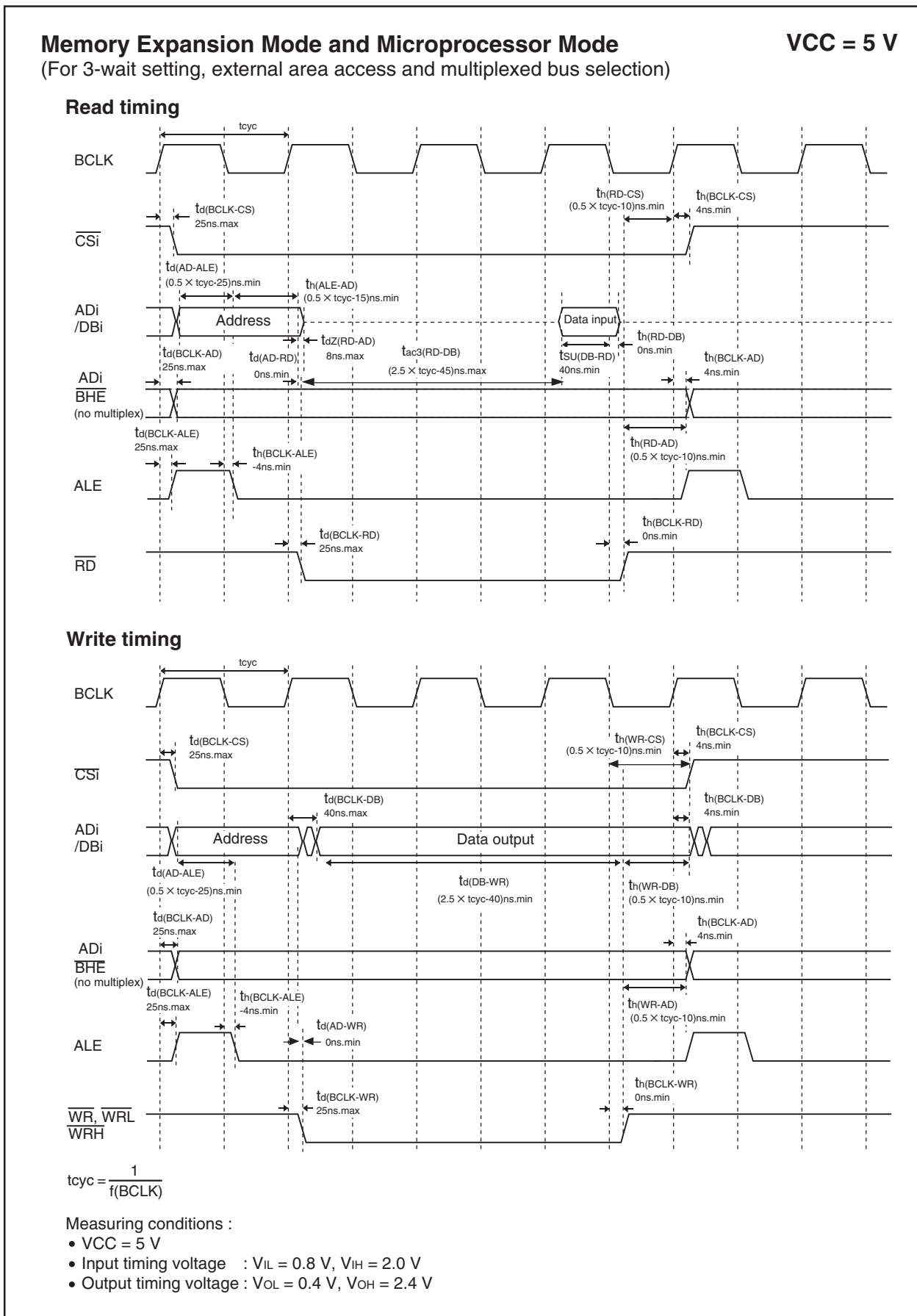


Figure 22.11 Timing Diagram (8)

22.2 Electrical Characteristics (Normal-ver.)

Table 22.26 Absolute Maximum Ratings

| Symbol | Parameter | | Condition | Rated Value | Unit |
|------------------|-------------------------------|--|-------------------------|-----------------|------|
| V _{CC} | Supply voltage (VCC1 = VCC2) | | VCC = AVCC | -0.3 to 6.5 | V |
| AV _{CC} | Analog supply voltage | | VCC = AVCC | -0.3 to 6.5 | V |
| V _i | Input voltage | RESET, CNVSS, BYTE, P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7, VREF, XIN | | -0.3 to VCC+0.3 | V |
| | | P7_1, P9_1 | | -0.3 to 6.5 | 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, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7, XOUT | | -0.3 to VCC+0.3 | V |
| | | P7_1, P9_1 | | -0.3 to 6.5 | V |
| P _d | Power dissipation | | T _{opr} = 25°C | 700 | mW |
| T _{opr} | Operating ambient temperature | During MCU operation | | -40 to 85 | °C |
| | | During flash memory program and erase operation | | 0 to 60 | |
| T _{stg} | Storage temperature | | | -65 to 150 | °C |

Table 22.27 Recommended Operating Conditions (1) ⁽¹⁾

| Symbol | Parameter | | Standard | | | Unit |
|-----------------------|---|--|---------------------|-----------------|----------------------|------|
| | | | Min. | Typ. | Max. | |
| V _{CC} | Supply voltage (V _{CC1} = V _{CC2}) | | 3.0 | 5.0 | 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} | HIGH input voltage | P3_1 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7, XIN, RESET, CNVSS, BYTE | 0.8 V _{CC} | | V _{CC} | V |
| | | P7_1, P9_1 | 0.8 V _{CC} | | 6.5 | V |
| | | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 (During single-chip mode) | 0.8 V _{CC} | | V _{CC} | V |
| | | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 (Data input during memory expansion and microprocessor modes) | 0.5 V _{CC} | | V _{CC} | V |
| V _{IL} | LOW input voltage | P3_1 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, XIN, RESET, CNVSS, BYTE | 0 | | 0.2 V _{CC} | V |
| | | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 (During single-chip mode) | 0 | | 0.2 V _{CC} | V |
| | | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 (Data input during memory expansion and microprocessor modes) | 0 | | 0.16 V _{CC} | V |
| I _{OH(peak)} | HIGH peak output current | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7 | | | -10.0 | mA |
| I _{OH(avg)} | HIGH average output current | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7 | | | -5.0 | mA |
| I _{OL(peak)} | LOW peak output current | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7 | | | 10.0 | mA |
| I _{OL(avg)} | LOW average output current | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7 | | | 5.0 | mA |

NOTES:

1. Referenced to V_{CC} = 3.0 to 5.5 V at T_{opr} = -40 to 85°C unless otherwise specified.
2. Average output current values during 100 ms period.
3. The total I_{OL(peak)} for ports P0, P1, P2, P8_6, P8_7, P9, and P10 must be 80 mA max.
The total I_{OL(peak)} for ports P3, P4, P5, P6, P7, and P8_0 to P8_4 must be 80 mA max.
The total I_{OH(peak)} for ports P0, P1, and P2 must be -40 mA max.
The total I_{OH(peak)} for ports P3, P4, and P5 must be -40 mA max.
The total I_{OH(peak)} for ports P6, P7, and P8_0 to P8_4 must be -40 mA max.
The total I_{OH(peak)} for ports P8_6, P8_7, P9, and P10 must be -40 mA max.

Table 22.28 Recommended Operating Conditions (2) ⁽¹⁾

| Symbol | Parameter | | | | Standard | | | Unit |
|--------------------------|---|---------|--|--------------------------|----------|--------|------------|------|
| | | | | | Min. | Typ. | Max. | |
| f(XIN) | Main clock input oscillation frequency ^{(2) (3) (4)} | No wait | Mask ROM version Flash memory version | VCC = 3.0 to 5.5 V | 0 | | 16 | MHz |
| f(XCIN) | Sub clock oscillation frequency | | | | | 32.768 | 50 | kHz |
| f(Ring) | On-chip oscillation frequency | | | | | 1 | | MHz |
| f(PLL) | PLL clock oscillation frequency | | | | 16 | | 24 | MHz |
| f(BCLK) | CPU operation clock | | | VCC = 3.0 to 5.5 V | 0 | | 24 | MHz |
| t _{su(PLL)} | PLL frequency synthesizer stabilization wait time | | | | | | 20 | ms |
| f _(ripple) | Power supply ripple allowable frequency (VCC) | | | | | | 10 | kHz |
| V _{P-P(ripple)} | Power supply ripple allowable amplitude voltage | | | VCC = 5 V VCC = 3.3 V | | | 0.5 0.3 | V |
| V _{CC(ΔV/ΔT)} | Power supply ripple rising/falling gradient | | | VCC = 5 V VCC = 3.3 V | | | 0.3 0.3 | V/ms |

NOTES:

1. Referenced to VCC = 3.0 to 5.5 V at Topr = -40 to 85°C unless otherwise specified.
2. Relationship between main clock oscillation frequency and supply voltage is shown right.
3. Execute program/erase of flash memory by VCC = 3.3 ± 0.3 V or VCC = 5.0 ± 0.5 V.
4. When using over 16 MHz, use PLL clock. PLL clock oscillation frequency which can be used is 16 MHz, 20 MHz or 24 MHz.

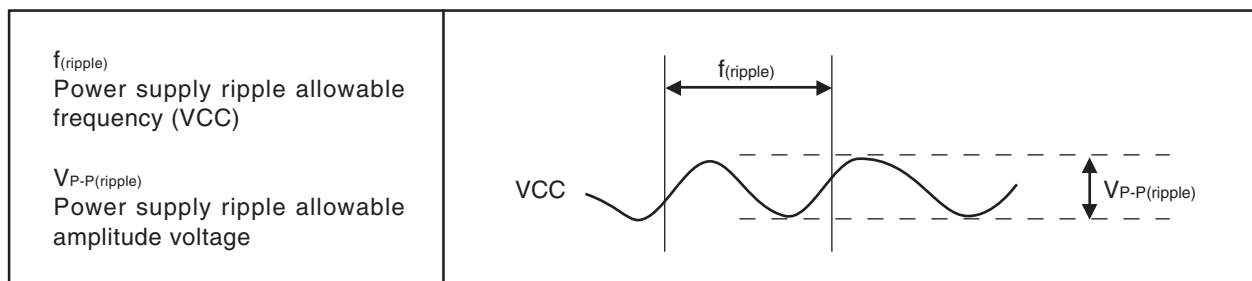
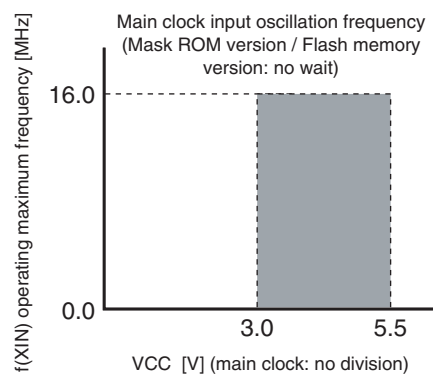


Figure 22.12 Voltage Fluctuation Timing

Table 22.29 A/D Conversion Characteristics ⁽¹⁾

| Symbol | Parameter | | Measuring Condition | | Standard | | | Unit | |
|--|--|----------------------------------|--|---|---|------|------------------|------|-----|
| | | | | | Min. | Typ. | Max. | | |
| – | Resolution | | VREF = VCC | | | | 10 | Bit | |
| INL | Integral nonlinearity error | 10 bits | VREF = VCC = 5 V | ANEX0, ANEX1 input, AN0 to AN7 input, AN0_0 to AN0_7 input, AN2_0 to AN2_7 input | | | ±3 | LSB | |
| | | | | External operation amp connection mode | | | ±7 | LSB | |
| | | 8 bits | VREF = AVCC = VCC = 5.0 V, 3.3 V | VREF = VCC = 3.3 V | ANEX0, ANEX1 input, AN0 to AN7 input, AN0_0 to AN0_7 input, AN2_0 to AN2_7 input | | | ±5 | LSB |
| | | | | External operation amp connection mode | | | ±7 | LSB | |
| – | Absolute accuracy | 10 bits | VREF = VCC = 5 V | ANEX0, ANEX1 input, AN0 to AN7 input, AN0_0 to AN0_7 input, AN2_0 to AN2_7 input | | | ±3 | LSB | |
| External operation amp connection mode | | | | | | ±7 | LSB | | |
| 8 bits | | VREF = AVCC = VCC = 5.0 V, 3.3 V | VREF = VCC = 3.3 V | ANEX0, ANEX1 input, AN0 to AN7 input, AN0_0 to AN0_7 input, AN2_0 to AN2_7 input | | | ±5 | LSB | |
| | | | External operation amp connection mode | | | ±7 | LSB | | |
| DNL | Differential nonlinearity error | | | | | | ±1 | LSB | |
| – | Offset error | | | | | | ±3 | LSB | |
| – | Gain error | | | | | | ±3 | LSB | |
| R _{LADDER} | Resistor ladder | | VREF = VCC | | 10 | | 40 | kΩ | |
| t _{CONV} | 10-bit conversion time, sample & hold available | | VREF = VCC = 5 V, φAD = 10 MHz | | 3.3 | | | μs | |
| | 8-bit conversion time, sample & hold available | | VREF = VCC = 5 V, φAD = 10 MHz | | 2.8 | | | μs | |
| t _{SAMP} | Sampling time | | | | 0.3 | | | μs | |
| V _{REF} | Reference voltage | | | | 2.0 | | V _{CC} | V | |
| V _{IA} | Analog input voltage | | | | 0 | | V _{REF} | V | |

NOTES:

1. Referenced to VCC = AVCC = VREF = 3.3 to 5.5 V, VSS = AVSS = 0 V, –40 to 85°C unless otherwise specified.
2. φAD frequency must be 10 MHz or less.
3. When sample & hold is disabled, φAD frequency must be 250 kHz or more in addition to a limit of NOTE 2.
When sample & hold is enabled, φAD frequency must be 1 MHz or more in addition to a limit of NOTE 2.

Table 22.30 D/A conversion Characteristics ⁽¹⁾

| Symbol | Parameter | Measuring Condition | Standard | | | Unit |
|-------------------|--------------------------------------|---------------------|----------|------|------|------|
| | | | Min. | Typ. | Max. | |
| – | Resolution | | | | 8 | Bits |
| – | Absolute accuracy | | | | 1.0 | % |
| t _{su} | Setup time | | | | 3 | μs |
| R _o | Output resistance | | 4 | 10 | 20 | kΩ |
| I _{VREF} | Reference power supply input current | (NOTE 2) | | | 1.5 | mA |

NOTES:

1. Referenced to VCC = AVCC = VREF = 3.3 to 5.5 V, VSS = AVSS = 0 V, –40 to 85°C unless otherwise specified.
2. This applies when using one D/A converter, with the DAi register (i = 0, 1) for the unused D/A converter set to 00h.
The resistor ladder of the A/D converter is not included. Also, the current I_{VREF} always flows even though VREF may have been set to be unconnected by the ADCON1 register.

Table 22.31 Power Supply Circuit Timing Characteristics

| Symbol | Parameter | Measuring Condition | Standard | | | Unit |
|--------------|---|---------------------|----------|------|------|---------|
| | | | Min. | Typ. | Max. | |
| $t_{d(P-R)}$ | Time for internal power supply stabilization during powering-on | VCC = 3.0 to 5.5 V | | | 2 | ms |
| $t_{d(R-S)}$ | STOP release time | | | | 150 | μ s |
| $t_{d(W-S)}$ | Low power dissipation mode wait mode release time | | | | 150 | μ s |

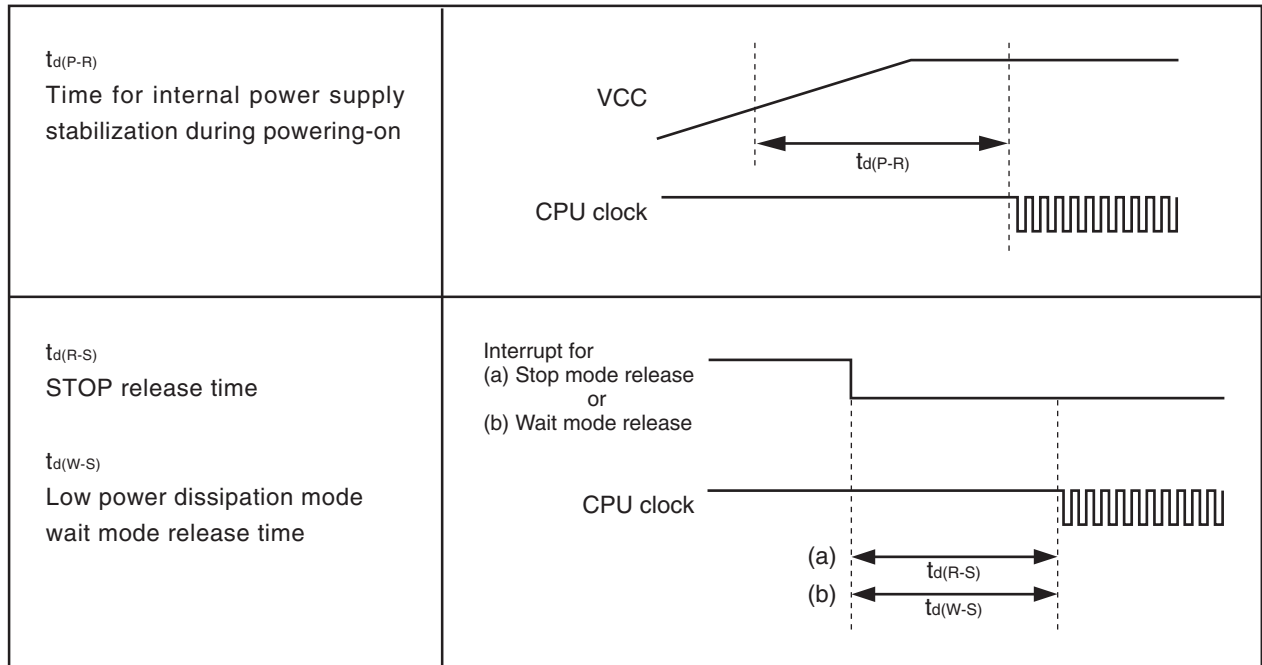


Figure 22.13 Power Supply Circuit Timing Diagram

Table 22.32 Electrical Characteristics (1) ⁽¹⁾

VCC = 5 V

| Symbol | Parameter | | Measuring Condition | Standard | | | Unit | |
|----------------------------------|-----------------------|--|---------------------------|---------------------------|------|-----------------|------|---|
| | | | | Min. | Typ. | Max. | | |
| V _{OH} | HIGH output voltage | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7 | I _{OH} = -5 mA | V _{CC} -2.0 | | V _{CC} | V | |
| V _{OH} | HIGH output voltage | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7 | I _{OH} = -200 μA | V _{CC} -0.3 | | V _{CC} | V | |
| V _{OH} | HIGH output voltage | XOUT | HIGHPOWER | I _{OH} = -1 mA | 3.0 | V _{CC} | V | |
| | | | LOWPOWER | I _{OH} = -0.5 mA | 3.0 | V _{CC} | | |
| | HIGH output voltage | XCOUT | HIGHPOWER | With no load applied | | 2.5 | | V |
| | | | LOWPOWER | With no load applied | | 1.6 | | |
| V _{OL} | LOW output voltage | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7 | I _{OL} = 5 mA | | | 2.0 | V | |
| V _{OL} | LOW output voltage | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7 | I _{OL} = 200 μA | | | 0.45 | V | |
| V _{OL} | LOW output voltage | XOUT | HIGHPOWER | I _{OL} = 1 mA | | 2.0 | V | |
| | | | LOWPOWER | I _{OL} = 0.5 mA | | 2.0 | | |
| | LOW output voltage | XCOUT | HIGHPOWER | With no load applied | | 0 | | V |
| | | | LOWPOWER | With no load applied | | 0 | | |
| V _{T+} -V _{T-} | Hysteresis | HOLD, RDY, TA0IN to TA4IN, TB0IN to TB5IN, INT0 to INT5, NMI, ADTRG, CTS0 to CTS2, SCL0 to SCL2, SDA0 to SDA2, CLK0 to CLK3, TA0OUT to TA4OUT, $\overline{KI}0$ to $\overline{KI}3$, RXD0 to RXD2, SIN3 | | 0.2 | | 1.0 | V | |
| V _{T+} -V _{T-} | Hysteresis | RESET | | 0.2 | | 2.5 | V | |
| I _{IH} | HIGH input current | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, XIN, RESET, CNVSS, BYTE | V _I = 5 V | | | 5.0 | μA | |
| I _{IL} | LOW input current | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, XIN, \overline{RESET} , CNVSS, BYTE | 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, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7 | V _I = 0 V | 30 | 50 | 170 | kΩ | |
| R _{FXIN} | Feedback resistance | XIN | | | 1.5 | | MΩ | |
| R _{FXCIN} | Feedback resistance | XCIN | | | 15 | | MΩ | |
| V _{RAM} | RAM retention voltage | | At stop mode | 2.0 | | | V | |

NOTES:

1. Referenced to VCC = 4.2 to 5.5 V, VSS = 0 V at T_{opr} = -40 to 85°C, f(BCLK) = 24 MHz unless otherwise specified.

Table 22.33 Electrical Characteristics (2) ⁽¹⁾

| Symbol | Parameter | | Measuring Condition | | Standard | | | Unit |
|-----------------|--|---|----------------------|---|--------------------------------|------|------|------|
| | | | | | Min. | Typ. | Max. | |
| I _{cc} | Power supply current (VCC = 3.0 to 5.5 V) | In single-chip mode, the output pins are open and other pins are VSS. | Mask ROM | f(BCLK) = 24 MHz, PLL operation, No division | | 20 | 36 | mA |
| | | | | On-chip oscillation, No division | | 1 | | mA |
| | | | Flash memory | f(BCLK) = 24 MHz, PLL operation, No division | | 22 | 38 | mA |
| | | | | On-chip oscillation, No division | | 1.8 | | mA |
| | | | Flash memory program | f(BCLK) = 10 MHz, VCC = 5 V | | 15 | | mA |
| | | | Flash memory erase | f(BCLK) = 10 MHz, VCC = 5 V | | 25 | | mA |
| | | | Mask ROM | f(BCLK) = 32 kHz, Low power dissipation mode, ROM ⁽²⁾ | | 25 | | μA |
| | | | Flash memory | f(BCLK) = 32 kHz, Low power dissipation mode, RAM ⁽²⁾ | | 25 | | μA |
| | | | | f(BCLK) = 32 kHz, Low power dissipation mode, Flash memory ⁽²⁾ | | 420 | | μA |
| | | | | Mask ROM Flash memory | On-chip oscillation, Wait mode | | 50 | |
| | | | Flash memory | f(BCLK) = 32 kHz, Wait mode ⁽³⁾ , Oscillation capacity High | | 8.5 | | μA |
| | | | | f(BCLK) = 32 kHz, Wait mode ⁽³⁾ , Oscillation capacity Low | | 3.0 | | μA |
| | | | | Stop mode, Topr = 25°C | | 0.8 | 3.0 | μA |

NOTES:

1. Referenced to VCC = 3.0 to 5.5 V, VSS = 0 V at Topr = -40 to 85°C, f(BCLK) = 24 MHz unless otherwise specified.
2. This indicates the memory in which the program to be executed exists.
3. With one timer operated using fC32.

Timing Requirements**VCC = 5 V****(Referenced to VCC = 5 V, VSS = 0 V, at Topr = –40 to 85°C unless otherwise specified)****Table 22.34 External Clock Input (XIN Input)**

| Symbol | Parameter | Standard | | Unit |
|-------------------|---------------------------------------|----------|------|------|
| | | Min. | Max. | |
| t _c | External clock input cycle time | 62.5 | | ns |
| t _{w(H)} | External clock input HIGH pulse width | 25 | | ns |
| t _{w(L)} | External clock input LOW pulse width | 25 | | ns |
| t _r | External clock rise time | | 15 | ns |
| t _f | External clock fall time | | 15 | ns |

Table 22.35 Memory Expansion Mode and Microprocessor Mode

| Symbol | Parameter | Standard | | Unit |
|----------------------------|--|----------|----------|------|
| | | Min. | Max. | |
| t _{ac1(RD-DB)} | Data input access time (for setting with no wait) | | (NOTE 1) | ns |
| t _{ac2(RD-DB)} | Data input access time (for setting with wait) | | (NOTE 2) | ns |
| t _{ac3(RD-DB)} | Data input access time (when accessing multiplexed bus area) | | (NOTE 3) | ns |
| t _{su(DB-RD)} | Data input setup time | 40 | | ns |
| t _{su(RDY-BCLK)} | RDY input setup time | 30 | | ns |
| t _{su(HOLD-BCLK)} | HOLD input setup time | 40 | | ns |
| t _{h(RD-DB)} | Data input hold time | 0 | | ns |
| t _{h(BCLK-RDY)} | RDY input hold time | 0 | | ns |
| t _{h(BCLK-HOLD)} | HOLD input hold time | 0 | | ns |

NOTES:

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 45 \text{ [ns]}$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{(n-0.5) \times 10^9}{f(\text{BCLK})} - 45 \text{ [ns]} \quad n \text{ is "2" for 1-wait setting, "3" for 2-wait setting and "4" for 3-wait setting.}$$

3. Calculated according to the BCLK frequency as follows:

$$\frac{(n-0.5) \times 10^9}{f(\text{BCLK})} - 45 \text{ [ns]} \quad n \text{ is "2" for 2-wait setting, "3" for 3-wait setting.}$$

Timing Requirements**VCC = 5 V****(Referenced to VCC = 5 V, VSS = 0 V, at Topr = -40 to 85°C unless otherwise specified)****Table 22.36 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 22.37 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 22.38 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 22.39 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 22.40 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 22.41 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 |

Timing Requirements**VCC = 5 V****(Referenced to VCC = 5 V, VSS = 0 V, at Topr = -40 to 85°C unless otherwise specified)****Table 22.42 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 22.43 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 22.44 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 22.45 A/D Trigger Input

| Symbol | Parameter | Standard | | Unit |
|--------------|---|----------|------|------|
| | | Min. | Max. | |
| $t_{c(AD)}$ | ADTRG input cycle time (trigger able minimum) | 1000 | | ns |
| $t_{w(ADL)}$ | ADTRG input LOW pulse width | 125 | | ns |

Table 22.46 Serial Interface

| 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 22.47 External Interrupt INTi Input

| Symbol | Parameter | Standard | | Unit |
|--------------|-----------------------------|----------|------|------|
| | | Min. | Max. | |
| $t_{w(INH)}$ | INTi input HIGH pulse width | 250 | | ns |
| $t_{w(INL)}$ | INTi input LOW pulse width | 250 | | ns |

Switching Characteristics

VCC = 5 V

(Referenced to VCC = 5 V, VSS = 0 V, at Topr = -40 to 85 °C unless otherwise specified)

Table 22.48 Memory Expansion Mode and Microprocessor Mode (for setting with no wait)

| Symbol | Parameter | Measuring Condition | Standard | | Unit |
|----------------------------|--|---------------------|----------|------|------|
| | | | Min. | Max. | |
| t _d (BCLK-AD) | Address output delay time | Figure 22.14 | | 25 | ns |
| t _h (BCLK-AD) | Address output hold time (in relation to BCLK) | | 4 | | ns |
| t _h (RD-AD) | Address output hold time (in relation to RD) | | 0 | | ns |
| t _h (WR-AD) | Address output hold time (in relation to WR) | | (NOTE 1) | | ns |
| t _d (BCLK-CS) | Chip select output delay time | | | 25 | ns |
| t _h (BCLK-CS) | Chip select output hold time (in relation to BCLK) | | 4 | | ns |
| t _d (BCLK-ALE) | ALE signal output delay time | | | 15 | ns |
| t _h (BCLK-ALE) | ALE signal output hold time | | -4 | | ns |
| t _d (BCLK-RD) | RD signal output delay time | | | 25 | ns |
| t _h (BCLK-RD) | RD signal output hold time | | 0 | | ns |
| t _d (BCLK-WR) | WR signal output delay time | | | 25 | ns |
| t _h (BCLK-WR) | WR signal output hold time | | 0 | | ns |
| t _d (BCLK-DB) | Data output delay time (in relation to BCLK) | | | 40 | ns |
| t _h (BCLK-DB) | Data output hold time (in relation to BCLK) ⁽³⁾ | | 4 | | ns |
| t _d (DB-WR) | Data output delay time (in relation to WR) | | (NOTE 2) | | ns |
| t _h (WR-DB) | Data output hold time (in relation to WR) ⁽³⁾ | | (NOTE 1) | | ns |
| t _d (BCLK-HLDA) | HLDA output delay time | | 40 | ns | |

NOTES:

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 10 \text{ [ns]}$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 40 \text{ [ns]} \quad f(\text{BCLK}) \text{ is } 12.5 \text{ MHz or less.}$$

3. This standard value shows the timing when the output is off, and does not show hold time of data bus.

Hold time of data bus varies with capacitor volume and pull-up (pull-down) resistance value.

Hold time of data bus is expressed in

$$t = -CR \times \ln(1 - V_{OL} / V_{CC})$$

by a circuit of the right figure.

For example, when V_{OL} = 0.2 V_{CC}, C = 30 pF, R = 1 kΩ, hold time of output “L” level is

$$t = -30 \text{ pF} \times 1 \text{ k}\Omega \times \ln(1 - 0.2 V_{CC} / V_{CC}) = 6.7 \text{ ns.}$$

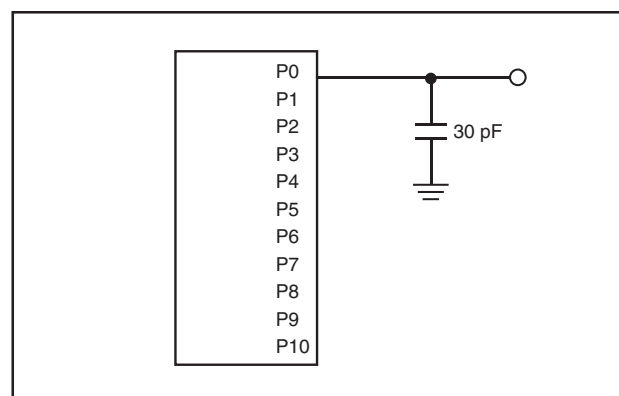
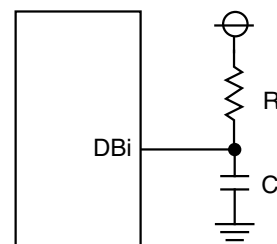


Figure 22.14 Port P0 to P10 Measurement Circuit

Switching Characteristics**VCC = 5 V****(Referenced to VCC = 5 V, VSS = 0 V, at Topr = -40 to 85 °C unless otherwise specified)****Table 22.49 Memory Expansion Mode and Microprocessor Mode (for 1- to 3-wait setting and external area access)**

| Symbol | Parameter | Measuring Condition | Standard | | Unit |
|----------------------------|--|---------------------|----------|------|------|
| | | | Min. | Max. | |
| t _d (BCLK-AD) | Address output delay time | Figure 22.14 | | 25 | ns |
| t _h (BCLK-AD) | Address output hold time (in relation to BCLK) | | 4 | | ns |
| t _h (RD-AD) | Address output hold time (in relation to RD) | | 0 | | ns |
| t _h (WR-AD) | Address output hold time (in relation to WR) | | (NOTE 1) | | ns |
| t _d (BCLK-CS) | Chip select output delay time | | | 25 | ns |
| t _h (BCLK-CS) | Chip select output hold time (in relation to BCLK) | | 4 | | ns |
| t _d (BCLK-ALE) | ALE signal output delay time | | | 15 | ns |
| t _h (BCLK-ALE) | ALE signal output hold time | | -4 | | ns |
| t _d (BCLK-RD) | RD signal output delay time | | | 25 | ns |
| t _h (BCLK-RD) | RD signal output hold time | | 0 | | ns |
| t _d (BCLK-WR) | WR signal output delay time | | | 25 | ns |
| t _h (BCLK-WR) | WR signal output hold time | | 0 | | ns |
| t _d (BCLK-DB) | Data output delay time (in relation to BCLK) | | | 40 | ns |
| t _h (BCLK-DB) | Data output hold time (in relation to BCLK) ⁽³⁾ | | 4 | | ns |
| t _d (DB-WR) | Data output delay time (in relation to WR) | | (NOTE 2) | | ns |
| t _h (WR-DB) | Data output hold time (in relation to WR) ⁽³⁾ | (NOTE 1) | | ns | |
| t _d (BCLK-HLDA) | HLDA output delay time | | 40 | ns | |

NOTES:

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 10 \text{ [ns]}$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{(n - 0.5) \times 10^9}{f(\text{BCLK})} - 40 \text{ [ns]}$$

n is "1" for 1-wait setting, "2" for 2-wait setting and "3" for 3-wait setting.
When n = 1, f(BCLK) is 12.5 MHz or less.

3. This standard value shows the timing when the output is off, and does not show hold time of data bus.

Hold time of data bus varies with capacitor volume and pull-up (pull-down) resistance value.

Hold time of data bus is expressed in

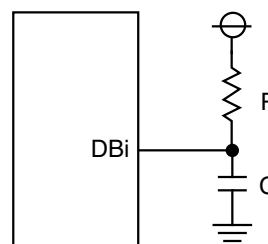
$$t = -CR \times \ln(1 - V_{OL} / V_{CC})$$

by a circuit of the right figure.

For example, when $V_{OL} = 0.2 V_{CC}$, $C = 30 \text{ pF}$,

$R = 1 \text{ k}\Omega$, hold time of output "L" level is

$$t = -30 \text{ pF} \times 1 \text{ k}\Omega \times \ln(1 - 0.2 V_{CC} / V_{CC}) = 6.7 \text{ ns.}$$



Switching Characteristics**VCC = 5 V****(Referenced to VCC = 5 V, VSS = 0 V, at Topr = -40 to 85 °C unless otherwise specified)****Table 22.50 Memory Expansion Mode and Microprocessor Mode
(for 2- to 3-wait setting, external area access and multiplexed bus selection)**

| Symbol | Parameter | Measuring Condition | Standard | | Unit |
|----------------------------|---|---------------------|----------|------|------|
| | | | Min. | Max. | |
| t _d (BCLK-AD) | Address output delay time | Figure 22.14 | | 25 | ns |
| t _h (BCLK-AD) | Address output hold time (in relation to BCLK) | | 4 | | ns |
| t _h (RD-AD) | Address output hold time (in relation to RD) | | (NOTE 1) | | ns |
| t _h (WR-AD) | Address output hold time (in relation to WR) | | (NOTE 1) | | ns |
| t _d (BCLK-CS) | Chip select output delay time | | | 25 | ns |
| t _h (BCLK-CS) | Chip select output hold time (in relation to BCLK) | | 4 | | ns |
| t _h (RD-CS) | Chip select output hold time (in relation to RD) | | (NOTE 1) | | ns |
| t _h (WR-CS) | Chip select output hold time (in relation to WR) | | (NOTE 1) | | ns |
| t _d (BCLK-RD) | RD signal output delay time | | | 25 | ns |
| t _h (BCLK-RD) | RD signal output hold time | | 0 | | ns |
| t _d (BCLK-WR) | WR signal output delay time | | | 25 | ns |
| t _h (BCLK-WR) | WR signal output hold time | | 0 | | ns |
| t _d (BCLK-DB) | Data output delay time (in relation to BCLK) | | | 40 | ns |
| t _h (BCLK-DB) | Data output hold time (in relation to BCLK) | | 4 | | ns |
| t _d (DB-WR) | Data output delay time (in relation to WR) | | (NOTE 2) | | ns |
| t _h (WR-DB) | Data output hold time (in relation to WR) | | (NOTE 1) | | ns |
| t _d (BCLK-HLDA) | HLDA output delay time | | | 40 | ns |
| t _d (BCLK-ALE) | ALE signal output delay time (in relation to BCLK) | | | 15 | ns |
| t _h (BCLK-ALE) | ALE signal output hold time (in relation to BCLK) | | -4 | | ns |
| t _d (AD-ALE) | ALE signal output delay time (in relation to Address) | | (NOTE 3) | | ns |
| t _h (ALE-AD) | ALE signal output hold time (in relation to Address) | (NOTE 4) | | ns | |
| t _d (AD-RD) | RD signal output delay from the end of Address | 0 | | ns | |
| t _d (AD-WR) | WR signal output delay from the end of Address | 0 | | ns | |
| t _{dZ} (RD-AD) | Address output floating start time | | 8 | ns | |

NOTES:

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 10 \text{ [ns]}$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{(n - 0.5) \times 10^9}{f(\text{BCLK})} - 40 \text{ [ns]} \quad n \text{ is "2" for 2-wait setting, "3" for 3-wait setting.}$$

3. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 25 \text{ [ns]}$$

4. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 15 \text{ [ns]}$$

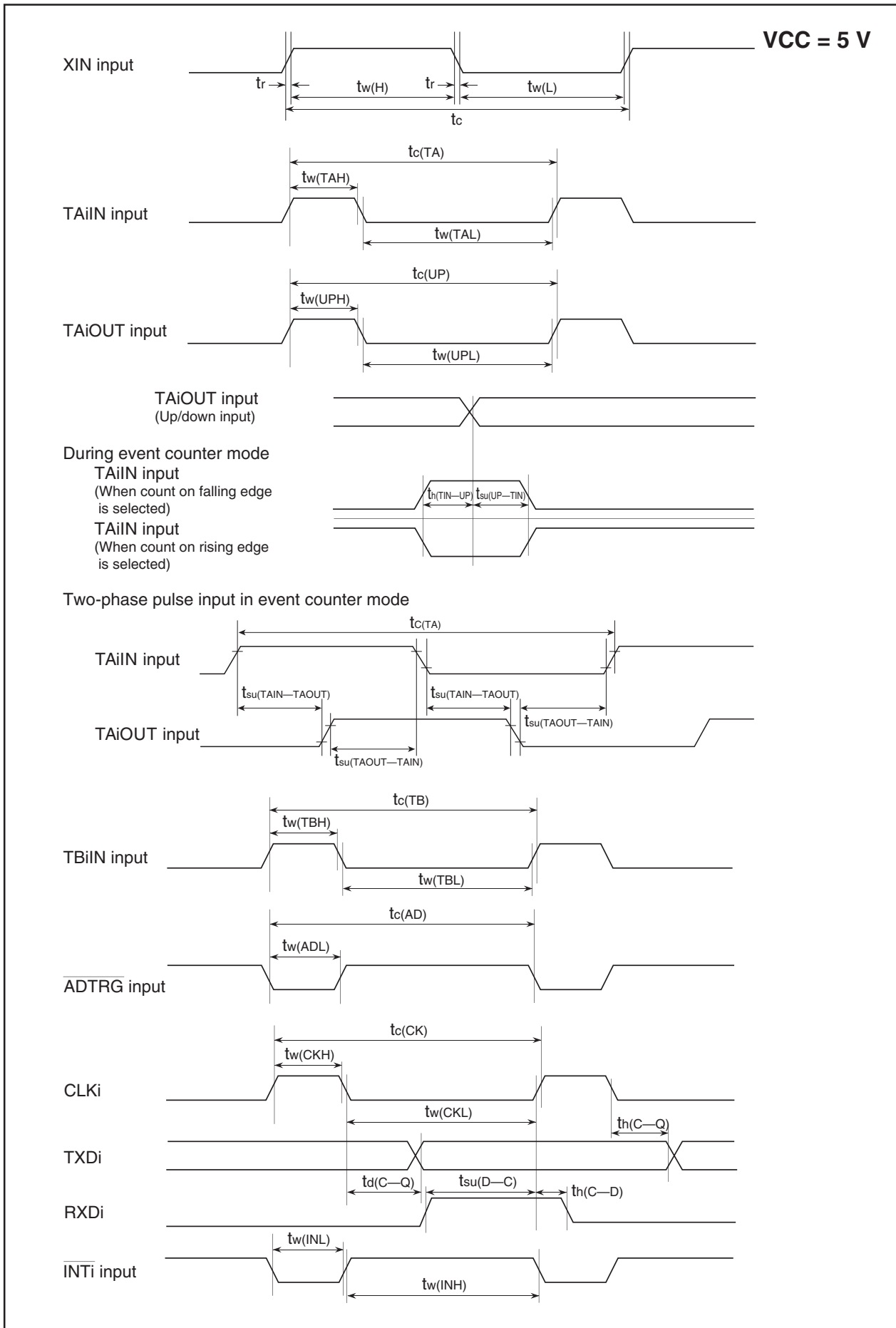


Figure 22.15 Timing Diagram (1)

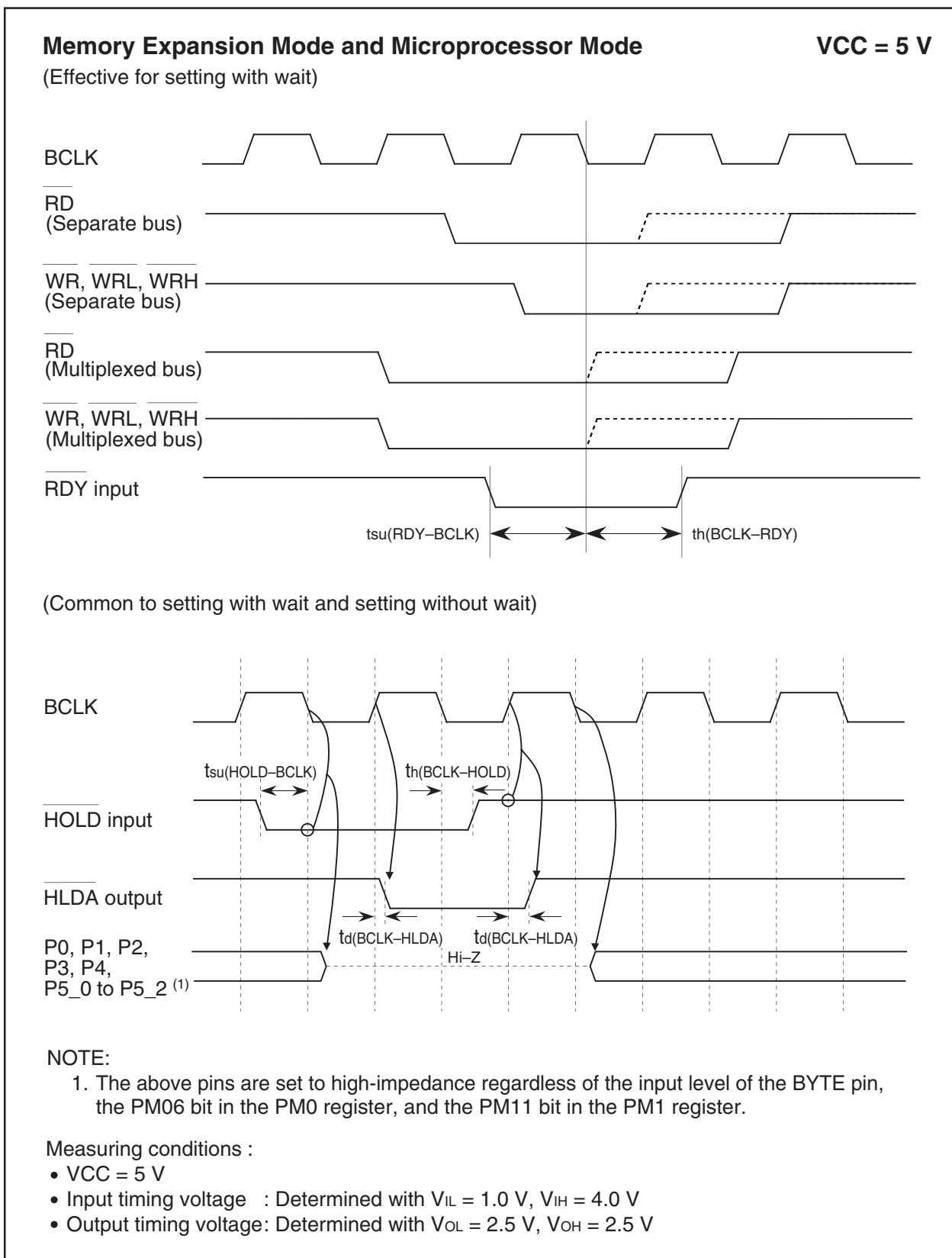


Figure 22.16 Timing Diagram (2)

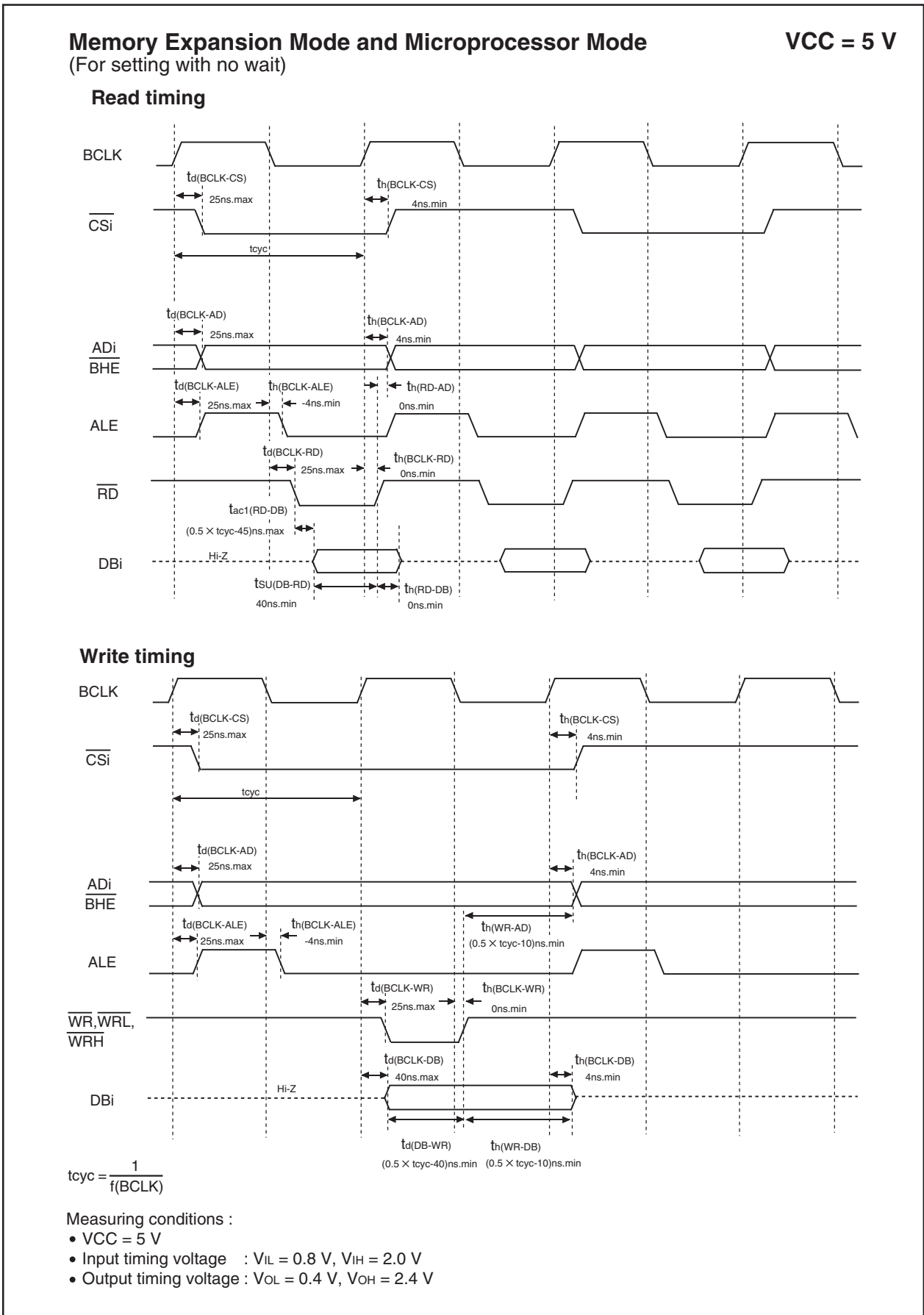


Figure 22.17 Timing Diagram (3)

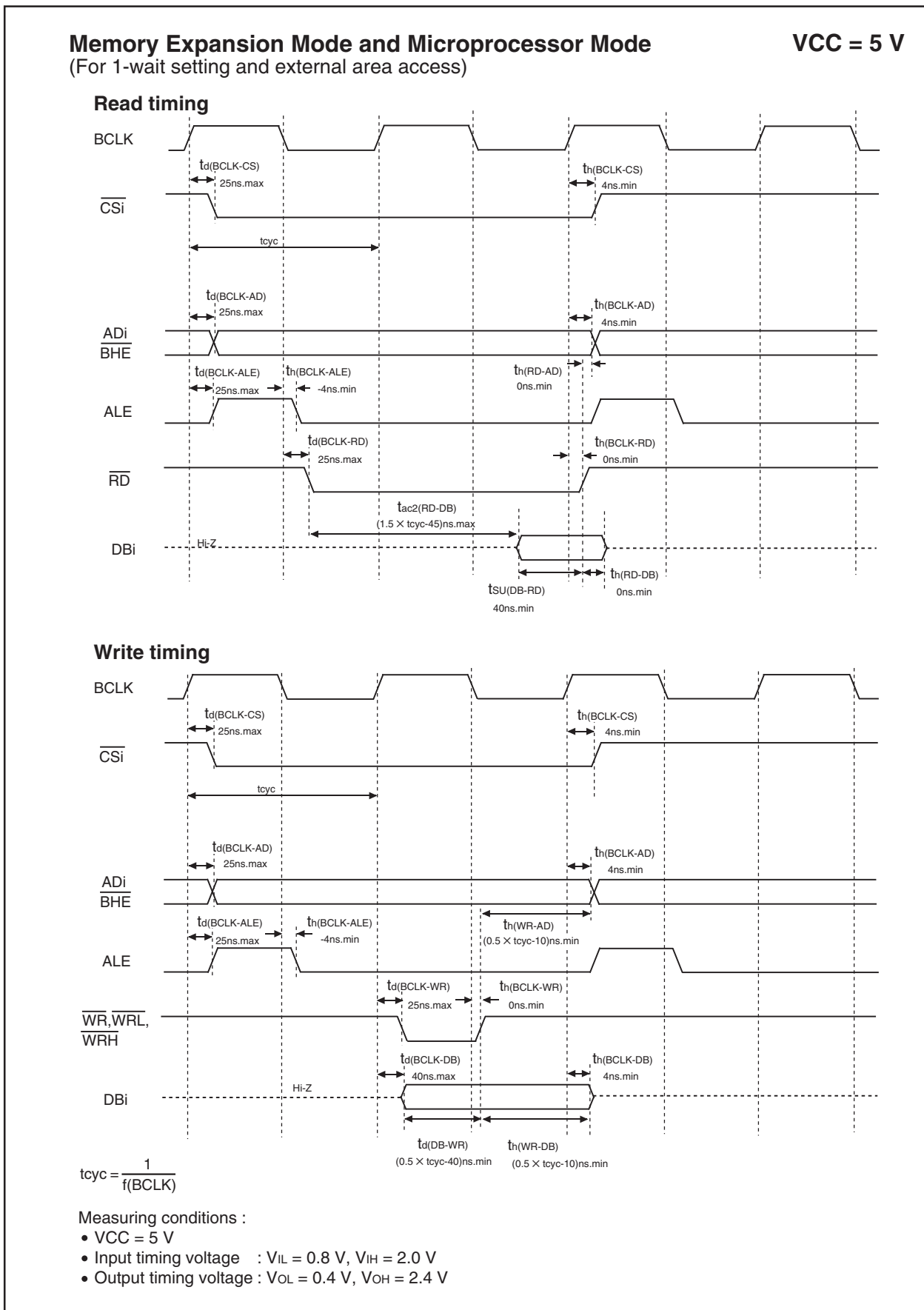


Figure 22.18 Timing Diagram (4)

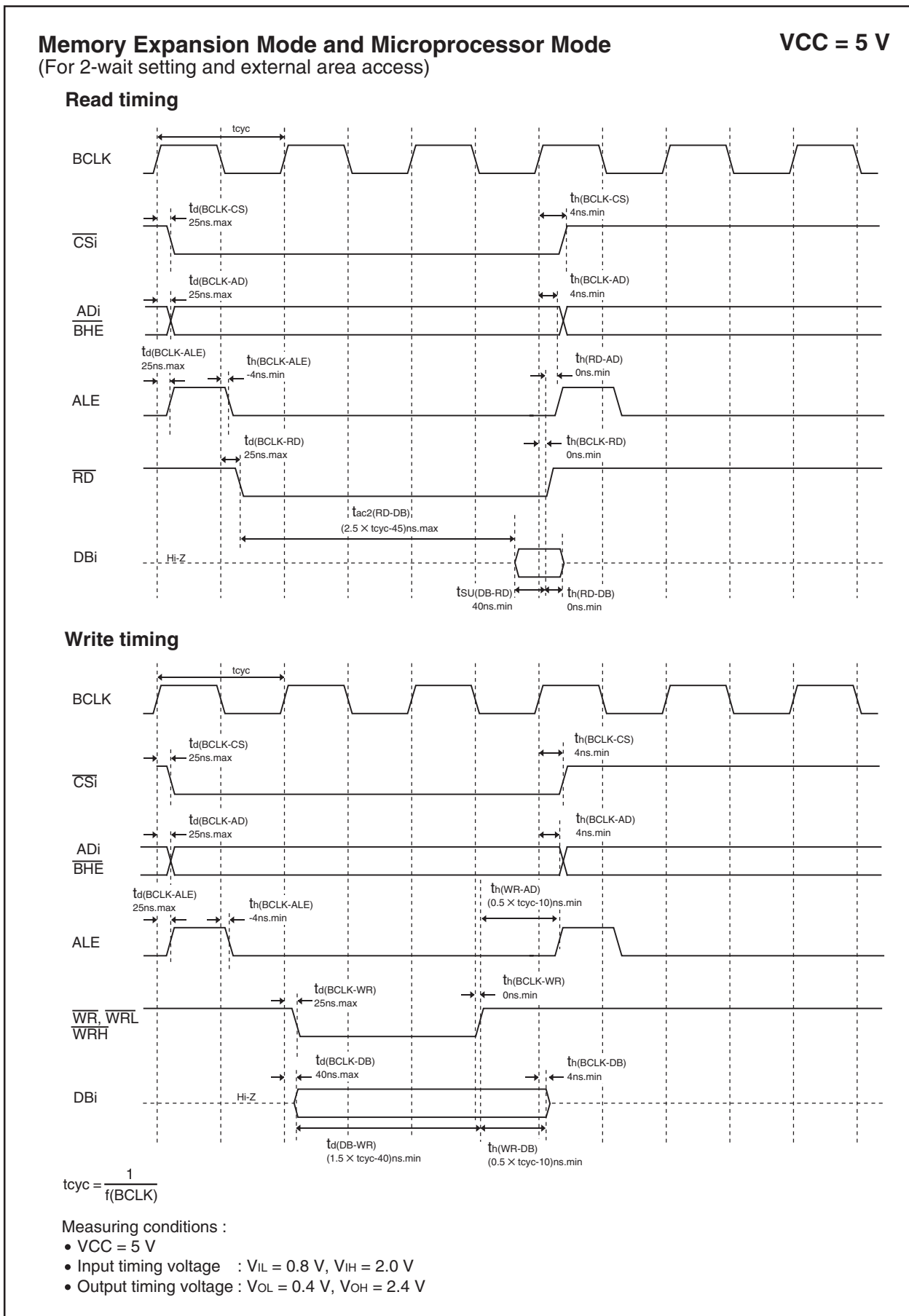


Figure 22.19 Timing Diagram (5)

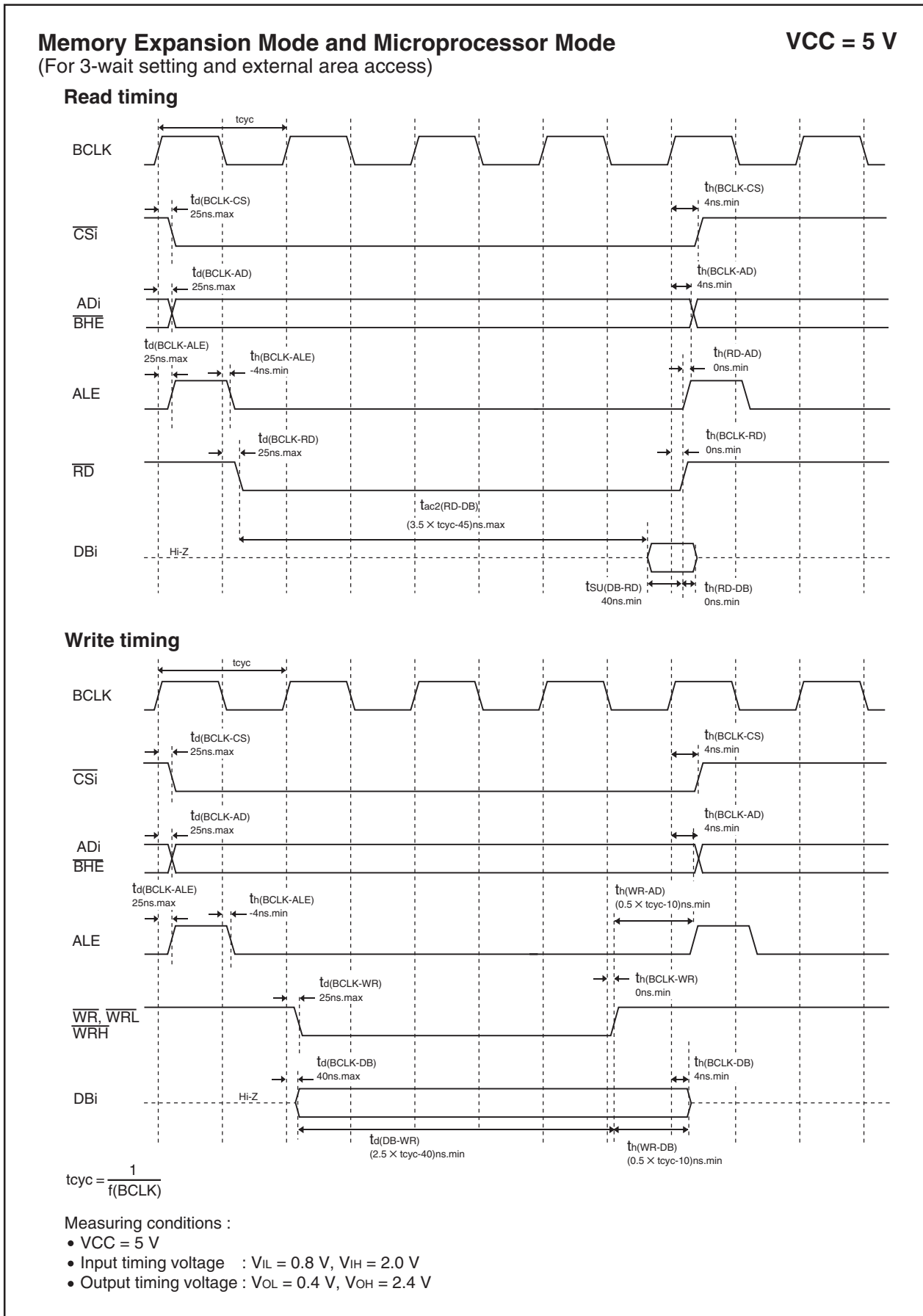


Figure 22.20 Timing Diagram (6)

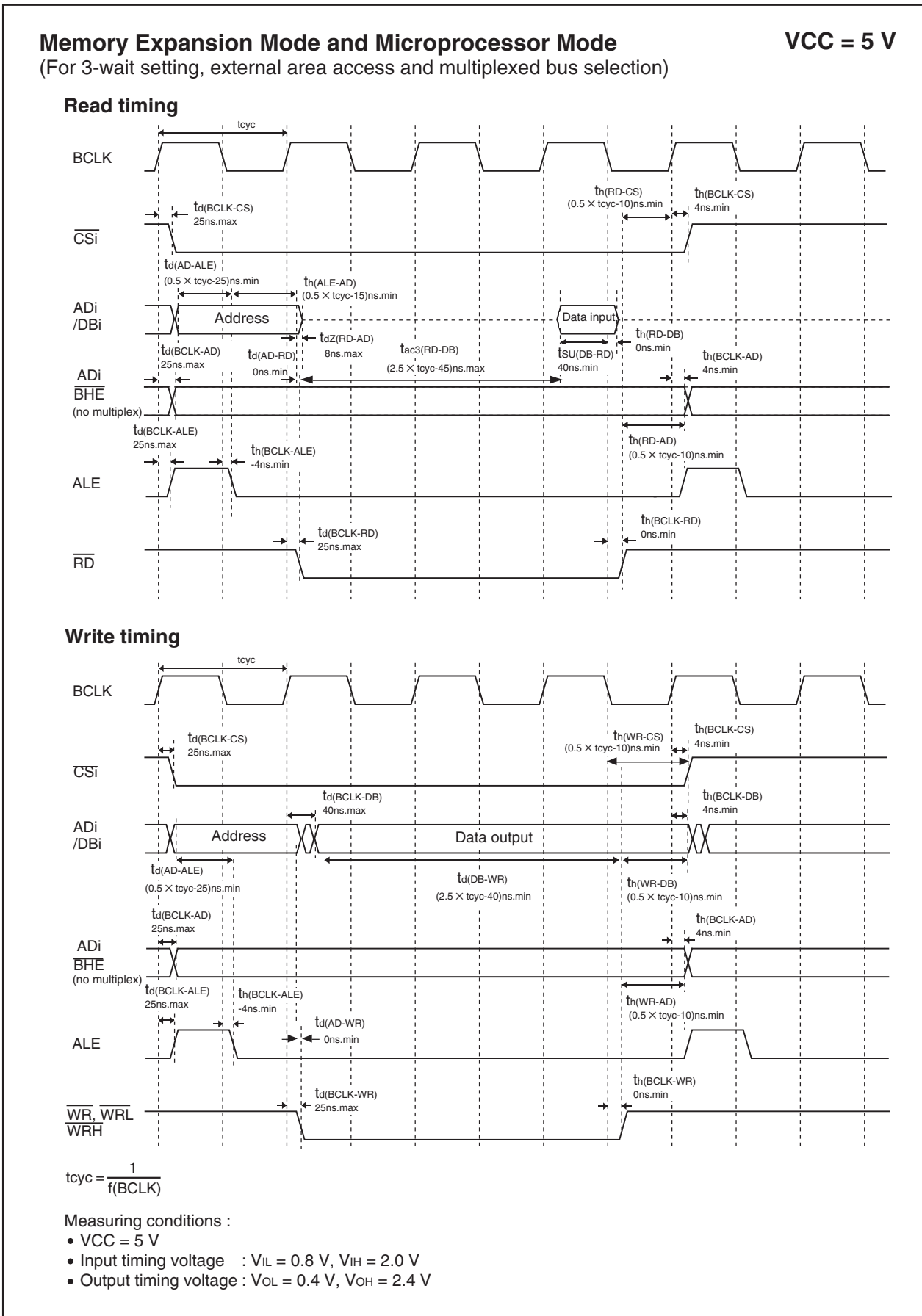


Figure 22.22 Timing Diagram (8)

Table 22.51 Electrical Characteristics ⁽¹⁾

VCC = 3.3 V

| Symbol | Parameter | | Measuring Condition | Standard | | | Unit |
|----------------------------------|-----------------------|---|-------------------------|---------------------------|----------------------|-----------------|------|
| | | | | Min. | Typ. | Max. | |
| V _{OH} | HIGH output voltage | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7 | I _{OH} = -1 mA | V _{CC} -0.5 | | V _{CC} | V |
| V _{OH} | HIGH output voltage | XOUT | HIGHPOWER | I _{OH} = -0.1 mA | V _{CC} -0.5 | V _{CC} | V |
| | | | LOWPOWER | I _{OH} = -50 μA | V _{CC} -0.5 | V _{CC} | |
| | HIGH output voltage | XCOUT | HIGHPOWER | With no load applied | | 2.5 | V |
| | | | LOWPOWER | With no load applied | | 1.6 | |
| V _{OL} | LOW output voltage | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7 | I _{OL} = 1 mA | | | 0.5 | V |
| V _{OL} | LOW output voltage | XOUT | HIGHPOWER | I _{OL} = 0.1 mA | | 0.5 | V |
| | | | LOWPOWER | I _{OL} = 50 μA | | 0.5 | |
| | LOW output voltage | XCOUT | HIGHPOWER | With no load applied | | 0 | V |
| | | | LOWPOWER | With no load applied | | 0 | |
| V _{T+} -V _{T-} | Hysteresis | HOLD, RDY, TA0IN to TA4IN, TB0IN to TB5IN, INT0 to INT5, NMI, ADTRG, CTS0 to CTS2, SCL0 to SCL2, SDA0 to SDA2, CLK0 to CLK3, TA0OUT to TA4OUT, KI0 to KI3, RXD0 to RXD2, SIN3 | | 0.2 | | 0.8 | V |
| V _{T+} -V _{T-} | Hysteresis | RESET | | 0.2 | | 1.8 | V |
| I _{IH} | HIGH input current | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, XIN, RESET, CNVSS, BYTE | V _I = 3.3 V | | | 4.0 | μA |
| I _{IL} | LOW input current | P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, XIN, RESET, CNVSS, BYTE | 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, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7 | V _I = 0 V | 50 | 100 | 500 | kΩ |
| R _{IXIN} | Feedback resistance | XIN | | | 3.0 | | MΩ |
| R _{IXCIN} | Feedback resistance | XCIN | | | 25 | | MΩ |
| V _{RAM} | RAM retention voltage | | At stop mode | 2.0 | | | V |

NOTES:

1. Referenced to VCC = 3.0 to 3.6 V, VSS = 0 V at Topr = -40 to 85°C, f(BCLK) = 24 MHz unless otherwise specified.

Timing Requirements**VCC = 3.3 V****(Referenced to VCC = 3.3 V, VSS = 0 V, at Topr = -40 to 85°C unless otherwise specified)****Table 22.52 External Clock Input (XIN Input)**

| Symbol | Parameter | Standard | | Unit |
|-------------------|---------------------------------------|----------|------|------|
| | | Min. | Max. | |
| t _c | External clock input cycle time | 62.5 | | ns |
| t _{w(H)} | External clock input HIGH pulse width | 25 | | ns |
| t _{w(L)} | External clock input LOW pulse width | 25 | | ns |
| t _r | External clock rise time | | 15 | ns |
| t _f | External clock fall time | | 15 | ns |

Table 22.53 Memory Expansion Mode and Microprocessor Mode

| Symbol | Parameter | Standard | | Unit |
|----------------------------|--|----------|----------|------|
| | | Min. | Max. | |
| t _{ac1(RD-DB)} | Data input access time (for setting with no wait) | | (NOTE 1) | ns |
| t _{ac2(RD-DB)} | Data input access time (for setting with wait) | | (NOTE 2) | ns |
| t _{ac3(RD-DB)} | Data input access time (when accessing multiplexed bus area) | | (NOTE 3) | ns |
| t _{su(DB-RD)} | Data input setup time | 50 | | ns |
| t _{su(RDY-BCLK)} | RDY input setup time | 40 | | ns |
| t _{su(HOLD-BCLK)} | HOLD input setup time | 50 | | ns |
| t _{h(RD-DB)} | Data input hold time | 0 | | ns |
| t _{h(BCLK-RDY)} | RDY input hold time | 0 | | ns |
| t _{h(BCLK-HOLD)} | HOLD input hold time | 0 | | ns |

NOTES:

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 60 \text{ [ns]}$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{(n-0.5) \times 10^9}{f(\text{BCLK})} - 60 \text{ [ns]} \quad n \text{ is "2" for 1-wait setting, "3" for 2-wait setting and "4" for 3-wait setting.}$$

3. Calculated according to the BCLK frequency as follows:

$$\frac{(n-0.5) \times 10^9}{f(\text{BCLK})} - 60 \text{ [ns]} \quad n \text{ is "2" for 2-wait setting, "3" for 3-wait setting.}$$

Timing Requirements**VCC = 3.3 V****(Referenced to VCC = 3.3 V, VSS = 0 V, at Topr = -40 to 85°C unless otherwise specified)****Table 22.54 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 22.55 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 22.56 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 22.57 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 22.58 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 22.59 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 | | μ s |
| $t_{su(TAIN-TAOUT)}$ | TAiOUT input setup time | 500 | | ns |
| $t_{su(TAOUT-TAIN)}$ | TAiIN input setup time | 500 | | ns |

Timing Requirements**VCC = 3.3 V****(Referenced to VCC = 3.3 V, VSS = 0 V, at Topr = -40 to 85°C unless otherwise specified)****Table 22.60 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 22.61 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 22.62 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 22.63 A/D Trigger Input

| Symbol | Parameter | Standard | | Unit |
|--------------|---|----------|------|------|
| | | Min. | Max. | |
| $t_{c(AD)}$ | ADTRG input cycle time (trigger able minimum) | 1500 | | ns |
| $t_{w(ADL)}$ | ADTRG input LOW pulse width | 200 | | ns |

Table 22.64 Serial Interface

| 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 22.65 External Interrupt INTi Input

| Symbol | Parameter | Standard | | Unit |
|--------------|-----------------------------|----------|------|------|
| | | Min. | Max. | |
| $t_{w(INH)}$ | INTi input HIGH pulse width | 380 | | ns |
| $t_{w(INL)}$ | INTi input LOW pulse width | 380 | | ns |

Switching Characteristics

VCC = 3.3 V

(Referenced to VCC = 3.3 V, VSS = 0 V, at Topr = -40 to 85 °C unless otherwise specified)

Table 22.66 Memory Expansion Mode and Microprocessor Mode (for setting with no wait)

| Symbol | Parameter | Measuring Condition | Standard | | Unit |
|----------------------------|--|---------------------|----------|------|------|
| | | | Min. | Max. | |
| t _d (BCLK-AD) | Address output delay time | Figure 22.23 | | 30 | ns |
| t _h (BCLK-AD) | Address output hold time (in relation to BCLK) | | 4 | | ns |
| t _h (RD-AD) | Address output hold time (in relation to RD) | | 0 | | ns |
| t _h (WR-AD) | Address output hold time (in relation to WR) | | (NOTE 1) | | ns |
| t _d (BCLK-CS) | Chip select output delay time | | | 30 | ns |
| t _h (BCLK-CS) | Chip select output hold time (in relation to BCLK) | | 4 | | ns |
| t _d (BCLK-ALE) | ALE signal output delay time | | | 25 | ns |
| t _h (BCLK-ALE) | ALE signal output hold time | | -4 | | ns |
| t _d (BCLK-RD) | RD signal output delay time | | | 30 | ns |
| t _h (BCLK-RD) | RD signal output hold time | | 0 | | ns |
| t _d (BCLK-WR) | WR signal output delay time | | | 30 | ns |
| t _h (BCLK-WR) | WR signal output hold time | | 0 | | ns |
| t _d (BCLK-DB) | Data output delay time (in relation to BCLK) | | | 40 | ns |
| t _h (BCLK-DB) | Data output hold time (in relation to BCLK) ⁽³⁾ | | 4 | | ns |
| t _d (DB-WR) | Data output delay time (in relation to WR) | | (NOTE 2) | | ns |
| t _h (WR-DB) | Data output hold time (in relation to WR) ⁽³⁾ | | (NOTE 1) | | ns |
| t _d (BCLK-HLDA) | HLDA output delay time | | 40 | ns | |

NOTES:

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 10 \text{ [ns]}$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 40 \text{ [ns]} \quad f(\text{BCLK}) \text{ is } 12.5 \text{ MHz or less.}$$

3. This standard value shows the timing when the output is off, and does not show hold time of data bus.

Hold time of data bus varies with capacitor volume and pull-up (pull-down) resistance value.

Hold time of data bus is expressed in

$$t = -CR \times \ln(1 - V_{OL} / V_{CC})$$

by a circuit of the right figure.

For example, when $V_{OL} = 0.2 V_{CC}$, $C = 30 \text{ pF}$,

$R = 1 \text{ k}\Omega$, hold time of output "L" level is

$$t = -30 \text{ pF} \times 1 \text{ k}\Omega \times \ln(1 - 0.2 V_{CC} / V_{CC}) = 6.7 \text{ ns.}$$

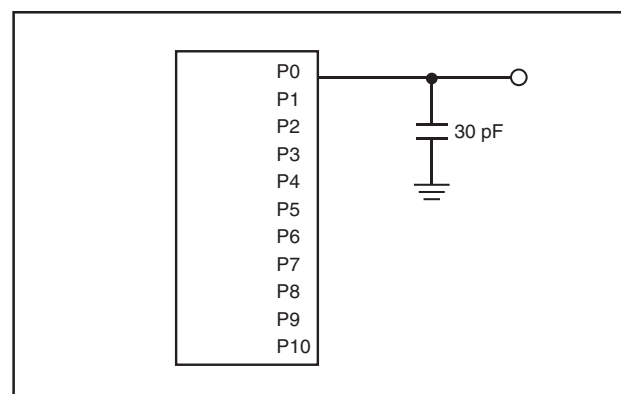
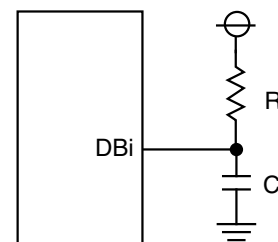


Figure 22.23 Port P0 to P10 Measurement Circuit

Switching Characteristics**VCC = 3.3 V****(Referenced to VCC = 3.3 V, VSS = 0 V, at Topr = -40 to 85 °C unless otherwise specified)****Table 22.67 Memory Expansion Mode and Microprocessor Mode (for 1- to 3-wait setting and external area access)**

| Symbol | Parameter | Measuring Condition | Standard | | Unit |
|----------------------------|--|---------------------|----------|------|------|
| | | | Min. | Max. | |
| t _d (BCLK-AD) | Address output delay time | Figure 22.23 | | 30 | ns |
| t _h (BCLK-AD) | Address output hold time (in relation to BCLK) | | 4 | | ns |
| t _h (RD-AD) | Address output hold time (in relation to RD) | | 0 | | ns |
| t _h (WR-AD) | Address output hold time (in relation to WR) | | (NOTE 1) | | ns |
| t _d (BCLK-CS) | Chip select output delay time | | | 30 | ns |
| t _h (BCLK-CS) | Chip select output hold time (in relation to BCLK) | | 4 | | ns |
| t _d (BCLK-ALE) | ALE signal output delay time | | | 25 | ns |
| t _h (BCLK-ALE) | ALE signal output hold time | | -4 | | ns |
| t _d (BCLK-RD) | RD signal output delay time | | | 30 | ns |
| t _h (BCLK-RD) | RD signal output hold time | | 0 | | ns |
| t _d (BCLK-WR) | WR signal output delay time | | | 30 | ns |
| t _h (BCLK-WR) | WR signal output hold time | | 0 | | ns |
| t _d (BCLK-DB) | Data output delay time (in relation to BCLK) | | | 40 | ns |
| t _h (BCLK-DB) | Data output hold time (in relation to BCLK) ⁽³⁾ | | 4 | | ns |
| t _d (DB-WR) | Data output delay time (in relation to WR) | | (NOTE 2) | | ns |
| t _h (WR-DB) | Data output hold time (in relation to WR) ⁽³⁾ | (NOTE 1) | | ns | |
| t _d (BCLK-HLDA) | HLDA output delay time | | 40 | ns | |

NOTES:

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 10 \text{ [ns]}$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{(n - 0.5) \times 10^9}{f(\text{BCLK})} - 40 \text{ [ns]}$$

n is "1" for 1-wait setting, "2" for 2-wait setting and "3" for 3-wait setting.
When n = 1, f(BCLK) is 12.5 MHz or less.

3. This standard value shows the timing when the output is off, and does not show hold time of data bus.

Hold time of data bus varies with capacitor volume and pull-up (pull-down) resistance value.

Hold time of data bus is expressed in

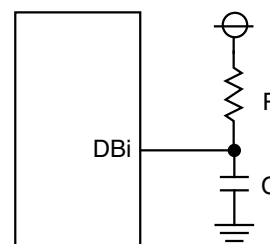
$$t = -CR \times \ln(1 - V_{OL} / V_{CC})$$

by a circuit of the right figure.

For example, when $V_{OL} = 0.2 V_{CC}$, $C = 30 \text{ pF}$,

$R = 1 \text{ k}\Omega$, hold time of output "L" level is

$$t = -30 \text{ pF} \times 1 \text{ k}\Omega \times \ln(1 - 0.2 V_{CC} / V_{CC}) = 6.7 \text{ ns.}$$



Switching Characteristics**VCC = 3.3 V****(Referenced to VCC = 3.3 V, VSS = 0 V, at Topr = –40 to 85 °C unless otherwise specified)****Table 22.68 Memory Expansion Mode and Microprocessor Mode
(for 2- to 3-wait setting, external area access and multiplexed bus selection)**

| Symbol | Parameter | Measuring Condition | Standard | | Unit |
|----------------------------|---|---------------------|----------|------|------|
| | | | Min. | Max. | |
| t _d (BCLK-AD) | Address output delay time | Figure 22.23 | | 50 | ns |
| t _h (BCLK-AD) | Address output hold time (in relation to BCLK) | | 4 | | ns |
| t _h (RD-AD) | Address output hold time (in relation to RD) | | (NOTE 1) | | ns |
| t _h (WR-AD) | Address output hold time (in relation to WR) | | (NOTE 1) | | ns |
| t _d (BCLK-CS) | Chip select output delay time | | | 50 | ns |
| t _h (BCLK-CS) | Chip select output hold time (in relation to BCLK) | | 4 | | ns |
| t _h (RD-CS) | Chip select output hold time (in relation to RD) | | (NOTE 1) | | ns |
| t _h (WR-CS) | Chip select output hold time (in relation to WR) | | (NOTE 1) | | ns |
| t _d (BCLK-RD) | RD signal output delay time | | | 40 | ns |
| t _h (BCLK-RD) | RD signal output hold time | | 0 | | ns |
| t _d (BCLK-WR) | WR signal output delay time | | | 40 | ns |
| t _h (BCLK-WR) | WR signal output hold time | | 0 | | ns |
| t _d (BCLK-DB) | Data output delay time (in relation to BCLK) | | | 50 | ns |
| t _h (BCLK-DB) | Data output hold time (in relation to BCLK) | | 4 | | ns |
| t _d (DB-WR) | Data output delay time (in relation to WR) | | (NOTE 2) | | ns |
| t _h (WR-DB) | Data output hold time (in relation to WR) | | (NOTE 1) | | ns |
| t _d (BCLK-HLDA) | HLDA output delay time | | | 40 | ns |
| t _d (BCLK-ALE) | ALE signal output delay time (in relation to BCLK) | | | 25 | ns |
| t _h (BCLK-ALE) | ALE signal output hold time (in relation to BCLK) | | –4 | | ns |
| t _d (AD-ALE) | ALE signal output delay time (in relation to Address) | | (NOTE 3) | | ns |
| t _h (ALE-AD) | ALE signal output hold time (in relation to Address) | | (NOTE 4) | | ns |
| t _d (AD-RD) | RD signal output delay from the end of Address | | 0 | | ns |
| t _d (AD-WR) | WR signal output delay from the end of Address | | 0 | | ns |
| t _{dZ} (RD-AD) | Address output floating start time | | | 8 | ns |

NOTES:

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 10 \text{ [ns]}$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{(n - 0.5) \times 10^9}{f(\text{BCLK})} - 50 \text{ [ns]} \quad n \text{ is "2" for 2-wait setting, "3" for 3-wait setting.}$$

3. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 40 \text{ [ns]}$$

4. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 15 \text{ [ns]}$$

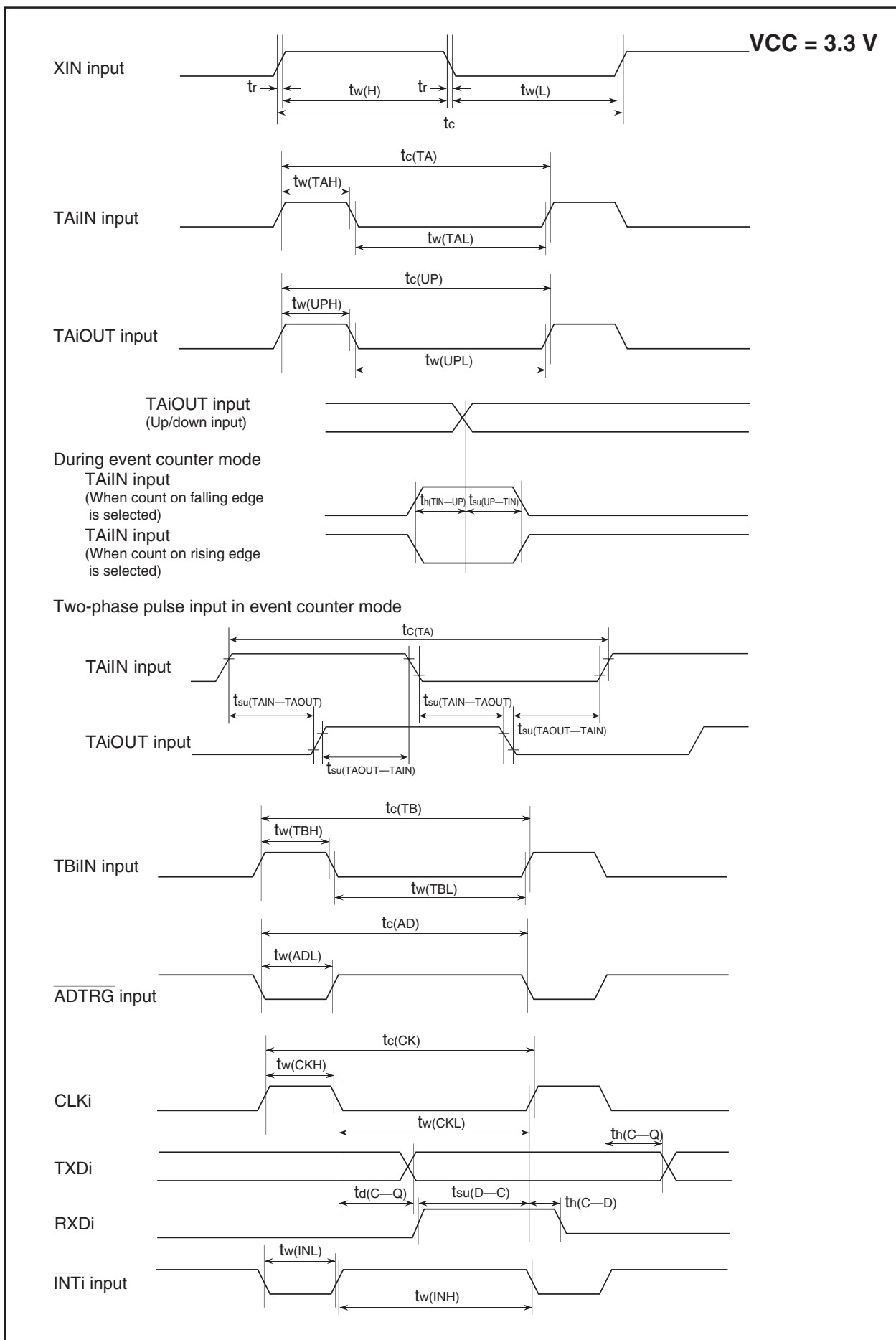


Figure 22.24 Timing Diagram (1)

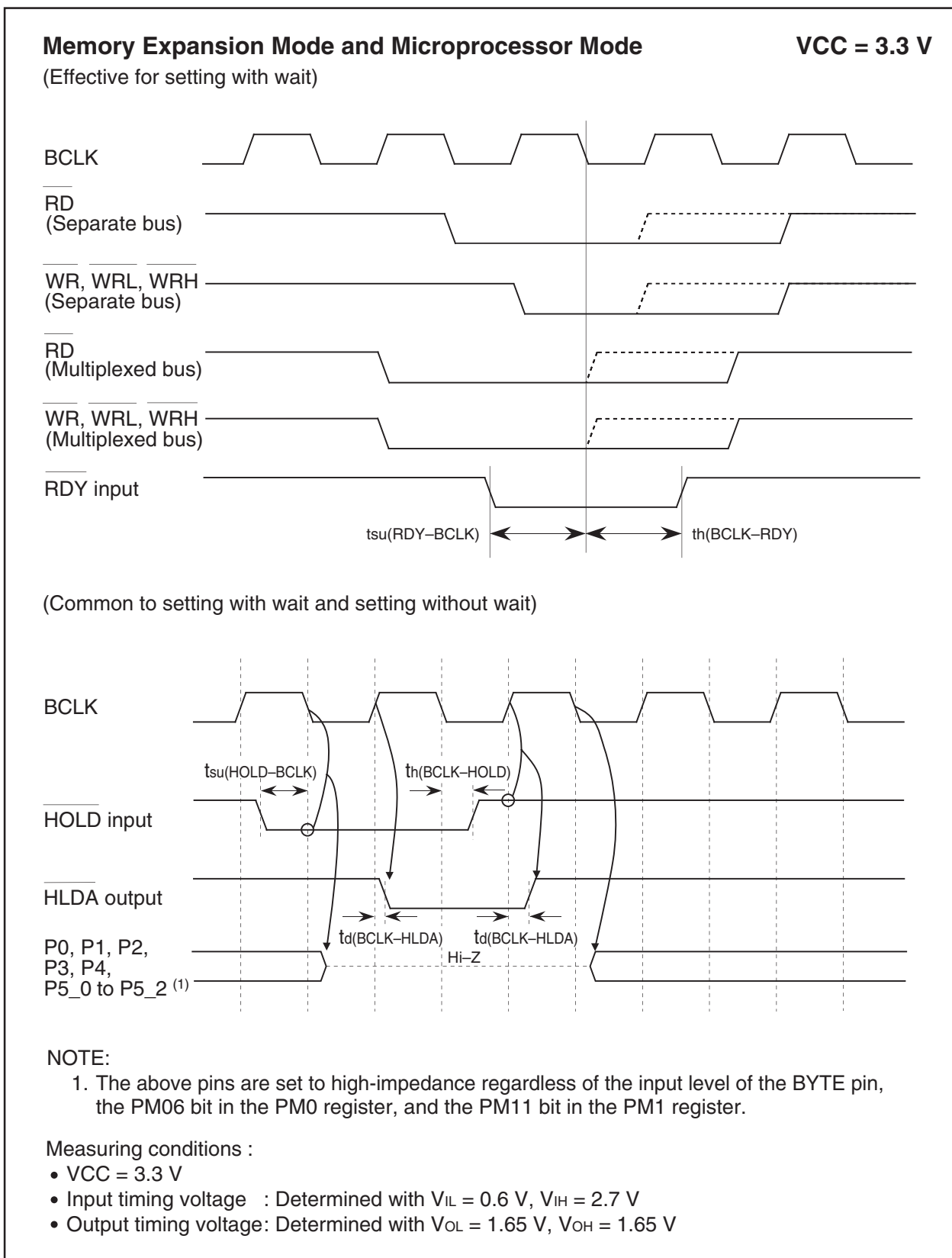


Figure 22.25 Timing Diagram (2)

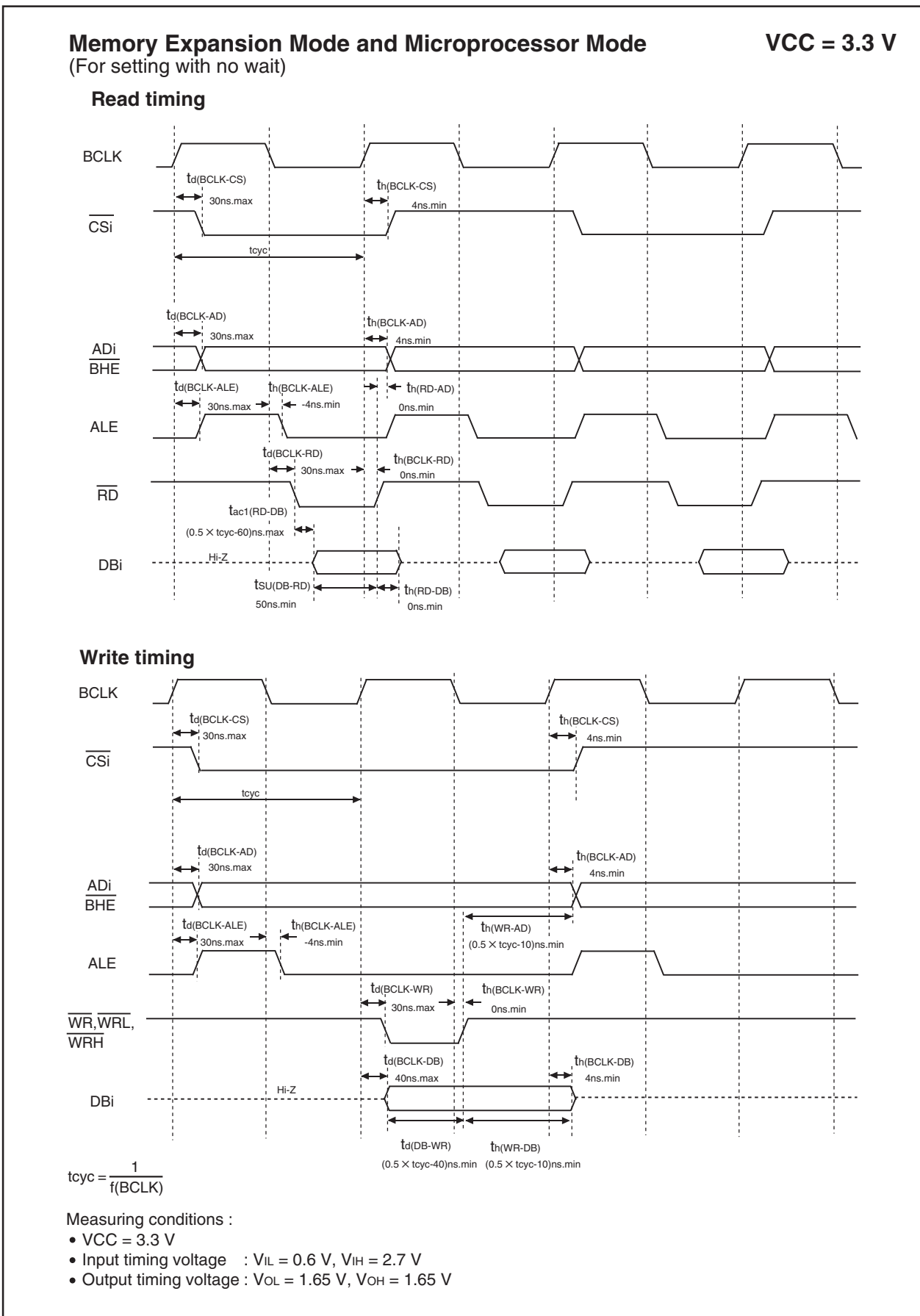


Figure 22.26 Timing Diagram (3)

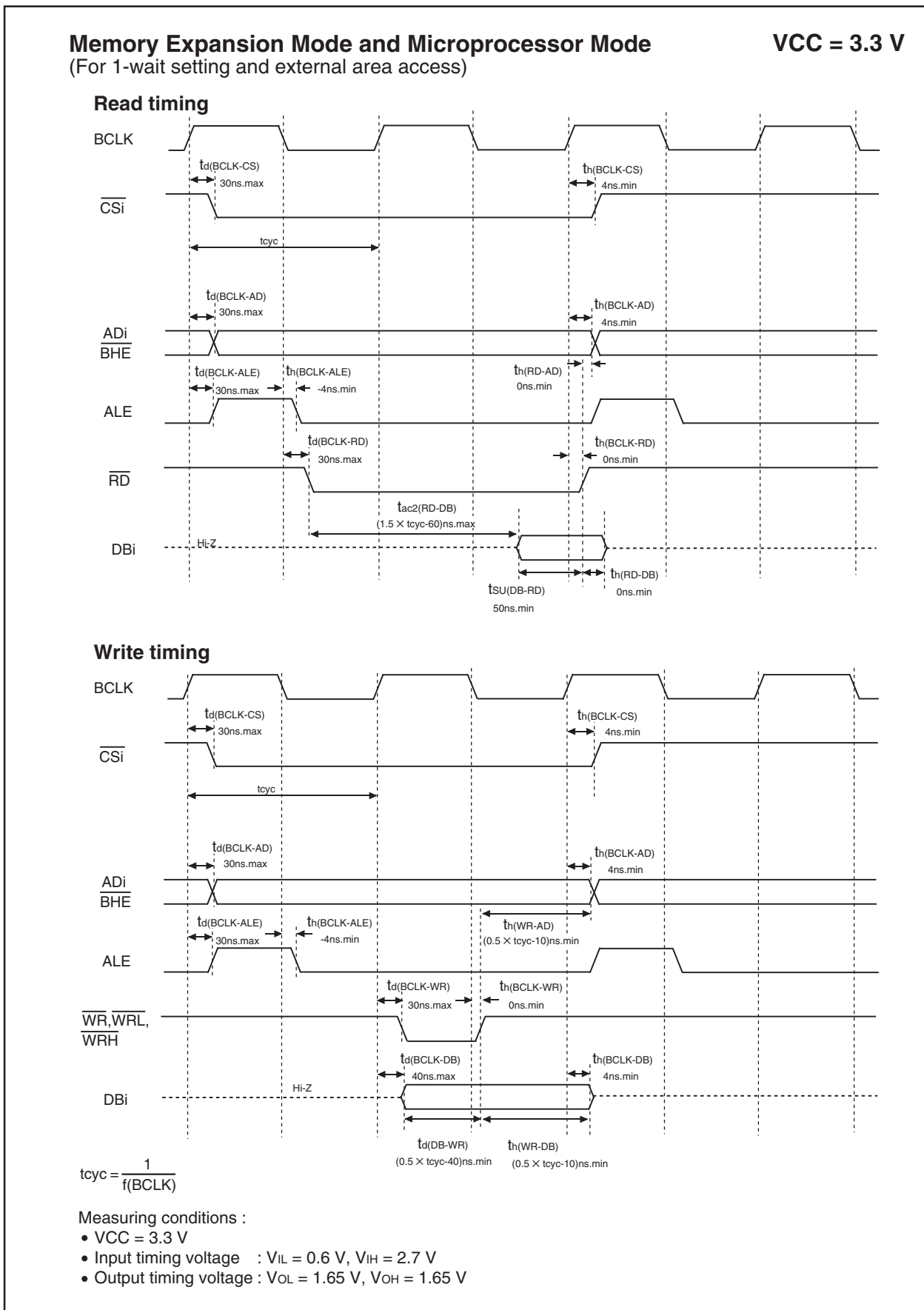


Figure 22.27 Timing Diagram (4)

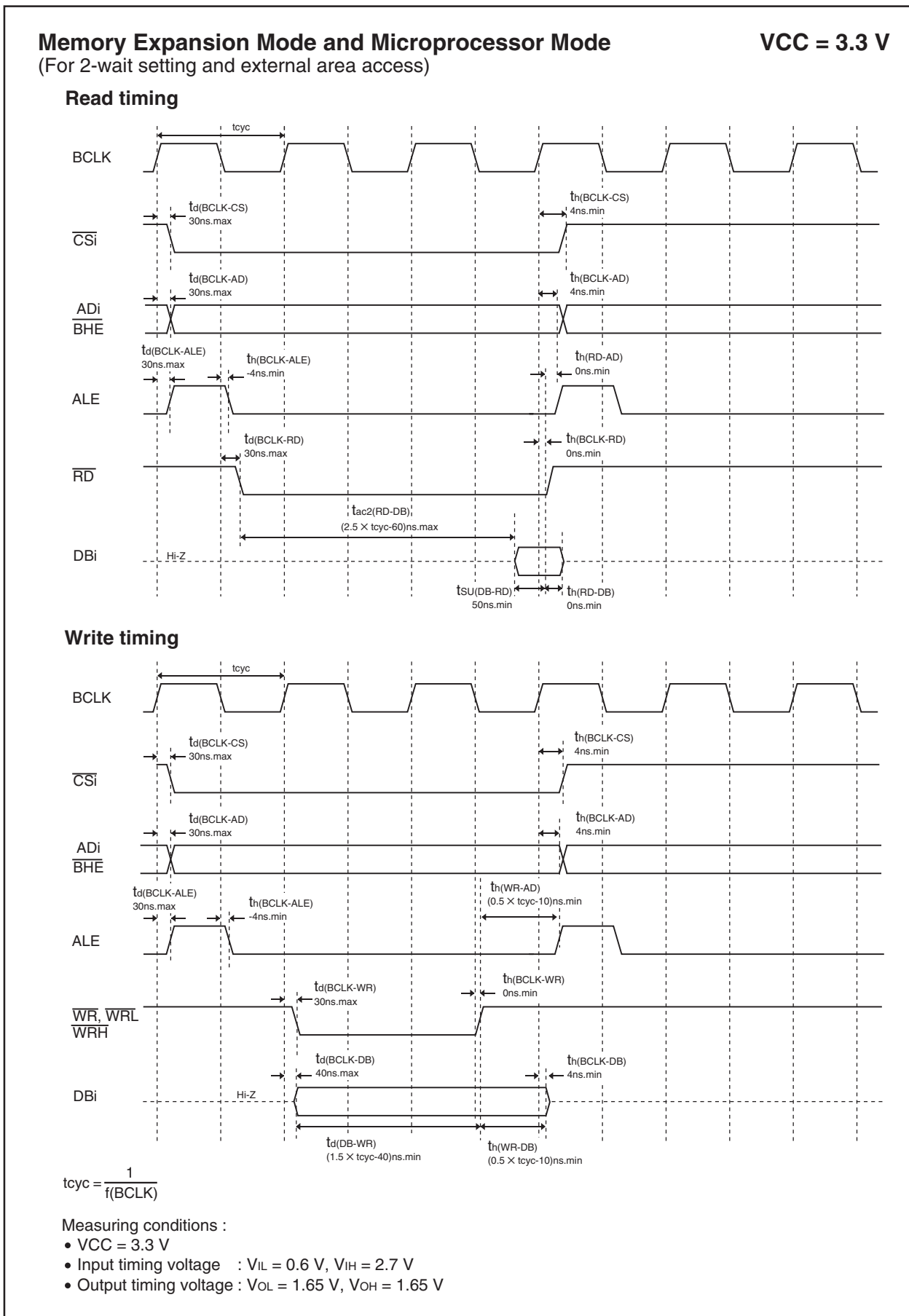


Figure 22.28 Timing Diagram (5)

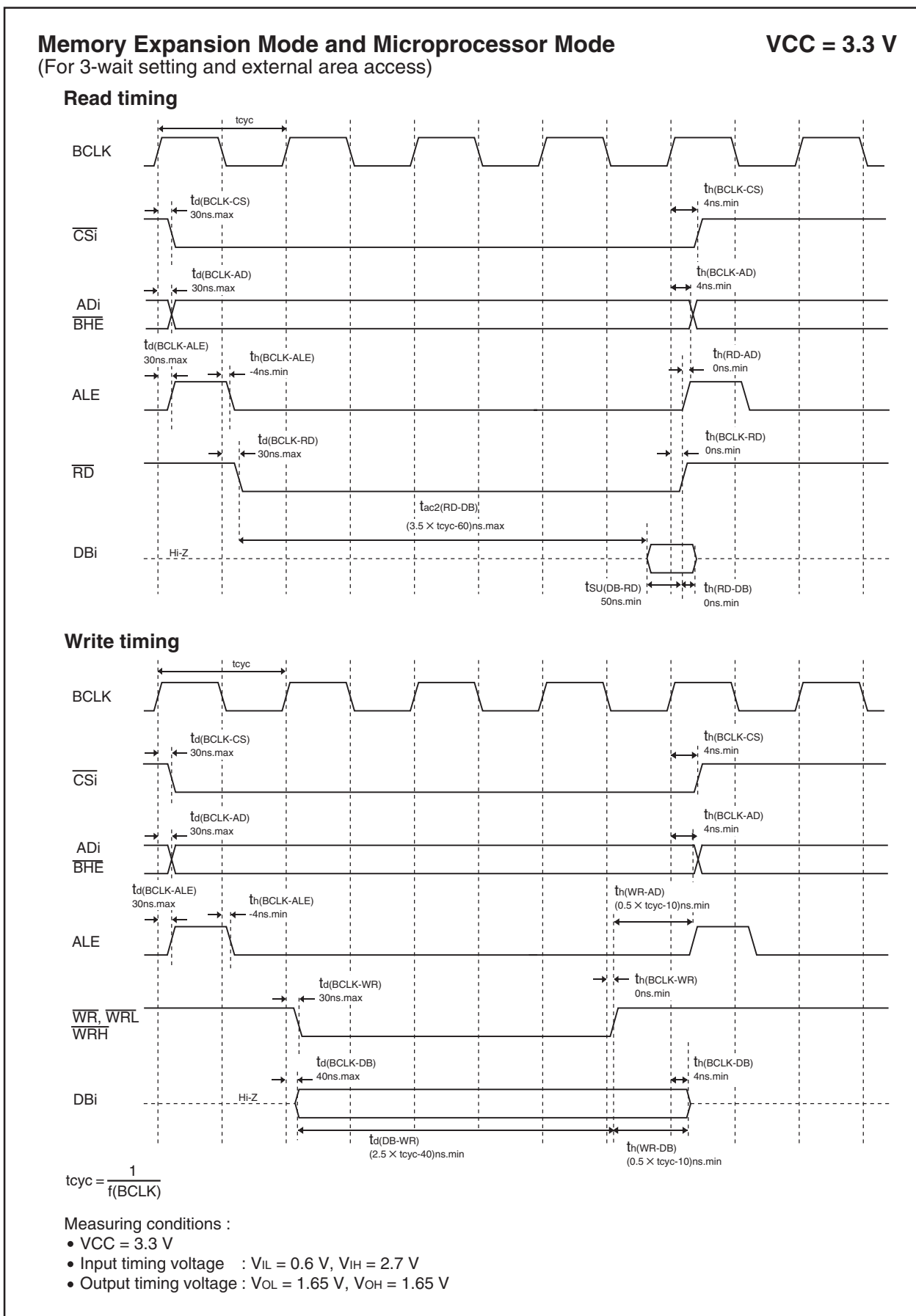


Figure 22.29 Timing Diagram (6)

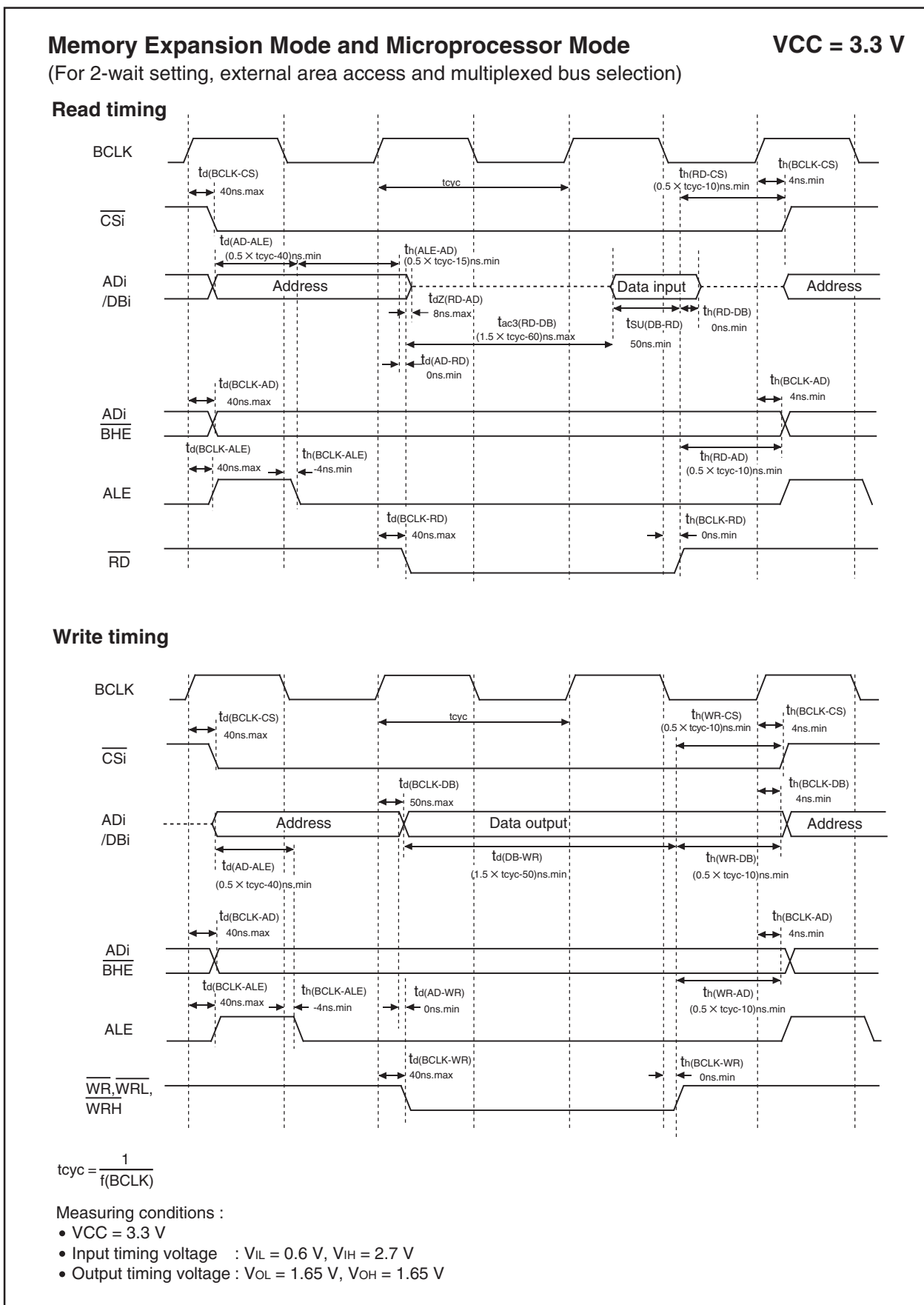


Figure 22.30 Timing Diagram (7)

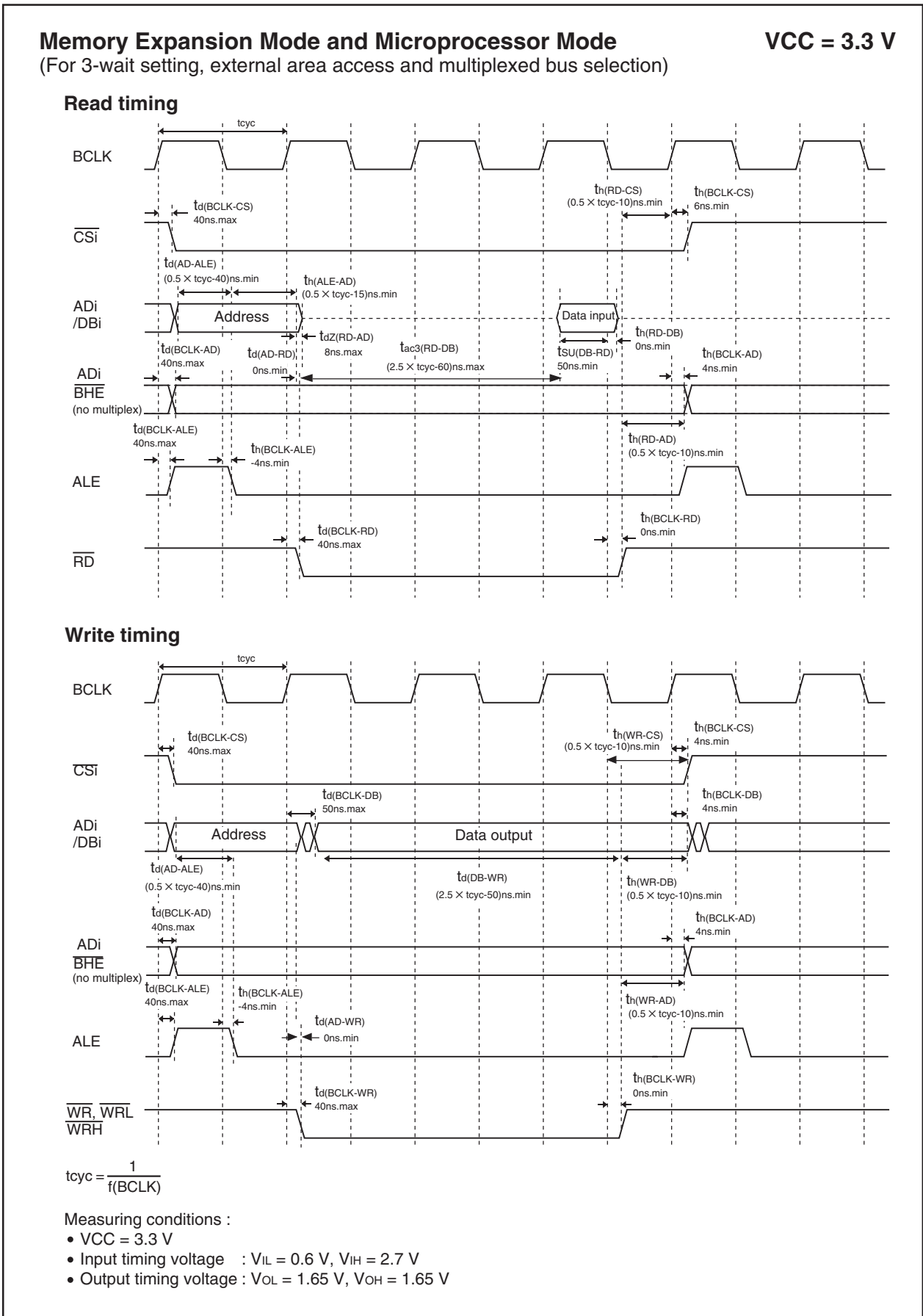


Figure 22.31 Timing Diagram (8)

23. Usage Notes

23.1 SFRs

There are the SFRs with write-only bits which can only be written to. Set these registers with undefined values. When establishing the next value by altering the present value, write the present value to the RAM as well as to the register. Transfer the next value to the register after making changes in the RAM.

Table 23.1 lists Registers with Write-only Bits.

Table 23.1 Registers with Write-only Bits

| Register Name | Symbol | Address |
|---|--------|--------------|
| Watchdog Timer Start Register | WDTS | 000Eh |
| Timer A1-1 Register | TA11 | 01C3h, 01C2h |
| Timer A2-1 Register | TA21 | 01C5h, 01C4h |
| Timer A4-1 Register | TA41 | 01C7h, 01C6h |
| Dead Time Timer | DTT | 01CCh |
| Timer B2 Interrupt Generation Frequency Set Counter | ICTB2 | 01CDh |
| SI/O3 Bit Rate Register | S3BRG | 01E3h |
| UART2 Bit Rate Register | U2BRG | 01F9h |
| UART2 Transmit Buffer Register | U2TB | 01FBh, 01FAh |
| Up-Down Flag | UDF | 0384h |
| Timer A0 Register | TA0 | 0387h, 0386h |
| Timer A1 Register | TA1 | 0389h, 0388h |
| Timer A2 Register | TA2 | 038Bh, 038Ah |
| Timer A3 Register | TA3 | 038Dh, 038Ch |
| Timer A4 Register | TA4 | 038Fh, 038Eh |
| UART0 Bit Rate Register | U0BRG | 03A1h |
| UART0 Transmit Buffer Register | U0TB | 03A3h, 03A2h |
| UART1 Bit Rate Register | U1BRG | 03A9h |
| UART1 Transmit Buffer Register | U1TB | 03ABh, 03AAh |

23.2 External Bus

When resetting CNVSS pin with “H” input, contents of internal ROM cannot be read out.

23.3 PLL Frequency Synthesizer

Stabilize supply voltage so that the standard of the power supply ripple is met. (Refer to **22. Electrical Characteristics.**)

23.4 Power Control

- When exiting stop mode by hardware reset, set $\overline{\text{RESET}}$ pin to “L” until a main clock oscillation is stabilized.
- Set the MR0 bit in the TAIMR register ($i = 0$ to 4) to 0 (pulse is not output) to use the timer A to exit stop mode.
- When entering wait mode, insert a JMP.B instruction before a WAIT instruction. Do not execute 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 roadstead 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

```

- When entering stop mode, describe as follows.
 - (1) To use the BSET instruction for entering stop mode:
Write the BSET instruction (BSET bit, base:16) as described below.
When entering stop mode, DMA transfer must be disabled.

```

                    BSET   0,CM1      ; Stop mode setting [bit, base:16]
                    JMP.B  L1          ;
L1:
                    NOP     ; Countermeasure to avoid the program from
                    NOP     ; stopping by reading instruction ahead
                    NOP     ; (insert 4 or more NOPs)
                    NOP     ;

```

- (2) To use the MOV instruction for entering stop mode:
Write the MOV instruction (MOV.B #IMM8, abs16) as described below.
When entering stop mode, DMA transfer must be disabled.
Change the **src** value (marked as “#21”), depending on your usage condition.

```

                    MOV.B  #21H,CM1   ; Stop mode setting [#IMM8, abs16]
                    JMP.B  L1          ;
L1:
                    NOP     ; Countermeasure to avoid the program from
                    NOP     ; stopping by reading instruction ahead
                    NOP     ; (insert 4 or more NOPs)
                    NOP     ;

```

- When entering medium-speed mode after transferring to stop mode from low-speed mode and low power dissipation mode, write the MOV instruction (MOV.W #IMM16, abs16) as described below. When entering stop mode and exiting from stop mode, DMA transfer must be disabled. Change the **src** value (marked as “#2118”) depending on your usage condition.

```

MOV.W #2118H,CM0 ; Stop mode setting [#IMM16, abs16]
JMP.S L1 ;
L1:
NOP ; Countermeasure to avoid the program from
NOP ; stopping by reading instruction ahead
NOP ; (insert 4 or more NOPs)
NOP ;

```

- Wait until the main clock oscillation stabilizes, before switching the clock source for CPU clock to the main clock. Similarly, wait until the sub clock oscillation stabilizes, before switching the clock source for CPU clock to the sub clock.
- Suggestions to reduce power consumption.

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 pass current flows in input ports that high-impedance state. When entering wait mode or stop mode, set non-used ports to input and stabilize the potential.

A/D converter

When A/D conversion is not performed, set the VCUT bit in the ADCON1 register to 0 (VREF not 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).

D/A converter

When not performing D/A conversion, set the DAiE bit ($i = 0, 1$) in the DACON register to 0 (input disabled) and DAi register to 00h.

Stopping peripheral functions

Use the CM02 bit in the CM0 register 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.

Switching the oscillation-driving capacity

Set the driving capacity to “LOW” when oscillation is stable.

23.5 Protection

Set the PRC2 bit in the PRCR register to 1 (write enabled) and then write to given address, and the PRC2 bit will be set 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 no DMA transfers will occur between the instruction in which the PRC2 bit is set to 1 and the next instruction.

23.6 Interrupts

23.6.1 Reading Address 00000h

Do not read the address 00000h 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 00000h during the interrupt sequence. At this time, the IR bit for the accepted interrupt is set to 0.

If the address 00000h is read in a program, the IR bit for the interrupt which has the highest priority among the enabled interrupts is set to 0. This causes a problem that the interrupt is canceled, or an unexpected interrupt request is generated.

23.6.2 Setting SP

Set any value in the SP (USP, ISP) before accepting an interrupt. The SP (USP, ISP) is set to 0000h after reset. Therefore, if an interrupt is accepted before setting any value in the SP (USP, ISP), the program may go out of control.

Especially when using $\overline{\text{NMI}}$ interrupt, set a value in the ISP at the beginning of the program. For the first and only the first instruction after reset, all interrupts including $\overline{\text{NMI}}$ interrupt are disabled.

23.6.3 $\overline{\text{NMI}}$ Interrupt

- The $\overline{\text{NMI}}$ interrupt cannot be disabled. If this interrupt is unused, connect the $\overline{\text{NMI}}$ pin to VCC via a resistor (pull-up).
- The input level of the $\overline{\text{NMI}}$ pin can be read by accessing the P8_5 bit in the P8 register. Note that the P8_5 bit can only be read when determining the pin level in $\overline{\text{NMI}}$ interrupt routine.
- 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 CM10 bit in the CM1 register is fixed to 0.
- 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.
- 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.

23.6.4 Changing Interrupt Source

If the interrupt source 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 source for an interrupt that needs to be used, be sure to set the IR bit for that interrupt to 0 (interrupt not requested).

Changing the interrupt source 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 source, polarity or timing of an interrupt, be sure to set 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 23.1 shows the Procedure for Changing Interrupt Source.

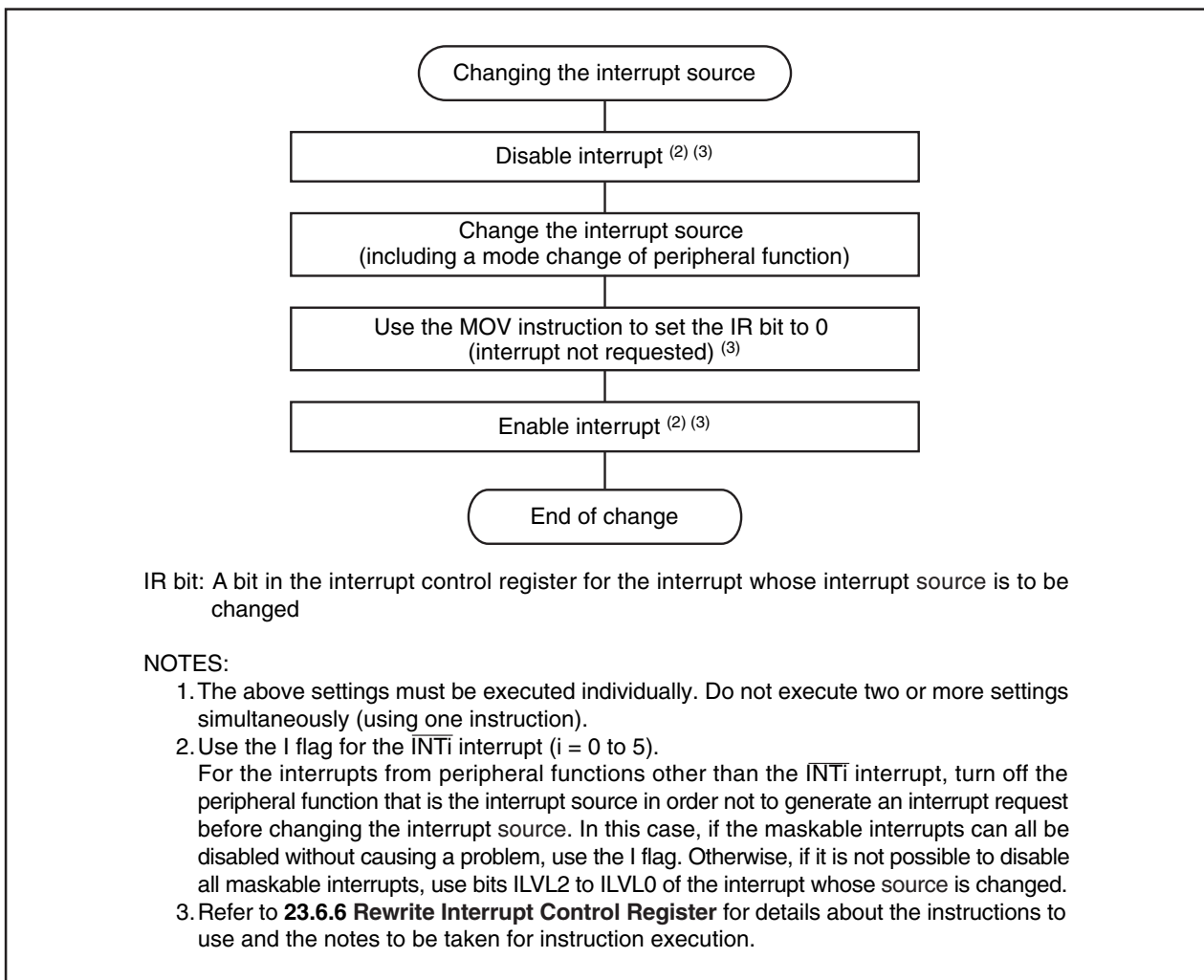


Figure 23.1 Procedure for Changing Interrupt Source

23.6.5 $\overline{\text{INT}}$ Interrupt

- Either an “L” level of at least $tW(\text{INH})$ or an “H” level of at least $tW(\text{INL})$ width is necessary for the signal input to pins $\overline{\text{INT}}_0$ to $\overline{\text{INT}}_5$ regardless of the CPU operation clock.
- If the POL bit in registers INT0IC to INT5IC or bits IFSR10 to IFSR17 in the IFSR1 register are changed, the IR bit may inadvertently set to 1 (interrupt requested). Be sure to set the IR bit to 0 (interrupt not requested) after changing any of those register bits.

23.6.6 Rewrite Interrupt Control Register

- (a) The interrupt control register for any interrupt should be modified in places where no requests for that interrupt may be generated. Otherwise, disable the interrupt before rewriting the interrupt control register.
- (b) To rewrite the interrupt control register for any interrupt after disabling that interrupt, care must be taken when selecting the instructions.

Changing any bit other than IR bit

If while executing an instruction, an interrupt request controlled by the register being modified is generated, the IR bit of 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 IR bit

Depending on the instruction used, the IR bit may not always be set to 0 (interrupt not requested). Therefore, be sure to use the MOV instruction to set the IR bit to 0.

- (c) 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 (b) 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 (interrupt enabled) before the interrupt control register is rewritten, owing to the effects of 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 00h.
  NOP
  NOP
  FSET   I           ; Enable interrupts.
```

The number of the NOP instruction is as follows.

- The PM20 bit in the PM2 register = 1 (1 wait) : 2
- The PM20 bit = 0 (2 waits) : 3
- When using HOLD function : 4

Example 2: Using the dummy read to the FSET instruction delay

```
INT_SWITCH2:
  FCLR   I           ; Disable interrupts.
  AND.B  #00h, 0055h ; Set the TA0IC register to 00h.
  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 00h.
  POPC   FLG        ; Enable interrupts.
```

23.6.7 Watchdog Timer Interrupt

Initialize the watchdog timer after the watchdog timer interrupt request is generated.

23.7 DMAC

23.7.1 Write to DMAE Bit in DMiCON Register (i = 0, 1)

When both of the conditions below are met, follow the steps below.

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.

Step 1: Write 1 to the DMAE bit and DMAS bit in the DMiCON register simultaneously ⁽¹⁾.

Step 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 while the instruction is being executing.

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.

23.8 Timers

23.8.1 Timer A

23.8.1.1 Timer A (Timer Mode)

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.

While counting is in progress, the counter value can be read out at any time by reading the TAI register. However, if the counter is read at the same time it is reloaded, the value FFFFh is read. Also, if the counter is read before it starts counting after a value is set in the TAI register while not counting, the set value is read.

If a low-level signal is applied to the $\overline{\text{NMI}}$ pin when the IVPCR1 bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on $\overline{\text{NMI}}$ pin enabled), pins TA1OUT, TA2OUT, and TA4OUT go to a high-impedance state.

23.8.1.2 Timer A (Event Counter Mode)

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 the TAI_iMR register, the UDF register, bits TAZIE, TA0TGL, and TA0TGH, and the TRGSR register are modified while the TAI_S bit remains 0 (count stops) regardless whether after reset or not.

While counting is in progress, the counter value can be read out at any time by reading the TAI register. However, FFFFh can be read in underflow, while reloading, and 0000h in overflow. When setting the TAI register to a value during a counter stop, the setting value can be read before a counter starts counting. Also, if the counter is read before it starts counting after a value is set in the TAI register while not counting, the set value is read.

If a low-level signal is applied to the $\overline{\text{NMI}}$ pin when the IVPCR1 bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on $\overline{\text{NMI}}$ pin enabled), pins TA1OUT, TA2OUT, and TA4OUT go to a high-impedance state.

23.8.1.3 Timer A (One-shot Timer Mode)

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 the TAI_{MR} register, bits TA0TGL and TA0TGH, and the TRGSR register are modified while the TAI_S bit remains 0 (count stops) regardless whether after reset or not.

When setting the TAI_S bit to 0 (count stops), 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 the TAI_{IC} register is set to 1 (interrupt request).

Output in one-shot timer mode synchronizes with a count source internally generated. When an external trigger has been selected, one-cycle delay of a count source as maximum occurs between a trigger input to TAI_{IN} pin and output in one-shot timer mode.

The IR bit is set to 1 when timer operating mode is set with any of the following procedures:

- Select one-shot timer mode after reset.
- Change an operating mode from timer mode to one-shot timer mode.
- Change an operating mode from event counter mode to one-shot timer mode.

To use the timer Ai interrupt (the IR bit), set the IR bit to 0 after the changes listed above have been made.

When a trigger occurs, while counting, a counter reloads the reload register to continue counting after generating a re-trigger and counting down once. To generate a trigger while counting, generate a second trigger between occurring the previous trigger and operating longer than one cycle of a timer count source.

If a low-level signal is applied to the $\overline{\text{NMI}}$ pin when the IVPCR1 bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on $\overline{\text{NMI}}$ pin enabled), pins TA1_{OUT}, TA2_{OUT}, and TA4_{OUT} go to a high-impedance state.

23.8.1.4 Timer A (Pulse Width Modulation (PWM) Mode)

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 the TAI_{MR} register, bits TA0TGL and TA0TGH, and the TRGSR register are modified while the TAI_S bit remains 0 (count stops) regardless whether after reset or not.

The IR bit is set to 1 when setting a timer operating mode with any of the following procedures:

- Select the PWM mode after reset.
- Change an operating mode from timer mode to PWM mode.
- Change an operating mode from event counter mode to PWM mode.

To use the timer Ai interrupt (the IR bit), set the IR bit to 0 by program after the above listed changes have been made.

When setting TAI_S bit to 0 (count stops) during PWM pulse output, the following action occurs:

- Stop counting.
- When TAI_{OUT} pin is output "H", output level is set to "L" and the IR bit is set to 1.
- When TAI_{OUT} pin is output "L", both output level and the IR bit remain unchanged.

If a low-level signal is applied to the $\overline{\text{NMI}}$ pin when the IVPCR1 bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on $\overline{\text{NMI}}$ pin enabled), pins TA1_{OUT}, TA2_{OUT}, and TA4_{OUT} go to a high-impedance state.

23.8.2 Timer B

23.8.2.1 Timer B (Timer Mode)

The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TBiMR (i = 0 to 5) register and TBi register before setting the TBiS bit ⁽¹⁾ 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.

NOTE:

1. Bits TB0S to TB2S are the bits 5 to 7 in the TABSR register, bits TB3S to TB5S are the bits 5 to 7 in the TBSR register.

A value of a counter, while counting, can be read in the TBi register at any time. FFFFh is read while reloading. Setting value is read between setting values in the TBi register at count stop and starting a counter.

23.8.2.2 Timer B (Event Counter Mode)

The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TBiMR (i = 0 to 5) register and TBi 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.

The counter value can be read out on-the-fly 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 FFFFh. If the TBi register is read after setting a value in it while not counting but before the counter starts counting, the read value is the one that has been set in the register.

23.8.2.3 Timer B (Pulse Period/pulse Width Measurement Mode)

The timer remains idle after reset. Set the mode, count source, etc. using the TBiMR (i = 0 to 5) register before setting the TBiS bit in the TABSR or 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 set the MR3 bit to 0 by writing to the TBiMR register while the TBiS bit = 1 (count starts), be sure to write the same value as previously written to bits TM0D0, TM0D1, MR0, MR1, TCK0, and TCK1 and, a 0 to the MR2 bit.

The IR bit in the TBiIC register goes to 1 (interrupt request), when an effective edge of a measurement pulse is input or timer Bi is overflowed. The interrupt source can be determined by use of the MR3 bit in the TBiMR register within the interrupt routine.

If the interrupt source 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.

To set the MR3 bit to 0 (no overflow), set the TBiMR register with setting the TBiS bit to 1 and counting the next count source after setting the MR3 bit to 1 (overflow).

Use the IR bit in the TBiIC register to detect only overflows. Use the MR3 bit only to determine the interrupt source.

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.

A value of the counter is undefined at the beginning of a count. The MR3 bit may be set to 1 and timer Bi interrupt request may be generated between a count start and an effective edge input.

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.

23.9 Serial Interface

23.9.1 Clock Synchronous Serial I/O Mode

23.9.1.1 Transmission/reception

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 $\overline{\text{CTS}}$ 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.

If a low-level signal is applied to the $\overline{\text{NMI}}$ pin when the IVPCR1 bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on $\overline{\text{NMI}}$ pin enabled), pins $\overline{\text{RTS2}}$ and CLK2 go to a high-impedance state.

23.9.1.2 Transmission

When an external clock is selected, the conditions must be met while if the CKPOL bit in the UiC0 register = 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 = 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 the UiC1 register = 1 (transmission enabled)
- The TI bit in the UiC1 register = 0 (data present in UiTB register)
- If CTS function is selected, input on the $\overline{\text{CTS}}$ pin = L

23.9.1.3 Reception

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 (i = 0 to 2) pin when receiving data.

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 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.

When successively receiving data, if all bits of the next receive data are prepared in the UARTi receive register while the RI bit in the UiC1 register = 1 (data present in the UiRB register), an overrun error occurs and the OER bit in the UiRB register 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 IR bit in the SiRIC register does not change state.

To receive data in succession, set dummy data in the lower-order byte of the UiTB register every time reception is made.

When an external clock is selected, the conditions must be met while if the CKPOL bit = 0, the external clock is in the high state; if the CKPOL bit = 1, the external clock is in the low state.

- The RE bit in the UiC1 register = 1 (reception enabled)
- The TE bit in the UiC1 register = 1 (transmission enabled)
- The TI bit in the UiC1 register = 0 (data present in the UiTB register)

23.9.2 Special Modes

23.9.2.1 Special Mode 1 (I²C Mode)

When generating start, stop and restart conditions, set the STSPSEL bit in the UiSMR4 register to 0 (start and stop conditions not output) and wait for more than half cycle of the transfer clock before setting each condition generate bit (bits STAREQ, RSTAREQ, and STPREQ) from 0 (clear) to 1 (start).

23.9.2.2 Special Mode 2

If a low-level signal is applied to the $\overline{\text{NMI}}$ pin when the IVPCR1 bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on $\overline{\text{NMI}}$ pin enabled), pins $\overline{\text{RTS2}}$ and $\overline{\text{CLK2}}$ go to a high-impedance state.

23.9.2.3 Special Mode 4 (SIM Mode)

A transmit interrupt request is generated by setting the U2IRS bit in the U2C1 register to 1 (transmission completed) and U2ERE bit in the U2C1 register to 1 (error signal output) after reset. Therefore, when using SIM mode, be sure to set the IR bit to 0 (no interrupt request) after setting these bits.

23.9.3 SI/O3

The SOUT3 default value which is set to the SOUT3 pin by the SM37 in the S3C register bit approximately 10 ns may be output when changing the SM33 bit in the S3C register from 0 (I/O port) to 1 (SOUT3 output and CLK3 function) while the SM32 bit in the S3C register to 0 (SOUT3 output) and the SM36 bit is set to 1 (internal clock). And then the SOUT3 pin is held high-impedance.

If the level which is output from the SOUT3 pin is a problem when changing the SM33 bit from 0 to 1, set the default value of the SOUT3 pin by the SM37 bit.

23.10 A/D Converter

Set the ADCON0 (except bit 6), registers ADCON1 and ADCON2 when A/D conversion is stopped (before a trigger occurs). After stopping A/D conversion, the VCUT bit in the ADCON1 register is changed from 1 (VREF connected) to 0 (VREF not connected),

When the VCUT bit is changed from 0 to 1, start A/D conversion after passing 1 μ s or longer.

To prevent noise-induced device malfunction or latch-up, as well as to reduce conversion errors, insert capacitors between the AVCC, VREF, and analog input pins (AN_i (i = 0 to 7), AN0_i, and AN2_i) each and the AVSS pin. Similarly, insert a capacitor between the VCC pin and the VSS pin.

Figure 23.2 shows the Use of Capacitors to Reduce Noise.

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 = 1 (external trigger), make sure the port direction bit for the ADTRG pin is set to 0 (input mode).

When using key input interrupt, do not use any of four pins AN4 to AN7 as analog inputs. (A key input interrupt request is generated when the A/D input voltage goes low.)

The ϕ AD frequency must be 10 MHz or less. Without sample and hold, limit the ϕ AD frequency to 250 kHz or more. With the sample and hold, limit the ϕ AD frequency to 1 MHz or more.

When changing an A/D operating mode, select analog input pin again in bits CH2 to CH0 in the ADCON0 register and bits SCAN1 to SCAN0 in the ADCON1 register.

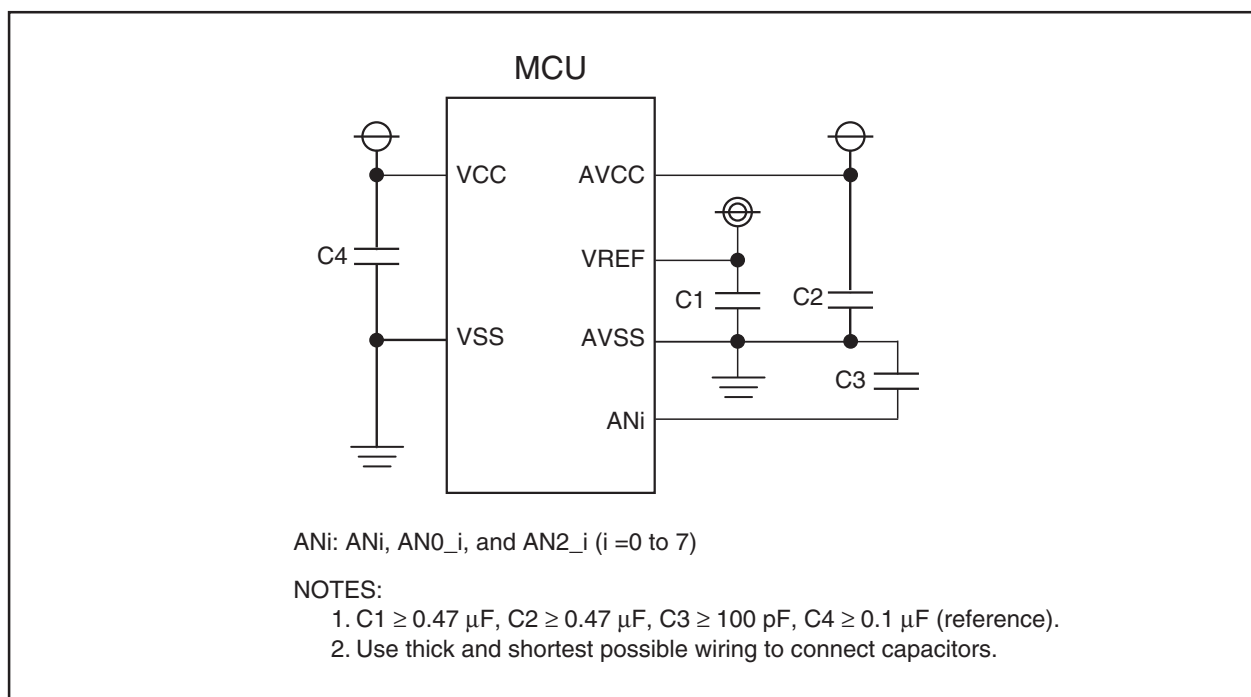


Figure 23.2 Use of Capacitors to Reduce Noise

If the CPU reads the ADi register (i = 0, 1) 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 sub clock is selected for CPU clock.

- When operating in one-shot or single-sweep mode

Check to see that A/D conversion is completed before reading the target ADi register. (Check the IR bit in the ADIC register 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.

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 register irrelevant to A/D conversion may also become undefined. If while A/D conversion is underway the ADST bit is set to 0 in a program, ignore the values of all ADi registers.

When setting the ADST bit to 0 in single sweep mode during A/D conversion and A/D conversion is aborted, disable the interrupt before setting the ADST bit to 0.

The applied intermediate potential may cause more increase in power consumption than other analog input pins (AN0 to AN3, AN0_0 to AN0_7, and AN2_0 to AN2_7), since the AN4 to AN7 are used with the K10 to K13.

23.11 CAN Module

23.11.1 Reading CiSTR Register (i = 0, 1)

The CAN module on the M16C/6N Group (M16C/6N4) updates the status of the CiSTR register in a certain period. When the CPU and the CAN module access to the CiSTR 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 23.3 When Updating Period of CAN Module Matches Access Period from CPU.**)

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:

- (a) There should be a wait time of 3fCAN or longer (see **Table 23.2 CAN Module Status Updating Period**) before the CPU reads the CiSTR register. (See **Figure 23.4 With Wait Time of 3 fCAN before CPU Read.**)
- (b) When the CPU polls the CiSTR register, the polling period must be 3 fCAN or longer. (See **Figure 23.5 When Polling Period of CPU is 3 fCAN or Longer.**)

Table 23.2 CAN Module Status Updating Period

| 3fCAN Period = 3 × XIN (Original Oscillation Period) × Division Value of CAN Clock (CCLK) | |
|---|--|
| (Example 1) Condition XIN 16 MHz CCLK: Divide-by-1 | 3 fCAN period = 3 × 62.5 ns × 1 = 187.5 ns |
| (Example 2) Condition XIN 16 MHz CCLK: Divide-by-2 | 3 fCAN period = 3 × 62.5 ns × 2 = 375 ns |
| (Example 3) Condition XIN 16 MHz CCLK: Divide-by-4 | 3 fCAN period = 3 × 62.5 ns × 4 = 750 ns |
| (Example 4) Condition XIN 16 MHz CCLK: Divide-by-8 | 3 fCAN period = 3 × 62.5 ns × 8 = 1.5 μs |
| (Example 5) Condition XIN 16 MHz CCLK: Divide-by-16 | 3 fCAN period = 3 × 62.5 ns × 16 = 3 μs |

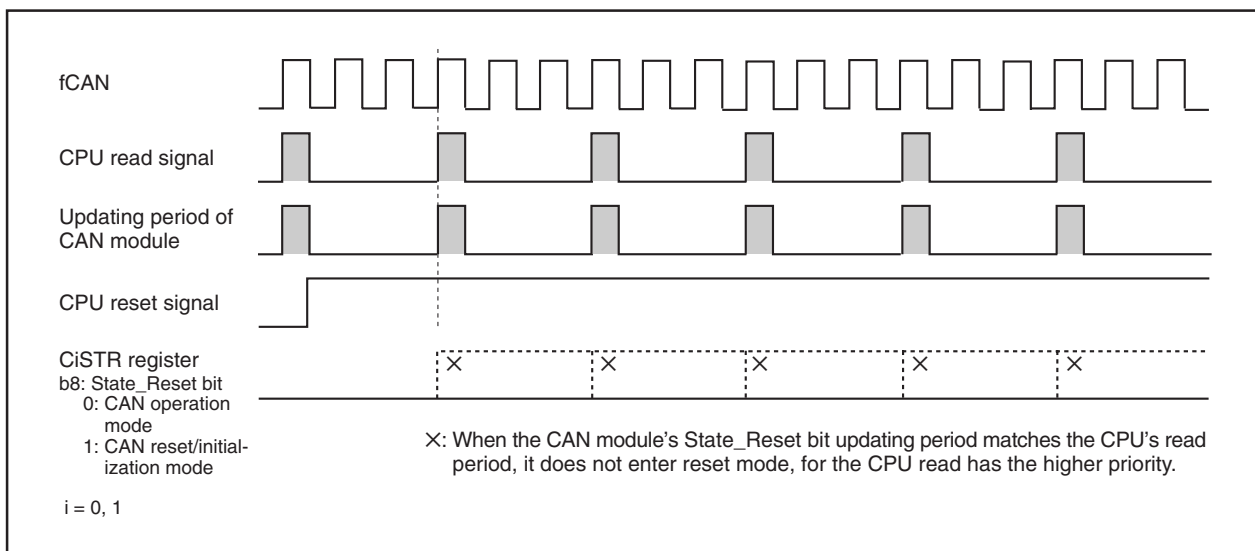


Figure 23.3 When Updating Period of CAN Module Matches Access Period from CPU

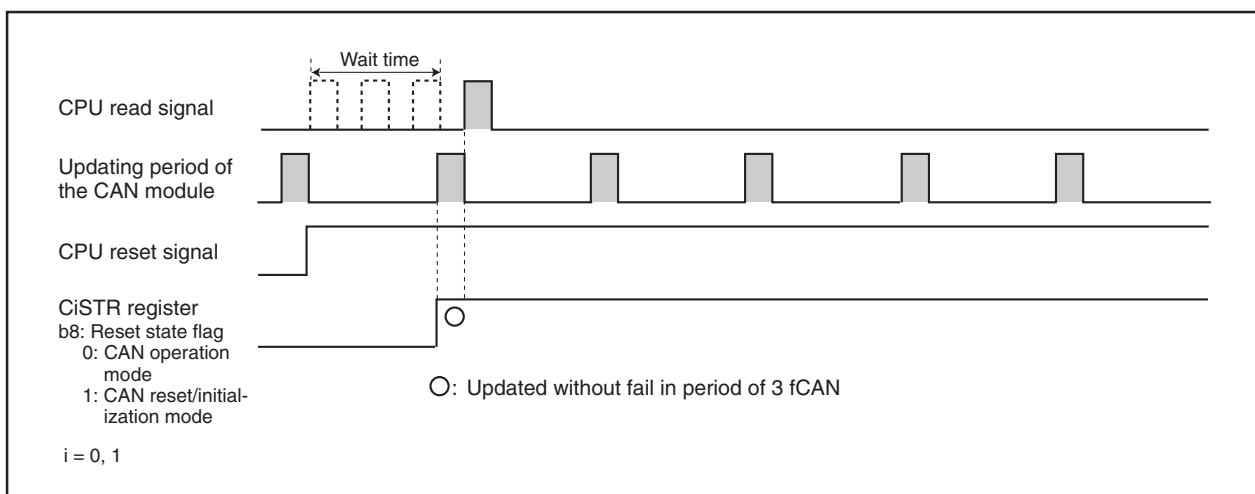


Figure 23.4 With Wait Time of 3 fCAN before CPU Read

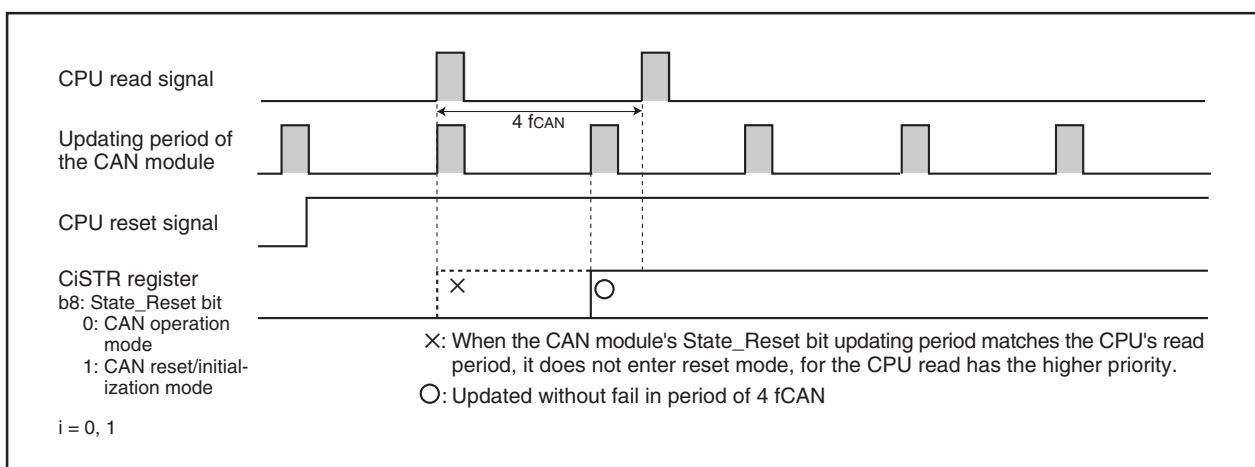


Figure 23.5 When Polling Period of CPU is 3 fCAN or Longer

23.11.2 Performing CAN Configuration

If the Reset bit in the CiCTRL register ($i = 0, 1$) is changed from 0 (operation mode) to 1 (reset/initialization mode) in order to place the CAN module from CAN operation mode into CAN reset/initialization mode, always be sure to check that the State_Reset bit in the CiSTR register is set to 1 (reset mode).

Similarly, if the Reset bit is changed from 1 to 0 in order to place the CAN module from CAN reset/initialization mode into CAN operation mode, always be sure to check that the State_Reset bit is set to 0 (operation mode).

The procedure is described below.

To Place CAN Module from CAN Operation Mode into CAN Reset/Initialization Mode

- Change the Reset bit from 0 to 1
- Check that the State_Reset bit is set to 1

To Place CAN Module from CAN Reset/Initialization Mode into CAN Operation Mode

- Change the Reset bit from 1 to 0
- Check that the State_Reset bit is set to 0

23.11.3 Suggestions to Reduce Power Consumption

When not performing CAN communication, the operation mode of CAN transceiver should be set to “standby mode” or “sleep mode”.

When performing CAN communication, the power consumption in CAN transceiver in not performing CAN communication can be substantially reduced by controlling the operation mode pins of CAN transceiver.

Tables 23.3 and 23.4 show the Recommended Pin Connections.

Table 23.3 Recommended Pin Connections (In case of PCA82C250: Philips product)

| | Standby Mode | High-speed Mode |
|---|-----------------------|-----------------|
| Rs pin ⁽¹⁾ | “H” | “L” |
| Power consumption in CAN transceiver ⁽²⁾ | less than 170 μ A | less than 70 mA |
| CAN communication | impossible | possible |
| Connection | | |

i = 0, 1

NOTES:

1. The pin which controls the operation mode of CAN transceiver.
2. In case of Ta = 25 °C
3. Connect to enabled port to control CAN transceiver.

Table 23.4 Recommended Pin Connections (In case of PCA82C252: Philips product)

| | Sleep Mode | Normal Operation Mode |
|---|----------------------|-----------------------|
| STB pin ⁽¹⁾ | “L” | “H” |
| EN pin ⁽¹⁾ | “L” | “H” |
| Power consumption in CAN transceiver ⁽²⁾ | less than 50 μ A | less than 35 mA |
| CAN communication | impossible | possible |
| Connection | | |

i = 0, 1

NOTES:

1. The pin which controls the operation mode of CAN transceiver.
2. Ta = 25 °C
3. Connect to enabled port to control CAN transceiver.

23.11.4 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 23.5 and 23.6 show the Pin Connections of CAN Transceiver.

Table 23.5 Pin Connections of CAN Transceiver (In case of PCA82C250: Philips product)

| | Standby Mode | High-speed Mode |
|-----------------------|--------------|-----------------|
| Rs pin ⁽¹⁾ | “H” | “L” |
| CAN communication | impossible | possible |
| Connection | | |

i = 0, 1

NOTES:

1. The pin which controls the operation mode of CAN transceiver.
2. Connect to enabled port to control CAN transceiver.

Table 23.6 Pin Connections of CAN Transceiver (In case of PCA82C252: Philips product)

| | Sleep Mode | Normal Operation Mode |
|------------------------|------------|-----------------------|
| STB pin ⁽¹⁾ | “L” | “H” |
| EN pin ⁽¹⁾ | “L” | “H” |
| CAN communication | impossible | possible |
| Connection | | |

i = 0, 1

NOTES:

1. The pin which controls the operation mode of CAN transceiver.
2. Connect to enabled port to control CAN transceiver.

23.12 Programmable I/O Ports

If a low-level signal is applied to the $\overline{\text{NMI}}$ pin when the IVPCR1 bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on $\overline{\text{NMI}}$ pin enabled), pins P7_2 to P7_5, P8_0, and P8_1 go to a high-impedance state.

Setting the SM32 bit in the S3C register to 1 causes the P9_2 pin to go to a high-impedance state.

The input threshold voltage of pins differs between programmable I/O ports and peripheral functions. Therefore, if any pin is shared by a programmable I/O port and a peripheral function and the input level at this pin is outside the range of recommended operating conditions VIH and VIL (neither “high” nor “low”), the input level may be determined differently depending on which side—the programmable I/O port or the peripheral function—is currently selected.

Undefined values are read from bits P3_7 to P3_4, and PD3_7 to PD3_4 by reading registers P3 and PD3 when bits PM01 to PM00 in the PM0 register are set to 01b (memory expansion mode) or 11b (microprocessor mode) and setting the PM11 bit to 1.

Use the MOV instruction when rewriting registers P3 and PD3 (including the case that the size specifier is “.W” and registers P2 and PD2 are rewritten).

When bits PM01 to PM00 are rewritten, “L” is output from pins P3_7 to P3_4 during 0.5 cycles of the BCLK by setting bits PM01 to PM00 in the PM0 register to 01b (memory expansion mode) or 11b (microprocessor mode) from 00b (single-chip mode) after setting the PM11 bit to 1.

23.13 Electrical Characteristic Differences between Mask ROM and Flash Memory Version MCUs

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.

23.14 Mask ROM Version

When using the masked ROM version, write nothing to internal ROM area.

23.15 Flash Memory Version

23.15.1 Functions to Prevent Flash Memory from Rewriting

ID codes are stored in addresses 0FFFFDFh, 0FFFFE3h, 0FFFFEBh, 0FFFFEFh, 0FFFFF3h, 0FFFFF7h, and 0FFFFFBh. If wrong data are written to these addresses, the flash memory cannot be read or written in standard serial I/O mode and CAN I/O mode.

The ROMCP register is mapped in address 0FFFFFFh. 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.

23.15.2 Stop Mode

When entering 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.

23.15.3 Wait Mode

When entering wait mode, set the FMR01 bit in the FMR0 register to 0 (CPU rewrite mode disabled) before executing the WAIT instruction.

23.15.4 Low Power Dissipation Mode and On-Chip Oscillator Low Power Dissipation Mode

If the CM05 bit is set to 1 (main clock stopped), do not execute the following commands:

- Program
- Block erase
- Erase all unlocked blocks
- Lock bit program
- Read lock bit status

23.15.5 Writing Command and Data

Write commands and data to even addresses in the user ROM area.

23.15.6 Program Command

By writing xx40h in the first bus cycle and data to the write address in the second bus cycle, an auto-program operation (data program and verify) will start. 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.

23.15.7 Lock Bit Program Command

By writing xx77h in the first bus cycle and xxD0h to the highest-order even address of a block in the second bus cycle, the lock bit for the specified block is set to 0. The address value specified in the first bus cycle must be the same highest-order even address of a block specified in the second bus cycle.

23.15.8 Operating Speed

Before entering CPU rewrite mode (EW0 or EW1 mode), set the CM11 bit in the CM1 register to 0 (main clock), select 10 MHz or less for CPU clock using the CM06 bit in the CM0 register and bits CM17 to CM16 in the CM1 register. Also, set the PM17 bit in the PM1 register to 1 (with wait state).

23.15.9 Prohibited Instructions

The following instructions cannot be used in EW0 mode because the CPU tries to read data in flash memory: the UND instruction, INTO instruction, JMPS instruction, JSRS instruction, and BRK instruction

23.15.10 Interrupts

EW0 Mode

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 request is generated. Allocate the jump addresses for each interrupt service routines to the fixed vector table. Flash memory rewrite operation is suspended when the $\overline{\text{NMI}}$ or watchdog timer interrupt request is generated. Execute the rewrite program again after exiting the interrupt routine.
- The address match interrupt is not available since the CPU tries to read data in the flash memory.

EW1 Mode

- Do not acknowledge any interrupts with vectors in the relocatable vector table or address match interrupt during auto-programming or auto-erasure.
- Do not use the watchdog timer interrupt.
- The $\overline{\text{NMI}}$ interrupt is available since registers FMR0 and FMR1 are forcibly reset when the interrupt request is generated. Allocate the jump address for the interrupt service routine to the fixed vector table. Flash memory rewrite operation is suspended when the $\overline{\text{NMI}}$ interrupt request is generated. Execute the rewrite program again after exiting the interrupt service routine.

23.15.11 How to Access

To set the FMR01, FMR02, or FMR11 bit to 1, write 1 after first setting the bit 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 while an “H” signal is applied to the $\overline{\text{NMI}}$ pin.

23.15.12 Rewriting in User ROM Area

EW0 Mode

If the supply voltage drops while rewriting the block where the rewrite control program is stored, the flash memory cannot be rewritten because the rewrite control program is not correctly rewritten. If this error occurs, rewrite the user ROM area while in standard serial I/O mode, parallel I/O mode, or CAN I/O mode.

EW1 Mode

Avoid rewriting any block in which the rewrite control program is stored.

23.15.13 DMA Transfer

In EW1 mode, do not perform a DMA transfer while the FMR00 bit in the FMR0 register is set to 0 (auto-programming or auto-erasure).

23.16 Flash Memory Programming Using Boot Program

When programming the on-chip flash memory using boot program, be careful about the pins state and connection as follows.

23.16.1 Programming Using Serial I/O Mode

CTX0 pin : This pin automatically outputs “H” level.

CRX0 pin : Connect to CAN transceiver or connect via resistor to VCC (pull-up)

Figure 23.6 shows the Pin Connection for Programming Using Serial I/O Mode.

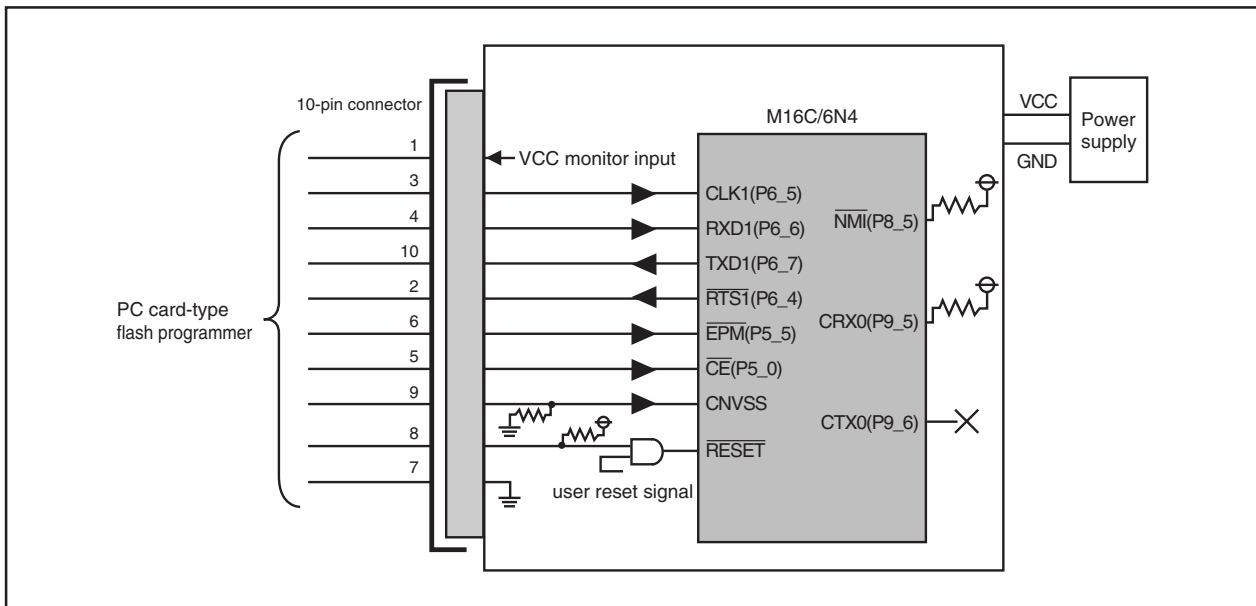


Figure 23.6 Pin Connection for Programming Using Serial I/O Mode

23.16.2 Programming Using CAN I/O Mode

RTS1 pin : This pin automatically outputs “H” and “L” level.

Figure 23.7 shows the Pin Connection for Programming Using CAN I/O Mode.

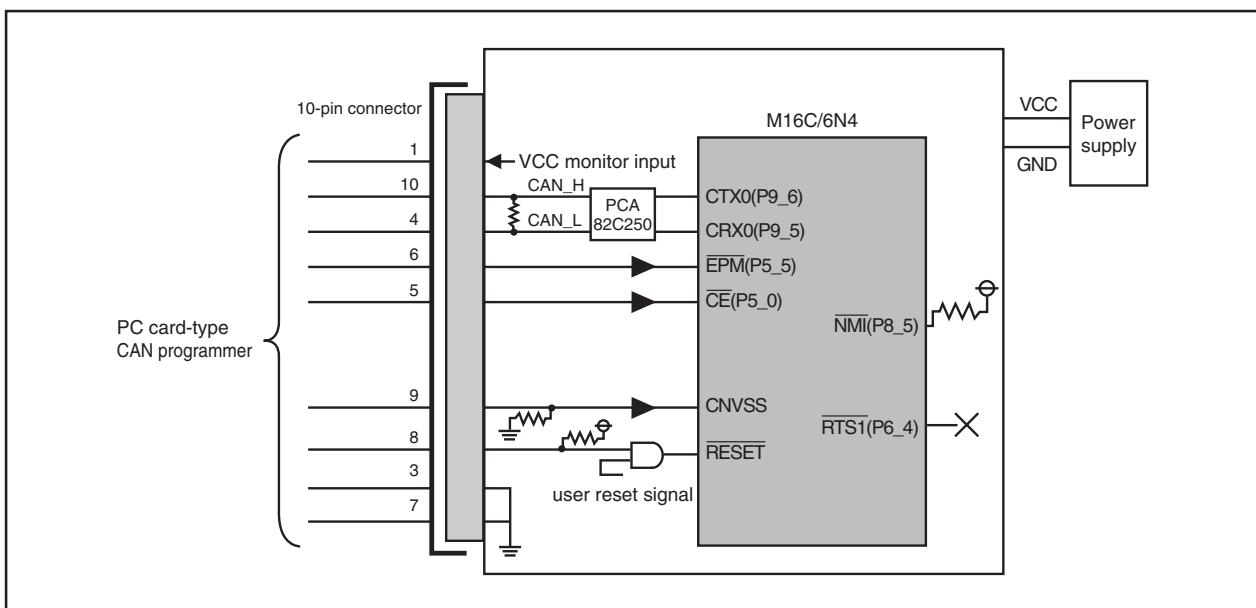


Figure 23.7 Pin Connection for Programming Using CAN I/O Mode

23.17 Noise

Connect a bypass capacitor (approximately 0.1 μF) across pins VCC1 and VSS, and pins VCC2 and VSS using the shortest and thicker possible wiring.

Figure 23.8 shows the Bypass Capacitor Connection.

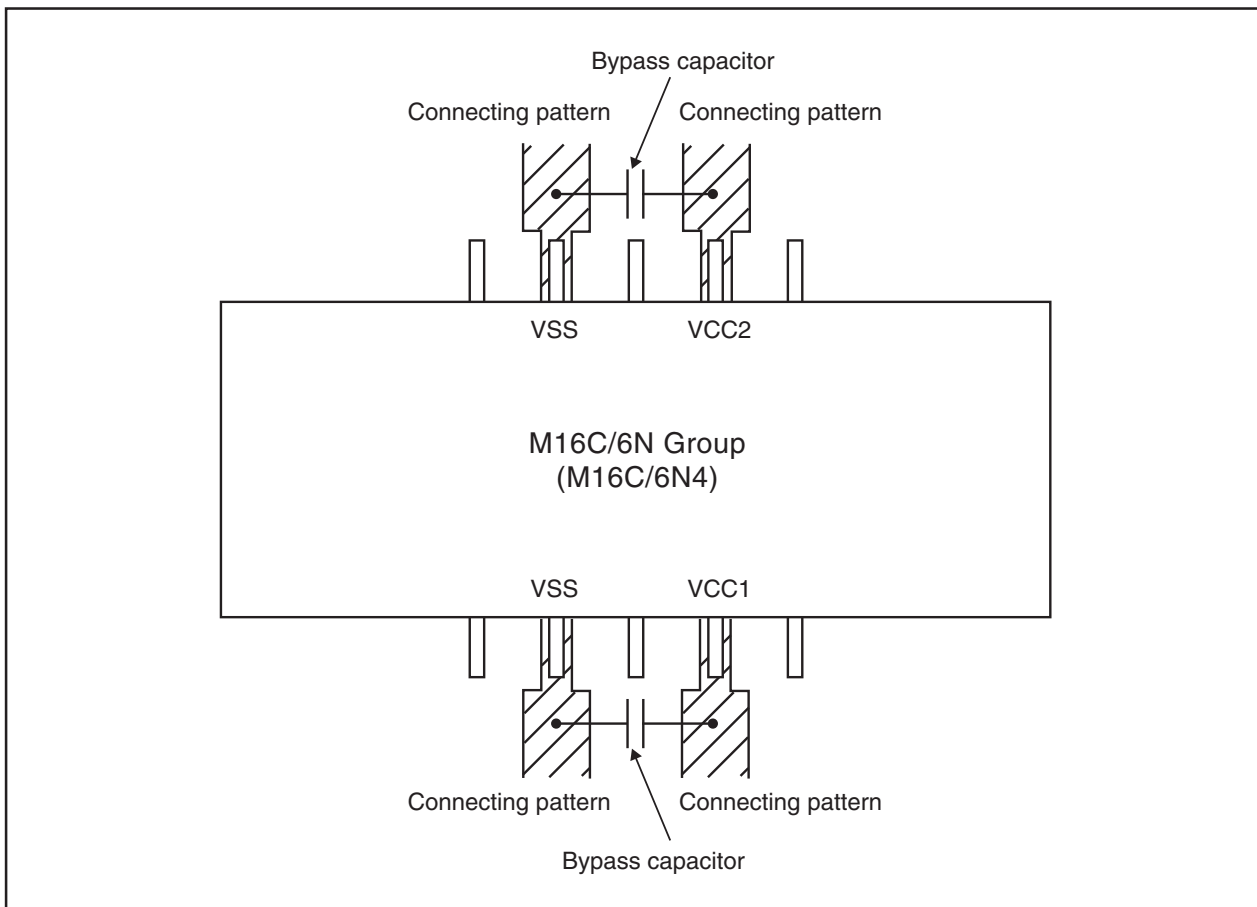
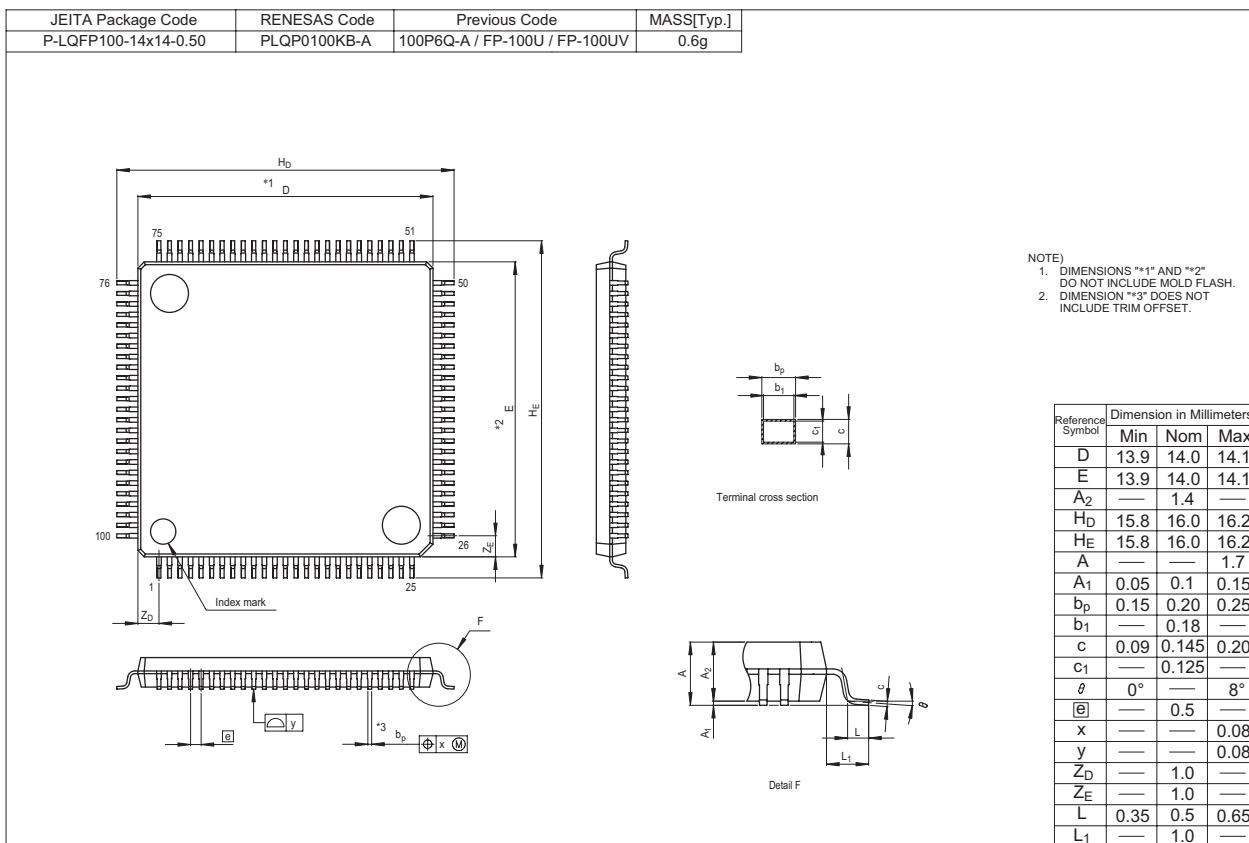
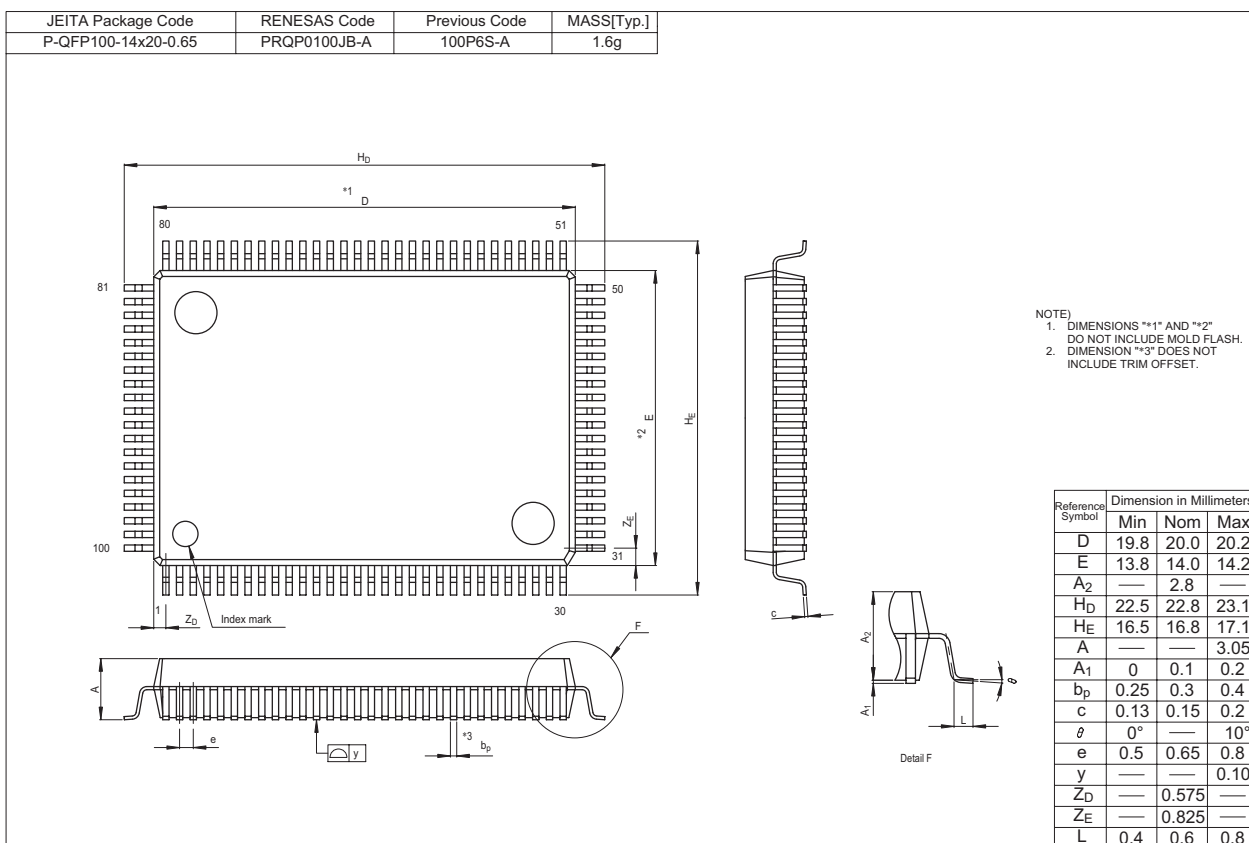


Figure 23.8 Bypass Capacitor Connection

Appendix 1. Package Dimensions



Register Index

| | | | | |
|-------------------------------------|--|---------------------------|---------------------------------------|--|
| A | | DAR0, DAR1 99 | T | |
| AD0 to AD7 196 | | DM0CON, DM1CON 98 | TA0 108 | |
| ADCON0 195,198,200,202,204,206 | | DM0IC, DM1IC 80 | TA0IC 80 | |
| ADCON1 195,198,200,202,204,206 | | DM0SL 97 | TA0MR 108,111,113,118,120 | |
| ADCON2 196 | | DM1SL 98 | TA1 108,135 | |
| ADIC 80 | | DTT 134 | TA11 135 | |
| AIER 92 | | | TA1IC 80 | |
| AIER2 92 | | F | TA1MR 108,111,113,118,120,138 | |
| | | FMR0 257 | TA2 108,135 | |
| C | | FMR1 257 | TA21 135 | |
| C01ERRIC 80 | | | TA2IC 80 | |
| C01WKIC 80 | | I | TA2MR ... 108,111,113,115,118,120,138 | |
| C0AFS, C1AFS 224 | | ICTB2 136 | TA3 108 | |
| C0CONR, C1CONR 223 | | IDB0, IDB1 134 | TA3IC 80 | |
| C0CTLR, C1CTLR 220 | | IFSR0 89 | TA3MR 108,111,113,115,118,120 | |
| C0GMR, C1GMR 218 | | IFSR1 89 | TA4 108,135 | |
| C0ICR, C1ICR 222 | | INT0IC to INT5IC 81 | TA41 135 | |
| C0IDR, C1IDR 222 | | INVC0 132 | TA4IC 80 | |
| C0LMAR, C1LMAR 218 | | INVC1 133 | TA4MR ... 108,111,113,115,118,120,138 | |
| C0LMBR, C1LMBR 218 | | | TABSR 109,124,137 | |
| C0MCTL0 to C0MCTL15 219 | | K | TB0 123 | |
| C0RECIC 80 | | KUPIC 80 | TB0IC 80 | |
| C0RECR, C1RECR 224 | | | TB0MR 123,125,126,128 | |
| C0SSTR, C1SSTR 225 | | O | TB1 123 | |
| C0STR, C1STR 221 | | ONSF 110 | TB1IC 80 | |
| C0TECR, C1TECR 224 | | | TB1MR 123,125,126,128 | |
| C0TRMIC 80 | | P | TB2 123,135 | |
| C0TSR, C1TSR 224 | | P0 to P10 246 | TB2IC 80 | |
| C1MCTL0 to C1MCTL15 219 | | PCLKR 56 | TB2MR 123,125,126,128,138 | |
| C1RECIC 81 | | PCR 248 | TB2SC 136 | |
| C1TRMIC 81 | | PD0 to PD10 245 | TB3 123 | |
| CAN0/1 Slot 0 to 15 | | PLC0 57 | TB3IC 80 | |
| : Time Stamp 216,217 | | PM0 35 | TB3MR 123,125,126,128 | |
| : Data Field 216,217 | | PM1 36 | TB4 123 | |
| : Message Box 216,217 | | PM2 57 | TB4IC 80 | |
| CCLKR 56 | | PRCR 74 | TB4MR 123,125,126,128 | |
| CM0 53 | | PUR0 to PUR2 247 | TB5 123 | |
| CM1 54 | | | TB5IC 80 | |
| CM2 55 | | R | TB5MR 123,125,126,128 | |
| CPSRF 110,124 | | RMAD0 to RMAD3 92 | TBSR 124 | |
| CRCD 212 | | ROMCP 254 | TCR0, TCR1 99 | |
| CRCIN 212 | | | TRGSR 110,137 | |
| CSE 47 | | S | | |
| CSR 41 | | S0RIC to S2RIC 80 | U | |
| | | S0TIC to S2TIC 80 | U0BCNIC to U2BCNIC 80 | |
| D | | S3BRG 189 | U0BRG to U2BRG 145 | |
| DA0, DA1 211 | | S3C 189 | U0C0 to U2C0 146 | |
| DACON 211 | | S3IC 81 | U0C1 to U2C1 147 | |
| | | S3TRR 189 | | |
| | | SAR0, SAR1 99 | | |

| | |
|------------------------|-----|
| U0MR to U2MR | 146 |
| U0RB to U2RB | 145 |
| U0SMR to U2SMR | 148 |
| U0SMR2 to U2SMR2 | 149 |
| U0SMR3 to U2SMR3 | 149 |
| U0SMR4 to U2SMR4 | 150 |
| U0TB to U2TB | 145 |
| UCON | 148 |
| UDF | 109 |

W

| | |
|------------|----|
| WDC | 94 |
| WDTS | 94 |

REVISION HISTORY

M16C/6N Group (M16C/6N4) Hardware Manual

| Rev. | Date | Description | |
|------|---------------|-------------|---|
| | | Page | Summary |
| 1.00 | May. 30, 2003 | – | First edition issued |
| 2.00 | Nov. 10, 2004 | – | Revised edition issued * Words standardizes (on-chip oscillator) * 100P6Q-A (100-pin version) is added. * Usage Notes Reference Book is added to Chapter 23 Usage Precaution. * Revised parts and revised contents are as follows (except for change of chapter composition, change of a layout, and an expressional change). |
| | | 1 | 1. Overview 3rd line: "and LQFP" is added. |
| | | 2 | Table 1.1 Performance outline of M16C/6N Group (M16C/6N4) • Operation Mode is added. • Address Space is added. • Power Consumption is revised. • "LQFP" is added to Package. |
| | | 4 | Table 1.2 Product List is revised. Figure 1.2 Type No., Memory Size, and Package: • "GP: Package 100P6Q-A" is added to Package type. |
| | | 5 | Figure 1.3 Pin Configuration (Top View) (1): "ZP" is added. |
| | | 6 | Figure 1.4 Pin Configuration (Top View) (2) is added. (100P6Q-A) |
| | | 8 | Table 1.4 Pin Description (2): "ZP" is added to Timer A. |
| | | 12 | 3. Memory • 5th to 6th lines: The description about the flash memory version (block A) is added. Figure 3.1 Memory Map: • Internal ROM (data area) is added. • NOTES 3, 4 are added and NOTE 5 is revised. |
| | | 13 | Table 4.1 SFR Information (1) • The value of After Reset in PM1 register is revised. • The value of After Reset in CM2 register is revised. |
| | | 19 | Table 4.7 SFR Information (7) • The value of After Reset in FMR0 register is revised. |
| | | 27 | Table 4.15 SFR Information (15) • The value of After Reset in U0C1 register is revised. • The value of After Reset in U1C1 register is revised. • NOTE 1 is added. |
| | | 28 | Table 4.16 SFR Information (16) • The value of After Reset in DA0, DA1 registers are revised. |
| | | 30 | Figure 5.1 Example Reset Circuit: NOTE 1 is added. |
| | | 34 | Figure 6.2 PM1 Register • The value of After Reset is revised. • NOTES 2, 6 are revised. |
| | | 37 | Figure 6.6 Memory Map and \overline{CS} Area in Memory Expansion Mode and Microprocessor Mode (3) • NOTE 2 is added. Figure 6.7 Memory Map and \overline{CS} Area in Memory Expansion Mode and Microprocessor Mode (4) • NOTE 1 is added. |

REVISION HISTORY

M16C/6N Group (M16C/6N4) Hardware Manual

| Rev. | Date | Description | |
|------|---------------|-------------|--|
| | | Page | Summary |
| 2.00 | Nov. 10, 2004 | 38 | Table 7.1 Difference between Separate Bus and Multiplexed Bus is added. |
| | | 39 | Figure 7.1 CSR Register: NOTE 2 is revised. |
| | | 46 | Table 7.8 Software Wait Related Bits and Bus Cycles <ul style="list-style-type: none"> • Bus Cycle of SFR (PM20 = 0) is revised from "2 BCLK cycles" to "3 BCLK cycles". • Bus Cycle of SFR (PM20 = 1) is revised from "3 BCLK cycles" to "2 BCLK cycles". • From bottom to 5th item in CSR Register: The value is revised from "1" to "0". • NOTE 5 is added. |
| | | 49 | Table 8.1 Clock Generating Circuit Specifications <ul style="list-style-type: none"> • Clock Frequency in PLL Frequency Synthesizer: 16 MHz is added. |
| | | 50 | Figure 8.1 Clock Generating Circuit: Block diagram (upper) is revised. |
| | | 51 | Figure 8.2 CM0 Register <ul style="list-style-type: none"> • Bit name of CM02 is revised. • NOTE 6 (2) and NOTE 8 are revised. |
| | | 52 | Figure 8.3 CM1 Register: NOTE 3 of CM11 bit is deleted. |
| | | 54 | Figure 8.6 CCLKR Register: Location of NOTE 2 is changed and NOTE 3 is added. |
| | | 55 | Figure 8.7 PM2 Register: NOTE 2 is revised. Figure 8.8 PLC0 Register: Function of 011b and 100b in PLC02 to PLC00 bits are revised from "Multiply by 6 and Multiply by 8" to "Do not set a value". |
| | | 58 | 8.1.4 PLL Clock 11th line: 16 MHz is added to PLL clock frequency. Table 8.2 Example for Setting PLL Clock Frequencies <ul style="list-style-type: none"> • PLL clock = 16 MHz is added. (8X2, 4X4) • 16 MHz is added to NOTE 1. |
| | | 59 | Figure 8.11 Procedure to Use PLL Clock as CPU Clock Source <ul style="list-style-type: none"> • 4th frame: "(To select a 16 MHz or higher PLL clock)" is revised to "(When PLL clock >16 MHz)". |
| | | 61 | 8.4.1.2 PLL Operation Mode: 1st line <ul style="list-style-type: none"> • The main clock multiplied is revised from "by 2, 4, 6 or 8" to "by 2 or 4". |
| | | 62 | Table 8.3 Setting Clock Related Bit and Modes <ul style="list-style-type: none"> • CM21 bit in Low Power Dissipation Mode: Value is revised from "-" to "0". • CM11 bit in Low-Speed Mode, Low Power Dissipation Mode, On-chip Oscillator Mode and On-chip Oscillator Low Power Dissipation Mode: Value is revised from "-" to "0". |
| | | 63 | 8.4.2 Wait Mode 4th line: "PLL clock" is deleted. Table 8.4 Pin Status During Wait Mode <ul style="list-style-type: none"> • Memory Expansion Mode, Microprocessor Mode in ALE: Value is revised from "H" to "L". |
| | | 64 | Table 8.5 Interrupts to Exit Wait Mode <ul style="list-style-type: none"> • CAN0/1 Wake-up Interrupt: "in CAN sleep mode" is added. |
| | | 65 | 8.4.3 Stop Mode <ul style="list-style-type: none"> • CAN0/1 Wake-up interrupt: "(when CAN sleep mode is selected)" is added. Table 8.6 Pin Status in Stop Mode <ul style="list-style-type: none"> • Memory Expansion Mode, Microprocessor Mode in ALE: Value is revised from " H" to "indeterminate". |

REVISION HISTORY

M16C/6N Group (M16C/6N4) Hardware Manual

| Rev. | Date | Description | |
|------|---------------|-------------|---|
| | | Page | Summary |
| 2.00 | Nov. 10, 2004 | 67 | Figure 8.12 State Transition to Stop Mode and Wait Mode <ul style="list-style-type: none"> • Figure is revised. • NOTE 3 is revised. |
| | | 68 | Figure 8.13 State Transition in Normal Operation Mode <ul style="list-style-type: none"> • Low-Speed and Low Power Dissipation Mode: "CM7 = 1" is revised to "CM7 = 0" (3 places). • NOTES 2, 6 are revised. |
| | | 71 | Figure 8.14 Procedure to Switch Clock Source from On-chip Oscillator to Main Clock is revised. |
| | | 77 | Table 10.2 Relocatable Vector Tables <ul style="list-style-type: none"> • Interrupt Source: "Software interrupt" is revised to "INT Instruction Interrupt" • NOTES 10, 11 are added. |
| | | 78 | Figure 10.3 Interrupt Control Registers (1): NOTES 5, 6, 7 are added. |
| | | 79 | Figure 10.4 Interrupt Control Registers (2) <ul style="list-style-type: none"> • NOTE 2 is added to C1RECIC/INT5IC, C1TRMIC/S3IC/INT4IC • NOTES 6, 7 are added. |
| | | 87 | Figure 10.11 (upper) IFSR0 Register: NOTE 3 is added. |
| | | 88 | 10.9 CAN0/1 Wake-up Interrupt is revised. Figure 10.13 CAN0/1 Wake-up Interrupt Block Diagram is revised. |
| | | 91 | Figure 11.1 Watchdog Timer Block Diagram: "RESET" is revised to "Internal RESET signal". |
| | | 108 | Figure 13.6 (upper and middle) ONSF Register, TRGSR Register: NOTE 2 is added. |
| | | 109 | Table 13.1 Specifications in Timer Mode <ul style="list-style-type: none"> • Specification of Divide Ratio: "TAiMR register" is revised to "TAi register". • Specification of Select Function: "When not counting, the pin outputs a low" is revised to "When TAiS bit is set to "0" (stop counting), the pin outputs a low". |
| | | 110 | Table 13.2 Specifications in Event Counter Mode (when not processing two-phase pulse signal) <ul style="list-style-type: none"> • Specification in Select Function: "When not counting, the pin outputs a low" is revised to "When TAiS bit is set to "0" (stop counting), the pin outputs a low". |
| | | 114 | 13.1.2.1 Counter Initialization by Two-Phase Pulse Signal Processing 4th line <ul style="list-style-type: none"> • "the INT2 pin" is revised to "the ZP pin". Figure 13.10 Two-phase Pulse (A phase and B phase) and Z Phase <ul style="list-style-type: none"> • "INT2 (Z phase)" is revised to "ZP". |
| | | 118 | Figure 13.12 TA0MR to TA4MR Registers in PWM Mode <ul style="list-style-type: none"> • Bit name and Function in MR0 bit is revised from "Set to "1" in PWM mode" to "Pulse Output Function Select Bit ⁽³⁾". • NOTE 3 is added. |
| | | 123 | Table 13.6 Specifications in Timer Mode <ul style="list-style-type: none"> • Specification in Divide Ratio: "TBiMR register" is revised to "TBi register". |
| | | 129 | Figure 14.1 Three-Phase Motor Control Timer Function Block Diagram is revised. |
| | | 130 | Figure 14.2 INVC0 Register is revised. |
| | | 131 | Figure 14.3 INVC1 Register: Function of INV13 bit is revised. |
| | | 132 | Figure 14.4 (upper) IDB0 and IDB1 Registers: (b7-b6) is revised. Figure 14.4 (lower) DTT Register: NOTE 2 is revised. |

REVISION HISTORY

M16C/6N Group (M16C/6N4) Hardware Manual

| Rev. | Date | Description | |
|------|--|-------------|--|
| | | Page | Summary |
| 2.00 | Nov. 10, 2004 | 134 | Figure 14.6 (upper) ICTB2 Register <ul style="list-style-type: none"> •(b7-b4) is revised. •NOTE 3 is added. |
| | | 135 | Figure 14.7 (upper) TRGSR Register: NOTE 2 is added. |
| | | 136 | Figure 14.8 (upper) TA1MR, TA2MR and TA4MR Registers <ul style="list-style-type: none"> •Function of MR1 bit: "Has no effect" is revised to "Set to "0" " . |
| | | 137 | Figure 14.9 Triangular Wave Modulation Operation is revised. |
| | | 139 | 15.1 UARTi: "UART0, UART1" in Special mode 3 is deleted. |
| | | 140, 141 | Figures 15.1 to 15.3 UART0 to 2 Block Diagram are revised. |
| | | 142 | Figure 15.4 UARTi Transmit/Receive Unit is revised. |
| | | 144 | Figure 15.6 (lower) U0C0 to U2C0 Registers: NOTES 3, 4 are revised. |
| | | 145 | Figure 15.7 (upper) U0C1, U1C1 Registers <ul style="list-style-type: none"> •The value of After Reset is revised. •(b5-b4) is revised from "When read, their contents are "0" " to "When read, their contents are indeterminate". •NOTE 1 is added. |
| | | | Figure 15.7 (lower) U2C1 Register: NOTE 1 is added. |
| | | 153 | 15.1.1.1 Counter Measure for Communication Error Occurs is added. |
| | | 154 | 15.1.1.4 Continuous Receive Mode: first to 4th lines are added. |
| | | 156 | 15.1.1.7 CTS/RTS Function is added. |
| | | 157 | Table 15.5 UART Mode Specifications: NOTE 3 is added. |
| | | 159 | Table 15.7 I/O Pin Functions <ul style="list-style-type: none"> •Method of Selection in TXDi: "Output dummy data" is revised to "Output "H" " . |
| | | 161 | 15.1.2.1 Bit Rates and Table 15.9 Example of Bit Rates and Settings are added. |
| | | 162 | 15.1.2.2 Counter Measure for Communication Error Occurs is added. |
| | | 164 | 15.1.2.6 CTS/RTS Function is added. |
| | | 176 | Table 15.15 Registers to Be Used and Settings in Special Mode 2 <ul style="list-style-type: none"> •"U2LCH" in UiC1 register is revised to "UiLCH". |
| | | 179 | Table 15.16 Registers to Be Used and Settings in IE Mode <ul style="list-style-type: none"> •"UiRRM" in UiC1 register is revised to "U2RRM". |
| | | 181 | Table 15.17 SIM Mode Specifications: NOTE 3 is added. |
| | | 189 | Figure 15.39 Polarity of Transfer Clock is revised. |
| | | 205 | 16.2.4 External Operation Amplifier (Op-Amp) Connection Mode: 6th line <ul style="list-style-type: none"> •"Note that the ANEX0 and ANEX1 pins cannot be directly connected to each other." is deleted. |
| | | 206 | 16.2.6 Output Impedance of Sensor under A/D Conversion is added. |
| | | 209 | Figure 17.2 (lower) DA0 and DA1 Registers: The value of After Reset are revised. |
| | | 216 | Figure 19.4 Bit Mapping of Mask Registers in Byte Access: NOTES 1, 2 are added. Figure 19.5 Bit Mapping of Mask Registers in Word Access: NOTES 1, 2 are added. |
| | | 217 | Figure 19.6 C0MCTLj and C1MCTLj Registers: NOTE 2 is revised. |
| | | 218 | Figure 19.7 C0CTLR and C1CTLR Registers (upper) <ul style="list-style-type: none"> •NOTE 1 (Rev.1.00) is deleted and NOTES 1, 2, 3 are added. |
| | Figure 19.7 C0CTLR and C1CTLR Registers (lower): NOTES 3, 4 are added. | | |

REVISION HISTORY

M16C/6N Group (M16C/6N4) Hardware Manual

| Rev. | Date | Description | |
|------|---------------|-------------|--|
| | | Page | Summary |
| 2.00 | Nov. 10, 2004 | 219 | Figure 19.8 C0STR and C1STR Registers (upper): NOTE 2 is added. |
| | | 223 | 19.5 Operational Modes <ul style="list-style-type: none"> • 1st line: "three operational modes" is revised to "four operational modes". • 5th line: "CAN Interface Sleep Mode" is added. Figure 19.12 Transition Between Operational Modes is revised. |
| | | 224 | 19.5.1 CAN Reset/Initialization Mode is revised. 19.5.2 CAN Operation Mode is revised. 19.5.3 CAN Sleep Mode is revised. 19.5.4 CAN Interface Sleep Mode is added. |
| | | 225 | 19.5.5 Bus Off State is revised. |
| | | 231 | 19.12 Return from Bus Off Function is revised. 19.14 Listen-Only Mode <ul style="list-style-type: none"> • last line: "When listen-only mode is selected, do not request the transmission." is added. |
| | | 233 | Figure 19.20 Timing of Receive Data Frame Sequence: Waveform of RecState bit is revised. 19.15.1 Reception: (4) (5) are revised. |
| | | 234 | Figure 19.21 Timing of Transmit Sequence <ul style="list-style-type: none"> • The position of the number corresponding to the text is revised. 19.15.2 Transmission: (1) to (4) are revised. |
| | | 251 | 21.2.1 ROM Code Protect Function is revised. 21.2.2 ID Code Check Function is revised. |
| | | 252 | Figure 21.2 ROMCP Register is revised. |
| | | 255 | Figure 21.4 (upper) FMR0 Register: The value of After Reset is revised. |
| | | 256 | 21.3.3.1 FMR00 Bit is revised. 21.3.3.8 FMR11 Bit is revised. 21.3.3.9 FMR16 Bit is revised. |
| | | 257 | Figure 21.5 Setting and Resetting of EW0 Mode is revised. Figure 21.6 Setting and Resetting of EW1 Mode: NOTE 3 is revised. |
| | | 258 | Figure 21.7 Processing Before and After Low Power Dissipation Mode: NOTE 4 is added. |
| | | 260 | 21.3.4.12 Low Power Dissipation Mode and On-chip Oscillator Low Power Dissipation Mode is revised. |
| | | 261 | Table 21.4 Software Commands: NOTE 2 is deleted. |
| | | 262 | 21.3.5.4 Program Command (40h) <ul style="list-style-type: none"> • From bottom to 3rd line: "read command" is revised to "read array command". |
| | | 265 | Figure 21.11 Read Lock Bit Status Command <ul style="list-style-type: none"> • "Locked", "Not locked" are revised to "Block is locked", "Block is not locked". |
| | | 266 | 21.3.7.1 Sequencer Status (SR7 and FMR00 Bits) is revised. |
| | | 271 | Table 21.7 Pin Functions for Standard Serial I/O Mode <ul style="list-style-type: none"> • "VCC" is revised to "VCC1", and "VCC2" is added. • VCC1, VCC2, VSS: VCC apply condition is added. |
| | | 273 | Figure 21.14 Pin Connections for Standard Serial I/O Mode (2) is added. |
| | | 274 | Figure 21.16 Circuit Application in Standard Serial I/O Mode 2: "RESET" is added. |
| | | 276 | Table 21.8 Pin Functions for CAN I/O Mode <ul style="list-style-type: none"> • "VCC" is revised to "VCC1", and "VCC2" is added. • VCC1, VCC2, VSS: VCC apply condition is added. |

REVISION HISTORY

M16C/6N Group (M16C/6N4) Hardware Manual

| Rev. | Date | Description | |
|------|---|-------------|--|
| | | Page | Summary |
| 2.00 | Nov. 10, 2004 | 278 | Figure 21.18 Pin Connections for CAN I/O Mode (2) is added. |
| | | 280 | Table 21.9 Flash Memory Version Electrical Characteristics <ul style="list-style-type: none"> • Parameter is added and the value of some item is revised. |
| | | 281 | Table 22.1 Absolute Maximum Ratings <ul style="list-style-type: none"> • "Flash Program Erase" in Operating Ambient Temperature is added. |
| | | 283 | Table 22.3 Recommended Operating Conditions (2) <ul style="list-style-type: none"> • Parameters of Power Supply Ripple are added. • NOTE 4 is revised. |
| | | | Figure 22.1 Timing of Voltage Fluctuation is added. |
| | | 284 | Table 22.4 Electrical Characteristics (1): Hysteresis <ul style="list-style-type: none"> • "CLK4" is revised to "CLK3", and "TA2OUT" is revised to "TA0OUT". • Max. of Standard in <u>RESET</u> is revised from "2.2" to "2.5". • XIN is added. |
| | | 286 | Table 22.6 A/D Conversion Characteristics: "Tolerance Level Impedance" is added. |
| | | 287 | Table 22.8 Power Supply Circuit Timing Characteristics: " $t_{d(M-L)}$ " is deleted. Figure 22.2 Power Supply Circuit Timing Diagram is added. |
| | | 288 | Table 22.10 Memory Expansion Mode and Microprocessor Mode: " $t_{d(BCLK-HLDA)}$ " is deleted. |
| | | 290 | Table 22.21 Serial I/O: Min. of standard in $t_{su(D-C)}$ is revised from "30" to "70". |
| | | 291 | Table 22.23 Memory Expansion Mode and Microprocessor Mode (for setting with no wait) <ul style="list-style-type: none"> • Max. of Standard in $t_{d(BCLK-ALE)}$ is revised from "25" to "15". • $t_{d(BCLK-HLDA)}$ is added. |
| | | 292 | Table 22.24 Memory Expansion Mode and Microprocessor Mode (for 1- to 3-wait setting and external area access) <ul style="list-style-type: none"> • Max. of Standard in $t_{d(BCLK-ALE)}$ is revised from "25" to "15". • $t_{d(BCLK-HLDA)}$ is added. |
| | | 293 | Table 22.25 Memory Expansion Mode and Microprocessor Mode (for 2- to 3-wait setting, external area access and multiplexed bus selection) <ul style="list-style-type: none"> • $t_{d(BCLK-HLDA)}$ is added. • Max. of Standard in $t_{d(BCLK-ALE)}$ is revised from "25" to "15". |
| | | 294 | Figure 22.4 Timing Diagram (1): "XIN input" is added. |
| | | 296, 297 | Figures 22.6 and 22.7 Timing Diagram (3) (4): "DB" in Read timing is revised to "DBi". |
| | | 298, 299 | Figures 22.8 and 22.9 Timing Diagram (5) (6): "DB" in Write timing is revised to "DBi". |
| | | 301 | Figure 22.11 Timing Diagram (8) <ul style="list-style-type: none"> • "ADi/DBi" in Read/Write timing is revised to "ADi/DBi". |
| | | 302 | 23.1 External Bus: The description of the external ROM version is deleted. |
| | | 303 | 23.2 PLL Frequency Synthesizer is revised. |
| | | 304 | 23.3 Power Control <ul style="list-style-type: none"> • 2nd item is added. (Set the MR0 bit in the TAI_{MR} register to ●●) • 4th item is revised. (Wait for main clock oscillation ●●) • Section of "External clock" is deleted. |
| 316 | 23.8.2.1 Special Mode 1 (I ² C Mode) is added. | | |
| 317 | 23.8.3 SI/O3 is added. | | |

REVISION HISTORY

M16C/6N Group (M16C/6N4) Hardware Manual

| Rev. | Date | Description | |
|------|---------------|--|---|
| | | Page | Summary |
| 2.00 | Nov. 10, 2004 | 319 322 323 327 328 330 331 332 | 23.9 A/D Converter: last item is added. (When setting the ADST bit to ●●●) 23.10.2 Performing CAN Configuration is added. 23.10.3 Suggestions to Reduce Power Consumption is added. 23.13 Mask ROM Version is added. 23.14.4 Low Power Dissipation Mode and On-Chip Oscillator Low Power Dissipation Mode is revised. 23.15 Flash Memory Programming Using Boot Program is added. 23.16 Noise is added. Appendix 1. Package Dimensions: 100P6Q-A is added. |
| 2.10 | Jun. 24, 2005 | – 2 4 19 53 68 217 218 219 222 233 | Revised edition issued * The contents of product are revised. (Normal-ver. is added.) * Revised parts and revised contents are as follows (except for expressional change). Table 1.1 Performance outline of M16C/6N Group (M16C/6N4) • Performance outline of Normal-ver. is added. Table 1.2 Product List is revised. (Normal-ver. is added.) Figure 1.2 Type No., Memory Size, and Package: • "(no): Normal-ver." is added to Characteristics. Figure 4.7 SFR Information (7): NOTE 1 is revised. Figure 8.4 CM2 Register: The value of After Reset is revised. Figure 8.13 State Transition in Normal Operation Mode: NOTE 7 is revised. Figure 19.6 C0MCTLj and C1MCTLj Registers • RemActive bit: Function is revised. • RspLock bit: Bit Name is revised. • NOTE 2 is revised. Figure 19.7 C0CTLR and C1CTLR Registers (upper) • LoopBack bit: The expression of Function is revised. • BasicCAN bit: The expression of Function is revised. Figure 19.7 C0CTLR and C1CTLR Registers (lower) • TSPreScale bit: Bit Symbol is revised. ("Bit1, Bit0" is deleted.) • TSReset bit: The expression of Function is revised. • RetBusOff bit: The expression of Function is revised. • RXOnly bit: The expression of Function is revised. Figure 19.8 C0STR and C1STR Registers (upper): NOTE 1 is deleted. Figure 19.8 C0STR and C1STR Registers (lower) • State_LoopBack bit: The expression of Function is revised. • State_BasicCAN bit: The expression of Function is revised. Figure 19.11 C0RECR, C1RECR Registers, C0TECR, C1TECR Registers, C0TSR, C1TSR Registers, and C0AFS, C1AFS Registers • C0RECR, C1RECR Registers: NOTE 2 is deleted. • C0TECR, C1TECR Registers: NOTE 1 is deleted. • C0TSR, C1TSR Registers: NOTE 1 is deleted. 19.15.1 Reception (1): "(refer to 19.15.2 Transmission)" is deleted. |

REVISION HISTORY

M16C/6N Group (M16C/6N4) Hardware Manual

| Rev. | Date | Description | |
|------|---|-------------|---|
| | | Page | Summary |
| 2.10 | Jun. 24, 2005 | 238 | Figure 20.1 I/O Ports (1): "P7_0" in 4th figure is deleted. |
| | | 240 | Figure 20.3 I/O Ports (3): "P7_0" is added to middle figure. |
| | | 242 | Figure 20.6 I/O Pins: NOTE 1 is deleted. |
| | | 284 | Table 22.4 Electrical Characteristics (1) <ul style="list-style-type: none"> Measuring Condition of V_{OL} is revised from "$L_{OL} = -200\mu A$" to "$L_{OL} = 200\mu A$". |
| | | 285 | Table 22.5 Electrical Characteristics (2): Mask ROM (5th item) <ul style="list-style-type: none"> "f(XCIN)" is changed to "f(BCLK)". |
| | | 286 | Table 22.6 A/D Conversion Characteristics: "Tolerance Level Impedance" is deleted. |
| 2.30 | Oct. 24, 2005 | – | Revised edition issued * Electric Characteristics of Normal-ver. is added. * Revised parts and revised contents are as follows (except for expressional change). |
| | | 1 | 1.1 Applications: Comment of Normal-ver. is added. |
| | | 4 | Table 1.2 Product List: NOTE 1 is added. |
| | | 7, 8 | Tables 1.3 and 1.4 Pin Characteristics (1)(2) are added. |
| | | 9 | Table 1.5 Pin Description (1) <ul style="list-style-type: none"> 3.0 to 3.6 V (Normal-ver.) is added to Description of Power supply input. |
| | | 31 to 33 | 5. Reset: Layout is changed. |
| | | 33 | 5.5 Internal Space is added. |
| | | 44 | 7.2.6 RDY Signal: Last sentence is revised. |
| | | 51 | Table 8.1 Clock Generating Circuit Specifications <ul style="list-style-type: none"> Clock Frequency in PLL Frequency Synthesizer: 24 MHz ⁽¹⁾ is added. NOTE 1 is added. |
| | | 57 | Figure 8.8 PLC0 Register <ul style="list-style-type: none"> PLC02 to PLC00 bits: Function of 011b is revised. NOTE 4 is added. |
| | | 58 | Figure 8.9 Examples of Main Clock Connection Circuit is revised. |
| | | 59 | Figure 8.10 Examples of Sub Clock Connection Circuit is revised. |
| | | 60 | 8.1.4 PLL Clock <ul style="list-style-type: none"> 9th line: The sentence (When the PLL ... to) is added. 12th line: 24 MHz and NOTE 1 is added to PLL clock frequency. NOTE 1 is added. Table 8.2 Example for Setting PLL Clock Frequencies <ul style="list-style-type: none"> 24 MHz is added to PLL clock. 24 MHz is added to NOTE 1. NOTES 2 and 3 are added. |
| | | 63 | 8.4.1.2 PLL Operation Mode <ul style="list-style-type: none"> 1st line: The main clock multiplied by "6" and NOTE 1 is added. |
| | | 64 | 8.4.1.6 On-chip Oscillator Mode: Last sentence (When the operation mode is ...) is added. 8.4.1.7 On-chip Oscillator Low Power Dissipation Mode: Last sentence (When the operation mode is ...) is deleted. |
| | | 67 | Table 8.6 Interrupts to Stop Mode and Use Conditions is added. |
| 70 | Figure 8.13 State Transition in Normal Operation Mode: NOTE 7 is deleted. | | |

REVISION HISTORY

M16C/6N Group (M16C/6N4) Hardware Manual

| Rev. | Date | Description | |
|------|---------------|-------------|---|
| | | Page | Summary |
| 2.30 | Oct. 24, 2005 | 86 | 10.5.8 Returning from an Interrupt Routine: Last sentence (Register bank ...) is added. 10.5.9 Interrupt Priority: First sentence (If two or more...) is revised. 10.5.10 Interrupt Priority Resolution Circuit: First sentence (The interrupt priority level ...) is revised. |
| | | 89 | Figure 10.11 IFSR1 Register (upper) <ul style="list-style-type: none"> • IFSR17: NOTE 2 is added to Bit Name. • NOTE 2 is revised. |
| | | 96 | Table 12.1 DMAC Specifications: DMA transfer Cycles is added. |
| | | 100 | 12.1.3 Effect of Software Wait: 3rd to 9th lines is moved from next section of 12.1.4. |
| | | 120 | Figure 13.12 TA0MR to TA4MR Registers in PWM Mode: b2 is revised from "1" to "(blank)". |
| | | 131 | Figure 14.1 Three-Phase Motor Control Timer Function Block Diagram is revised. |
| | | 132 | Figure 14.2 INVC0 Register: NOTES 5 and 6 are revised. |
| | | 145 | Figure 15.5 U0BRG to U2BRG Registers (lower): NOTE 3 is added. |
| | | 146 | Figure 15.6 U0C0 to U2C0 Registers (lower): NOTE 5 is added. |
| | | 163 | Table 15.9 Example of Bit Rates and Settings: 24 MHz and NOTE 1 is added. |
| | | 189 | Figure 15.37 S3C Register (upper): NOTE 5 is added. Figure 15.37 S3BRG Register (middle): NOTE 3 is added. |
| | | 193 | Table 16.1 A/D Converter Performance <ul style="list-style-type: none"> • Performance of Integral Nonlinearity Error: "When AVCC = VREF = 3.3 V" is added. |
| | | 194 | Figure 16.1 A/D Converter Block Diagram <ul style="list-style-type: none"> • ADGSEL1 to ADGSEL0 (right/lower) is revised from "10b" to "11b". |
| | | 208 | 16.2.6 Output Impedance of Sensor under A/D Conversion <ul style="list-style-type: none"> • 10th line: f(XIN) is revised to f(ϕAD). |
| | | 209 | Figure 16.10 Analog Input Pin and External Sensor Equivalent Circuit <ul style="list-style-type: none"> • fAD is revised to ϕAD. |
| | | 210 | Figure 17.1 D/A Converter Block Diagram is revised. |
| | | 211 | Figure 17.2 DA0 and DA1 Registers: Setting Range is added. Figure 17.3 D/A Converter Equivalent Circuit: NOTE 2 is added. |
| | | 213 | Figure 18.3 CRC Calculation: Details of CRC operation is revised. |
| | | 224 | Figure 19.11 C0TECR, C1TECR Registers (2nd register): NOTE 1 is added. |
| | | 229 | Table 19.2 Examples of Bit-rate: 24 MHz and NOTE 2 is added. |
| | | 247 | Figure 20.9 PUR1 Register (middle): Value of After Reset is revised. |
| | | 252 | Figure 21.1 Flash Memory Block Diagram is revised. |
| | | 254 | Figure 21.2 ROMCP Register is revised. |
| | | 255 | Table 21.3 EW0 Mode and EW1 Mode: NOTE 1 is revised. |
| | | 256 | 21.3.2 EW1 Mode: Last sentence (When an erase/program ...) is added. |
| | | 258 | 21.3.3.4 FMSTP Bit <ul style="list-style-type: none"> • 8th line: Procedure to change the FMSTP bit setting (1) to (4) are added. |
| | | 261 | Figure 21.7 Processing Before and After Low Power Dissipation Mode or On-chip Oscillator Low Power Dissipation Mode <ul style="list-style-type: none"> • Title, First and second frames (left) and top of right: "on-chip oscillator low power dissipation mode" is added. |

REVISION HISTORY

M16C/6N Group (M16C/6N4) Hardware Manual

| Rev. | Date | Description | |
|------|---|-------------|--|
| | | Page | Summary |
| 2.30 | Oct. 24, 2005 | 263 | 21.3.4.11 Stop Mode is revised. |
| | | | 21.3.4.12 Low Power Dissipation Mode and On-chip Oscillator Low Power Dissipation Mode is partly revised. |
| | | 266 | 21.3.5.5 Block Erase Command: Last sentence (Also execute ...) is added. Figure 21.9 Block Erase Command: NOTES 2 and 3 are added. |
| | | 272 | Figure 21.12 Full Status Check and Handling Procedure for Each Error • Erase error: (4) is added. |
| | | 274 | Table 21.7 Pin Functions for Standard Serial I/O Mode • Description of VCC1, VCC2, VSS is revised. • Description of P8_4 is revised. • NOTE 1 is revised. • NOTE 2 is added. |
| | | 277 | Figures 21.15 and 21.16 Circuit Application in Serial I/O Mode 1/2 • "VCC1" and "VCC2" are added. |
| | | 279 | Table 21.8 Pin Functions for CAN I/O Mode • Description of VCC1, VCC2, VSS is revised. • Description of P8_4 is revised. • NOTE 1 is added. |
| | | 282 | Figure 21.19 Circuit Application in CAN I/O Mode: "VCC1" and "VCC2" are added. |
| | | 283 | Table 21.9 Flash Memory Version Electrical Characteristics • Measuring condition is revised in word program time and block erase time. |
| | | 284 | 21.7.2 Electrical Characteristics (Normal-ver.) is added. |
| | | 306 to 341 | 22.2 Electrical Characteristics (Normal-ver.) is added. |
| | | 344 | 23.3 Power Control: 3rd and 4th items (When entering wait mode ... / When entering stop mode ...) are revised. |
| | | 359 | Figure 23.2 Use of Capacitors to Reduce Noise is partly revised. |
| | | 360 | 23.9 A/D Converter: Last item (The applied intermediate ...) is added. |
| | | 366 | 23.11 Programmable I/O Ports: 4th and 5th items (Indeterminate values ... / When the PM01 ...) are added. |
| 369 | 23.14.2 Stop Mode is revised. 23.14.4 Low Power Dissipation Mode and On-Chip Oscillator Low Power Dissipation Mode is partly revised. 23.14.8 Operation Speed is revised. | | |
| 2.40 | Apr.14, 2006 | – | Revised edition issued * Revised parts and revised contents are as follows (except for expressional change). |
| | | 4 | Table 1.2 Product Information: Note 2 is added. |
| | | 22 | Table 4.8 SFR Information (8) • The value of After Reset in IDB0 register is revised. • The value of After Reset in IDB1 register is revised. |
| | | 69 | Figure 8.12 State Transition to Stop Mode and wait Mode is revised. |
| | | 100 | 12.1.3 Effect of Software Wait: 3rd to 9th lines (Figure 12.5 shows ... required.) is moved to next section of 12.1.4. |

REVISION HISTORY

M16C/6N Group (M16C/6N4) Hardware Manual

| Rev. | Date | Description | |
|------|--------------|-------------|---|
| | | Page | Summary |
| 2.40 | Apr.14, 2006 | 111 | Figure 13.7 Registers TA0MR to TA4MR in Timer Mode: Note 2 is added. |
| | | 118 | Figure 13.11 Registers TA0MR to TA4MR in One-shot Timer Mode: Note 3 is added. |
| | | 120 | Figure 13.12 Registers TA0MR to TA4MR in PWM Mode: Note 4 is added. |
| | | 125 | Figure 13.18 Registers TB0MR to TB5MR in Timer Mode: Note 1 is added. |
| | | 128 | Figure 13.20 Registers TA0MR to TA4MR in Pulse Period and Pulse Width Measurement Mode: Note 2 is added. |
| | | 133 | Figure 14.3 INVC1 Register: Note 6 is added. |
| | | 134 | Figure 14.4 Registers IDB0 and IDB1 (upper): The value of After Reset is revised. |
| | | 138 | Figure 14.8 Registers TA1MR, TA2MR, TA4MR (upper): Note 1 is added. Figure 14.8 TB2MR Register (lower): Note 1 is added. |
| | | 142, 143 | Figures 15.1 to 15.3 are revised. |
| | | 145 | Figure 15.5 Registers U0RB to U2RB (middle): Note 3 is added. |
| | | 146 | Figure 15.6 Registers U0C0 to U2C0 (lower): Note 6 is added. |
| | | 151 | Table 15.1 Clock Synchronous Serial I/O Mode Specifications <ul style="list-style-type: none"> • Transfer clock: "fj/2(n+1)" is revised to "fj/(2(n+1))". • Note 3 is revised. |
| | | 154 | Figure 15.11 Transmit and Receive Operation is revised. |
| | | 159 | Table 15.5 UART Mode Specifications <ul style="list-style-type: none"> • Transfer clock: "fj/16(n+1)" is revised to "fj/(16(n+1))" and "fEXT/16(n+1)" is revised to "fEXT/(16(n+1))" . • Note 2 is revised. |
| | | 162 | Figure 15.17 Transmit Operation is revised. |
| | | 163 | Table 15.9 Example of Bit Rates and Settings: "Actual Time" is revised to "Bit Rate". |
| | | 167 | Table 15.10 I ² C Mode Specifications <ul style="list-style-type: none"> • Transfer clock: "fj/2(n+1)" is revised to "fj/(2(n+1))". |
| | | 169 | Table 15.11 Registers to Be Used and Settings in I ² C Mode: Note 3 is added. |
| | | 176 | Table 15.14 Special Mode 2 Specifications <ul style="list-style-type: none"> • Transfer clock: "fj/2(n+1)" is revised to "fj/(2(n+1))". |
| | | 183 | Table 15.17 SIM Mode Specifications <ul style="list-style-type: none"> • Transfer clock: "fj/16(n+1)" is revised to "fj/(16(n+1))" and "fEXT/16(n+1)" is revised to "fEXT/(16(n+1))". |
| | | 185 | Figure 15.32 Transmit and Receive Timing in SIM Mode is revised. |
| | | 187 | 15.1.6.2 Format is revised. |
| | | 189 | Figure 15.37 S3C Register (upper): Note 6 is added. |
| | | 190 | Table 15.19 SI/O3 Specifications <ul style="list-style-type: none"> • Transfer clock: "fj/2(n+1)" is revised to "fj/(2(n+1))". |
| | | 191 | Figure 15.38 SI/O3 Operation Timing: Cycle and Note 1 is revised. (1.5 -> 0.5 to 1.0) |
| | | 192 | 15.2.3 Functions for Setting SOUT3 Initial Value: 2nd item (However...) is added. |
| | | 211 | Figure 17.3 D/A Converter Equivalent Circuit is revised. |
| | | 220 | Figure 19.7 Registers C0CTLR and C1CTLR (upper): NOTE 4 is added. |
| | | 224 | Figure 19.11 Registers C0TSR and C1TSR (3rd register): Note 1 is added. |
| | | 225 | Figure 19.12 Transition between Operational Modes is revised. |

REVISION HISTORY

M16C/6N Group (M16C/6N4) Hardware Manual

| Rev. | Date | Description | |
|------|--------------|-------------|---|
| | | Page | Summary |
| 2.40 | Apr.14, 2006 | 226 | 19.5.3 CAN Sleep Mode <ul style="list-style-type: none"> • 1st item: “and Reset bit to 0” is deleted. |
| | | 229 | Table 19.2 Examples of Bit-rate is revised. |
| | | 249 | Table 20.2 Unassigned Pin Handling in Memory expansion Mode and Microprocessor Mode <ul style="list-style-type: none"> • Pin Name: “P0 to P7” is revised to “P6, P7”. |
| | | 288 | Table 22.4 Electrical Characteristics (1): Hysteresis XIN is deleted. |
| | | 311 | Table 22.32 Electrical Characteristics (1): Hysteresis XIN is deleted. |
| | | 327 | Table 22.51 Electrical Characteristics: Hysteresis XIN is deleted. |
| | | 342 | 23.1 SFR is added. |
| | | 345 | 23.4 Power Control <ul style="list-style-type: none"> • 4th item: Notes when entering stop mode is revised. |
| | | 346 | <ul style="list-style-type: none"> • 5th item: Notes is added. |
| | | 360 | 23.10 A/D Converter <ul style="list-style-type: none"> • 1st item: “After stopping ...” is added. |
| | | | |

M16C/6N Group (M16C/6N4) Hardware Manual

**Publication Data : Rev.1.00 May 30, 2003
Rev.2.40 Apr 14, 2006**

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

M16C/6N Group (M16C/6N4) Hardware Manual



Renesas Electronics Corporation

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

REJ09B0009-0240